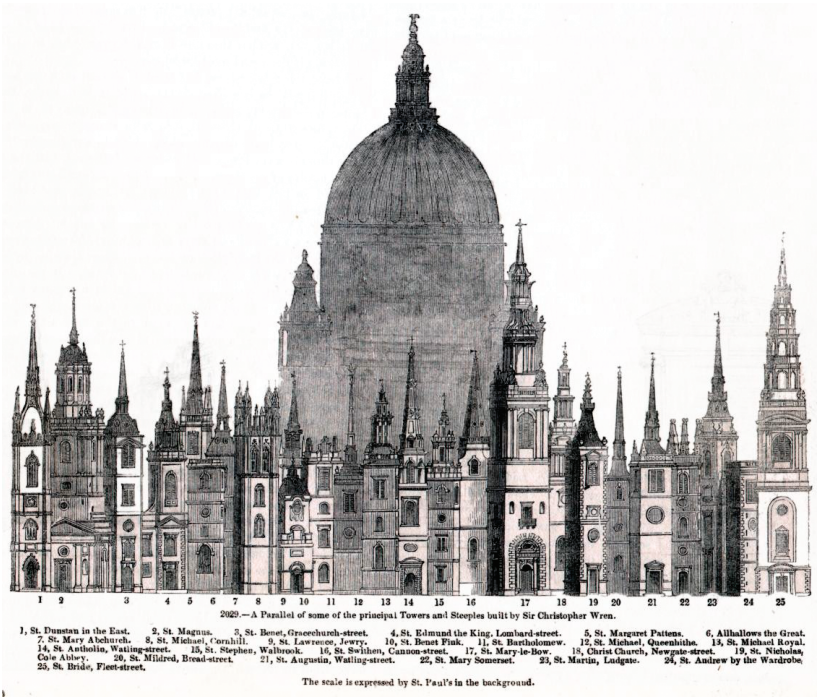
Intelligent Linguistic Architectures

Variations on Themes by Ronald M. Kaplan



edited by
Miriam Butt, Mary Dalrymple,
& Tracy Holloway King

INTELLIGENT LINGUISTIC ARCHITECTURES

CSLI Lecture Notes
Number 179

INTELLIGENT LINGUISTIC ARCHITECTURES

VARIATIONS ON THEMES BY RONALD M. KAPLAN

EDITED BY
MIRIAM BUTT, MARY DALRYMPLE,
& TRACY HOLLOWAY KING

CSLI
PUBLICATIONS
Center for the Study of
Language and Information
STANFORD, CALIFORNIA

CSLI Publications
Center for the Study of Language and Information
Leland Stanford Junior University
Printed in the United States
03 02 01 00 99 1 2 3 4 5

Library of Congress Cataloging-in-Publication Data

Intelligent linguistic architectures : variations on themes by Ronald M. Kaplan / edited by Miriam Butt, Mary Dalrymple, and Tracy Holloway King.

p. cm. – (CLSI lecture notes ; no. 179)
Includes bibliographical references and index.

ISBN-13: 978-1-57586-532-4 (pbk. : alk. paper)
ISBN-10: 1-57586-532-7 (pbk. : alk. paper)

1. Computational linguistics. 2. Kaplan, Ronald M. I. Butt, Miriam, 1966- II. Dalrymple, Mary. III. King, Tracy Holloway, 1966- IV. Title. V. Series.

P98.B88 2006
410'.285–dc22 2006018834
CIP

eISBN: 1-57586-779-6 (electronic)

CSLI Publications gratefully acknowledges a generous gift from Jill and Donald Knuth in support of scholarly publishing that has made the production of this book possible.

∞ The acid-free paper used in this book meets the minimum requirements of the American National Standard for Information Sciences—Permanence of Paper for Printed Library Materials, ANSI Z39.48-1984.

CSLI was founded in 1983 by researchers from Stanford University, SRI International, and Xerox PARC to further the research and development of integrated theories of language, information, and computation. CSLI headquarters and CSLI Publications are located on the campus of Stanford University.

CSLI Publications reports new developments in the study of language, information, and computation.
Please visit our web site at
http://cslipublications.stanford.edu/
for comments on this and other titles, as well as for changes and corrections by the author and publisher.

Contents

Contributors **ix**

Preface **xiii**

I Generation and Translation 1

1 Translation, Meaning and Reference 3
Martin Kay

2 Efficient Generation from Packed Input 19
John T. Maxwell III

3 Grammatical Machine Translation 35
Stefan Riezler and John T. Maxwell III

4 On Some Formal Properties of LFG Generation 53
Jürgen Wedekind

II Grammar Engineering and Applications 73

5 Using XLE in an Intelligent Tutoring System 75
Richard R. Burton

6 How Much Can Part-Of-Speech Tagging Help Parsing? 91
Mary Dalrymple

7 Rapid Treebank-Based Acquisition of Multilingual LFG Resources 111
Josef van Genabith

8 Hand-crafted Grammar Development – How Far Can It Go? 137
Christian Rohrer and Martin Forst

9 Using a Large, External Dictionary in an LFG Grammar: The STO Experiments 167
Beau Sheil and Bjarne Ørsnes

III Constraints on Syntax and Morphology 199

10 Agentive Nominalizations in Gĩkũyũ and the Theory of Mixed Categories 201
Joan Bresnan and John Mugane

11 Restriction for Morphological Valency Alternations: The Urdu Causative 235
Miriam Butt and Tracy Holloway King

12 A (Discourse-)Functional Analysis of Asymmetric Coordination 259
Anette Frank

13 The Insufficiency of Paper-and-Pencil Linguistics: the Case of Finnish Prosody 287
Lauri Karttunen

14 Gender Resolution in Rumanian 301
Louisa Sadler

15 Animacy and Syntactic Structure: Fronted NPs in English 323
Neal Snider and Annie Zaenen

16 Accounting for Discourse Relations: Constituency and Dependency 339
Bonnie Webber

IV Semantics and Inference 361

17 Direct Compositionality and the Architecture of LFG 363
Ash Asudeh

18 Packed Rewriting for Mapping Text to Semantics and KR 389
Dick Crouch

Index 417

Contributors

Ash Asudeh: Institute of Cognitive Science & School of Linguistics and Applied Language Studies, Carleton University, 2201 Dunton Tower, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada, asudeh@ccs.carleton.ca.

Joan Bresnan: Department of Linguistics, Stanford University, Stanford, California 94305, USA, bresnan@stanford.edu.

Richard Burton: Acuitus, Inc., 650 Castro Street, Suite 280, Mountain View, CA 94340, USA, burton@acuitus.com.

Miriam Butt: FB Sprachwissenschaft, D 184, Universität Konstanz, 78457 Konstanz, Germany, miriam.butt@uni-konstanz.de.

Dick Crouch: Natural Language Theory and Technology Area, Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304, USA, crouch@parc.com.

Mary Dalrymple: Centre for Linguistics and Philology, Oxford University, Walton Street, Oxford OX1 2HG, UK, mary.dalrymple@ling-phil.ox.ac.uk.

Martin Forst: Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Azenbergstraße 12, 70174 Stuttgart, Germany, forst@ims.uni-stuttgart.de.

Anette Frank: Language Technology Lab, DFKI, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, frank@dfki.de.

Lauri Karttunen: Natural Language Theory and Technology Area, Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304, USA, karttunen@parc.com.

Martin Kay: Department of Linguistics, Stanford University, Stanford, CA 94305-2150, USA, kay@csli.stanford.edu.

Tracy Holloway King: Natural Language Theory and Technology Area, Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA, thking@parc.com.

John T. Maxwell III: Natural Language Theory and Technology Area, Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304, USA, maxwell@parc.com.

John Mugane: Department of African and African American Studies, Harvard University, Barker Center Rm 244, 12 Quincy Street, Cambridge, MA 02138, USA, mugane@fas.harvard.edu.

Stefan Riezler: Natural Language Theory and Technology Area, Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304, USA, riezler@parc.com.

Christian Rohrer: Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Azenbergstraße 12, 70174 Stuttgart, Germany, Christian.Rohrer@ims.uni-stuttgart.de.

Louisa Sadler: Department of Language and Linguistics, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK, louisa@essex.ac.uk.

Beau Sheil: Acuitus, Inc., 650 Castro Street, Suite 280, Mountain View, CA 94340, USA, sheil@best.com.

Neal Snider: Department of Linguistics, Stanford University, Stanford CA 94305, USA, snider@stanford.edu.

Josef van Genabith: National Centre for Language Technology (NCLT), School of Computing, Dublin City University, Dublin 9, Ireland and IBM Dublin Center for Advanced Studies (CAS), josef@computing.dcu.ie.

Bonnie Webber: School of Informatics, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9lW, UK, bonnie@inf.ed.ac.uk.

Jürgen Wedekind: Center for Language Technology, University of Copenhagen, Njalsgade 80, 2300 Copenhagen, Denmark, juergen@cst.dk.

Annie Zaenen: Natural Language Theory and Technology Area, Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304, USA, zaenen@parc.com.

Bjarne Ørsnes: Department of Computational Linguistics, Copenhagen Business School, Dalgas Have 15, 2000 Frederiksberg, Denmark, boe.id@cbs.dk.

Preface

In a world where navigating in and around linguistic theories is often reminiscent of finding one's way around quicksand, working within Lexical-Functional Grammar (LFG) and other architectures designed by Ron Kaplan gives one the feeling of being in a safe and secure home.

Guided by the strong desiderata that emerged in the 1970s for the design of a good linguistic theory, Ron Kaplan joined forces with Joan Bresnan in 1977 to design an architecture that aimed to be formally responsible (and hence, implementable), linguistically forward-looking, and psycholinguistically realistic. Around the same time, Ron and his colleagues Martin Kay and Lauri Karttunen worked on making finite-state technology more efficient and applying the new technology to linguistic problems such as phonological alternations, morphological analysis, and even to problems that linguistics thinks of as being so "low-level" as to be beneath its notice, e.g. tokenization.

However, this is a misapprehension, as anyone who has ever listened to Ron Kaplan present the newest version of his tokenizer (most recently in the spring of 2006) quickly comes to realize. A tokenization talk by Ron is a real treat. Tokenization turns out to be a creature of beauty and elegance — a problem that demands extremely clever and compact rule interaction. One understands everything about the problem and the notation at one moment, and then nothing the next, until one has gone back and picked apart the intricate yet ruthlessly efficient edifice built by Ron.

One of the papers in this volume (Crouch) cites Ron Kaplan as preaching that "notation matters". Anyone who has engaged with Ron in painstaking discussions on linguistic notation, and particularly on whether a proposed formal device should be added to the inventory of LFG, will immediately recognize this to be a crisp characterization of Ron's central strategy. His formal intuitions are one of a kind, as is his

gift for finding the right abstract formal characterization of the messy linguistic problems which linguists are confronted with.

As testament to Ron Kaplan's design skills, the original architectural design of LFG has stood the test of time. Not only do linguists still find it a useful and structurally sound framework for linguistic analysis, but it has also emerged intact from an integration into large-scale grammar engineering applications. LFG grammars can now parse robustly and generate grammatical sentences from underspecified (or even bad) input; they are used for text condensation, machine translation, and even knowledge representation. A statistical component as well as a version of Optimality Theory have been integrated without violating or changing the original LFG kernel. No major shift in the theory has been necessary to accommodate these changes. Rather, theoretical and computational linguists have been able to build on the solid foundations laid by Ron in the late 1970s and early 1980s. It is no mean accomplishment to build a house that can accommodate a large family of disparate interests (theoretical linguists, field researchers, mathematicians/logicians, computational linguists, and engineers) and to make sure that it has a sound and very definite structure, but is still designed flexibly enough to accommodate change in the form of extra rooms being added on (e.g., optimization over f-structures), or entire extensions being built (e.g., Linking or Lexical-Mapping Theory, glue semantics). Most other linguistic houses need to be pulled down at least partially, if not razed, before change is possible.

This volume collects papers on a number of Ron's interests. Part I, on Generation and Translation, includes contributions by colleagues who have long-standing collaborations with Ron on how to design the best architecture for machine translation (Kay) and the best parser and generator (Wedekind, Maxwell). In addition to these long-standing issues, this part also includes a contribution that builds a novel machine translation system by taking newer developments into account: the recent focus on combining statistical methods with deep approaches to natural language processing (Riezler and Maxwell).

Part II, on Grammar Engineering and Applications, focuses on practical matters of natural language processing: using the LFG grammar development platform XLE for implementing tutoring systems (Burton), building large lexicons and grammars (Rohrer and Forst, Sheil and Ørsnes), exploring interactions of tagging and parsing (Dalrymple), and building large grammars and lexical resources from treebanks (van Genabith). All of these are issues with which Ron Kaplan has been centrally involved, and his suggestions and insights have served to push work in this field demonstrably forward.

In Part III, deep formal issues are discussed in light of difficult linguistic data (or rather, to be true to Ron's perspective: difficult linguistic data are analyzed with respect to high formal standards). These papers are wide-ranging, from contributions in OT-based and finite-state treatments of Finnish prosody (Karttunen) and the analysis of mixed-category constructions (Bresnan and Mugane) to theories of discourse in a formal perspective (Webber) and the importance of animacy in defining constraints on left displacement (Snider and Zaenen). Many of the papers in this section address foundational issues in Lexical Functional theory, including interactions between morphology and syntax in the treatment of complex predicates with the Restriction Operator (Butt and King), coordination and its interactions with agreement (Sadler), and the resolution of coordination asymmetries via f-structure analysis and the Principle of Economy (Frank).

Part IV, on Semantics and Inference, contains two contributions on formal semantics and its treatment in LFG: the issue of compositionality in syntactic and semantic theory (Asudeh) and the theoretical and practical issues in mapping from linguistic structures to knowledge representations (Crouch).

The vast range of topics dealt with in this volume do credit to the person who inspired them, namely Ron Kaplan. The alacrity with which the papers were offered up to this volume and the degree of cooperativeness in responding to the reviews and other requests for changes does immense credit to every author who has contributed. We would like to thank them, as well as the reviewers, who worked quickly, willingly, and thoroughly.

We would also like to thank Daniela Decheva Valeva and Rita Megerle, who helped with standardizing the bibliographies (not a simple or quick job!) and Tina Bögel, who helped with some mop up actions. Daniela Decheva Valeva in particular did a huge job, pitching in even while on vacation in Bulgaria. Finally, as always, our thanks go to Dikran for being there and giving us CSLI Publications.

Part I

Generation and Translation

1

Translation, Meaning and Reference

Martin Kay

President Nixon hoped to disengage the United States from the war in Vietnam by increasing training and equipment for the South Vietnamese forces and turning more and more of the responsibility for fighting the war over to them. A minor problem with the plan was that Vietnamese soldiers could not read the manuals that came with the equipment. Translating it all would be unthinkable unless a large part of the work could be automated. At the Rand Corporation, I secured a small machine translation contract to work on this and hired Ron Kaplan to help. After a short time, we were asked to work on Korean instead for reasons that remain unclear. Fortunately, for us, one language was like another. We were linguists, after all! We have worked together, on and off, ever since, on problems more or less closely related to machine translation. In this essay, I reflect, as I doubtless should have done already in Nixon's day, on just what translation is all about anyway, and why we did not always achieve as much as we had hoped.

Imagine a venerable scholar with a document in each hand. He is getting manifestly quite unhappy as he looks from one text to the other and back again. Finally, he throws both documents aside, declaring "This is rubbish! These are not translations!" The chances are that the venerable reader has allowed himself a little scholar's license in his choice of words. That is alright: he is angry and no one is listening anyway. The chances are that the documents are translations of one another, but they are such poor examples as to make him want to withhold recognition of them as such.

A poem that purports to be the translation of a poem in another

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

language may, like Fitzgerald's "Rubaiyat of Omar Khayyam", depart in so many and in such extreme ways from the original as to cast doubt on the claim that it is a translation, rather than a new work inspired by an older one. But, in more mundane cases, it is quite difficult to imagine the quality of a translation degenerating so much as to call in question whether it should be taken to be a translation at all. Degrees of translationhood are not necessarily the same as degrees of translation quality. We rarely find ourselves faced with the problem of judging the degree of translationhood achieved by a pair of utterances or documents, but the *gedanken* experiment that involves imagining ourselves in that position may be useful for what it says about the nature of translation.

In this essay, I offer some preliminary reflections on the question of when one text should be allowed to count as a translation of another. Though logically prior to just about all other questions concerning translation, it is one that is rarely addressed. A commonly accepted criterion, namely that the texts should express the same meanings, will quickly prove to be inadequate, though the intuition remains strong that there is some property of an original text that must be preserved in any translation. If not the meaning, then the question is, what is that property? The view for which I shall attempt to argue is that what must be preserved is the sequence of mental states through which each text leads its readers. Just what a mental state is will remain somewhat elusive but, in this, it will not differ from the meanings on which the commoner view rests.

For the purposes of this discussion, it will often be convenient to refer to one of a pair of documents as the *original* and the other as the *translation*. However, we will generally be thinking of translation as a symmetrical relationship. From the point of view of the translator, which document serves as the source makes many and crucial differences, but our concern here will be mainly with static relationships that exist between the documents themselves and not with the process that brought one or the other of them into being. It is sometimes possible to tell, especially in the case of poor translations, which must be the original and, while this is interesting, it will not be at the center of our concerns.

A natural first requirement to make of a translation is that it tell the same story as the original. To make this a general requirement, we clearly must construe the word "story" very broadly. Consider, for example, the case of a document that arrives in a box along with bits and pieces intended to be assembled into some object or contraption. The story that document tells is about a sequence of events in which

pieces are connected in specific ways and in a specific order. If two people were observed constructing such objects from boxes with identical contents except for the language in which the instructions were written, a *prima facie* case would surely have been made that the documents were translations of one another. The metaphor of a text as a set of instructions for assembling pieces into a complete object is perhaps more widely applicable than that of a story. After all, what a speaker or a writer is generally aiming to do is indeed to construct an object, but in the head of the hearer or reader rather than in the real world.

If texts in different languages lead to parallel sequences of events, we may be encouraged to accept them as translations, especially if other sequences could have led to the same result using the same pieces. But it may not be so. How the pieces go together might be so obvious to both constructors as to render the instructions superfluous. On the other hand one set of instructions may be altogether more detailed than the other, perhaps because it is intended for a reader with more experience in this kind of construction. It could nevertheless give rise to the same sequence of events. We will see an example of this kind shortly.

The requirements we have placed on the translation of an assembly instruction sheet might seem too narrow from a purely functional point of view. The success of the overall enterprise — that of achieving a correct final assembly — is surely paramount. The particular sequence of events plays a secondary role. Let us consider a specific example. For the sake of simplicity, both texts are in English.

> You can get to the airport on the RER, line B, from the Gare du Nord. You can reach the Gare du Nord by taking the Metro from Place Monge. The Place Monge is just up the hill from the apartment.
>
> Go up the hill from the apartment to Place Monge. Take the Metro from there to the Gare du Nord. From the Gare du Nord, take the RER line B to the airport

They both tell the reader how to get from some, presumably contextually given, apartment to the airport. While one may seem more natural than the other in some way, they both leave the reader in possession of the same mental construct, one that connects the apartment to the airport in a particular way. They might therefore be said to tell the same story.

In both versions, the story has three episodes corresponding to the three legs of the journey. The principal difference between the two versions is that the order of the episodes in one version is the reverse of

what it is in the other. As a consequence, the reader is in possession of different constructs, or partial constructs, after reading just one or two episodes, depending on the version, even though he has the same complete construct by the end of the story.

The reader's mental states, or partial constructs, are the life blood of literature. An author's skill consists, in large measure, in manipulating them in subtle ways, and a translator's skill consists in leaving them as he found them. But there is no subtle manipulation of mental states going on in our example. Indeed, if we take it that the original is on the left and the translation is on the right in the above display, we might be inclined to commend the translator for rearranging things so that the order of the mental states in the translation corresponds to the order of the physical states that would occur if the instructions were carried out. In a case like this, there might be some tension in the mind of the translator between the desire to leave the order of things under the control of the author and the desire to get the reader to the airport as reliably as possible.

As we look at larger and larger texts, the requirement to maintain the identity and the relative order of mental states dominates more and more. A redesign at this level would be seen as rewriting the story and not translation. In the translating of *belles lettres*, it is particularly important to respect the author's intended sequence of mental states as closely as possible. In these situations, the translator can be sure neither how important the sequence really is nor, indeed, exactly what the states are. So the safest policy is to translate the smallest pieces one can, consistent with maintaining the smoothness of the result, and to keep them as nearly as possible in the order that corresponds to the original. This is also the easiest thing to do. In translating more mundane texts there is rarely any cause to do otherwise except, perhaps, if one culture routinely places the ingredients in a recipe before the method, and another puts them the other way round, or something of that kind.

So we have arrived, by a somewhat circuitous route, at the notion that a translation should tell the same story as the original and, furthermore, that it should consist of as many elementary sections as possible, each being a translation of its opposite number in the other text.

Now let us examine another pseudo-translation, from English into English, that meets these requirements more nearly than our previous one:

Go out of the front door and turn left. You will pass three turnings on the right, the third being only for pedestrians. Turn right at the next possibility following this one and continue straight ahead, until you have the possibility of turning half left along a wide boulevard at the end of which you will see an impressive building with a lot of gold leaf on the roof. That is the building you are looking for.	Go west along the river and cross at the Pont du Carrouselle. Go through the Louvre and up the Avenue de l'Opéra. The opera house is at the end of that street.

This will doubtless seem a great deal less plausible as a translation than our earlier example. With a few exceptions, such as the word *go*, none of the words or phrases in the translation seems to translate a word or phrase in the original. However, both texts describe the same route from a hotel called "Les Rives de Nôtre Dame" in Paris, to the old Opera house. Since the mental states in the description correspond to the physical places mentioned, and in that same order, then these texts should surely be allowed to count as translations of one another. But maybe it is not sufficient that the observable behavior of the people following the two sets of instructions should be essentially indistinguishable. To the best of our ability, we must also look at their internal states — at the sequences of partial mental constructs that are assembled in their heads.

Both readers know that they will leave the hotel by the front door, one because the text says so, and the other because he knows that he is going to have to leave the building and, *ceteris paribus*, that is the best way to do it. But the reader of the longer text constructs a model with three turnings on the right, a half-left turn, and a wide boulevard. These will be part of the other reader's mental model only if he knows his way around Paris well enough to make the instructions largely redundant. Here, as always, the model constructed is a function both of the text and the previously constructed models that the reader has available. In this case, we can argue that similar models would be constructed only under special circumstances and the two texts should not therefore be allowed to count as translations.

Let us consider still another example taken from the magazine of the Accor hotel chain (Accor 2005).

Ci-contre: Vestiges du III^{e} millénaire avant J-C, surplombés de colonnes romaines.	*Left*: Ruins dating from the third millenium BC, surrounded by Roman columns.
Ci-contre: Théâtre romain (Odéon) construit au III^{e} siècle apr. J.-C.	*Right*: Roman theatre (Odeon) built in the third century AD.

The important point to note is, of course, that "Ci-contre" is translated first as "left" and then as "right". One would not wish to be forced into the position that it has both of these meanings. In fact, of course, it has neither. Taken in isolation, we might translate it as "opposite", or "on the facing page". In the situation in question, what was being referred to was not on the facing page but on the opposite side of the same page. In the first case, the text was to the right of the picture it referred to and, in the second case, on the left. The translator achieved the required mental state but using a word with opposite meanings in the two cases. The important thing — the only important thing — is to cause the reader to look at the correct picture. Whether "ci-contre" has a meaning that is in any way related to those of "left" or "right" is of secondary importance.

These examples illustrate that language is essentially *situated* in the sense of being grounded not simply in meanings such as a dictionary would supply for each of the words, but in complete situations that allow for coherent sets and sequences of mental states. To make this point, we will consider an extended example, not of a translation, but of a monolingual extract from the autobiography of the physicist Richard Feynman (Feynman 1985). This passage explains how a certain kind of combination lock, a kind often used on small safes and filing cabinets, works. The lock has a single dial which is turned a certain number of times alternately in clockwise and counterclockwise directions, stopping each time when a particular number on the dial is at the top. The question is, how does the mechanism cause the lock to open in response to just one such sequence of events?

I will discuss only a few phrases from the beginning of this passage in detail, but I invite the reader — especially the reader who does not know how these locks work — to read the whole passage once or twice. I hope that this will demonstrate that the passage achieves the author's intentions for it, which is presumably that the reader should come to know how the device works.

> There are three discs on a single shaft, one behind the other; each has a notch in a different place. The idea is to line up the

> notches so that when you turn the wheel to ten, the little friction drive will draw the bolt down into the slot generated by the notches of the three discs.
>
> Now, to turn the disks, there's a pin sticking out from the back of the combination wheel, and a pin sticking up from the first disk at the same radius. Within one turn of the combination wheel, you've picked up the first disc.
>
> On the back of the first disk, there's a pin at the same radius as a pin on the front of the second disc, so by the time you've spun the combination wheel around twice, you've picked up the second disc as well.
>
> Keep turning the wheel, and a pin on the back of the second disk will catch a pin on the front of the third disc, which you now set into the proper position with the first number of the combination.
>
> Now you have to turn the combination wheel the other way one full turn to catch the second disc from the other side, and then continue to the second number of the combination to set the second disc.
>
> Again you reverse direction and set the first disc to its proper place. Now the notches are lined up, and by turning the wheel to ten, you open the cabinet.

This is an extremely informal piece of writing. The picture of the mechanism that is in the words is casual and impressionistic, but the one that is constructed in the mind of the attentive reader is very precise. This is because the picture is constructed from components some of which are contributed by the text while others are contributed by the reader.

Let us begin at the beginning:

> There are three discs on a single shaft ...

A disk is a circular piece of material, quite thin relative to its diameter. The word "shaft" has several meanings. Among various other things, it can be (1) a long, usually vertical, space in the ground or in a building, such as an elevator or mine shaft, or (2) a solid cylinder, usually of metal, much longer than its diameter, intended to convey rotational force, as in the drive shaft of a car, or to support rotating wheels. In the interest of simplicity, let us suppose that these are the only possibilities. The first of the two meanings seems hard to involve in the workings of a lock, and maybe this is why the second immediately

seems right. The difficult question is to decide in just what sense the disks are "on" the shaft. They could be screwed or welded to it so that the shaft would lie across the surface of each disk, perhaps at the diameter. Or they could be screwed or welded to the end, or ends, of the shaft. It was immediately clear to me, as to others to whom I put the question, that neither of these is intended. What we have somehow to understand is that there is a hole in the center of each disk that the shaft passes through. The disks can therefore rotate on the shaft so that they become, essentially, wheels. There is only the gentlest of invitations to this interpretation, in that the meaning we are betting on for "shaft" makes of it something intended to carry wheels and the disks are reasonable candidates for this role.

Now the disks are

> . . . one behind the other. . .

If the shaft passes through a hole in the center of each disk, as I am betting they do, then are they not one beside the other or, even one on top of the other? Of course, they are one behind the other from the point of view of a person who is approaching the lock from the canonical angle, that is, from outside and in front of the safe or file cabinet. Such a person sees that disk directly in front, behind which the shaft extends away from him, carrying the disks, one behind the other. But there is no absolute or neutral position that justifies the word "behind" here, and it plays no role in understanding how the mechanism works.

Now for a real puzzle.

> . . . each has a notch in a different place. . . .

A notch is a small cut in the edge of something. For me, the word carries with it the suggestion that the cut has been made in a casual manner and may therefore be irregular in shape. However, I am prepared to abandon this last condition as being almost certainly inapplicable to a precisely engineered mechanism like a combination lock. The real problem comes with the phrase "in a different place". A disk with a hole in the center has only two edges in which to put a notch: the outside edge, and the one around the hole through which the shaft passes. If one disk had a notch on one of these edges and one on the other, that would be two notches in clearly different places. But, now, what of the third one? The third disk must surely have a notch in the same place as one of the others because there are simply no other alternatives. All other things being equal — and I am claiming they must be for things to make sense — the notches should be in corresponding edges of each

disk.

It will turn out that the notches all have to be in the outer edge, and I had no difficulty placing them there when I first read the piece. But how, then, can they be in different places given that disks are, by definition, objects of wonderful symmetry? The problem is somewhat less perplexing if one thinks of the disks, not as they would be when first manufactured, or when removed from the lock and lined up carefully on the work bench, but as they might appear when one first opened the mechanism and looked inside. Each notch would then be in a different place, not relative to its disk, but relative to the mechanism as a whole.

Questions like these arise throughout the whole of the text. I mention a few more, without discussing them at length.

> The idea is to line up the notches ...

What idea? "Line up" in what sense?

> when you turn the wheel to ten

What wheel? And what does it mean to turn a wheel "to ten"?

And, please, what are we to make of the following?

> the slot generated by the notches of the three discs.

In two places, the operator of the lock is assured that he will "pick up" one disk or another. What meaning of this verb is being invoked here? How can we pick up a disk that we cannot even see and whose existence we are learning about now for the first time?

Let me reiterate that I take this to be a remarkably successful piece of writing which, for me at any rate, succeeded immediately in its presumed goal of conveying how one of these combination locks works. But it does it by inviting the reader to participate in a mental journey in which the text serves only to gently suggest which way to take at each branch in the road. Each signpost makes sense only to one who has been involved from the start and who knows where we are going, and why.

The burden of this discussion was summarized by Jean Delisle in his *L'analyse du discours comme méthode de traduction*, (p. 73) where he says "Le texte d'un message ne contient pas le sens, il ne fait que pointer vers lui." [1] If everything were included that would be required so that even the most perverse reader could not misinterpret the writer's

[1] The text of a message does not contain the meaning, it only points to it.

intention, it would be so heavy and complicated as to defeat that very purpose. One might be able to argue that it was strictly correct, even if it were totally incomprehensible.

Now here is the problem that this poses for the translator. The grammar and lexicon of every language requires certain kinds of information to be made explicit that can be omitted in some others. Consequently the part of the intended message that is made explicit in one language can rarely be exactly what it is in another. This means that the translator can, and often does, leave some of the information in the original implicit, allowing the momentum of the mental journey to supply it. It also means that the translator must frequently make explicit information that was left implicit in the original. Needless to say, this is only possible if the translator is being carried forward by the momentum of the text in just the way intended by the author. Most of the time, this does not put an inordinate strain on the translator, though it would presumably be entirely beyond the reach of any current machine-translation system.

Vinay and Darbelnet (1958) give many examples in which information that is required in French is optional in English, or the other way round. Consider the French question "Où voulez-vous que je me mette?" which can be glossed as "Where do you want me to put myself?". However, this is something no English speaker would ever say. Better translations are available, but there are indefinitely many of them and the choice among them depends on where we stand in the mental journey. Some possibilities are:

Where do you want me to sit
stand
park
tie up my horse
sign my name
draw up my regiment
hang my pictures
...

Finding a good translation in a case like this requires the momentum of the mental journey to carry one forward to the next state where the words themselves are inadequate to do it. There is often no alternative to this. In this particular case, a cunning translator who was not carried forward strongly by the momentum might write "Where do you want me?", but such a possibility may not be available in every case.

As another example, the French noun "promenade" describes move-

ments through space that a person undertakes for recreational purposes. By default, one would probably translate it as "go for a walk", or "take a walk". But it could also be "go for a ride" if the context made it clear that a horse or bicycle was involved, or "go sailing" in situations where walking would require superhuman powers, and so on.

As a final example, a chair in French must be specified as either "chaise" (straight chair) or "fauteuil" (easy chair). Leaving other alternatives aside, let us ponder what considerations would be involved in the following examples of the word in use:

I found this change purse on a chair in the kitchen.	J'ai trouvé ce porte-monnaie sur une chaise dans la cuisine.	(1)
I found this change purse in a chair in the living room.	J'ai trouvé ce porte-monnaie dans un fauteuil dans le salon.	(2)
Let's put Mary in the chair at the other end of the table.	Mettons Marie dans la chaise à l'autre bout de la table.	(3)
There is plenty to eat. That is not the problem. The problem is that we don't have enough chairs.	Il y a assez à manger. Ça, ce n'est pas le problème. Le problème, c'est que nous n'avons pas assez de chaises.	(4)
There are plenty of barbers. That is not the problem. The problem is that we don't have enough chairs.	Il y a assez de coiffeurs. Ça, ce n'est pas le problème. Le problème, c'est que nous n'avons pas assez de fauteuils.	(5)

In each case, a larger context could change our judgement about which word to use for "chair", but given only what is here, the following considerations, at least, seem relevant. We might expect to find more straight chairs in the kitchen and more easy chairs in the living room, but the key distinction between (1) and (2) lies in the preposition. The arms of an easy chair give it more the aspect of a container, so that you would find things *in* it, whereas you find things *on* the surface of a straight chair. Likewise for *dans* and *sur* in French. This is the kind of clue that a statistical machine translation system might easily learn to pick up because the preposition is only one word removed from the word "chair". But, then, it would probably get (3) wrong. Presumably what is happening here is that decisions are being made about where to seat people around a table. This is a situation in which the preposition *in* is generally used in English, regardless of the kind of chair involved. This is *chair* as a position round a table, rather than *chair* as an article of furniture.

In both (4) and (5), there are three sentences. And the second one

can be thought of as standing for an arbitrary amount of intervening material, just so long as it does not upset the connection between the first sentence and the third. In both cases, there is a problem in that we do not have enough chairs of some kind. In (4), the momentum carries the reader naturally to the idea that the food of which we have sufficient quantities will be consumed by people who will be seated, presumably around one or more tables. They will be sitting on straight chairs because those are the kinds of chairs one sits on while seated around a table eating. In (5) we are presumably envisaging some number of barbers, presumably attending in some appropriate way to the hair of their clients. It is usual for a barber to do this while the client is seated in a chair and that chair is called a *fauteuil* in French, however comfortable or uncomfortable it may be.

I have argued for the view that a text and its translation must consist of sequences of corresponding segments each of which should be as short as possible while honoring grammatical and stylistic constraints. Such corresponding segments should add similar pieces to the mental structure that is under construction in each reader's mind. Any requirement to preserve meaning must be subservient to these. But, if this effect is to be achieved, there must presumably be some properties of segments that are preserved. The most important one is not far to seek. It is the referential properties of the segment, where we construe the term "referential" very broadly. The reference, of course, is to objects in the mental model that is under construction rather than to objects in the world.

When we come upon the words "There are three disks on a single shaft", we are already committed to a collaboration with the author in which we will do our best to make a replica in our minds of a picture that he has in his. The references will not be to anything in the real world — it is a generic lock we are talking about and not any particular instantiation — and we will give everything we hear the interpretation that seems best fitted to advancing that enterprise. We know quite a lot even before the first word arrives because the explanation we are being given is of a device that we are already familiar with from the outside and our understanding of the first words comes partly from their meanings and grammatical relationships and partly from that prior knowledge. Suppose the text had contained the following sequence:

> The Ministry keeps its archives in a tunnel about thirty feet underground, and reachable by two special elevators, one in each of its office buildings. There are three disks on a single shaft.

We have no context beyond what the words themselves provide. The first sentence invites us to imagine a ministry — perhaps part of a government — that occupies a pair of buildings from each of which it is possible to take an elevator to a subterranean tunnel in which the ministry stores its archives. We need to bring only some very general knowledge of the world to the task of constructing the image from the words. But the next sentence is more puzzling because we need to interpret it as a blueprint for an addition to the existing structure and, indeed, it does seem to refer to some parts of that structure where the new material might be added, namely at the shafts through which the elevators presumably move. But now we must somehow install three disks on one of these shafts. We can only hope that the immediately following text will provide some assistance in doing this because I, for one, am entirely at a loss. If we were translating the text into a language that had different words for elevator shafts and the mechanical engineer's shafts, we might lean towards the former, but presumably without complete confidence. We desperately need to have a mental model for the disks, and of the situation in which several of them are on an elevator shaft.

"Ci-contre" must achieve the same reference as "left" in one context, and the same as "right" in another. The problem is not hard because the reference is to the real world in this case; indeed it is to the very paper on which the words appear. In the case of "Où voulez-vous que je me mette?", we need a replacement for x in "Where do you want me to x" that achieves the same reference as the French. But if x refers to anything here it must surely be an action that has not been performed yet and is therefore not available in the real world to be referred to. But it is available in the mental model that we are constructing as we read the text. This is why we need to construe the notion of reference very broadly. In particular, we must construe it against the background of a very promiscuous ontology (Hobbs 1985). In "Où voulez-vous que je me mette?", the actions that the last three words are most likely to refer to are narrowly constrained by the surroundings in the real world, or as established by the foregoing text. If we are contemplating a table around which people are being seated in preparation for a meal, it is natural to take it as referring to an act of sitting down, or taking one's place, and we can achieve the same reference in English by saying "Where do you want me to sit?", but if I am addressing the official at a polling station before casting my vote, a better rendering would be "Where do you want me to sign?". I can even change the meaning and say something like "Where do I sign?" or "I have to sign somewhere here, don't I?" so long as I can be reasonably sure that the effect on

the reader's mental model will be the same.

It goes without saying that there can be quite a lot of guesswork in translating. An author may be unconscious of potential ambiguities in his text, or may abstain from clarification that would be cumbersome or that might seem to reflect adversely on the intelligence or good sense of the reader.

Consider the following sentences:

I just got back from Dallas/Prague. I had forgotten how good beer tastes.	Ich komme eben von Dallas/Prague zurück. Ich hatte vergessen, wie gutes Bier schmeckt.	(6)
I just got back from Provo/Riyadh. I had forgotten how good beer tastes.	Ich komme eben von Provo/Riyadh zurück. Ich hatte vergessen, wie gut Bier schmeckt.	(7)

The second sentence in each English pair can be paraphrased either as "I had forgotten how good it is that beer tastes" and "I had forgotten the taste of good beer". In the syntax underlying the first, but not the second interpretation, "good" is an attributive adjective modifying "beer". The corresponding German requires agreement between the adjective and the noun, and hence the form "gutes". In the other interpretation, "good" is predicative and therefore has the uninflected form "gut".

In these examples, I take it that, all other things being equal, the first interpretation will seem more natural for Utah and Riyadh, and the second for Texas and Prague based on common stereotypes according to which beer is hard, if not impossible, to obtain in Utah and Saudi Arabia whereas in Texas and the Czech republic, it is not only readily available, but it is especially good.[2] The point is simply that an infusion of information, based on fact or fancy, stereotype or surmise, is indispensable for determining the morphology of the German adjective. Moreover, any connection between these inferences and the meanings of the words is mediated through the mental model that the hearer constructs, and not directly through the meanings of the words.

It has been my intension to cast some doubt on the rarely questioned claim that a translation is successful to the extent that it preserves the meaning of the source text, *modulo* requirements on fluency, register, and the like. I have suggested replacing meaning in this role with a notion of a sequence of mental states. A mental journey is a sequence

[2]Ivan Sag, to whom I owe this set of examples, points out that the stereotypes, particularly concerning Utah, are inaccurate.

of mental states, and a mental state is a partially constructed mental model. A mental model is inhabited by objects, properties, beliefs and so forth, which may or may not have correspondents in some real or imaginary world. From a purely logical point of view, we do not need to ascribe any properties to these objects beyond that of distinguishability. As with the *atoms* of some programming languages, like Lisp, we need to be able to say of a pair of variables that have atoms as their values if they are the same or different.

It has been proposed, notably by Johnson-Laird (1983), that models of inference based on mental models constitute a superior theory of the way humans solve logical tasks. Clearly, translation involves solving many and diverse logical tasks, but it also serves to underline how little of the information required for the construction of the mental model is typically furnished by the text itself, and how much must be supplied from the hearer's preexisting stock of models and partial models. This is where the designers of machine translation systems need to concentrate their efforts because, without some means of storing such models, and constructing new ones as text is processed, high quality machine translation will remain an illusion.

References

Accor. 2005. "À livre ouvert" (An open book). *Accor Magazine*, No. 68, Summer 2005.

Feynman, Richard P. 1985. *Surely You're Joking, Mr. Feynman*. New York: W. W. Norton.

Hobbs, Jerry R. 1985. Ontological promiscuity. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pages 61–69. Chicago.

Johnson-Laird, Philip N. 1983. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge: Cambridge University Press.

Vinay, J-.P. and J. Darbelnet. 1958. *Stylistique comparée du français et de l'anglais*. Harrap.

2

Efficient Generation from Packed Input

John T. Maxwell III

Kay (1996) introduces chart generation, a simple and relatively efficient algorithm for generating strings from an unambiguous meaning representation. This paper extends Kay's algorithm to efficiently generate strings from packed meaning representations which encode a large number of meanings in a small amount of data. This is useful when trying to efficiently translate all of the possible meanings of a source sentence into a target language, or when trying to present a user with alternate paraphrases that distinguish the different meanings of a sentence. The new algorithm will typically generate in polynomial time in the size of the packed input when the grammar is context-free, the number of dependencies per semantic variable is bounded, and the disjunctions are relatively independent.

2.1 Introduction

Parsing is the process of mapping a string of words to a set of possible meanings. In our terminology, generation is the process of going from a meaning to a set of strings that express that meaning (this is called "realization" by some researchers). Thus, generation is just the inverse of parsing. Although generation is the inverse of parsing, it has not been easy to take advantage of the extensive work that has been done on using chart parsers to parse efficiently (Earley 1970, Thompson 1983, Kay 1989, Nederhof 1993, Sikkel and Op den Akker 1993, van Lohuizen 1997, Penn and Munteanu 2003, etc.). This is because when parsing,

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

the string of words is given and the meaning is unknown. When generating, on the other hand, the meaning is given and the string of words is unknown. Chart parsers work by indexing edges by the substrings that they cover. Since there is only a quadratic number of substrings, the time that it takes to parse is polynomial in the size of the input for context-free grammars. Since the words are unknown during generation, it is not possible to index on the substrings. Thus chart parsing techniques are not directly applicable to generation.

2.1.1 Operating on packed representations

A chart parser for a context-free grammar represents all possible meanings of a string of words in a packed representation called a parse forest. The representation is "packed" in the sense that there can be an exponential number of meanings encoded in a polynomial amount of space. Many parsers use a beam search to pick the most likely meanings from the set of meanings given by the parse forest. The most likely meanings are then used as the output of the parser. However, this does not always produce the best results, since the most likely meanings do not always include the correct meaning. Enumerating all of the meanings is not practical since there can be an exponential number of meanings. A possible alternative is to use the parse forest as the output of the parser, and to change the clients of the parser to operate efficiently on parse forests. For instance, Dymetman and Tendeau (2000) give an algorithm for applying transfer rules to a packed meaning representation that is like a parse forest. The output of the algorithm is a packed meaning representation for a different language. However, Dymetman and Tendeau do not give an algorithm for generating from the resulting packed meaning representation.

2.1.2 Other approaches to generation

There have been several algorithms devised for generating from a meaning representation. Shieber et al. (1989) describe a semantic head-driven generation algorithm, but it does not use a chart in an interesting way and so is typically exponential in the size of the input. The Shake and Bake generation algorithm developed in Brew (1992) has the same problem. The first paper that used a chart in a meaningful way during generation is Kay (1996). Kay's algorithm uses a generation chart to efficiently generate from an unambiguous meaning representation. If the number of dependencies per semantic variable is bounded and the underlying grammar is context-free, then the time that it takes to generate using Kay's algorithm is polynomial in the size of the input meaning. Carroll et al. (1999) describe a variation on Kay's algorithm

that handles modifiers more efficiently by processing them in a second pass after the non-modifiers have been generated. However, it is more complicated to generalize this algorithm to the case of packed meaning representations, so this paper will focus on extending Kay's algorithm. Shemtov (1996) describes an algorithm for generating from packed input that is based on Kay's algorithm, but Shemtov's algorithm is typically exponential in the number of disjunctions.

The remainder of this paper will describe an algorithm for efficiently generating from a packed meaning representation. The packed meaning representation is usually derived from the parse forest produced by a parser, but the algorithm does not depend on this. The algorithm is an extension of Kay's algorithm for generating from an unambiguous meaning representation. Section 2.2 gives a overview of Kay's algorithm. Section 2.3 describes how Kay's algorithm can be extended to handle packed meaning representations. Section 2.4 illustrates the extended algorithm with a simple example. Section 2.5 discusses the efficiency of the algorithm. Section 2.6 describes possible applications of the algorithm.

2.2 Generation from an Unpacked Representation

The key ideas of the algorithm for generating described in Kay (1996) are:

1. Annotate each edge in the generation chart with its syntactic category and the semantic facts that it covers.
2. Index each edge by its active semantic variable.
3. Combine edges based on a shared semantic variable.
4. When a semantic variable is no longer accessible from an edge, then discard the edge if the semantic variable is missing any semantic facts.

The algorithm starts by seeding the generation chart with any lexical items that match the input meaning. Then edges are combined according to the syntactic rules given in the grammar that is being used. If the combination is missing semantic facts for internal variables, it is discarded. When all of the edges have been processed, then the solution is the edge that covers all of the semantic facts and that has the root syntactic category and whose first semantic variable corresponds to the root semantic variable of the input.

2.2.1 An example of Kay's algorithm

Suppose that we wanted to generate from the meaning representation for *John ran*:

John(1) run(0,1) past(0)

using the following lexical entries:

word	cat	semantics
John	np(x)	John(x)
ran	v(x,y)	run(x,y) past(x)

and the following syntactic rules:

s(x) ⟶ np(y) vp(x,y)
vp(x,y) ⟶ v(x,y)

First we would seed the generation chart with lexical entries that matched the input:

id	cat	semantics	source
1	np(1)	John(1)	*John*
2	v(0,1)	run(0,1) past(0)	*ran*

Then we would apply the syntactic rules to create new edges:

id	cat	semantics	source
3	vp(0,1)	run(0,1) past(0)	2
4	s(0)	John(1) run(0,1) past(0)	1+3

Each line represents an edge. The number in the first column is the edge's id. This is followed by the edge's syntactic category and semantics in the second and third columns. The semantics is an unordered multi-set of semantic facts. Finally, we indicate where the edge came from in the fourth column. This information is used during the read-out phase to produce the generation strings.

Edge 3 is obtained from edge 2 using the syntactic rule "vp(x,y) ⟶ v(x,y)". Edge 4 is obtained from edge 1 and edge 3 using the syntactic rule "s(x) ⟶ np(y) vp(x,y)" because of the shared semantic variable y (which is instantiated to 1 in this case). Since edge 4 contains all of the semantic facts in the input and has the root category and the first semantic variable corresponds to the root semantic variable of the input, it is the root edge for the generation chart. We can then find the generated strings by recursively following the subtrees given in the last field until we reach the terminal edges, and then concatenating the strings. This produces *John ran*.

In Kay's paper, the edges are indexed by their active semantic variables. These indices form the vertices of the generation chart, and are crucial for efficiently determining which edges to combine. Our extension to Kay's algorithm indexes the edges in exactly the same way.

However we have left the vertices out of the tables that describe the edges so that we can focus the paper on the parts of Kay's algorithm that we are extending.

2.2.2 The efficiency of Kay's algorithm

Kay's algorithm works well when there are no modifiers represented in the semantics. Modifiers are a problem since if a noun or verb has N modifiers, there can be $N!$ different ways of attaching the modifiers. Most of the time, though, there are only a few modifiers per noun or verb, so this does not cause much of a problem. However, modifiers can still cause problems even when there are a bounded number of modifiers per noun or verb. This is because the modifiers are optional in the syntax, and the algorithm described so far does not force the modifiers to be attached. This means that if an input meaning had N nouns with just one modifier attached to each noun, then the algorithm would generate 2^N root edges. Kay's algorithm avoids this problem by noticing when an edge has a semantic variable that is inaccessible to the syntax and checking whether the semantic variable has any missing semantic facts. If the semantic variable is missing a semantic fact, then the edge is discarded. This is safe since the semantic variable is inaccessible to the syntax and so the semantic fact cannot be added later. If the number of dependencies per semantic variable is bounded and the generation grammar is context-free, then Kay's algorithm will generate all possible strings for a given input in time that is polynomial in the size of the input.

2.3 Generating from a Packed Representation

A simple way to extend Kay's algorithm so that it can handle packed input is to allow an edge to have any combination of semantic facts that is licensed by the packed input. However, since the packed input can have an exponential number of different possible combinations of semantic facts, this makes the algorithm exponential in the size of the input. The major problem is how to make sure that exactly one choice is made from each of the disjunctions in the input without an explicit list of the facts that each edge covers. We need some way to put members of a disjunction into an equivalence class.

2.3.1 Putting disjuncts into equivalence classes

One way to put members of a disjunction into an equivalence class is to convert the packed input into a set of semantic rewrite rules that have a pseudo-fact that represents a disjunction on the left-hand side and semantic facts (possibly including other pseudo-facts) on the right-

hand side. For instance, we might convert a disjunction that represents the ambiguity of the verb *saw* into the following semantic rewrite rules:

P ⟶ see(0,1,2) past(0)
P ⟶ saw(0,1,2) pres(0)

The semantic facts on the right-hand side of each rule are unordered, and correspond to the semantic facts of one disjunct in the disjunction. Whenever an edge has some semantic facts that correspond to the right-hand side of a semantic rewrite rule, a new edge is created with the semantic facts replaced by the pseudo-fact that is on the left-hand side of the rule. The main advantage of this is that replacing semantic facts with pseudo-facts will cause edges to be collapsed together in the chart because they have the same semantics even though they were derived from different semantic facts from different disjuncts. When the semantics is reduced to just the root pseudo-fact, then the semantics is complete. The semantic rewrite rules will be explained in more detail in the example in section 2.4.

2.3.2 Avoiding spurious ambiguity

We can extend Kay's algorithm to handle packed representations by allowing semantic rewrite rules to apply at the same time that syntactic rewrite rules apply. However, this can lead to spurious ambiguities. For instance, if a semantic rewrite rule and a syntactic rewrite rule can both apply to an edge, then they can apply in two different orders to ultimately produce the same edge. We solve this problem by only allowing a semantic rewrite rule to apply to a new semantic fact or a new combination of semantic facts. Also, if two non-overlapping semantic rules can apply to the same edge, then they can apply in two different orders to produce the same new edge. We solve this problem by ordering the semantic rewrite rules, and disallowing an earlier rewrite rule from applying if a later rewrite rule has already applied without an intervening syntactic rule applying.

2.4 A Simple Example of Packed Generation

Suppose we wanted to generate from all possible meanings of *You saw trees with binoculars*. Here is a possible packed representation of the meaning of this sentence using the contexted notation described in Maxwell and Kaplan (1989):

you(1)	*you*
p1 → see(0,1,2) past(0)	past tense of *see*
p2 → saw(0,1,2) pres(0)	present tense of *saw*
tree(2) pl(2)	*trees*
q1 → with(0,3)	*with* attaches to *saw*
q2 → with(2,3)	*with* attaches to *trees*
binoculars(3)	*binoculars*
TRUE ↔ oneof(p1,p2)	
TRUE ↔ oneof(q1,q2)	

This representation has two parts: a set of contexted facts of the form "context → facts" and a set of propositional constraints of the form "context ↔ oneof(a1,a2,...,an)", where the ai are boolean variables. Contexts are boolean combinations of the boolean variables. Facts that have no context are implicitly in the "TRUE" context. Solutions can be extracted by finding a set of assignments to the boolean variables that satisfies the propositional constraints and then evaluating the context of each contexted fact to see if is true or false. Contexted facts with false contexts are discarded.

We can represent the packed ambiguity as a grammar made up of rewrite rules whose right-hand sides are unordered. The categories of the rules are pseudo-facts that represent a set of alternative meanings. The top-level pseudo-fact is named "ALL" to indicate that it covers all of the facts needed for one complete meaning. There is one rule for each simple context plus an "ALL" rule for the "TRUE" context. Each rule lists the facts and disjunctions that are defined in its context. Contexts use the same rule category if and only if they are from the same disjunction. Here is the grammar for the packed input given above:

ALL ⟶ you(1) P tree(2) pl(2) Q binoculars(3)
P ⟶ see(0,1,2) past(0)
P ⟶ saw(0,1,2) pres(0)
Q ⟶ with(0,3)
Q ⟶ with(2,3)

The first rule says that ALL can be rewritten as a set of semantic facts that are in all of the analyses plus the pseudo-facts P and Q. The next two rewrite rules say that the pseudo-fact P can be rewritten as either see(0,1,2) past(0) or saw(0,1,2) pres(0). Similarly, the last two rewrite rules say that the pseudo-fact Q can be rewritten as with(0,3) or with(2,3).

Starting with ALL and non-deterministically replacing pseudo-facts according to the rewrite rules given above until there are no more

pseudo-facts left produces all of the valid meanings for *You saw trees with binoculars.*

The following is a set of simplified lexical entries for English:

word	cat	semantics
you	np(x)	you(x)
saw	v(x,y,z)	see(x,y,z) past(x)
saw	v(x,y,z)	saw(x,y,z) pres(x)
sawn	ppt(x,y,z)	saw(x,y,z)
seen	ppt(x,y,z)	see(x,y,z)
are	aux(x)	pres(x)
were	aux(x)	past(x)
trees	np(x)	tree(x) pl(x)
with	p(x,y)	with(x,y)
by	by(x)	
binoculars	np(x)	binoculars(x)

Here is a set of simplified syntactic rules for English:

s(x) ⟶ np(y) vp(x,y)
vp(x,y) ⟶ v(x,y,z) np(z)
vp(x,z) ⟶ aux(x) ppt(x,y,z) byp(y)
byp(x) ⟶ by(x) np(x)
pp(x) ⟶ p(x,y) np(y)
vp(x,y) ⟶ vp(x,y) pp(x)
s(x) ⟶ pp(x) s(x)
np(x) ⟶ np(x) pp(x)

We start by seeding a generation chart with instantiated versions of the lexical items that match the input. This produces:

id	cat	semantics	source
1	np(1)	you(1)	*you*
2	v(0,1,2)	see(0,1,2) past(0)	*saw*
3	v(0,1,2)	saw(0,1,2) pres(0)	*saw*
4	ppt(0,1,2)	saw(0,1,2)	*sawn*
5	ppt(0,1,2)	see(0,1,2)	*seen*
6	aux(0)	pres(0)	*are*
7	aux(0)	past(0)	*were*
8	np(x)	tree(2) pl(2)	*trees*
9	p(0,3)	with(0,3)	*with*
10	p(2,3)	with(2,3)	*with*
11	by(1)		*by*
12	np(3)	binoculars(3)	*binoculars*

This uses the same notation as the *John ran* example given earlier.

Whenever a new edge is added to the generation chart it must be processed further, since it may give rise to more new edges. This is usually managed using an agenda. When we process a new edge, we first apply any semantic rewrite rules that match the new semantic facts or combinations of facts to produce new edges with reduced semantics. Then we apply the syntactic rewrite rules for English that match the edge's category along with perhaps some other edges to produce new edges. If we apply the semantic rewrite rules to the edges given above, we get the following new edges:

id	cat	semantics	source
13	v(0,1,2)	P	2,3
14	p(0,3)	Q	9
15	p(2,3)	Q	10

Edge 13 is produced from edge 2 by replacing "see(0,1,2) past(0)" with P based on the semantic rewrite rule "P $\longrightarrow$ see(0,1,2) past(0)". It is also obtained from edge 3 by replacing "saw(0,1,2) pres(0)" with P based on the semantic rewrite rule "P $\longrightarrow$ saw(0,1,2) pres(0)". The fact that edge 13 can be derived two different ways is reflected in the source information at the end using a comma to separate the different sources.

Edge 14 is derived from edge 9 by reducing the semantics using the semantic rule "Q $\longrightarrow$ with(0,3)". Edge 15 is derived in a similar way. We produce new edges rather than changing the semantics of edges 9 and 10 in case there are other ways to reduce the semantic facts. If there is only one way to reduce a semantic fact then it is acceptable to change an edge's semantics, although we do not do that in this paper.

Now we use syntactic rewrite rules to combine edges to produce new edges. The rule "pp(x) $\longrightarrow$ p(x,y) np(y)" produces the following edges:

16	pp(0)	Q binoculars(3)	14+12
17	pp(2)	Q binoculars(3)	15+12

Edge 16 and edge 17 correspond to *with binoculars*.

If we were to combine edge 9 and edge 12 to produce a new edge, the new edge would never be part of a valid generation solution. This is because we require semantic rewrite rules to apply only when they first become applicable to avoid spurious ambiguity. We know that the semantic fact "with(0,3)" on edge 9 must be reduced by a semantic rewrite rule since it does not appear in the semantic rewrite rule for "ALL". The only semantic rewrite rule that reduces "with(0,3)" is "Q $\longrightarrow$ with(0,3)". This rewrite rule first becomes applicable on edge 9,

so it cannot apply to the semantic facts of any edge built with edge 9. If we were to combine edge 9 and edge 12, the semantic facts would never be reduced to "ALL". So we do not combine edge 9 and edge 12 to produce a new edge.

Now we use the rule "np(x) ⟶ np(x) pp(x)" to produce a new edge from edge 8 and edge 17:

18	np(2)	tree(2) pl(2) Q binoculars(3)	8+17

Edge 18 corresponds to *trees with binoculars*.

Now we use the rule "vp(x,y) ⟶ v(x,y,z) np(x)" to produce a new edge from edge 13 and edge 8 that corresponds to *saw trees*:

19	vp(0,1)	P tree(2) pl(2)	13+8

This same rule can be used to produce a new edge from edge 13 and edge 18 that corresponds to *saw trees with binoculars*:

20	vp(0,1)	P tree(2) pl(2) Q binoculars(3)	13+18

Edge 20 can also be produced by combining edge 19 and edge 17 using the rule "vp(x,y) ⟶ vp(x,y) pp(x)". This corresponds to *saw trees with binoculars*, but where *with binoculars* modifies *saw* instead of *trees*. The new way of constructing edge 20 is added to the last field of the edge:

20	vp(0,1)	P tree(2) pl(2) Q binoculars(3)	13+18,19+17

Edge 20 is the place where the two different ways of attaching *with binoculars* meet to produce the same syntax and the same semantics.

Now we use the rule "s(x) ⟶ np(y) vp(x,y)" to produce a new edge from edge 1 and edge 20 that corresponds to *you saw trees with binoculars*:

21	s(0)	you(1) P tree(2) pl(2) Q binoculars(3)	1+20

The semantics of this edge can be reduced using the semantic rewrite rule "ALL ⟶ you(1) P tree(2) pl(2) Q binoculars(3)". This produces:

22	s(0)	ALL	21

This edge is the root edge of the chart, since it has the root pseudo-fact "ALL" and it has the root syntactic category and its first semantic variable corresponds to the root semantic variable of the input. All of the original parses are included in the generation forest rooted in edge 22.

We can also get some paraphrases of the original meanings by completing the chart using some of the other syntactic rules. For instance,

if we apply the syntactic rule "s(x) ⟶ np(y) vp(x,y)" to edge 1 and edge 19 and then apply "s(x) ⟶ pp(x) s(x)" to the result we get:

23	s(0)	you(1) P tree(2) pl(2)	1+19
21	s(0)	you(1) P tree(2) pl(2) Q binoculars(3)	1+20,16+23

Edge 23 corresponds to *you saw trees*. Edge 21 now has two ways of being constructed. The new way corresponds to *With binoculars you saw trees*.

We can also get passive paraphrases. First we apply the syntactic rule "byp(x) ⟶ by(x) np(x)" to get a passive agent (*by you*):

24	byp(1) you(1)	11+1

Then we use the syntactic rule "vp(x,z) ⟶ aux(x) ppt(x,y,z) byp(y)" to build passive constructions:

25	vp(0,2)	you(1) pres(0) saw(0,1,2)	6+4+24
26	vp(0,2)	you(1) past(0) see(0,1,2)	7+5+24

Edge 25 corresponds to *are sawn by you* and edge 26 corresponds to *were seen by you*.

The semantics of edge 25 can be reduced with the semantic rewrite rule "P ⟶ saw(0,1,2) pres(0)". The semantics of edge 26 can be reduced with the semantic rewrite rule "P ⟶ see(0,1,2) past(0)". These reductions produce the same semantics, so we end up with only one new edge:

27	vp(0,2)	you(1) P	25,26

Edge 27 is the place where the distinction between *are sawn* and *were seen* is eliminated because it is no longer relevant.

Next, we can combine edge 12 with edge 27 to complete the passive construction using the rule "s(x) ⟶ np(y) vp(x,y)":

28	s(0)	you(1) P tree(2) pl(2)	12+27

This corresponds to *trees* {*were seen*|*are sawn*} *by you*.

We can also combine edge 18 with edge 27 using the rule "s(x) ⟶ np(y) vp(x,y)" to produce another way to construct edge 21:

21	s(0)	you(1) P tree(2) pl(2) Q binoculars(3)	1+20,16+23, 18+27

This corresponds to *trees with binoculars* {*were seen*|*are sawn*} *by you*.

We can also combine edge 16 with edge 28 using the syntactic rule "s(x) ⟶ pp(x) s(x)" to produce yet another way to construct edge 21:

21	s(0)	you(1) P tree(2) pl(2) Q binoculars(3)	1+20,16+23, 18+27,16+28

This corresponds to *with binoculars trees* {*were seen*|*are sawn*} *by you*.

Finally, we can combine edge 27 with edge 16 using the rule "vp(x,y) ⟶ vp(x,y) pp(x)" to produce a new edge that corresponds to {*were seen*|*are sawn*} *by you with binoculars*. The new edge can be combined with edge 12 using the rule "s(x) ⟶ np(y) vp(x,y)" to produce another way to construct edge 21:

29	vp(0,2)	you(1) P Q binoculars(3)	27+16
21	s(0)	you(1) P tree(2) pl(2) Q binoculars(3)	1+20,16+23, 18+27,16+28,12+29

This corresponds to *trees* {*were seen*|*are sawn*} *by you with binoculars*.

2.4.1 Reading strings out of the generation forest

The generation forest that we have produced is a compact representation of all of the different ways that each of the input meanings can be generated. The well-formed output strings can be generated by starting at the root edge (edge 22) and recursively choosing one of the subtrees in the source field of each edge until we get to the terminal words. This produces the following sentences:

You saw trees with binoculars
With binoculars you saw trees
Trees with binoculars were seen by you
Trees with binoculars are sawn by you
With binoculars trees were seen by you
With binoculars trees are sawn by you
Trees were seen by you with binoculars
Trees are sawn by you with binoculars

2.4.2 How packed is the generation forest?

The packed generation forest for this example has 29 edges and 36 subtrees. The average number of edges and subtrees for each individual meaning in the input is 21.5 edges and 27.5 subtrees. Thus, it only takes about 35 percent more edges and 30 percent more subtrees to generate from 4 meanings as it does to generate from 1 meaning. This is because there is so much structure sharing between the different meanings.

2.5 Efficiency

It is important to notice that the generation of the P meanings is independent of the generation of the Q meanings in the example given

above. If there were M different P meanings and N different Q meanings in this example, then the generation time would be proportional to N+M instead of N*M. This is crucial for the efficiency of the algorithm, since if the processing time were proportional to the product of the number of disjuncts in general, then the processing time would be exponential overall.

If the number of dependencies per semantic variable is bounded and the grammar is context-free, then Kay's algorithm will generate in time that is polynomial in the size of the (unambiguous) input. Similarly, if the number of dependencies per semantic variable is bounded and the grammar is context-free, then our extension of Kay's algorithm will typically generate in time that is polynomial in the size of the *packed* input if the disjunctions are relatively independent of each other. The disjunctions are typically independent of each other if they are derived from a chart parser, and so the number of edges in the generation chart is usually proportional to the sum of the size of the disjunctions instead of being proportional to the product of the size of the disjunctions. This makes the algorithm typically polynomial overall.

2.6 Applications

The most obvious application of this algorithm is to use it to produce a packed representation of all possible translations of all possible meanings of a source sentence, and then use statistics to choose the most likely translation. This could be done by taking the packed output of a parser, applying a set of transfer rules to it, and then giving the new packed representation as input to a generator. It is not clear, though, whether this will work much better than doing the same thing with a beam search, since a beam search may be faster than the algorithm and it may often get the same sentence or a sentence of similar quality.

A more interesting application is to detect sentences that preserve an ambiguity in the source sentence. Sometimes an ambiguity in the source language can be preserved in the target language if the right construction is used. Shemtov (1997) describes an algorithm for detecting ambiguity-preserving translations when there is a packed representation of the generation output. Emele and Dorna (1998) describe another approach to ambiguity-preserving translation. Ambiguity preservation could be added as a feature to a statistical model, so that target sentences that preserved an ambiguity in the source sentence would be given a higher score than target sentences that did not. This information is not available in a standard statistical model.

Another possible application is to use the algorithm to produce a

packed representation of all possible translations and then let the user browse the packed set of translations to find the one that makes the most sense given the current context. This could be useful when the user knows a subject domain well, but does not know the source language.

Finally, the algorithm could be used to find sentences that distinguish different meanings of a given sentence so that a user can be questioned about the correct meaning of the sentence. For instance, suppose that you wanted to ask what *You saw trees with binoculars* meant. You could first determine where *with binoculars* attached by asking whether *With binoculars you saw trees* was true. If the answer is no, this means that "with binoculars" must have been attached to "trees" rather than "saw". If the answer is yes, "with binoculars" was probably intended to attach to "saw". If the answer is yes, you can then ask whether *With binoculars trees are sawn by you* is true. If the answer to this question is yes, then "saw" must have been the present tense form of "to saw". If the answer to this question is no, then "saw" must have been the past tense of "to see". Thus, we can determine the intended meaning of the original sentence by asking the user about the validity of carefully chosen paraphrases of it. The sentences used to distinguish meanings can be found by using a variant of Shemtov's technique for finding ambiguity-preserving sentences. Instead of looking for ambiguity-preserving sentences, look for sentences that don't preserve ambiguity.

2.7 Conclusion

The algorithm given in this paper works for arbitrary packed input and will typically produce all possible generations in polynomial time if the number of dependencies per semantic variable is bounded, the grammar is context-free, and the disjunctions are relatively independent. If we combine this with a parser that produces a packed representation and a transfer component that operates on packed represenations, then we can produce a packed representation of all possible translations in typically polynomial time. We can then pick the most probable translation from the packed representation or let the user browse the packed set of translations. Finally, we may be able to determine whether there are ambiguity-preserving translations. If ambiguity-preserving translations exist, then we can avoid the need to disambiguate the source sentence.

Acknowledgments

Ron Kaplan was instrumental in getting me started in computational linguistics, and we have had a very fruitful collaboration for more than

twenty years. He has been a wonderful colleague and manager. I am delighted to dedicate this paper to him.

References

Brew, Chris. 1992. Letting the cat out of the bag: Generation for shake-and-bake MT. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, pages 610–616. Nantes, France.

Carroll, John, Ann Copestake, Dan Flickinger, and Victor Poznanski. 1999. An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG'99)*, pages 86–95. Toulouse, France.

Dymetman, Marc and Frederic Tendeau. 2000. Context-free grammar rewriting and the transfer of packed linguistic representations. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'00)*, pages 1016–1020. Saarbrücken, Germany.

Earley, Jay Clark. 1970. An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery (ACM)* 13(2):94–102.

Emele, Martin C. and Michael Dorna. 1998. Ambiguity preserving machine translation using packed representations. In C. Boitet and P. Whitelock, eds., *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL'98) and 17th International Conference on Computational Linguistics (COLING'98)*, pages 365–371. Montreal, Canada.

Kay, Martin. 1989. Head driven parsing. In *Proceedings of the International Workshop on Parsing Technologies (IWPT'89)*, pages 52–62. Pittsburgh, PA.

Kay, Martin. 1996. Chart generation. In A. Joshi and M. Palmer, eds., *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL'96)*, pages 200–204. San Francisco, CA: Morgan Kaufmann.

Maxwell, John T., III and Ronald M. Kaplan. 1989. An overview of disjunctive constraint satisfaction. In *Proceedings of the International Workshop on Parsing Technologies (IWPT'89)*, pages 18–27. Pittsburgh, PA. (Also published as "A method for disjunctive constraint satisfaction" in M. Tomita, editor, *Current Issues in Parsing Technology*, Kluwer Academic Publishers, 1991).

Nederhof, Mark-Jan. 1993. Generalized left-corner parsing. In *Proceedings of the 6th European Chapter of the Association for Computational Linguistics (EACL'93)*, pages 305–314. Utrecht, The Netherlands.

Penn, Gerald and Cosmin Munteanu. 2003. A tabulation-based parsing method that reduces copying. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*, pages 200–207. Sapporo, Japan.

Shemtov, Hadar. 1996. Generation of paraphrases from ambiguous logical forms. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, pages 919–924. Copenhagen, Denmark.

Shemtov, Hadar. 1997. *Ambiguity Management in Natural Language Generation*. Ph.D. thesis, Stanford University.

Shieber, Stuart M., Gertjan van Noord, Robert C. Moore, and Fernando C. N. Pereira. 1989. A semantic-head-driven generation algorithm for unification-based formalisms. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL'89)*, pages 7–17. Vancouver, Canada.

Sikkel, Klaas and Rieks Op den Akker. 1993. Predictive head-corner chart parsing. In *Proceedings of the 3rd International Workshop on Parsing Technologies (IWPT'93)*, pages 267–275. Tilburg, The Netherlands; Durbuy, Belgium.

Thompson, Henry S. 1983. MCHART: A flexible, modular chart parsing system. In *In Proceedings of the 3rd National Conference on Artificial Intelligence (AAAI-83)*, pages 408–410. Washington, DC.

van Lohuizen, Marcel P. 1997. Survey of parallel context-free parsing techniques. Parallel and Distributed Systems Reports Series PDS-1997-003, Delft University of Technology, Delft, The Netherlands.

3

Grammatical Machine Translation

Stefan Riezler and John T. Maxwell III

3.1 Introduction

Recent approaches to statistical machine translation (SMT) piggyback on the central concepts of phrase-based SMT (Och et al. 1999, Koehn et al. 2003) and at the same time attempt to improve on some of its shortcomings by incorporating syntactic knowledge in the translation process. Phrase-based translation with multi-word units excels at modeling local ordering and short idiomatic expressions; however, it lacks a mechanism to learn long-distance dependencies and is unable to generalize to unseen phrases that share non-overt linguistic information. Publicly available statistical parsers can provide the syntactic information that is necessary for linguistic generalizations and for the resolution of non-local dependencies. This information source is deployed in recent work either for *pre-ordering* source sentences before they are input to a phrase-based system (Xia and McCord 2004, Collins et al. 2005), or for *re-ordering* the output of translation models by statistical ordering models that access linguistic information on dependencies and part-of-speech (Lin 2004, Ding and Palmer 2005, Quirk et al. 2005).[1]

While these approaches deploy dependency-style grammars for parsing source and/or target text, a utilization of grammar-based generation on the output of dependency-based translation models has not yet been attempted. Instead, simple target language realization models

This is an extended version of a paper for HLT/NAACL 2006.

[1] A notable exception to this kind of approach is Chiang (2005) who introduces syntactic information into phrase-based SMT via hierarchical phrases rather than by external parsing.

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

that can easily be trained to reflect the ordering of the reference translations in the training corpus are preferred. The advantage of such models over grammar-based generation seems to be supported, for example, by Quirk et al. (2005)'s improvements over phrase-based SMT as well as over an SMT system that deploys a grammar-based generator (Menezes and Richardson 2001) on n-gram based automatic evaluation scores (Papineni et al. 2001, Doddington 2002). Another data point, however, is given by Charniak et al. (2003) who show that parsing-based language modeling can improve grammaticality of translations, even if these improvements are not recorded under n-gram based evaluation measures.

In this paper we would like to step away from n-gram based automatic evaluation scores for a moment, and investigate the possible contributions of incorporating a grammar-based generator into a dependency-based SMT system. We present a dependency-based SMT model that integrates the idea of multi-word translation units from phrase-based SMT into a transfer system for dependency structure snippets. The statistical components of our system are modeled on the phrase-based system of Koehn et al. (2003), and component weights are adjusted by minimum error rate training (Och 2003). In contrast to phrase-based SMT and to the above cited dependency-based SMT approaches, our system feeds dependency-structure snippets into a grammar-based generator, and determines target language ordering by applying n-gram and distortion models after grammar-based generation. The goal of this ordering model is thus not foremost to reflect the ordering of the reference translations, but to improve the grammaticality of translations.

Since our system uses standard SMT techniques to learn about correct lexical choice and idiomatic expressions, it allows us to investigate the contribution of grammar-based generation to dependency-based SMT.[2] In an experimental evaluation on the test-set that was used in Koehn et al. (2003), we show that for examples that are in coverage of the grammar-based system, we can achieve state-of-the-art quality on n-gram based evaluation measures. To discern the factors of grammaticality and translational adequacy, we conducted a manual evaluation on 500 in-coverage and 500 out-of-coverage examples. This showed that incorporation of a grammar-based generator into an SMT framework provides improved grammaticality over phrase-based SMT on in-coverage examples. Since in our system it is determinable whether

[2] A comparison of the approaches of Quirk et al. (2005) and Menezes and Richardson (2001) with respect to ordering models is difficult because they differ from each other in their statistical and dependency-tree alignment models.

an example is in-coverage, this opens the possibility for a hybrid system that achieves improved grammaticality at state-of-the-art translation quality.

3.2 Phrase-based SMT

Phrase-based SMT starts with a sentence-aligned bilingual corpus of translations. The words are aligned using a noisy channel model (IBM model 4: Brown et al. 1999). The word alignment is then improved by intersecting alignment matrices for both translation directions and refining the intersection alignment by adding directly adjacent alignment points and alignment points that align previously unaligned words (Och et al. 1999). This produces a many-to-many alignment between words in the source and the target sentences that is more suitable for extracting phrase translations than just the noisy channel model.

Next, phrase translations are extracted by collecting all aligned phrase pairs that are consistent with the improved word alignment. The words of a legal phrase pair are only aligned to each other, and not to words outside the phrase pair. For instance, suppose our corpus contains the following aligned sentences (this example is taken from our experiments on German-to-English translation):

Dafür bin ich zutiefst dankbar.
I have a deep appreciation for that.

Suppose further that the following many-to-many bi-directional word alignment has been created

Dafür{*6 7*} *bin*{*2*} *ich*{*1*} *zutiefst*{*3 4 5*} *dankbar*{*5*}

indicating for example that *Dafür* is aligned with words 6 and 7 of the English sentence (*for* and *that*). From this, the following primitive phrase translations can be extracted:

Dafür → for that
bin → have
ich → I
zutiefst dankbar → a deep appreciation

Note that *zutiefst → a deep appreciation* is not allowed because *appreciation* is also aligned with *dankbar*.

In addition, more complex phrase translations can be extracted which are just combinations of the primitive phrase translations:

bin ich → I have
bin ich zutiefst dankbar → I have a deep appreciation

Dafür bin ich zutiefst dankbar → I have a deep appreciation for that

Note that the "phrases" do not have to correspond to constituents (e.g. *bin ich → I have*). They are just snippets of the original sentences.

Once the phrase translations have been extracted, a sentence in German can be translated into English by non-deterministically applying all of the phrase translations that match on the German side, allowing the English outputs to be rearranged, and then using a statistical model to pick the best English translation. A beam decoder is used to make this process more efficient.

The Pharaoh system is a freely available phrase-based SMT system (Koehn 2004) that is useful as a benchmark system for our work. Its statistical model has eight components. The first two measure the relative frequency of phrase translations in the source-to-target and target-to-source directions. This is simply the number of times that a particular phrase translation appears in a training corpus divided by the number of times either the source phrase appears (for source-to-target) or the target phrase appears (for target-to-source). The counts are not smoothed in any way. This means that long phrase translations usually have a relative frequency of 1.

The second two components measure lexical frequency in the source-to-target and target-to-source directions. Lexical frequency is just the average of the relative alignment frequencies for each word on the source side (for source-to-target) or on the target side (for target-to-source). The lexical frequencies help measure the quality of the phrase translations, especially those that have a relative frequency of 1.

The next component counts the number of phrase translations. In general, fewer phrase translations produce better results because the phrases are longer.

The next two components measure the language model probability and the word count of each translation. The language model probability gives a measure of the likelihood of a particular string of words. By itself, it is biased toward short sentences since they have fewer probabilities multiplied together. The word count component is used to offset this bias.

The last component measures the distortion probability. This is a measure of how far each phrase gets moved from its default position. It is not lexicalized. In general, less movement is better than more movement.

The Pharaoh system works by applying translation rules to snippets of sentences. It is successful in spite of the simplistic linguistic model

because of the sophisticated statistical model. However, the simplistic linguistic model often causes it to produce garbage translations. This led us to wonder whether it was possible to get better translations by applying a similar statistical model to snippets of dependency-based f-structures instead of snippets of strings, thus improving the linguistic model without losing any of the benefits of the statistical model.

3.3 Extracting F-Structure Snippets

Our method for extracting transfer rules for dependency structure snippets operates on the paired sentences of a sentence-aligned bilingual corpus. Similar to phrase-based SMT, our approach starts with an improved word-alignment that is created by intersecting alignment matrices for both translation directions, and refining the intersection alignment by adding directly adjacent alignment points and alignment points that align previously unaligned words (see Och et al. 1999). Next, source and target sentences are parsed using source and target LFG grammars to produce a set of possible f(unctional) dependency structures for each side (see Riezler et al. (2002) for the English grammar and parser, and Butt et al. (2002) and Rohrer and Forst (this volume) for German). The two f-structures that most preserve dependencies are selected for further consideration. Selecting the most similar instead of the most probable f-structures is advantageous for rule induction since it provides for higher coverage with simpler rules.

In the third step, the many-to-many word alignment created in the first step is used to define many-to-many correspondences between the substructures of the f-structures selected in the second step. The parsing process maintains an association between words in the string and particular predicate features in the f-structure, and thus the predicates on the two sides are implicitly linked by virtue of the original word alignment. The word alignment is extended to f-structures by setting into correspondence the f-structure units that immediately contain linked predicates. These f-structure correspondences are the basis for hypothesizing candidate transfer rules.

To illustrate, consider the aligned sentences that we discussed earlier:

Dafür bin ich zutiefst dankbar.
I have a deep appreciation for that.

We use the same many-to-many bi-directional word alignment that the Pharaoh system uses:

Dafür{6 7} bin{2} ich{1} zutiefst{3 4 5} dankbar{5}

This results in the links between the predicates of the source and target f-structures shown in Figure 1.

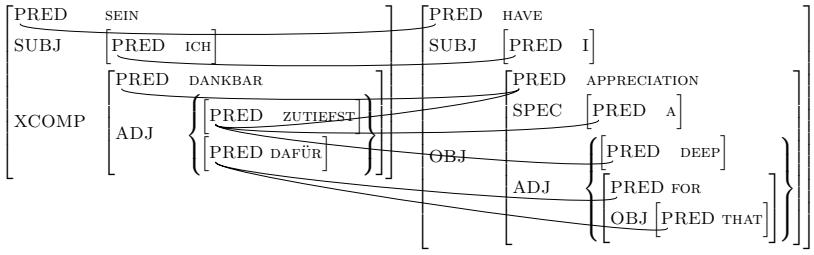


FIGURE 1 F-structure alignment for induction of German-to-English transfer rules.

From these source-target f-structure alignments, transfer rules are extracted in two steps. In the first step, primitive transfer rules are extracted directly from the alignment of f-structure units. These include simple rules for mapping lexical predicates such as:

```
PRED(%X1, ich)   ==>   PRED(%X1, I)
```

and somewhat more complicated rules for mapping local f-structure configurations. For example, the rule shown below is derived from the alignment of the outermost f-structures. It maps any f-structure whose pred is *sein* to an f-structure with pred *have*, and in addition interprets the subj-to-subj link as an indication to map the subject of a source with this predicate into the subject of the target and the xcomp of the source into the object of the target. Features denoting number, person, type, etc. are not shown; variables %X denote f-structure values.

```
PRED(%X1,sein)          PRED(%X1,have)
SUBJ(%X1,%X2)    ==>    SUBJ(%X1,%X2)
XCOMP(%X1,%X3)          OBJ(%X1,%X3)
```

The following rule shows how a single source f-structure can be mapped to a local configuration of several units on the target side, in this case the single f-structure headed by *dafür* into one that corresponds to an English preposition+object f-structure.

```
                              PRED(%X1,for)
PRED(%X1, dafür)   ==>        OBJ(%X1,%X2)
                              PRED(%X2,that)
```

Transfer rules are required to operate only on contiguous units of the f-structure that are consistent with the word alignment. This *transfer contiguity constraint* states that

1. source and target f-structures are each connected;
2. f-structures in the transfer source can only be aligned with f-structures in the transfer target, and vice versa.

This constraint on f-structures is analogous to the constraint on contiguous and alignment-consistent phrases employed in phrase-based SMT. It prevents the extraction of a transfer rule that would translate *dankbar* directly into *appreciation* since *appreciation* is aligned also to *zutiefst* and its f-structure would also have to be included in the transfer. Thus, the primitive transfer rule for these predicates must be:

```
PRED(%X1,dankbar)          PRED(%X1,appreciation)
ADJ(%X1,%X2)         ==>   SPEC(%X1,%X2)
in_set(%X3,%X2)            PRED(%X2,a)
PRED(%X3,zutiefst)         ADJ(%X1,%X3)
                           in_set(%X4,%X3)
                           PRED(%X4,deep)
```

In the second step, rules for more complex mappings are created by combining primitive transfer rules that are adjacent in the source and target f-structures. For instance, we can combine the primitive transfer rule that maps *sein* to *have* with the primitive transfer rule that maps *ich* to *I* to produce the complex transfer rule:

```
PRED(%X1,sein)          PRED(%X1,have)
SUBJ(%X1,%X2)    ==>    SUBJ(%X1,%X2)
PRED(%X2,ich)           PRED(%X2,I)
XCOMP(%X1,%X3)          OBJ(%X1,%X3)
```

In the worst case, there can be an exponential number of combinations of primitive transfer rules, so we allow at most three primitive transfer rules to be combined. This produces $O(n^2)$ transfer rules in the worst case, where n is the number of f-structures in the source.

Other points where linguistic information comes into play is in morphological stemming in f-structures, and in the optional filtering of f-structure phrases based on consistency of linguistic types. For example, the extraction of a phrase-pair that translates *zutiefst dankbar* into *a deep appreciation* is valid in the string-based world, but would be prevented in the f-structure world because of the incompatibility of the types *A* and *N* for adjectival *dankbar* and nominal *appreciation*. Similarly, a transfer rule translating *sein* to *have* could be dispreferred because of a mismatch in the verbal types *V/A* and *V/N*. However, the transfer of *sein zutiefst dankbar* to *have a deep appreciation* is licensed by compatible head types *V*.

3.4 Parsing-Transfer-Generation

We use LFG grammars, producing c(onstituent)-structures (trees) and f(unctional)-structures (attribute value matrices) as output, for parsing source and target text (Riezler et al. 2002, Butt et al. 2002, Rohrer and Forst, this volume). To increase robustness, the standard grammar is augmented with a FRAGMENT grammar. This allows sentences that are outside the scope of the standard grammar to be parsed as well-formed chunks specified by the grammar, with unparsable tokens possibly interspersed. The correct parse is determined by a fewest-chunk method.

Transfer converts source into target f-structures by applying all of the induced transfer rules non-deterministically and in parallel. Each fact in the German f-structure must be transferred by exactly one transfer rule. For robustness a default rule is included that transfers any fact as itself. Similar to parsing, transfer works on a chart. The chart has an edge for each combination of facts that have been transferred. When the chart is complete, the outputs of the transfer rules are unified to make sure they are consistent (for instance, that the transfer rules did not produce two determiners for the same noun). Selection of the most probable transfer output is done by beam-decoding on the transfer chart.

LFG grammars can be used bidirectionally for parsing and generation; thus, the existing English grammar used for parsing the training data can also be used for generation of English translations. For in-coverage examples, the grammar specifies c-structures that differ in the linear precedence of subtrees for a given f-structure and realizes the terminal yield according to morphological rules. In order to guarantee non-empty output for the overall translation system, the generation component has to be fault-tolerant in cases where the transfer system operates on a fragmentary parse, or produces non-valid f-structures from valid input f-structures. For generation from unknown predicates, a default morphology is used to inflect the source stem correctly for English. For generation from unknown structures, a default grammar is used that allows any attribute to be generated in any order as any category, with optimality marks set so as to prefer the standard grammar over the default grammar.

3.5 Statistical Models and Training

The statistical components of our system are modeled on the statistical components of the phrase-based system Pharaoh, described in Koehn et al. (2003) and Koehn (2004). Pharaoh integrates the eight statistical components discussed earlier: relative frequency of phrase translations

in source-to-target and target-to-source direction, lexical weighting in source-to-target and target-to-source direction, phrase count, language model probability, word count, and distortion probability.

Correspondingly, our system computes the following statistics for each translation:

1. log-probability of source-to-target transfer rules, where the probability $r(\texttt{e}|\texttt{f})$ of a rule that transfers source snippet `f` into target snippet `e` is estimated by the relative frequency
$$r(\texttt{e}|\texttt{f}) = \frac{\text{count}(\texttt{f ==> e})}{\sum_{\texttt{e'}} \text{count}(\texttt{f ==> e'})}$$
2. log-probability of target-to-source rules
3. log-probability of lexical translations from source to target snippets, estimated from Viterbi alignments $\hat{a}$ between source word positions $i = 1, \ldots, n$ and target word positions $j = 1, \ldots, m$ for stems f_i and e_j in snippets `f` and `e` with relative word translation frequencies $t(e_j|f_i)$:
$$l(\texttt{e}|\texttt{f}) = \prod_j \frac{1}{|\{i|(i,j) \in \hat{a}\}|} \sum_{(i,j)\in\hat{a}} t(e_j|f_i)$$
4. log-probability of lexical translations from target to source snippets
5. number of transfer rules
6. number of transfer rules with frequency 1
7. number of default transfer rules (translating source features into themselves)
8. log-probability of strings of predicates from root to frontier of target f-structure, estimated from predicate trigrams in English f-structures
9. number of predicates in target f-structure
10. number of constituent movements during generation based on the original order of the head predicates of the constituents (for example, `AP[2] BP[3] CP[1]` counts as two movements since the head predicate of CP moved from the first position to the third position)
11. number of generation repairs
12. log-probability of target string as computed by trigram language model
13. number of words in target string

These statistics are combined into a log-linear model whose parameters are adjusted by minimum error rate training (Och 2003).

3.6 Experimental Evaluation

The setup for our experimental comparison is German-to-English translation on the Europarl[3] parallel data set. For quick experimental turnaround we restricted our attention to sentences with 5 to 15 words, resulting in a training set of 163,141 sentences and a development set of 1,967 sentences. Final results are reported on the test set of 1,755 sentences of length 5-15 that was used in Koehn et al. (2003). To extract transfer rules, an improved bidirectional word alignment was created for the training data from the word alignment of IBM model 4 as implemented by GIZA++ (Och et al. 1999). Training sentences were parsed using German and English LFG grammars (Riezler et al. 2002, Butt et al. 2002). The grammars obtain 100% coverage on unseen data. 80% receive full parses; 20% receive FRAGMENT parses. Around 700,000 transfer rules were extracted from f-structures pairs chosen according to a dependency similarity measure. For language modeling, we used the trigram model of Stolcke (2002).

When applied to translating unseen text, the system operates on n-best lists of parses, transferred f-structures, and generated strings. For minimum-error-rate training on the development set, and for translating the test set, we considered 1 German parse for each source sentence, 10 transferred f-structures for each source parse, and 1,000 generated strings for each transferred f-structure. Selection of most probable translations proceeds in two steps: First, the most probable transferred f-structure is computed by a beam search on the transfer chart using the first 10 features described above. These features include tests on source and target f-structure snippets related via transfer rules (features 1-7) as well as language model and distortion features on the target c- and f-structures (features 8-10). In our experiments, the beam size was set to 20 hypotheses. The second step is based on features 11-13, which are computed on the strings that were generated from the selected n-best f-structures.

We compared our system to IBM model 4 as produced by GIZA++ (Och et al. 1999) and a phrase-based SMT model as provided by Pharaoh (Koehn 2004). The same improved word alignment matrix and the same training data were used for phrase-extraction for phrase-based SMT as well as for transfer-rule extraction for LFG-based SMT. Minimum-error-rate training was done using Koehn's implementation of Och (2003)'s minimum-error-rate model. To train the weights for phrase-based SMT, we used the first 500 sentences of the development set; the weights of the LFG-based translator were adjusted on the 750

[3] http://people.csail.mit.edu/koehn/publications/europarl/

TABLE 1 NIST scores on test set for IBM model 4 (M4), phrase-based SMT (P), and the LFG-based SMT (LFG) on the full test set and on in-coverage examples for LFG. Results in the same row that are not statistically significant from each other are marked with a ∗.

	M4	LFG	P
in-coverage	5.13	*5.82	*5.99
full test set	*5.57	*5.62	6.40

TABLE 2 Preference ratings of two human judges for translations of phrase-based SMT (P) or LFG-based SMT (LFG) under criteria of fluency/grammaticality and translational/semantic adequacy on 500 in-coverage examples. Ratings by judge 1 are shown in rows, for judge 2 in columns. Agreed-on examples are shown in boldface in the diagonals.

	adequacy			grammaticality		
j1\j2	P	LFG	equal	P	LFG	equal
P	**48**	8	7	**36**	2	9
LFG	10	**105**	18	6	**113**	17
equal	53	60	**192**	51	44	**223**

sentences that were in coverage of our grammars.

For automatic evaluation, we use the NIST metric (Doddington 2002) combined with the approximate randomization test (Noreen 1989), providing the desired combination of a sensitive evaluation metric and an accurate significance test (see Riezler and Maxwell 2005). In order to avoid a random assessment of statistical significance in our three-fold pairwise comparison, we reduced the per-comparison significance level to .01 so as to achieve a standard experimentwise significance level of .05 (see Cohen 1995). Table 1 shows results for IBM model 4, phrase-based SMT, and LFG-based SMT, where examples that are in coverage of the LFG-based systems are evaluated separately. Out of the 1,755 sentences of the test set, 44% were in coverage of the LFG-grammars; for 51% the system had to resort to the FRAGMENT technique for parsing and/or repair techniques in generation; in 5% of the cases our system timed out. Since our grammars are not set up with punctuation in mind, punctuation is ignored in all evaluations reported below. For in-coverage examples, the difference between NIST scores for the LFG system and the phrase-based system is statistically not significant. On the full set of test examples, the suboptimal quality on out-of-coverage examples overwhelms the quality achieved on

TABLE 3 Preference ratings of two human judges for translations of phrase-based SMT (P) or LFG-based SMT (LFG) under criteria of fluency/grammaticality and translational/semantic adequacy on 500 out-of-coverage examples. Ratings by judge 1 are shown in rows, for judge 2 in columns. Agreed-on examples are shown in boldface in the diagonals.

	adequacy			grammaticality		
j1\j2	P	LFG	equal	P	LFG	equal
P	**156**	1	19	**121**	1	20
LFG	6	**53**	7	0	**23**	11
equal	69	38	**152**	54	21	**250**

in-coverage examples, resulting in a statistically not significant result difference in NIST scores between the LFG system and IBM model 4.

In order to investigate further the quality of in-coverage examples, we randomly selected 500 examples that were in coverage of the grammar-based generator for a manual evaluation. Two independent human judges were presented with the source sentence and the output of the phrase-based and LFG-based systems in a blind test. This was achieved by displaying the system outputs in random order. The judges were asked to indicate a preference for one system translation over the other, or whether they thought them to be of equal quality. These questions had to be answered separately under the criteria of grammaticality/fluency and translational/semantic adequacy. As shown in Table 2, both judges express a preference for the LFG system over the phrase-based system for both adequacy and grammaticality. If we look only at sentences where judges agree, we see a net improvement on translational adequacy of 57 sentences, which is an improvement of 11.4% over the 500 sentences. If this were part of a hybrid system, this would amount to a 5% overall improvement in translational adequacy. Similarly we see a net improvement on grammaticality of 77 sentences, which is an improvement of 15.4% over the 500 sentences or 6.7% overall in a hybrid system. Result differences on agreed-on ratings are statistically significant, where significance was assessed by approximate randomization via stratified shuffling of the preferences between the systems (Noreen 1989). Examples from the manual evaluation are shown in the appendix.

Along the same lines, a further manual evaluation was conducted on 500 randomly selected examples that were out of coverage of the LFG-based grammars. The two judges agreed on a preference for the phrase-based system in 156 cases and for the LFG-based system in 53

cases under the measure of translational adequacy, and on a preference for the phrase-based system in 121 cases and for the LFG-based system in 23 cases under the measure of grammaticality. Across the combined set of 1,000 in-coverage and out-of-coverage sentences, this resulted in an agreed-on preference for the phrase-based system in 204 cases and for the LFG-based system in 158 cases under the measure of translational adequacy. Under the grammaticality measure the phrase-based system was preferred by both judges in 157 cases and the LFG-based system in 136 cases.

3.7 Discussion

The evaluation of the LFG-based translator presented above shows promising results for examples that are in coverage of the employed LFG grammars. However, a back-off to robustness techniques in parsing and/or generation results in a considerable loss in translation quality. The high percentage of examples that fall out of coverage of the LFG-based system can partially be explained by the accumulation of errors in parsing the training data where source and target language parser each produce FRAGMENT parses in 20% of the cases. Together with errors in rule extraction, this results in a large number of ill-formed transfer rules that force the generator to back-off to robustness techniques. In applying the parse-transfer-generation pipeline to translating unseen text, parsing errors can cause erroneous transfer, which can result in generation errors. Similar effects can be observed for errors in translating in-coverage examples. Here disambiguation errors in parsing and transfer propagate through the system, producing suboptimal translations. An error analysis on 100 suboptimal in-coverage examples from the development set showed that 69 suboptimal translations were due to transfer errors, 10 of which were due to errors in parsing.

The discrepancy between NIST scores and manual preference rankings can be explained by the suboptimal integration of transfer and generation in our system, making it infeasible to work with large n-best lists in training and application. Moreover, despite our use of minimum-error-rate training and n-gram language models, our system cannot be adjusted to maximize n-gram scores on reference translation in the same way as phrase-based systems since statistical ordering models are employed in our framework *after* grammar-based generation, thus giving preference to grammaticality over similarity to reference translations.

3.8 Conclusion

We presented an SMT model that marries phrase-based SMT with traditional grammar-based MT by incorporating a grammar-based generator into a dependency-based SMT system. Under the NIST measure, we achieve results in the range of the state-of-the-art phrase-based system of Koehn et al. (2003) for in-coverage examples of the LFG-based system. A manual evaluation of a large set of such examples shows that on in-coverage examples our system achieves significant improvements in grammaticality and also translational adequacy over the phrase-based system. Fortunately, it is determinable when our system is in-coverage, which opens the possibility for a hybrid system that achieves improved grammaticality at state-of-the-art translation quality. Future work thus will concentrate on improvements of in-coverage translations, e.g. by stochastic generation. Furthermore, we intend to apply our system to other language pairs and larger data sets.

Acknowledgments

This paper is dedicated to Ron Kaplan, who gave the first author the great opportunity to work with LFG grammars and the people who invented them by bringing him onto his team five years ago, and who has been a long-time collaborator of the second author. Ron always believed in the power of the combination of deep linguistic knowledge and broad statistical methods. This view was justified in several papers on statistical parsing which we co-authored over the last years. This paper is an attempt to go a step further and show that deep LFG grammars can be deployed to build a hybrid statistical machine translation system that provides improved translational adequacy and grammaticality. Besides the fact that Ron's intuitions always proved to be right, it was always great fun to work with him.

We would also like to thank Sabine Blum for her help with the manual evaluation which was a crucial ingredient of this paper.

Appendix: Examples from manual evaluation

Examples from manual evaluation: Preference for LFG-based system (LFG) over phrase-based system (P) under both adequacy and grammaticality (the first five), preference of phrase-based system over LFG (the second five), together with source (src) sentences and human reference (ref) translations. All ratings are agreed on by both judges.

src: in diesem fall werde ich meine verantwortung wahrnehmen
ref: then i will exercise my responsibility

LFG: in this case i accept my responsibility
P: in this case i shall my responsibilities

src: die politische stabilität hängt ab von der besserung der lebensbedingungen
ref: political stability depends upon the improvement of living conditions
LFG: the political stability hinges on the recovery the conditions
P: the political stability is rejects the recovery of the living conditions

src: und schließlich muß dieser agentur eine kritische haltung gegenüber der kommission selbst erlaubt sein
ref: moreover the agency must be able to criticise the commission itself
LFG: and even to the commission a critical stance must finally be allowed this agency
P: finally this is a critical attitude towards the commission itself to be agency

src: nach der ratifizierung werden co2 emissionen ihren preis haben
ref: after ratification co2 emission will have a price tag
LFG: carbon dioxide emissions have its price following the ratification
P: after the ratification co2 emissions are a price

src: die lebensmittel müssen die sichere ernährung des menschen gewährleisten
ref: man's food must be safe to eat
LFG: food must guarantee the safe nutrition of the people
P: the people of the nutrition safe food must guarantee

src: was wir morgen beschließen werden ist letztlich material für das vermittlungsverfahren
ref: whatever we agree tomorrow will ultimately have to go into the conciliation procedure
LFG: one tomorrow we approved what is ultimately material for the conciliation procedure
P: what we decide tomorrow is ultimately material for the conciliation procedure

src: die verwaltung muß künftig schneller reagieren können
ref: in future the administration must be able to react more quickly
LFG: more in future the administration must be able to react
P: the administration must be able to react more quickly

src: das ist jetzt über 40 jahre her
ref: that was over 40 years ago
LFG: on 40 years ago it is now
P: that is now over 40 years ago

src: das ist schon eine seltsame vorstellung von gleichheit
ref: a strange notion of equality
LFG: equality that is even a strange idea
P: this is already a strange idea of equality

src: frau präsidentin ich beglückwünsche herrn nicholson zu seinem ausgezeichneten bericht
ref: madam president i congratulate mr nicholson on his excellent report
LFG: madam president i congratulate mister nicholson on his report excellented
P: madam president i congratulate mr nicholson for his excellent report

References

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1999. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19(2):263–311.

Butt, Miriam, Helge Dyvik, Tracy H. King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar project. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), Workshop on Grammar Engineering and Evaluation*, pages 1–7. Taipei, ROC.

Charniak, Eugene, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for statistical machine translation. In *Proceedings of the 9th Machine Translation Summit (MTS IX)*, pages 40–46. New Orleans, LA.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 236–270. Ann Arbor, MI.

Cohen, Paul R. 1995. *Empirical Methods for Artificial Intelligence*. Cambridge, MA: The MIT Press.

Collins, Michael, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540. Ann Arbor, MI.

Ding, Yuan and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 541–548. Ann Arbor, MI.

Doddington, George. 2002. Automatic evaluation of machine translation quality using N-gram co-occurrence statistics. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 128–132. San Diego, CA.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, pages 127–133. Edmonton, Canada.

Koehn, Philipp. 2004. PHARAOH. A beam search decoder for phrase-based statistical machine translation models. User manual. Tech. rep., USC Information Sciences Institute, Marina del Rey, CA.

Lin, Dekang. 2004. A path-based transfer model for statistical machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*, pages 625–630. Geneva, Switzerland.

Menezes, Arul and Stephen D. Richardson. 2001. A best-first alignment algorithm for automatic extraction of transfer-mappings from bilingual corpora. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01), Workshop on Data-Driven Machine Translation*, pages 39–46. Toulouse, France.

Noreen, Eric W. 1989. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. New York, NY: Wiley.

Och, Franz Josef, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proceedings of the 1999 Conference on Empirical Methods in Natural Language Processing (EMNLP'99)*, pages 20–28. College Park, MD.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, pages 160–167. Edmonton, Canada.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. Tech. Rep. IBM Research Division Technical Report, RC22176 (W0190-022), Yorktown Heights, NY.

Quirk, Chris, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279. Ann Arbor, MI.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 271–278. Philadelphia, PA.

Riezler, Stefan and John T. Maxwell, III. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 57–64. Ann Arbor, MI.

Stolcke, Andreas. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*. Denver, CO.

Xia, Fei and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*, pages 508–514. Geneva, Switzerland.

4

On Some Formal Properties of LFG Generation

Jürgen Wedekind

4.1 Introduction

This paper gives a brief survey of the formal properties of LFG generation that we examined in the past. Most of the results presented here were achieved in close collaboration with Ron, and this is perhaps already enough reason for presenting this survey in this volume. But this paper is also intended to encourage and stimulate further research on some challenging open problems in LFG generation, since an empirically adequate subclass of LFG grammars that does not suffer from the negative results we provided in earlier work has still not yet been agreed on, and the possibilities that the positive results open up have not been fully exploited. This applies (at least partially) to other higher-order grammatical formalisms, like PATR or HPSG (Shieber et al. 1983, Pollard and Sag 1994), which share the property of assigning feature structures to sentences.

We begin with the preliminaries to make the problems more precise. Similar to grammars of other unification-based formalisms, an LFG grammar G assigns to every string in its language at least one f(eature) structure (Kaplan and Bresnan 1982). The f-structure encodes its morphosyntactic features, its predicate-argument structure, and in some cases also a complete description of its semantic interpretation (commonly referred to as its semantic representation). Each LFG grammar G thus defines a binary derivation relation Δ_G between terminal strings s and f-structures F as given in (1).

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

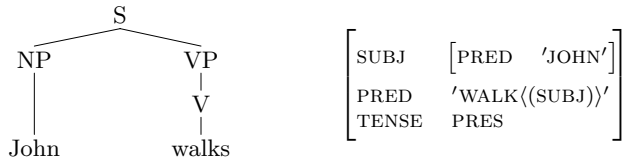


FIGURE 1 The components of an LFG representation.

(1) $\Delta_G(s, F)$ iff G assigns to the string s the f-structure F

Unlike HPSG, but similar to PATR grammars, an LFG grammar G establishes the relation between a terminal string s and an f-structure F on the basis of its annotated phrase structure rules. These create, in addition to the f-structure, at least one valid c-structure for s representing its surface constituency configurations. Thus, an LFG representation consists of a valid c-structure c for s and its f-structure F. A simple example is depicted in Figure 1. This representation is derivable with the grammar in (2).

(2) a. S → NP VP
(↑ SUBJ) = ↓ ↑ = ↓

b. VP → V
↑ = ↓

c. NP → John
(↑ PRED) = 'JOHN'

d. V → walks
(↑ PRED) = 'WALK⟨(SUBJ)⟩'
(↑ TENSE) = PRES

A representation consisting of a phrase structure tree c with terminal string s and an f-structure F is derivable with G iff both c and F are licensed by the rules of G. The phrase structure tree c is, of course, licensed or well-formed if the label of the root node is S and if we can assign to each nonterminal node n with label A and daughters $n_1..n_m$ with labels $X_1..X_m$, respectively, a rule r of G with context-free skeleton $A \rightarrow X_1..X_m$. Such an assignment is called *licensing rule-mapping* in the following, and notated by ρ. If we assume that the root node of the tree in Figure 1 is *root*, its left daughter n_1, its right daughter n_2, and that n_2's daughter is n_3, then G licenses this tree because of the rule-mapping ρ given in (3).

(3) $\rho(root) = (2a)$, $\rho(n_1) = (2c)$, $\rho(n_2) = (2b)$, $\rho(n_3) = (2d)$

The f-structure is licensed if it is a minimal solution of the f-description induced by the annotations of all licensing rules. If in the following we abbreviate $\rho(n)$ by ρ_n then the f-description FD for a given licensing rule-mapping ρ is the union of the instantiated descriptions of all justifying rules: $FD = \bigcup_{n \in Dom(\rho)} Inst(\rho_n, (n, dts(n)))$, where $Inst(\rho_n, (n, dts(n)))$ is the description that we obtain from the annotations of ρ_n by substituting the $\uparrow$ symbol in all annotations by n, and the $\downarrow$ symbol in the annotations of all j $(j = 1, ..m)$ daughters $dts(n) = n_1..n_m$ by n_j.[1] For the licensing rule-mapping in (3) we thus obtain the f-description in (4).

(4) $$\left\{ \begin{array}{c} (root\ \textsc{subj}) = n_1, root = n_2, n_2 = n_3, \\ (n_1\ \textsc{pred}) = {'\textsc{john}'}, \\ (n_3\ \textsc{pred}) = {'\textsc{walk}\langle(\textsc{subj})\rangle'}, (n_3\ \textsc{tense}) = \textsc{pres} \end{array} \right\}$$

An f-description is solvable if it is consistent and if no atomic feature value corefers with some other distinct constant (atomic feature value or node) or a complex term. If an f-description FD is solvable (and there are only simple equations involved) it has an (up to isomorphism unique) minimal model. Such a model consists of a universe and an interpretation function that assigns unary (partial) functions to the attributes (e.g., SUBJ, PRED) and elements of the universe to the atomic feature values (e.g., PRES, JOHN), as well as to the nodes in the description. However, such a minimal model M cannot directly be taken to be the f-structure. This is because it still interprets the nodes from which an f-structure is assumed to abstract. To obtain the f-structure, we thus have to remove from M the nodes and their interpretation. We accomplish this by restricting it to the language L_{AV} consisting of the attributes A and atomic feature values V. This restriction is denoted by $M|L_{AV}$. Hence, the f-structure F of an LFG representation is licensed (or derivable) if $M|L_{AV}$ is isomorphic to F.[2]

We can now turn to some of the generation problems that arise from LFG's derivation relation. We concentrate here on LFG grammars with only equational annotations. This is sufficient for the negative results, since they also hold for LFG grammars that make use of a wider range of formal devices. For the positive results, on the other hand, it is

[1] Without loss of generality for the results presented here, we instantiate the $\uparrow$ and $\downarrow$ symbols with the nodes themselves instead of additional f-structure variables (Kaplan and Bresnan 1982) or even more complex terms involving the ϕ projection (Kaplan 1995).

[2] To be mathematically precise, an f-structure has to be regarded as an equivalence class of isomorphic structures. However, for all practical purposes we can work on representatives, as long as the independence of the representatives chosen can be shown.

in some cases not immediately obvious whether they extend to these more elaborate grammars. In some cases they do extend, but it would require a longer and more technical presentation than we can provide in this paper. Wedekind and Kaplan (2006) give a more comprehensive treatment of the latter cases.

For our present purposes, an LFG grammar G (over L_{AV}) is a 4-tuple (N, T, S, R) where N is a finite set of nonterminal categories, T is a finite set of terminal symbols, $\mathrm{S} \in N$ is the root category, and R is a finite set of annotated rules of the form

$$\begin{array}{lcl} A & \rightarrow & X_1 \; .. \; X_m \\ & & D_1 \quad D_m \end{array}$$

with $A \in N$ and $X_1..X_m \in (N \cup T)^*$. Each annotated description D_j $(j = 1, .., m)$ is a (possibly empty) finite set of equalities between terms of the form $(\uparrow \sigma)$, $(\downarrow \sigma)$ or v, where v is a value of L_{AV} and σ is a possibly empty sequence of attributes of L_{AV}.

4.2 Generation from Underspecified F-Structures

If we consider only single f-structures then generation from arbitrarily underspecified input f-structures is clearly the most general problem. An instance of this problem is, for example, generation from semantic representations, at least when they are contained in the f-structures. Semantic representations are then encoded in the f-structures as the value of a specific attribute, like SEM or CONT, or they are the value of a projection (Halvorsen and Kaplan 1988), which is formally reconstructable by such a distinguished attribute.[3] Since the f-structures assigned to the sentences are always subsumed by the semantic representations they contain, a generator for a grammar G has to compute for any input representation F' a sentence s with a feature structure F that is subsumed by the input (notated by $F' \sqsubseteq F$). The formal problem we are concerned with is thus an instance of the problem of whether we can decide (5) for any given input F'.

(5) $\{s \in T^* \mid \exists F(F' \sqsubseteq F \wedge \Delta_G(s, F))\} = \emptyset$

The undecidability of this problem has been well-known for many years. Dymetman (1991) provided a proof for definite clause grammars using a reduction of Hilbert's Tenth Problem, van Noord (1993) proved it for PATR grammars by reduction of Post's Correspondence Problem, and we showed it for LFG and PATR grammars using the

[3] If the input to generation consists of a projected (sub)structure or an f-structure that a projection maps to some other representation, we have, formally, an instance of the same problem.

emptiness problem of the intersection of arbitrary context-free languages (Wedekind 1999). In Wedekind (1999) we observed already a close relationship between the emptiness problem of the languages of lexical-functional and other unification-based grammars ($L(G) = \emptyset$) and the generation problem in (5). We know now that this is because the undecidability of the emptiness problem of $L(G)$ trivially implies the undecidability of the generation problem (5). Therefore, it might not be just a simple coincidence that the same reductions were used to prove the undecidability of the emptiness problem for lexical-functional languages.[4] However, the close relation between these two problems allows us to sketch here an even simpler proof by directly reducing the emptiness problem of lexical-functional languages to the generation problem (5).

Thus, let $G' = (N', T', \mathrm{S}', R')$ be an arbitrary LFG grammar. With a new start symbol S and a new feature SEM we construct an LFG grammar $G = (N, T, \mathrm{S}, R)$ by $N = \{\mathrm{S}\} \cup N'$, $T = T'$, and

$$R = \left\{ \begin{array}{ccc} \mathrm{S} & \rightarrow & \mathrm{S}' \\ & & \uparrow = \downarrow \\ & & (\uparrow \textsc{sem}) = 1 \end{array} \right\} \cup R'.$$

By this construction, $\{s \in T^* \mid \exists F([\textsc{sem}\ 1] \sqsubseteq F \wedge \Delta_G(s, F))\} = \emptyset$ if and only if $L(G') = \emptyset$. Thus, LFG's emptiness problem, which is in general undecidable, reduces to LFG's generation problem for underspecified input structures, and this generation problem must therefore be undecidable as well.

Theorem 1 *For LFG grammars G and functional structures F' it is in general not decidable whether there is an f-structure F and a terminal string s with $F' \sqsubseteq F$ and $\Delta_G(s, F)$.*

For the proof of Theorem 1 we assumed the input structures to be structurally unrelated to the f-structures they subsume, and this seems very unrealistic from a cognitive point of view. As we pointed out in Wedekind (1999), it seems much more plausible that natural language grammars belong to a specific subclass of LFG grammars that satisfy conditions which bound the size of an f-structure assigned to a string by the size of its subsuming semantic representation. If we refer to all these conditions together as the *off-line generability restriction* to express the affinity with the corresponding off-line parsability restriction that guarantees the decidability of LFG's parsing (recognition) problem (Kaplan

[4] Hilbert's Tenth Problem was used by Roach (1983), Nishino (1991) reduced Post's Correspondence Problem, and in Wedekind (1999) we used the emptiness problem of the intersection of arbitrary context-free languages to establish the same result.

and Bresnan 1982), then such a condition would force the f-structures of the sentences realizing a given semantic representation to be included in a finite and computable set of structurally related structures. However, it has still not yet been agreed on how this off-line generability restriction has to be defined so that it establishes a structural relation between a semantic representation and its functional extensions which is both cognitively plausible and effectively computable.

The few existing proposals, which are unfortunately neither published nor readily available to a wide audience (e.g., Prescher 1997), all exploit LFG's semantic forms, feature values that are uniquely instantiated for each instance of their use. To enforce that the length of a derivation is bounded by the size of the input structure, XLE (Crouch et al. 2006), for example, currently uses an off-line generability restriction that consists of the following two conditions: (i) every category in a derivation must be associated with a functional unit in the input, and (ii) if a mother-daughter category pair occurs twice on a path of the c-structure of a derivation, the licensing rule-mapping assigns to these category pairs the same annotations, and the licensed node pairs labeled with these category pairs are associated with the same functional units of the input, then at least one semantic form must be introduced somewhere in between the two instances.

XLE's off-line generability condition certainly ensures that there is only a finite number of possible trees for any (underspecified) input, but from a linguistic point of view this condition might be considered too strong. For example, for LFG grammars which treat auxiliaries as main verbs that introduce their own predicate, derive complex analytical tense forms by means of recursive VP rules, and produce flat semantic representations (without semantic forms corresponding to the predicates of the auxiliaries), it would block the generation of well-formed strings when the input is a semantic representation for a sentence with a complex tense form. Similar problems arise when grammars codescribe f-structures for sentences in different languages that stand in a translation relation to one another. If the source language realizes complex tenses periphrastically (e.g., English), but the target language morphologically (e.g., French), and auxiliaries are treated as main verbs, then for each VP recursion there would be a semantic form in the source, but not in the target structure. For a target f-structure of a sentence with a morphologically realized complex tense as input, generation with such a grammar would thus violate XLE's off-line generability condition. This would be unfortunate, since the components of such an LFG-based machine translation system would work only from the source to the target language but not vice versa.

We do not want to judge here whether or not these grammars are linguistically motivated, we merely want to notice that there are still some controversies on LFG's syntax-semantic interface that require further research to resolve.

4.3 Generation from Fully Specified F-Structures

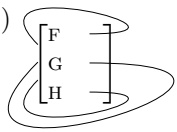
In the general case above we permitted the f-structure of the generated sentence to be subsumed by the input and thus arbitrarily larger structures to be hypothesized by the generator. This is excluded if we assume the input to be fully specified. In such a situation, a generator for G should produce for an input f-structure F the set $Gen_G(F)$ consisting of all terminal strings that are related to F by the grammar:

(6) $Gen_G(F) = \{s \in T^* \mid \Delta_G(s, F)\}$.

However, the set of terminal strings that an LFG grammar relates to an f-structure can still be quite expressive. This is illustrated by the LFG grammar $G = (\{\mathrm{S}, \mathrm{A}, \mathrm{B}\}, \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}, \mathrm{S}, R)$ with the annotated rules R in (7).[5]

(7)
$$\begin{array}{lcl}
\mathrm{S} & \rightarrow & \mathrm{a}\ \ \mathrm{A}\ \ \mathrm{c} \\
 & & (\uparrow \textsc{f}) = \uparrow \\
 & & (\uparrow \textsc{g}) = \downarrow \\
 & & (\uparrow \textsc{f}) = (\downarrow \textsc{f}) \\
\mathrm{A} & \rightarrow & \mathrm{a}\ \ \mathrm{A}\ \ \mathrm{c} \\
 & & (\uparrow \textsc{g}) = \downarrow \\
 & & (\uparrow \textsc{f}) = (\downarrow \textsc{f}) \\
\mathrm{A} & \rightarrow & \mathrm{B} \\
 & & (\uparrow \textsc{f}) = (\uparrow \textsc{h}) \\
 & & (\uparrow \textsc{h}\ \textsc{g}) = \downarrow \\
\mathrm{B} & \rightarrow & \mathrm{b}\ \ \mathrm{B} \\
 & & (\uparrow \textsc{g}) = \downarrow \\
\mathrm{B} & \rightarrow & \mathrm{b} \\
 & & (\uparrow \textsc{h}\ \textsc{f}) = (\uparrow \textsc{h}\ \textsc{g})
\end{array}$$

This grammar derives with the input f-structure in (8)

(8)


the set of strings $\{\mathrm{a}^n\mathrm{b}^n\mathrm{c}^n \mid 1 \leq n\}$, a language that is known not to be context-free. This shows that LFG generation from fully specified

[5] Another example of such a grammar can be found in Wedekind and Kaplan (2006).

f-structures has greater generative power than the class of context-free grammars. Although we do not know yet exactly which class of languages an arbitrary LFG grammar generates for arbitrary input f-structures, we know that LFG generation is decidable for these inputs.

Theorem 2 *For any LFG grammar G, f-structure F, and terminal string s, it is decidable whether s belongs to $Gen_G(F)$.*

Wedekind (1995) provides a proof where this results directly from the following shrinking lemma.

Lemma 3 *Let G be an LFG grammar and F be an f-structure. Then there is a constant l, only depending on G and F, such that for every derivation of s with F of length greater than l there is a derivation of s' with F of length less than l.*

To determine whether a terminal string belongs to $Gen_G(F)$, one thus has only to consider a finite set of 'short' derivations. Since the proof of the shrinking lemma in Wedekind (1995) is rather complicated and the lemma itself not particularly instructive with respect to the generative power of LFG generation, we dispense here with a more detailed description. However, we must nevertheless notice that it is still a research topic to determine how the family of languages that LFG grammars are able to relate to f-structures fits into the system of the well-known language families. Moreover, in case the generative power of LFG generation actually coincides with the generative capacity of one of the well-known families of grammars then a constructive proof of such a result could considerably simplify the arguments presented in Wedekind (1995).

The solution to these problems might be considered a purely academic exercise, since from a cognitive perspective it is again not very plausible that an f-structure can be realized by sentences which together form a non-context-free language. We will see below that such languages are only generable when the input structures contain cycles. Thus, one might take this as a formal argument to support the hypothesis that acyclic structures are in fact the only f-structures which are motivated for linguistic analysis. By entirely disregarding cyclic structures as inputs to generation, we could then at this stage dispense with attempts to restrict the expressiveness by additional reinforcements of the off-line generability condition under discussion. Thus, let us consider generation from acyclic f-structures more thoroughly.

4.4 Acyclic Input F-Structures

If the input to generation is assumed to consist only of acyclic f-structures then the set of strings that an LFG grammar relates to a par-

ticular input is a context-free language. Kaplan and Wedekind (2000) and Wedekind and Kaplan (2006) prove this in a constructive way by providing an algorithm that creates, for an arbitrary LFG grammar G and an arbitrary acyclic f-structure F, a context-free grammar G_F that describes exactly the set of strings that the given LFG grammar relates to the input.

The proof is based on the key insight that a finite set of canonical terms is sufficient to maintain the functional discriminations made in every partial f-description of every derivation for an acyclic F, even if there is an infinite number of valid c-structures for F and thus no fixed upper bound on the number of nodes that may occur in the f-descriptions for F. As a simple example consider the LFG grammar G in (9).

(9)
$$\begin{array}{lcl}
\mathrm{S} \rightarrow \mathrm{a}\ \mathrm{A}\ \mathrm{b} & & \\
\quad (\uparrow \textsc{h}) = \downarrow & & \\
\mathrm{A} \rightarrow \mathrm{a}\ \mathrm{A}\ \mathrm{b} & \qquad & \mathrm{A} \rightarrow \mathrm{A}\ \mathrm{c} \\
\quad \uparrow = \downarrow & & \quad (\downarrow \textsc{h}') = \uparrow \\
\mathrm{A} \rightarrow \mathrm{a}\ \mathrm{b} & & \mathrm{A} \rightarrow \mathrm{c} \\
\quad (\uparrow \textsc{h}) = \textsc{v} & & \quad (\uparrow \textsc{h}') = \textsc{v}'
\end{array}$$

This grammar generates for the input f-structure F in (10)

(10) $\left[\textsc{h}\ \left[\textsc{h}\ \textsc{v}\right]\right]$

the infinite context-free language $\{\mathrm{a}^n\mathrm{b}^n \mid 1 < n\}$. Since the size of the c-structures that G provides for F is not finitely bounded and all non-terminals are annotated, there is also no fixed finite upper bound on the number of nodes that occur in the f-descriptions for F.

However, for every input f-structure F we can identify a finite set of canonical terms $\mathcal{T}_F$ that allows us to construct from the f-description of an arbitrary derivation of F an equivalent description that is free of the nodes other than *root*. For the definition of the set of canonical terms $\mathcal{T}_F$ for F we associate a distinct canonical constant a_{a} with each element a of the universe of F that is not denoted by an atomic feature value. We use these new constants as pointers to the substructures they are associated with. On the basis of the structure we obtain from F by adding the pointers with their intended interpretation, we then define the set $\mathcal{T}_F$ as the set that consists of all denoting terms that do not corefer with an atomic feature value, the terms that we obtain from them by substituting *root* for the constant symbols, plus the dummy constant $\perp$. If we assume that the universe of the f-structure in (10) contains the elements a and b, and a is the root and b its H value, then the set of canonical terms for (10) is the finite set given in (11).

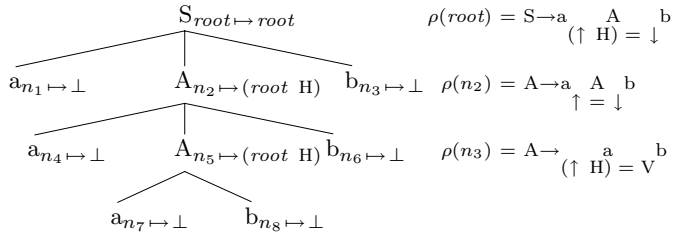


FIGURE 2 A sample derivation together with its ψ substitution.

(11) $\{a_{\mathtt{a}}, a_{\mathtt{b}}, (a_{\mathtt{a}}\ \mathrm{H})\} \cup \{root, (root\ \mathrm{H})\} \cup \{\perp\}$

On the basis of $\mathcal{T}_F$ we can then reduce the f-description *FD* of a given derivation c and ρ of s with F in G to an equivalent node-free description. In general, two descriptions D and D' are said to be *equivalent* iff the restrictions of their minimal models to L_{AV} are isomorphic. For the reduction we make use of the fact that some daughters are definable in terms of their mother. The definability relation our reduction relies on is defined as follows.

Let r be an m-ary LFG rule, t be a term and $a_1..a_m$ be a sequence of constants of length m each of them not occurring in t. A constant a_j is *m(other)-definable* in $Inst(r, (t, a_1..a_m))$ iff there is a (possibly empty) sequence of attributes σ such that $Inst(r, (t, a_1..a_m)) \vdash a_j = (t\ \sigma)$.[6]

We then define a substitution ψ from the nodes of c into $\mathcal{T}_F$ by induction on the depth of c (i.e. top-down) such that $\psi(root) = root$. And for any other node n_j with mother n we define ψ such that it satisfies the following conditions. If n_j does not occur in *FD* then $\psi(n_j) = \perp$. If n_j occurs in *FD* and n_j is m-definable in $Inst(\rho_n, (\psi(n), dts(n)))$ then ψ assigns to n_j some term $t_j \in \mathcal{T}_F$ that defines n_j in $Inst(\rho_n, (\psi(n), dts(n)))$ in terms of $\psi(n)$. Otherwise, ψ maps n_j to the pointer (canonical constant) that is associated with the functional unit of F that the node n_j corresponds to. Figure 2 illustrates the ψ substitution for the derivation of 'aaabbb' with f-structure (10) of the grammar in (9). The substitution is indicated by assigning the canonical terms to the nodes.

The substitution ψ produces from the original f-description *FD* a description $FD[\psi]$ that is free of nodes except for *root*. This description is equivalent to the original, since a given constant is substituted for two nodes only if those two nodes denote the same unit of F, and definable nodes can always be replaced by their defining terms without

[6]Here, we assume a more general definition of the *Inst* function which allows arbitrary terms to be substituted for the ↑/↓ metavariables.

a change of the model's underlying f-structure. The ψ mapping for the derivation in Figure 2, for example, produces from the derived f-description depicted in (12a) the equivalent description in (12b).

(12) a. $\left\{\begin{array}{c}(\mathit{root}\ \textsc{h}) = n_2,\\ n_2 = n_5,\\ (n_5\ \textsc{h}) = \textsc{v}\end{array}\right\}$

b. $\left\{\begin{array}{c}(\mathit{root}\ \textsc{h}) = (\mathit{root}\ \textsc{h}),\\ (\mathit{root}\ \textsc{h}\ \textsc{h}) = \textsc{v}\end{array}\right\}$

If we associate with each node n the set of all instantiated rules IR that license the subderivation from n, except that the licensed nodes are replaced by their ψ values

$$\{(\rho_{n'}, (n', \mathit{dts}(n'))[\psi]) \mid n' \in \mathit{Dom}(\rho) \text{ and } n \text{ dominates } n'\}$$

we observe the following.

First, the instantiated description provided by IR_{root} (the set of all licensing rules with the nodes replaced by their ψ values) is equal to $FD[\psi]$ and thus equivalent to the original f-description FD.

Moreover, the instantiated rules in IR_{root} are *appropriately instantiated* by terms of $\mathcal{T}_F$. That is, they are tuples $(r, (t, t_1..t_m))$ consisting of an m-ary LFG rule r ($m \geq 0$) and a pair $(t, t_1..t_m) \in \mathcal{T}_F \times \mathcal{T}_F^m$ that satisfy, for any given sequence $a_1..a_m$ ($|a_1..a_m| = m$) of pair-wise distinct constants not in $\mathcal{T}_F$, the following conditions:

(i) if $t_j = \perp$ then a_j does not occur in $\mathit{Inst}(r, (t, a_1..a_m))$,
(ii) if a_j is m-definable in $\mathit{Inst}(r, (t, a_1..a_m))$ then $\mathit{Inst}(r, (t, a_1..a_m)) \vdash a_j = t_j$,
(iii) otherwise, t_j is a canonical constant, not identical to any of the other terms.[7]

In this definition, the constants $a_1..a_m$ provide the same discriminations as the daughter nodes of any local tree licensed by the rule. Thus, if a daughter is definable in terms of its mother, it is replaced by a defining term for that daughter (condition (ii)). If a daughter node does not occur in the instantiated description of the rule, it might (but need not) be replaced by the dummy constant $\perp$ (condition (i)). This is because such a daughter node might be introduced into the f-description of a derivation by the rule expanding that daughter. Finally, a daughter node that is not definable in terms of its mother and is not instantiated by $\perp$ must biuniquely be instantiated by a canonical constant (condition (iii)). This is because distinct nodes occurring in an f-description of a derivation of an acyclic f-structure cannot denote

[7] That is, $t_j \neq t$ and $t_j \neq t_i$ for all $i = 1, .., m$ with $i \neq j$.

the same f-structure unit, if they are not definable in terms of their mothers (for a proof see Wedekind and Kaplan 2006). Appropriate instantiations of the start rule of the grammar in (9) are, for example, the instantiated rules in (13).

(13) a. $\left(\begin{smallmatrix}\mathrm{S}\rightarrow\mathrm{a} & & \mathrm{A} & & \mathrm{b}\\ & (\uparrow\ \mathrm{H}) = \downarrow & & & \end{smallmatrix}, (root, \perp\ (root\ \mathrm{H})\ \perp)\right)$

b. $\left(\begin{smallmatrix}\mathrm{S}\rightarrow\mathrm{a} & & \mathrm{A} & & \mathrm{b}\\ & (\uparrow\ \mathrm{H}) = \downarrow & & & \end{smallmatrix}, (a_{\mathrm{a}}, \perp\ (a_{\mathrm{a}}\ \mathrm{H})\ a_{\mathrm{b}})\right)$

The rules (14a–c), on the other hand, are not appropriately instantiated. They violate the conditions (i)–(iii), respectively.

(14) a. $\left(\begin{smallmatrix}\mathrm{S}\rightarrow\mathrm{a} & & \mathrm{A} & & \mathrm{b}\\ & (\uparrow\ \mathrm{H}) = \downarrow & & & \end{smallmatrix}, (root, \perp\ \perp\ \perp)\right)$

b. $\left(\begin{smallmatrix}\mathrm{S}\rightarrow\mathrm{a} & & \mathrm{A} & & \mathrm{b}\\ & (\uparrow\ \mathrm{H}) = \downarrow & & & \end{smallmatrix}, (root, \perp\ a_{\mathrm{b}}\ \perp)\right)$

c. $\left(\begin{smallmatrix}\mathrm{A}\rightarrow & & \mathrm{A} & & \mathrm{c}\\ & (\downarrow\ \mathrm{H}') = \uparrow & & & \end{smallmatrix}, (a_{\mathrm{b}}, a_{\mathrm{b}}\ \perp)\right)$

Finally we observe that undefinable daughters of rules which license distinct local trees are always instantiated by distinct canonical constants. This is again a consequence of the fact that two distinct non-mother-definable nodes of an f-description can never refer to the same functional unit. Two instantiated rules which satisfy this condition are said to be *compatible*.

If we now augment the category label X of each node n of the c-structure of an arbitrary derivation of F in G by the term component $\psi(n)$ and the associated rule component IR (written $X{:}\psi(n){:}IR$) we obtain a context-free derivation. The context-free rules that license this derivation have the form:

(15) $A{:}t{:}IR \rightarrow X_1{:}t_1{:}IR_1..X_m{:}t_m{:}IR_m$ such that

(a) there is an $r \in R$ expanding A to $X_1..X_m$,

(b) $IR = \{(r, (t, t_1..t_m))\} \cup \bigcup_{j=1}^{m} IR_j$,

(c) if the rule $(r, (t, t_1..t_m))$ or a rule $(r', \tau') \in IR_i$ is also contained in some IR_j and $i \neq j$ $(i, j = 1, .., m)$ then it is compatible with itself.

The category projection *Cat* of the terminal string of this derivation is identical to the terminal string of the original LFG derivation.[8]

Moreover, the rule component of the root category IR_{root} is included in the set IRD_F consisting of all sets of appropriately instantiated and

[8]The category projection maps every symbol of the form $X{:}t{:}IR$ in a string or a set of strings back to the refined symbol X of G.

pair-wise compatible rules that yield a description of F. Both the set IRD_F and its elements are finite, since R and $\mathcal{T}_F$ are finite. Thus, if we introduce a new start symbol S_F and create a start rule of the form

(16) $\mathrm{S}_F \rightarrow \mathrm{S}{:}root{:}IR_{root}$, with $IR_{root} \in IRD_F$

we obtain the rules of a context-free grammar that derives the original string as its *Cat* projection.

If for an LFG grammar $G = (N, T, \mathrm{S}, R)$ and an acyclic f-structure F, we now define a context-free grammar $G_F = (N_F, T_F, \mathrm{S}_F, R_F)$ whose rule set R_F contains all rules of the form (16) and all rules of the form (15), where IR is a subset of one of the elements of IRD_F, we have enough rules to simulate all derivations of F in G. The vocabulary of this grammar consists of the nonterminals $N_F = \{\mathrm{S}_F\} \cup (N \times \mathcal{T}_F \times \bigcup\{Pow(IR) \mid IR \in IRD_F\})$, the terminals $T_F = T \times \mathcal{T}_F \times \{\emptyset\}$, and the new start symbol S_F.

That G_F does not simulate derivations of f-structures other than F can also be seen. From a derivation from $\mathrm{S}{:}root{:}IR_{root}$ in G_F, we first read off a c-structure c and a substitution ψ. We obtain the c-structure by taking the *Cat* projection of every category, and ψ by setting $\psi(n) = t$, for each node n with label $X{:}t{:}IR$. Using a bottom-up argument, then the following is also easy to verify: (i) if n is a nonterminal node we can assign to n an LFG rule that is instantiated by the term components of the licensing G_F rule contained in the rule component IR of n's label and that licenses the corresponding mother-daughter configuration in c, and (ii) IR contains all instantiated rules of G that are required to license the corresponding subderivation from n in c, except that the nodes are replaced by their ψ values. The substitution ψ must thus reduce the f-description that is induced by this rule-assignment to the description provided by IR_{root}. Moreover, these two descriptions must be equivalent, since the appropriateness and compatibility conditions ensure that ψ replaces m-definable daughters by defining terms, but exclude that two different non-m-definable, but denoting daughters, are replaced by the same constant.

This completes the informal proof of the following theorem. The proof is presented in full detail in Wedekind and Kaplan (2006).

Theorem 4 *Let G be an arbitrary LFG grammar. Then for any acyclic f-structure F, $Gen_G(F) = Cat(L(G_F))$.*

From Theorem 4 it follows immediately that the set $Gen_G(F)$ is a context-free language.

Corollary 5 *For any LFG G and any acyclic f-structure F, $Gen_G(F)$ is a context-free language.*

As an illustration, consider the grammar with the rules in (9). For this grammar and the f-structure in (10), the set IRD_F contains as one of its elements the set in (17)

$$(17)\quad \left\{ \begin{array}{l} \left(\begin{array}{c} \mathrm{S}\rightarrow\mathrm{a} \\ \end{array} \begin{array}{c} \mathrm{A} \\ (\uparrow\ \mathrm{H}) = \downarrow \end{array}^{\mathrm{b}}, (\mathit{root}, \bot\ (\mathit{root}\ \mathrm{H})\ \bot) \right), \\ \left(\begin{array}{c} \mathrm{A}\rightarrow\mathrm{a} \\ \end{array} \begin{array}{c} \mathrm{A} \\ \uparrow = \downarrow \end{array}^{\mathrm{b}}, ((\mathit{root}\ \mathrm{H}), \bot\ (\mathit{root}\ \mathrm{H})\ \bot) \right), \\ \left(\begin{array}{c} \mathrm{A}\rightarrow \\ \end{array} \begin{array}{c} \mathrm{a} \\ (\uparrow\ \mathrm{H}) = \mathrm{V} \end{array}^{\mathrm{b}}, ((\mathit{root}\ \mathrm{H}), \bot\ \bot) \right) \end{array} \right\}$$

that provides for the structure in (10) the description in (18).

$$(18)\quad \left\{ \begin{array}{l} (\mathit{root}\ \mathrm{H}) = (\mathit{root}\ \mathrm{H}), \\ (\mathit{root}\ \mathrm{H}\ \mathrm{H}) = \mathrm{V} \end{array} \right\}$$

There are of course other sets contained in IRD_F which provide either the same or alternative descriptions of F. These differ from the one in (17) in that the constants a_{a} and a_{b} (biuniquely) instantiate one or two terminal daughters, or the constant a_{a} replaces *root* and one terminal daughter is instantiated by a_{b}.

Based on the set of instantiated rules in (17), our construction produces a context-free grammar that includes the rules in (19). For convenience we use here r_1, r_2, and r_3 to denote the instantiated rules $\left(\begin{array}{c} \mathrm{S}\rightarrow\mathrm{a} \\ \end{array} \begin{array}{c} \mathrm{A} \\ (\uparrow\ \mathrm{H}) = \downarrow \end{array}^{\mathrm{b}}, (\mathit{root}, \bot\ (\mathit{root}\ \mathrm{H})\ \bot) \right)$, $\left(\begin{array}{c} \mathrm{A}\rightarrow\mathrm{a} \\ \end{array} \begin{array}{c} \mathrm{A} \\ \uparrow = \downarrow \end{array}^{\mathrm{b}}, ((\mathit{root}\ \mathrm{H}), \bot\ (\mathit{root}\ \mathrm{H})\ \bot) \right)$, and $\left(\begin{array}{c} \mathrm{A}\rightarrow \\ \end{array} \begin{array}{c} \mathrm{a} \\ (\uparrow\ \mathrm{H}) = \mathrm{V} \end{array}^{\mathrm{b}}, ((\mathit{root}\ \mathrm{H}), \bot\ \bot) \right)$, respectively.

$$\begin{array}{lcl} (19)\ \mathrm{S} & \rightarrow & \mathrm{S}{:}\mathit{root}{:}\{r_1, r_2, r_3\} \\ \mathrm{S}{:}\mathit{root}{:}\{r_1, r_2, r_3\} & \rightarrow & \mathrm{a}{:}\bot{:}\emptyset\ \ \mathrm{A}{:}(\mathit{root}\ \mathrm{H}){:}\{r_2, r_3\}\ \ \mathrm{b}{:}\bot{:}\emptyset \\ \mathrm{A}{:}(\mathit{root}\ \mathrm{H}){:}\{r_2, r_3\} & \rightarrow & \mathrm{a}{:}\bot{:}\emptyset\ \ \mathrm{A}{:}(\mathit{root}\ \mathrm{H}){:}\{r_2, r_3\}\ \ \mathrm{b}{:}\bot{:}\emptyset \\ \mathrm{A}{:}(\mathit{root}\ \mathrm{H}){:}\{r_2, r_3\} & \rightarrow & \mathrm{a}{:}\bot{:}\emptyset\ \ \mathrm{A}{:}(\mathit{root}\ \mathrm{H}){:}\{r_3\}\ \ \mathrm{b}{:}\bot{:}\emptyset \\ \mathrm{A}{:}(\mathit{root}\ \mathrm{H}){:}\{r_3\} & \rightarrow & \mathrm{a}{:}\bot{:}\emptyset\ \ \mathrm{b}{:}\bot{:}\emptyset \end{array}$$

These derive the set of terminal strings $\{\mathrm{a}{:}\bot{:}\emptyset^n \mathrm{b}{:}\bot{:}\emptyset^n \mid 1 < n\}$ whose *Cat* projection is $\{\mathrm{a}^n\mathrm{b}^n \mid 1 < n\}$, the set of terminal strings that the grammar in (9) relates to (10). In addition to the rules in (19), the context-free grammar for (10) contains only productions that are either useless in that they do not combine with others to produce terminal strings, or redundant because they differ from the ones in (19) in that some of the terminals are biuniquely instantiated with the canonical constants a_{a} and a_{b} instead of $\bot$ and thus do not allow the derivation of strings other than the ones already derivable by (19).

Our grammar construction procedure produces for an LFG grammar G and an acyclic f-structure a context-free grammar G_F that is

a specialization of the context-free backbone of G. This specialization simulates exactly those derivations of G whose derived strings get assigned F. The context-free grammar G_F that compactly represents all those derivations can be seen as a chart representation of $Gen_G(F)$, if we follow Lang's (1994) characterization of a chart (see also Billot and Lang 1989). For context-free chart-parsing, Lang points out that a chart for an input string s and a context-free grammar G can be seen as a specialization G_s of G that derives the empty language if $s \notin L(G)$, or otherwise just s, with effectively the same parse trees that G assigns to s.

The characterization of charts as grammars offers a number of possibilities to exploit our grammar construction procedure. Since the correctness of the output grammar G_F can be shown for every LFG G and acyclic f-structure F (Wedekind and Kaplan (2006) contains an exact proof), our approach provides a general theoretical framework in which existing chart-based generation algorithms can be examined, compared and improved.

For simplicity, we presented the grammar construction here as an abstract procedure and did not take into account all the details of the data structures and their control on which an efficient implementation will depend. This does not, of course, exclude that our approach might also provide the basis for an efficient implementation of a chart-based generator. Some of the optimizations that can be made in order to accomplish this are presented in Wedekind and Kaplan (2006).

Moreover, for selecting the preferred sentence from the set of possibilities, such a chart-generation algorithm can easily be combined with different statistical models. This includes simple n-gram models, but extends also to more sophisticated discriminative models (e.g., maximum entropy models), since G_F records enough information on the derivations of F in G so that these can be systematically recovered. The training of the more sophisticated models requires treebanks which pair f-structures with their preferred surface realizations. These are different from most of the existing treebanks, which pair strings with their optimal f-structures, because these treebanks have been produced for the parsing direction. However, for LFG grammars that come already with a probabilistic context-free skeleton, like the ones induced from automatically annotated (phrase-structure) treebanks (see, for example, O'Donovan et al. 2005 and van Genabith, this volume), there is a way to do without a training treebank for generation. From such a grammar G we obtain a generation model in three steps. First we remove from G_F all useless rules with standard context-free tools. Then we distribute the probabilities of the original context-free skeletons uniformly over

their specializations. And finally we normalize the rule probabilities so that the sum of the rules expanding a nonterminal is equal to 1. However, how such an approach relates to others has not been tested yet, although only a relatively small generation treebank would be required for evaluation.

Wedekind and Kaplan (2006) show that the grammar-construction procedure can be extended to LFGs that make use of most of the other formal devices of the LFG formalism. However, for grammars with the restriction operator (Kaplan and Wedekind 1993) the context-freeness result cannot be established. In the general case, with no further limits imposed on the use of this operator, it is undecidable whether or not there are any strings associated with an f-structure that might be derived by restricting information off of larger structures.

4.5 Grammars with the Restriction Operator

The restriction operator (notated by $\backslash$) can be used to remove functional information associated with intermediate nodes of a c-structure. If the removed information is not considered to be part of the f-structure assigned to the sentence as a whole and therefore also not included in the input for generation, then generation from restricted f-structures is undecidable. This is similar to the general case, namely generation from arbitrarily underspecified inputs. Wedekind and Kaplan (2006) show this by a reduction of LFG's emptiness problem, similar to the proof of Theorem 1. Thus, let $G' = (N', T', \mathrm{S}', R')$ be an arbitrary LFG grammar. With two new nonterminals S and A we construct a new grammar $G = (N, T, \mathrm{S}, R)$ by $N = N' \cup \{\mathrm{S}, \mathrm{A}\}$, $T = T'$ and

$$R = \left\{ \begin{array}{ccccc} \mathrm{S} & \rightarrow & \mathrm{A} & \mathrm{A} \rightarrow & \mathrm{S}' \\ & & (\uparrow \textsc{empty}) = 0, & & (\uparrow \textsc{dep}) = \downarrow \\ & & \uparrow = \downarrow \backslash \textsc{dep} & & \end{array} \right\} \cup R'.$$

By this construction $Gen_G([\textsc{empty}\ 0]) = \emptyset$ if and only if $L(G') = \emptyset$. Thus, LFG's emptiness problem reduces to the generation problem for LFG grammars with the restriction operator.

Theorem 6 *For LFG grammars G with the restriction operator and f-structures F it is in general not decidable whether $Gen_G(F)$ is empty.*

In the proof, we permitted the restriction operator to eliminate substructures whose size is not bounded by the size of the remaining f-structure. Thus, arbitrarily larger structures have to be hypothesized by the generator, similar to the situation where the inputs are arbitrarily underspecified. Since there are interesting cases in which it might be appropriate to remove certain information from larger structures (Butt

et al. 2003, Butt and King this volume, Wedekind and Ørsnes 2003, 2004), further research is needed to identify formally appropriate and linguistically plausible limitations on the use of this operator that ensure that both the number of discarded structures and their depth are always finitely bounded.

4.6 Ambiguity Preservation

So far, we considered only problems where the input consists of single structures. In the context of machine translation, it is particularly useful to have a generator that is able to produce a sentence that exactly expresses the ambiguity represented by a set of f-structures (Maxwell, this volume). Such a generator would permit it to (partially) bypass the difficult disambiguation process whenever a target sentence can be found that expresses exactly the same readings as an ambiguous source sentence (or at least the majority of them). Moreover, if the target sentence preserves most of the readings of the source sentence, it provides the translation that is in general felt to be the most natural one.

However, the reason that we cannot appeal to a general algorithm for solving the problem of ambiguity-preserving generation is again the expressiveness of the language $Gen_G(F)$. In Wedekind and Kaplan (1996) we showed that it is in general undecidable whether or not there are any strings that have exactly the readings represented by a set of f-structures or their packed representation. For the proof it is sufficient to realize that an LFG grammar might relate an arbitrary context-free language to a given f-structure. Then, by constructing for two arbitrary context-free grammars $G_1 = (N_1, T_1, \mathrm{S}_1, R_1)$ and $G_2 = (N_2, T_2, \mathrm{S}_2, R_2)$ with $N_1 \cap N_2 = \emptyset$, an LFG grammar $G = (N, T, \mathrm{S}, R)$ with $N = N_1 \cup N_2 \cup \{\mathrm{S}\}$, $T = T_1 \cup T_2$, and

$$R = R_1 \cup R_2 \cup \left\{ \begin{array}{ccc} \mathrm{S} & \rightarrow & \mathrm{S}_1 \\ & & (\uparrow \textsc{a}) = 1' \end{array} \quad \begin{array}{ccc} \mathrm{S} & \rightarrow & \mathrm{S}_2 \\ & & (\uparrow \textsc{a}) = 2 \end{array} \right\}$$

we obtain an LFG grammar that assigns [A 1] to all strings in $L(G_1)$ and [A 2] to all strings in $L(G_2)$. Since only strings in the intersection $L(G_1) \cap L(G_2)$ are derived ambiguously with [A 1] and [A 2], the emptiness problem of the intersection of two arbitrary context-free languages reduces to LFG's ambiguity-preserving generation problem. Since the former problem is known to be undecidable, as mentioned above, LFG's ambiguity-preserving generation problem must be undecidable too.

Theorem 7 *For LFG grammars G and functional structures $F_1, .., F_n$ $(n > 1)$ it is in general not decidable whether* $\bigcap_{i=1}^{n} Gen_G(F_i) = \emptyset$.

Here, one might again point out that it is cognitively implausible that an LFG grammar relates strings that form a context-free or even more expressive language to an f-structure. It seems more appropriate to assume that there is again some structural relationship, now between the c-structure and the f-structure, that bounds the size of the c-structure by the size of the f-structures associated with it, and that this relation reduces the problem to the intersection of finite sets. The off-line generability condition that was previously called for to enforce an appropriate structural relation between a semantic representation and its subsuming f-structures should thus also extend to the c-structures which are set in correspondence with the f-structures by the grammar. XLE's off-line generability condition is certainly strong enough to guarantee that only a finite number of trees is generable for any fully specified input f-structure. But similar to underspecified inputs, it might again be too strong for some grammatical analyses of complex tenses. If auxiliaries are not treated as main verbs, but as functional, and thus non-PRED-bearing categories, and complex analytical tenses are derived by virtue of recursive rules, generation of sentences with complex tense forms will again fail.

However, at least for acyclic inputs, there is, under certain recognizable circumstances, a way to manage without such a restriction. Under the plausible assumption that only a finite number of sentences is related to an f-structure by a natural language grammar, our context-free grammar construction does enable us to determine whether there are any strings derivable that express exactly the readings represented by a collection of f-structures $F_1, .., F_n$. For a set of f-structures $\{F_1, .., F_n\}$, we construct the context-free grammars G_{F_i} and inspect them with standard context-free tools to determine whether $L(G_{F_i})$ is finite $(i = 1, .., n)$. If there are no infinite and thus defective candidate sets, we can then intersect the (finite) languages $\bigcap_{i=1}^{n} L(G_{F_i})$ to determine whether there are any sentences that are derived ambiguously with f-structures $F_1, .., F_n$. For cyclic inputs, on the other hand, it is not clear yet whether a similar procedure can be found. This depends, among others, on the still open question whether the finiteness of $Gen_G(F)$ is decidable for any given LFG G and f-structure F.

Acknowledgments

I am indebted to Ron for the fruitful collaboration that we had on these and other topics over almost the last two decades. Also, I wish to thank John Maxwell and two anonymous reviewers for their comments on an earlier version of this paper.

References

Billot, Sylvie and Bernard Lang. 1989. The structure of shared forests in ambiguous parsing. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL'89)*, pages 143–151. Vancouver, Canada.

Butt, Miriam, Tracy H. King, and John T. Maxwell, III. 2003. Complex predicates via Restriction. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2003 (LFG'03)*, pages 92–104. Albany, NY: CSLI Publications.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, John T. Maxwell, III, and Paula S. Newman. 2006. XLE Documentation. Palo Alto Research Center.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell, III, and Annie Zaenen, eds. 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Dymetman, Marc. 1991. Inherently reversible grammars, logic programming and computability. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL'91), Workshop on Reversible Grammar in Natural Language Processing*, pages 20–30. Berkeley, CA.

Halvorsen, Per-Kristian and Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'88)*, pages 1116–1122. Tokyo, Japan. Reprinted in Dalrymple et al. (1995), pages 279–292.

Kaplan, Ronald M. and Joan Bresnan. 1982. A formal system for grammatical representation. In J. Bresnan, ed., *The Mental Representation of Grammatical Relations*, pages 173–281. Cambridge, MA: MIT Press. Reprinted in Dalrymple et al. 1995, pages 29–130.

Kaplan, Ronald M. and Jürgen Wedekind. 1993. Restriction and correspondence-based translation. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL'93)*, pages 193–202. Utrecht, The Netherlands.

Kaplan, Ronald M. 1995. The formal architecture of Lexical-Functional Grammar. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell, III, and A. Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, pages 7–27. Stanford, CA: CSLI Publications.

Kaplan, Ronald M. and Jürgen Wedekind. 2000. LFG generation produces context-free languages. In *Proceedings of the 18th International Conference*

on Computational Linguistics (COLING'00), pages 425–431. Saarbrücken, Germany.

Lang, Bernard. 1994. Recognition can be harder than parsing. *Computational Intelligence* 10(4):486–494.

Nishino, Tetsuro. 1991. Mathematical analysis of Lexical-Functional Grammars. *Language Research* 27(1):119–141.

O'Donovan, Ruth, Michael Burke, Aoife Cahill, Josef van Genabith, and Andy Way. 2005. Large-scale induction and evaluation of lexical resources from the Penn-II and Penn-III treebanks. *Computational Linguistics* 31(3):329–365.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago, IL: The University of Chicago Press.

Prescher, Detlef. 1997. *Generierung in unifikationsbasierten Grammatiken*. Master's thesis, University of Stuttgart.

Roach, Kelly. 1983. LFG languages over a one-letter alphabet. Manuscript, Xerox PARC, Palo Alto, CA.

Shieber, Stuart M., Hans Uszkoreit, Fernando Pereira, Jane Robinson, and Mabry Tyson. 1983. The formalism and implementation of PATR-II. In B. J. Grosz and M. Stickel, eds., *Research on Interactive Acquisition and Use of Knowledge*, pages 39–79. Menlo Park, CA: SRI International. SRI Final Report 1894.

van Noord, Gertjan. 1993. *Reversibility in Natural Language Processing*. Ph.D. thesis, Rijksuniversiteit Utrecht.

Wedekind, Jürgen. 1995. Some remarks on the decidability of the generation problem in LFG- and PATR-style unification grammars. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL'95)*, pages 45–52. Dublin, Ireland.

Wedekind, Jürgen and Ronald M. Kaplan. 1996. Ambiguity-preserving generation with LFG- and PATR-style grammars. *Computational Linguistics* 22(4):555–558.

Wedekind, Jürgen. 1999. Semantic-driven generation with LFG- and PATR-style grammars. *Computational Linguistics* 25(2):277–281.

Wedekind, Jürgen and Bjarne Ørsnes. 2003. Restriction and verbal complexes in LFG. A case study for Danish. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2003 (LFG'03)*, pages 424–450. Stanford, CA: CSLI Online Publications.

Wedekind, Jürgen and Bjarne Ørsnes. 2004. An LFG account of the Danish verbal complex and its topicalization. *Acta Linguistica Hafniensia* 36:35–64.

Wedekind, Jürgen and Ronald M. Kaplan. 2006. LFG generation by grammar specialization. To appear. University of Copenhagen/Palo Alto Research Center.

Part II

Grammar Engineering and Applications

5

Using XLE in an Intelligent Tutoring System

Richard R. Burton

5.1 Introduction

I first met Ron Kaplan in 1973 while he was consulting at Bolt, Beranek and Newman, Inc. on Bill Woods' LUNAR natural language understanding project. He was finishing his dissertation at Harvard at the time and also working with Martin Kay on what would become LFG (Lexical-Functional grammar). I was working with John Seely Brown on intelligent tutoring systems. Roughly, I was trying to get a computer to teach students like good human tutors do. At the time, the state-of-the-art means of communicating with students was teletypes or character-based CRT display terminals. If you wanted to have a free flowing interaction with students, natural language was pretty much the only option. Thus began my interest in understanding natural language.

In the next few years, I built a natural language interface for the intelligent tutoring system SOPHIE that allowed a student to interact with the computer to learn electronic troubleshooting (Brown et al. 1982, Burton and Brown 1986). I was fortunate to be able to work closely with Bill Woods's Natural Language Understanding group. I learned a lot about formalisms for representing information about language and algorithms for manipulating them. They were working on complex linguistic phenomena such as conjunction, relative clauses, and the logical structure of quantification. As we started using SOPHIE with real students, we encountered a different set of linguistic problems.

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

The students were not using particularly complex sentences. They were, however, using conversational constructs such as ellipsis ("what about T2?") and pronominal reference ("What is it for T1?"), and making plain old spelling mistakes ("What is the voltage at the base of the power limiting transitor?"). Mostly the problems I faced were engineering ones: incomplete language coverage for what students actually said; the system not being fast enough; and poor semantic mapping between existing domains and mine. In the end, the system worked pretty well. Well enough to show that the idea of tutoring students in natural language was possible, at least within a limited domain.

Eventually, I concluded that my goals and hence my problems were sufficiently different from the ones driving natural language understanding that I should be using a different name to label my efforts. I started calling my work natural language engineering. I characterized natural language engineering as using current natural language machinery to provide a "natural language" interface to a computer application. About this time, bit-mapped graphic displays were developed, opening the possibility of graphical interfaces that were inexpensive, easy to program and full of opportunities for instructional interfaces. I stopped pursuing natural language interfaces and began developing graphical user interfaces. I moved to PARC and was fortunate to able to work with Ron for several years developing Interlisp-D. We each saw Interlisp-D as a necessary platform for pursuing our research; natural language for him, graphical user interfaces and intelligent tutoring systems for me.

A few years ago, I was approached by Acuitus, Inc. to provide natural language input capability to a new digital tutor they were developing. I had kept in touch with Ron and knew he had worked very hard refining and developing his ideas about how computers should handle natural language. I jumped at the chance to find out how much progress Ron has made and to see what could be done with the three orders of magnitude more computation that is available from today's computers. This paper describes my experiences using XLE in my most recent natural language engineering effort.

5.1.1 An Intelligent Tutor for Network Administration

The current efforts are focused on building a computer-based course to teach network administration. Our subject matter can be roughly characterized as the networking fundamentals and troubleshooting techniques necessary to find and fix any problems that would prevent a user's computer from being able to browse a web site. Behind this simple description lies a large body of content that includes computer and

networking hardware, ethernet and internet protocols and their implementations, networking services, client applications, web servers, and troubleshooting skills.

The course is a mixture of presentation of material, interactive activities to learn about commands, and most importantly, troubleshooting exercises in which students find and fix problems on real (not simulated) systems. The exercises are performed on a three-machine network: a client machine with web browser, a server machine with a web server, and a name server machine. During the exercises, a computer-based tutor monitors their activities and provides help either when the student asks for it or when the tutor decides to step in based on its observations.

Our goal is for our digital tutor to do as well as an excellent human tutor does when working one-on-one with a student. Figure 1 shows a systems-level view of the tutor. It shows how the tutor monitors the students actions and how XLE fits in.

5.2 Why Use Natural Language?

The tutor is written in Java and has (or could have) access to any of Java's interactive graphical and multimedia capabilities. On top of all this, why go to all the trouble of accepting natural language? In fact, the large majority of interactions that the tutor has with students are multiple choice or short answer questions (which do not use natural language). But there are some things that natural language can do that are not available otherwise.

The primary advantage of natural language is that it forces articulation of ideas onto a blank slate. This creates a different learning experience than multiple choice in which the listed choices define the set of allowable answers. The choices shape the student's thinking and allow them to use an elimination process rather than a creation process when they are not sure of the answer. Further, having the students express their thoughts in their own words is an important learning step.

In addition, natural language allows a much larger set of answers than is feasible with multiple choice. Hundreds or thousands of alternative answers can be supported with no change to the interface.

Another reason for using natural language in the interface is that learning how to express concepts and ideas about unix system administration/networking problems is one aspect of the curriculum. The course is teaching students to be system administrators. Part of being a system administrator is being able to write up what you found and changed so that you can communicate with other system administra-

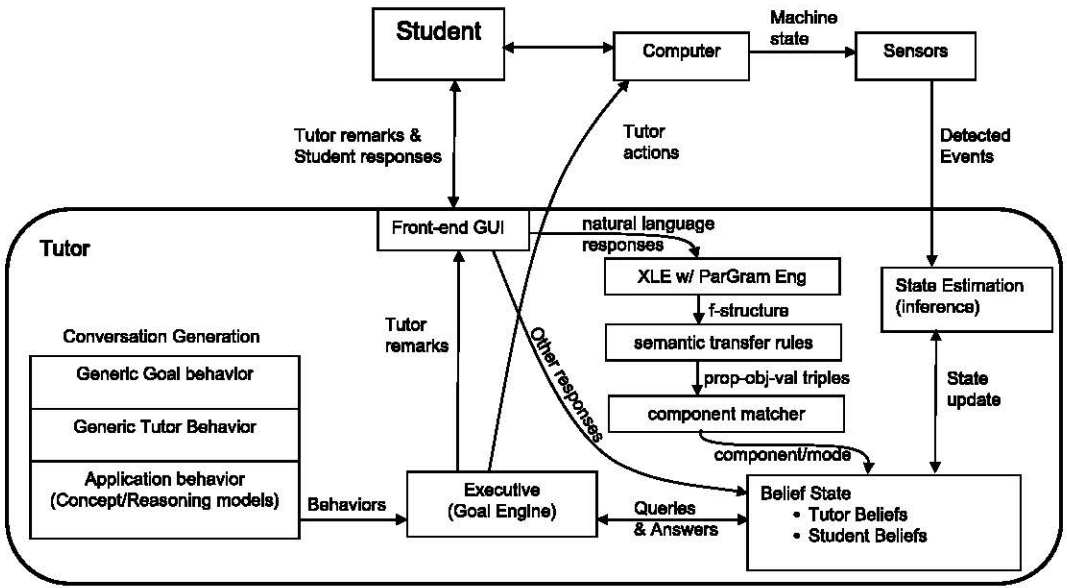


FIGURE 1 System diagram of the digital tutor showing XLE.

tors. Natural language interaction provides students with the opportunity to practice their system administrator speak.

The final reason we pursued natural language is that eventually we will want the tutor to be able to focus on meta-problem solving. Schoenfeld (1992) has demonstrated that a distinguishing characteristic of master mathematical problem solvers is awareness of their own problem solving strategies. He has been able to teach this behavior to undergraduates in a semester course. To get students to focus on meta-problem solving, he reserves the right any time the students are working on a problem to ask the following three questions:

- What (exactly) are you doing? (Can you describe it precisely?)
- Why are you doing it? (How does it fit into the solution?)
- How does it help you? (What will you do with the outcome when you obtain it?)

We eventually want to incorporate this type of behavior into our tutor and believe that interacting in natural language will be required for this ability.

5.2.1 When to Use Natural Language

Having suggested reasons why natural language is critical, it is important to keep in mind that these reasons do not apply to all interactions. Many interactions are well handled by multiple-choice and short answer questions. In fact, our initial prototype of the tutor did not include natural language at all. In the transcripts of trials with this prototype, we looked for places where natural language might be particular useful.

The interaction that stood out as the least satisfying was when the tutor asked the student "what do you think is wrong with the system?". In most sessions, the tutor asks this question early in a problem when the student has performed some tests on the system but has not yet attempted to fix anything. In this situation, the student may be almost ready to fix the problem, totally lost, or anywhere in between. Thus, responding accurately to what the student says is critical for the tutor to start off in the right direction.

Without natural language, this interaction was handled as a multiple choice question with 13 possible answers. This seemed like a good place for natural language. So, we decided to begin our use of XLE by handling student answers to this question.

5.2.2 The Range of Answers

Our tutor has a model of the things that can go wrong with the system. The model's central structure is a hierarchy of functionally-based

chunks called components. Since the problems all deal with ways browsing a web page can be broken, the top level component is "the user's browser reads a web page from the server". Just below the top component are several things including "web service works on server". An example of a lower level component is "the client's current IP address." Basically, the hierarchy represents all the things that can be broken in the system.

Each component has a number of ways that it can be faulted. The most common fault is just the generic "broken" but some components can have more specific faults. For example, files are components and may have a fault of "does not exist".

This provides a very nice range of meanings for natural language answers to the question "What's wrong?", that being the set of all components and the ways they can be faulted. Any answer to the question should identify a component and a fault mode.

Example Sentences and Their Interpretations

Here are some examples of responses to the question "what's wrong with the system?" and their interpretation as component/fault mode pairs.

> "You cannot send packets back and forth between the client and the server by using their names." means that the component "the client connects to the server by name" is "broken".
>
> "The http server is not responding to requests." means that the component "web service works on server" is "broken".
>
> Both "The IP address for badmojo is incorrect in the hosts file on goodmojo." and "The apache server's entry in the client's hosts file has the wrong IP address." have the meaning that component "the IP address of the server's entry in the client's hosts file" is "broken".

5.3 How XLE is Used

When doing natural language understanding for SOPHIE in the 1970s, the starting point was a string of characters. This time around, XLE[1] allowed me to start with the deep structure functional groupings and relationships between word meanings in the sentence called f-structures. This is a significant improvement. To get a sense of how large an improvement this is, let's look at an example of an f-structure.

[1]When I use the term XLE, I am referring to both the XLE parsing/generating framework and to the ParGram English grammar (Crouch et al. 2006, Kaplan et al. 2004, Riezler et al. 2002).

5.3.1 Sample F-Structure

Let's consider the sentence "The apache server's entry in the client's hosts file has the wrong IP address." The f-structure produced by XLE is shown in Figure 2 and directly represents the sentence's functional relationships.

There are many relationships shown in Figure 2 but the important ones for our purposes are:

The top level is a 'have' relationship between an 'entry' and an 'address'.
The 'entry' is for a 'server'.
The 'server' is an 'apache' server.
The 'entry' is 'in' a 'file'.
The 'file' is a 'hosts' file.
The 'file' is a 'client' file.
The 'address' is an 'ip' address.
The 'address' is 'wrong'.

XLE includes the ability to specialize existing grammars, lexicons and morphologies (Kaplan et al. 2002). When I was getting started, Tracy King used this capability to create a grammar with a few domain specific features. For example, the grammar was modified to make NP an acceptable top level constituent. (Since we are asking "what's wrong?", it is reasonable to accept a noun phrase as being the thing that is wrong.) Another modification was an addition to the lexicon to allow unix specific features. As can been seen from Figure 2, the f-structure contains lexical information about the words in the sentence. By far the vast majority of this information comes from standard features in the lexicon such as HUMAN, NUM and PERS. The domain specific features shown in Figure 2 are that 'hosts' is a unix configuration file (unix-cfile) and 'apache' is a unix application (unix-app). The resulting collection of relationships provides a very good beginning to understanding the meaning of this sentence.

Initial Corpus

To guide development of the tutor's natural language understanding component, I created a corpus of 810 sentences. It was basically all the interestingly different ways I could think of to say how each of the fifty components was broken. Then I put the corpus through XLE.[2] After debugging the domain specific grammar, morphology and lexicon additions, only four of the 810 sentences had parsing difficulties. I never

[2]I was using the March 2005 release of XLE with a slightly modified ParGram English grammar from November 2004.

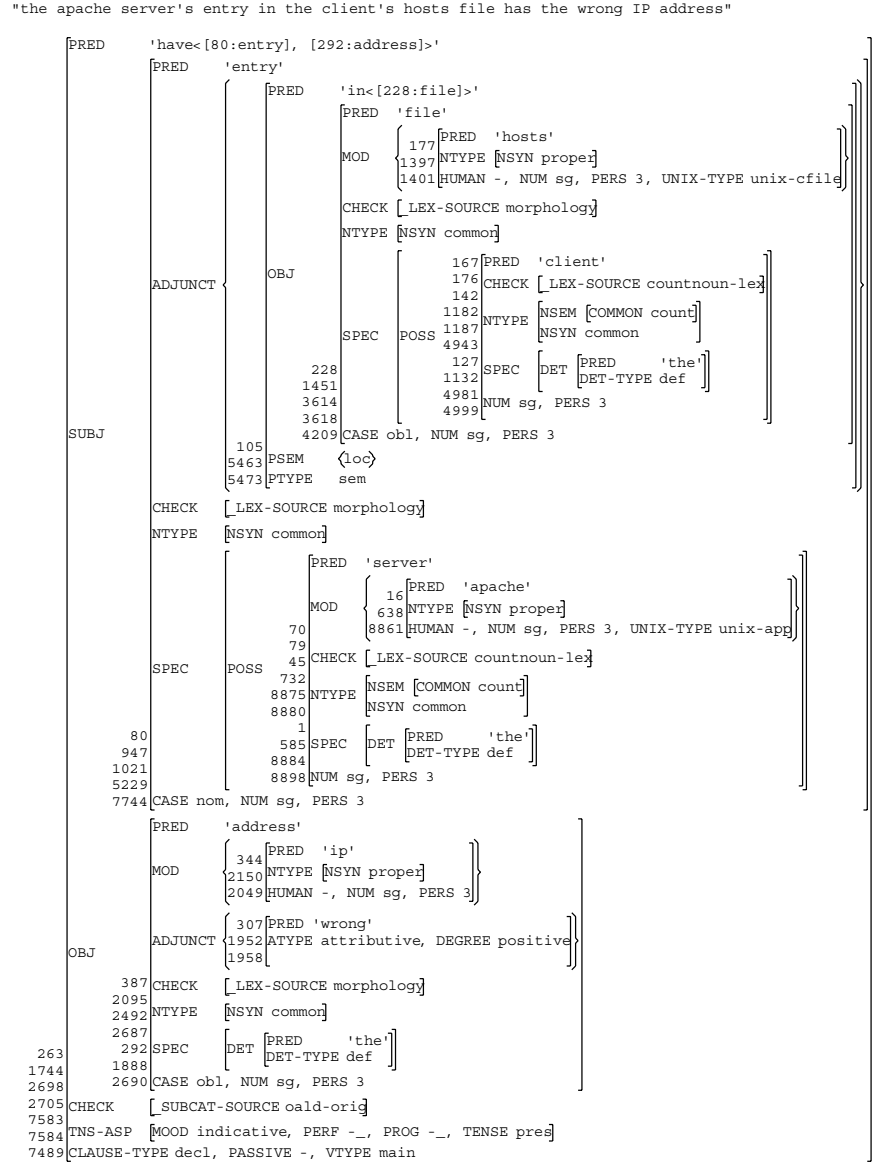


FIGURE 2 A sample f-structure from ParGram English for the sentence "The apache server's entry in the client's hosts file has the wrong IP address."

Component(var(1),EntryInFile)
ComponentFile(var(1),hosts)
EntryField(var(1),ipaddress)
OnMachine(var(1),clientmach)
OfMachine(var(1),servermach)
ComponentState(var(1),incorrect)

FIGURE 3 The property-object-value representation of the component DestIPEntryInHosts.destination-server.machine-client.

expect to see a student enter any of these four. I considered this to be an powerful testimonial to the capabilities of XLE.

Semantic Interpretation

Armed with usable parses, I began to explore how to get from f-structures into a representation that would mean something to the tutor. F-structures are still a good ways away from the component and fault mode that we are looking for as the meaning for our domain. Covering this distance requires understanding a little bit more about how components are structured.

Components are typed objects. Each type has a specific set of properties. For example, an EntryInFile is a type of component which represents an entry in a configuration file. It has properties:

OnMachine (the machine on which the file resides),
ComponentFile (the name of the file containing the entry),
OfMachine (the machine that the entry is about), and
EntryField (the field of the entry).

A component is determined by its type and the values of its properties. An example of an EntryInFile component is "the IP address in the entry of the server in the hosts file of the client." (Internally, we refer to components by name. This one's name is DestIPEntryInHosts.destination-server.machine-client.) This component's properties are ComponentFile='Hosts', EntryField='IPAddress', OnMachine='client', and OfMachine='server'. It could be described by the English phrases "the server's address in the hosts file on the client machine", or "the client machine's host file entry for the address of the server". The property-object-value triples representation is shown in Figure 3.

The first step in semantic interpretation is to go from the functional relationships of f-structures to property-object-value triples that are used to represent components. As one possible way of doing semantic interpretation, XLE provides a transfer rule language (Crouch 2005, this volume) that rewrites f-structures. We use transfer rules to match

```
PRED(%s,have),
arg(%s,1,%cmp1), +Component(%cmp1,EntryInFile),
arg(%s,2,%cmp2), Component(%cmp2,IPAddress),
Mode(%cmp2,%mode)
  ==> ComponentState(%cmp1,%mode),
      EntryField(%cmp1,IPAddress).
```

FIGURE 4 An example transfer rule.

pieces of the f-structure and rewrite them into domain specific property-object-value triples. Transfer rules are also used to remove f-structure information that is not relevant to our domain. The resulting triples are then used as constraints to determine the component.

The fault mode is represented by the property ComponentState. It is determined by looking at different pieces of the f-structure. Modifiers such as "working" or "good" get transferred into a "faultless" fault mode. Modifiers such as "bad", "broken", "wrong" or "incorrect" are transferred into a "broken" fault mode. If negation is present, the fault mode is switched from "faultless" to "broken" or vice versa. If the student enters a component as a noun phrase without modifiers rather than a complete sentence, the fault mode is left out. (In this case, when the triples are converted into a component, the tutor uses "broken" as a fault mode because the student was asked "what's wrong?".)

5.3.2 Transfer Rules

Figure 4 provides an example of a transfer rule to make things a little more concrete. It handles sentences roughly of the form "<entry> has the <mode> IP address" such as "badmojo's entry in goodmojo's hosts file has the wrong IP address."

Transfer rules[3] consist of a matching part, followed by "==>", followed by the replacement triples. Variables begin with a percent sign (%) e.g., %s or %cmp1. This rule is looking for an f-structure %s that has a PRED='have', a first argument (%cmp1) that is an EntryInFile Component, and a second argument (%cmp2) that is an IPAddress Component which also has a Mode.

When a transfer rule matches, all of the triples in the matching part are removed and the replacement triples are added. That is, the matching part is rewritten as the replacement part. Triples in the matching part are not removed if they are marked with a plus sign (+) as is done in the first Component clause in this rule. The effect of this rule

[3]The XLE documentation (Crouch et al. 2006) contains a complete description of transfer rules and how to use them.

2%	"flatten" depth by changing set representation and removing intermediate features.
7%	recognize machine names and equate terms that are the same within our domain.
16%	determine component parts from nouns and noun-noun modifications.
7%	produce triples from prepositional phrases and possessives.
4%	handle fault modes.
7%	build component triples from noun phrases.
41%	build component and fault mode from main verb.
6%	attach triples that are not connected to main component.
2%	handle negation.
7%	remove f-structure features that are not triples.

FIGURE 5 Functional groupings of transfer rules and the size of each group listed in the order in which they are applied.

would be to add to the EntryInFile Component the properties EntryField='IPAddress' and ComponentState=%mode. %mode will be either 'faultless' or 'broken' depending upon whether address was modified by 'right' or 'wrong' in the sentence.

Transfer rules are ordered. The first rule is applied, then the second rule is applied to the output of the first rule, and so on. In our example, the triples specifying Component and Mode are not part of the f-structure produced by XLE. They are added as a result of earlier transfer rules. The final set of rules removes the features of the f-structure that are not property-object-value triples. The resulting set of domain specific triples specify the component and fault mode.

The system has 1206 transfer rules to cover the corpus. They are summarized in Figure 5. The transfer rule file is slightly more than 225K characters. The execution time is very acceptable. The average time it takes to parse and interpret a sentence is about .6 seconds.

5.3.3 Getting to a Component

The set of triples that comes out of the application of the transfer rules is then matched against the actual set of components to determine which one was being referenced. This allows a separation between natural language issues and the practical issues of tutoring. For example, if you have been following closely, you may have figured out the triples for the phrase "the server's IP address in the host name file on the client" are:

Component(var(1),EntryInFile)
ComponentFile(var(1),hostname)
EntryField(var(1),ipaddress)
OnMachine(var(1),clientmach)
OfMachine(var(1),servermach).

However, it turns out the client's host name file does not contain IP addresses. It contains names. And it does not contain any information about the server. Having the interface between the natural language processing and the tutor be property triples gives the tutor the chance to see this misconception and, possibly, address it with the student.

This separation is also useful in cases where the set of triples is ambiguous in the sense that it represents more than one component. For example, the sentence "the host name file is wrong" does not say whether it is the client or the server host name file. Depending upon the context, the tutor may want to ask the student which machine, or, if it is clear that the student is focused on one machine, just fill it in.

5.3.4 Mishandled Sentences

After developing rules to get the correct semantics for 810 corpus sentences, the natural language understanding component was incorporated into our tutor. The tutor logs all sentences that it receives. Any that are not correctly handled are examined by hand.

The mishandled sentences we have seen fall into several categories. Some are 'word salad' like "server no ip" or "below application broken". XLE produces useable f-structures for the large majority of the sensical word salad we have seen so far. Our strategy for these has been that if XLE provides a useful f-structure and a human can determine what the student meant, transfer rules are written to pull out the semantics. If either XLE did not produce a usable f-structure or we could not figure out what the student meant, the system is not changed. In these cases, the tutor responds as if the student had said "I don't know" and asks directed questions to get at what the student knows.

Many of the mishandled sentences contain misspellings. For the unambiguously wrong ones, we added a character rewriting pre-pass. For example, 'resoution' gets changed to 'resolution'. This is effective at picking up misspellings that have occurred before. We have left the problem of incorporating a general spelling correction solution to the future. Most of the other mishandled sentences are added to the corpus and handled by adding transfer rules.

Over twelve months of development and testing with students, the corpus has grown to about 1400 sentences. Its growth so far has been nearly linear at a rate of about 50 sentences per month. Since the

number of students testing the system has been increasing over time, the rate of new sentences per student is decreasing. This is what we expect. Adding 50 new sentences takes about four days of work. Most of this time is spent adding transfer rules but it also includes substantial time to test the new rules in the system.

5.4 Coverage Issues

5.4.1 Ambiguity

XLE is a very powerful system with an extensive lexicon and grammar for English. While the sentences we have encountered so far do not need all of this power, it is nice to know that if a student types in a complex sentence, XLE will probably handle it. The downside of all the coverage is ambiguity. To simplify the integration of XLE with the tutor, we started with an assumption that the semantics would not deal with multiple interpretations. The system uses the first (most probable) parse (even if one of the later ones may be more correct) and the transfer rules do not build multiple interpretations. This has generally worked well.

A common place where ambiguity arises is nominal compounds such as "http server type entry". In our case this refers to the entry in the httpd configuration file that has 'servertype' as a key. In our approach, we need a transfer rule that puts these four nouns together to create the right semantic triples. As long as the most probable parse always has the same modifying relations, a single rule will work. For this phrase, the most probable parse has 'http' modifying 'server', and 'server' and 'type' modifying 'entry'. Thus we have a rule that matches the most probable parse f-structure and creates the appropriate triples. We have yet to encounter a case where we needed multiple rules for the nominal compounds in our domain. If, in the future, the most probable parses become more problematic, XLE provides a way of calculating the measures used to determine "most probable" and we could specialize it to our corpus.

Just as our students occasionally type in sentences the tutor cannot handle, they more rarely but still occasionally type in a sentence that XLE has trouble with. One recent example is that in the f-structure for "the problem is with the transport layer or farther down" 'down' is associated with the top level 'be' relationship rather than with 'farther'. In this case, a transfer rule finds the 'down' clause and produces the right meaning. This sort of thing has not happened often, and Tracy King has fixed the problems in the next grammar release. Overall, there are less than a dozen rules out of more than 1200 that look for misplaced constituents.

does	-N XLE.
like	-N XLE.
out	-V XLE.
fail	-N XLE.
work	-N XLE.
wrong	-N XLE.
can	-N XLE; -V XLE.
name	-A XLE.
on	-A XLE.
but	-ADV XLE; -N XLE.
or	-ADV XLE; -N XLE.
and	-ADV XLE; -N XLE.

FIGURE 6 List of the word senses that were removed.

5.4.2 Improving parsing by reducing coverage

More troublesome were cases where the first parse used a word sense that clearly makes sense for English in general but not in our domain. For example, the network administration domain does not use 'out' as a verb nor 'can' as a noun, and 'does' is never used as the plural of 'doe'. Fortunately, the XLE lexical routines have a way of removing word senses. Figure 6 lists the words we have had to de-sense. Tracy King suggested that much of the effort to find cases where removing word senses might help could be automated by taking all the technical terms, seeing what their morphological analyses are, and removing any that seem unlikely. At this stage for us, doing it by hand has worked fine.

5.4.3 Do Students Use Proper English?

One of the questions I was asked when I started was "will students type real English sentences into your system?" Based on our experiences, mostly the answer is yes. We have encountered abbreviations that were new to us (e.g. idk for "I don't know") and if text messaging stays popular we expect to get more. It is possible in XLE to create lexical entries for most abbreviations that allows them to be parsed in the normal way. As described earlier, we have also seen some word salad but XLE produced a useable f-structure for most of that. So far, our strategy of treating sentences that the system does not understand as if the student had said "I don't know" is producing appropriate tutorial interactions.

5.5 Conclusion

XLE and the ParGram English grammar are amazing! XLE has never crashed during real use. The grammar has parsed everything we needed it to parse. The transfer rules provide a good mechanism for translating f-structures into domain concepts. There are a lot of them but they are organized enough to continue to be extensible. In summary, our experience with XLE has surpassed our expectations. We believe XLE will continue to work well as our application grows.

My main concern with XLE is the amount that must be known to use it. You need to know morphology and XLE's language for representing morphology. You need to know lexicography and XLE's language for representing it. Thanks to Tracy King, I have not had to modify the grammar but I did need to learn lots of details about the grammar such as what the difference is between an adjunct_x, a mod_x, and an XCOMP. (For our domain, adjunct_x and mod_x are treated the same. Adjunct_x and mod_x are deep structure relations while XCOMP is a surface structure relation. And mostly, we only need consider deep structure relations.) If you want to include domain specific morphology, you need to learn Finite State Morphology, a task that begins with the book of the same name by Beesley and Karttunen (2003). You need to decide how to do semantic interpretation. This will probably involve learning yet another language such as the transfer rule language (which I recommend). XLE will shortly contain a transfer rule based semantics along with its English Grammar that promises to reduce the number of transfer rules needed. This will help.

Much of the application effort for XLE has been targeted at natural language translation. And I suspect that XLE's learning curve in this application is less. From the standpoint of a builder of interactive applications, XLE is a collection of well built, mostly complete parts that can be assembled in different ways. Each application needs to be custom built. We have yet to discover the right point of view on natural language use in interactive applications to make it easier to use. But until we do, XLE has the workbench of tools and parts to make any natural language engineer happy.

Acknowledgments

This paper is dedicated to Ron Kaplan for many years of warmth, friendship and good ideas. Thanks for the progress on all the intractable problems. This work was supported in part by the Defense Advanced Projects Agency within the DARWARS program (Contract #N00014-030C-0295). I would like to express gratitude to Ralph Chatham for his

support. Special thanks to Tracy King for her extensive, quick, knowledgeable support and for helping me get started with a customized grammar. Thanks to Tracy King, Rich Levinson and an anonymous reviewer for comments on this paper.

References

Beesley, Kenneth R. and Lauri Karttunen. 2003. *Finite State Morphology*. Stanford, CA: CSLI Publications.

Brown, John S., Richard R. Burton, and Johan deKleer. 1982. Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman and J. S. Brown, eds., *Intelligent Tutoring Systems*, pages 227–282. New York, NY: Academic Press.

Burton, Richard R. and John S. Brown. 1986. Toward a natural language capability for computer-aided instruction. In B. J. Grosz, ed., *Readings in Natural Language Processing*, pages 605–625. Los Altos, CA: Morgan Kaufmann.

Crouch, Richard. 2005. Packed rewriting for mapping semantics to KR. In *Proceedings of the 6th International Workshop on Computational Semantics (IWCS-6)*, pages 103–114. Tilburg, The Netherlands.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, John T. Maxwell, III, and Paula Newman. 2006. XLE Documentation. Palo Alto Research Center.

Kaplan, Ronald M., Tracy H. King, and John T. Maxwell, III. 2002. Adapting existing grammars: The XLE experience. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), Workshop on Grammar Engineering and Evaluation*, pages 29–35. Taipei, ROC.

Kaplan, Ronald M., Stefan Riezler, Tracy H. King, John T. Maxwell, III, Alexander Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of the Human Language Technology Conference and the 4th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04)*, pages 97–104. Boston, MA.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 271–278. Philadelphia, PA.

Schoenfeld, Alan H. 1992. Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. In D. A. Grouws, ed., *Handbook for Research on Mathematics Teaching and Learning*, chap. 15, pages 334–370. New York, NY: MacMillan.

6

How Much Can Part-Of-Speech Tagging Help Parsing?

Mary Dalrymple

6.1 Introduction

Systems that perform deep linguistic analysis generally operate by tokenizing the input string, performing morphological analysis, and handing off the tokenized, morphologically analyzed result as input to a syntactic parser. It is often argued that additional refinements in the input to the parser can improve performance: in particular, that including part-of-speech tagging as a preprocessing step removes incorrect syntactic analyses from consideration by the parser, speeding up the parsing process and reducing ambiguity in the output of the parser. However, it is not clear exactly how much tagging can help to disambiguate or reduce ambiguity in parser output in naturally-occurring text. The two ends of the spectrum of possibilities are illustrated by two well-known ambiguous sentences in English.

The two parses of sentence (1) are associated with two different tag sequences:

(1) Time flies.

In the most natural reading for this sentence, *time* is a noun and *flies* is a verb, but there is another parse in which *time* is a verb and *flies* is a noun. Given the information that *time* is a noun (or that *flies* is a verb) in the relevant context, the sentence can be disambiguated completely.

An earlier version of this paper with the same title appeared in *Natural Language Engineering*, vol. 12, 2006, copyright Cambridge University Press.

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

Another type of ambiguity is illustrated by example (2):

(2) I watched the man with the telescope.

This sentence has multiple parses because the prepositional phrase *with the telescope* has more than one possibility for attachment. The ambiguity is not reflected in different part-of-speech tags for the words in the sentence, and tagging does not help to disambiguate this example.

The study reported here grew out of a conversation with Ron Kaplan in the fall of 2002 about the potential benefits of introducing tagging as a pre-processing step in linguistic analysis. In the course of this conversation, we realized that this could be determined by examining the full parse-forest output of a large grammar, and determining whether ambiguity in naturally-occurring English sentences is commonly reflected in different part-of-speech tags for different well-formed parses. If different parses tend to be associated with different tag sequences (the *Time flies* case), assigning the correct tag sequence as a preprocessing step before parsing greatly reduces the number of parses that are produced. If different parses do not tend to be associated with different tag sequences (the *telescope* case), tagging would not help to reduce the output of the parser.

More concretely, this study can be thought of as measuring the disambiguating effect of a "perfect tagger", a hypothetical tagger which never makes a tagging mistake. This is done by examining the full parse-forest output of a parser, and partitioning the output parses into equivalence classes based on the tag sequences for each parse. Roughly speaking, a large number of tag sequence equivalence classes for each sentence means that different parses tend to be distinguished by their tags; a small number means that tagging would probably not help much in reducing ambiguity. In this way, it is possible to determine the benefits of tagging in the best case, if the perfect tagger were available to pick out the correct tag sequence for each sentence. Results of this study show that if a perfect tagger were available, an average of about 50% of the potential parses for a sentence would be eliminated. Somewhat surprisingly, the perfect tagger would not eliminate any parses for about 30% of the sentences in the corpus that was examined, since all of the parses for these sentences shared the same tag sequence.

Given the tag sequences for all successful parses of the input, more fine-grained questions about the utility of tagging can also be addressed: for example, which parts of speech play the biggest role in creating multiple tag sequence equivalence classes for a sentence. Assigning correct tags in these categories would have the greatest effect on disambiguation. This is the topic of Section 6.5 below.

It is important to note that the current study addresses only the question of the utility of tagging as it relates to ambiguity reduction. It does not answer the question of the effect of a tagger on *speed*; that is, it cannot determine whether a parser can perform faster on tagged input. This is because the data used in the study is a *parse forest* — a packed representation of all full, well-formed parses — and not a chart. Neither incomplete edges nor edges that fail to contribute to a full, well-formed parse are present for consideration in the data being examined. Since there is no way to determine how much work was done on tag sequences which do not ultimately contribute to a full and complete parse, there is no way for a study such as this one to address questions of increased efficiency or speed when parser input is pretagged. This question has been addressed in other studies, which conclude that tagging in a preprocessing step does in fact speed up the parsing process; see in particular Prins and van Noord (2001, 2003), discussed below.

6.2 Previous work

Previous research has attempted to determine the benefits of pretagging in linguistic analysis by incorporating a tagger as a preprocessor to a parser, and seeing whether the accuracy of the parser improves as a result; some of this research is discussed in the next section. This approach to the problem is potentially confusing, however, since any positive effects are inevitably obscured by the negative effect of tagging mistakes introduced by the tagger. No tagger is perfect: a recent best-case scenario for taggers trained and tested on the Wall Street Journal corpus (Marcus et al. 1994a) is around 97.2% correct (Toutanova et al. 2003). Various ways of dealing with the problem of mistags have been suggested; for example, Copperman and Segond (1996) suggest that tagging should be used not as a preprocessor to a parser, but to eliminate unlikely parts of speech for a particular domain or genre from a general-purpose lexicon. This may be of help, but does not address the general question of how much tagging as a preprocessing step could in principle help in disambiguation. An additional problem that has plagued several previous studies is incompatibility between the tags assigned by the tagger and the preterminal symbols used by the grammar. Requiring an additional mapping between the tagset for the tagger and the preterminal categories employed in syntactic analysis introduces additional errors and further obscures the results. Studies that train the tagger on the output of the parser do not suffer from this problem (Prins and van Noord 2001, 2003), but must still contend with the

harmful effect of mistags on the result.

One of the first studies to address the question of whether tagging helps in parsing was reported by Pulman (1992). In this study, a tagger was trained on the LOB corpus and used as a preprocessor to the Core Language Engine (Alshawi 1992). This resulted in a loss in accuracy in parsing, though it did increase parsing speed. Accuracy was regained by the use of a multiple tagger, a tagger that returns more than one tag for each word. However, to regain the original level of accuracy, each word had to be assigned a large enough number of tags that most of the speed gain obtained from pretagging the input was lost. Interestingly, this result goes against the findings of Charniak et al. (1996), whose work indicated that a multiple tagger does not significantly increase accuracy when used as a preprocessor to a probabilistic context-free phrase structure grammar relative to a single tagger, which assigns only one tag per word.

Subsequently, Wauschkuhn (1995) reported on a study in which two German corpora were studied; one was hand-tagged, and the other was statistically tagged, with an error rate of 3.5% to 4%. Both of these corpora were parsed twice: once with tags, and once without tags but with a morphological analyzer. There was no syntactic gold standard for either corpus, so the metric of success for this study was the number of sentences receiving a single parse in each case. This study suffered from several problems. First, the tags assigned by the morphological analyzer were not the same as the tags used for hand-tagging, which made comparison of the results difficult. Second, tagging alone cannot completely disambiguate a sentence; a sentence may be structurally ambiguous (the *telescope* case), even with the same tags, so using a metric which defines success as obtaining a single parse does not seem appropriate. Third, the grammar used in the test seems to be quite small, perhaps too small for a fair trial: the majority of sentences got either zero or one parse for both the tagged and untagged corpus.

A subsequent study was conducted by Voutilainen (1998) on the basis of a system which uses a finite-state syntactic disambiguator to discard impossible syntactic analyses. Voutilainen added a morphological disambiguator to discard impossible tags before syntactic analysis is performed. The conclusion of the study was that tagging helps to reduce ambiguity, but increases the number of sentences with no parse. As in Wauschkuhn's study, no syntactic gold standard was available to determine whether the correct parse was among those parses that were discarded, which makes it hard to determine the benefit of the addition of a tagger to the system.

More recently, Prins and van Noord (2001, 2003) addressed this ques-

tion by adding a morphological disambiguator trained on the output of the parser. This method solves one of the problems that plagues other approaches, that of incompatibility between the tagset used by the tagger and the preterminal categories of the grammar. Prins and van Noord also were able to evaluate their results against a syntactic gold standard, showing whether the correct parse is present in the output when a tagger is used. Like the work reported by Pulman (1992), Prins and van Noord concluded that tagging does in fact help to reduce ambiguity if a multiple tagger is used. Prins and van Noord also show conclusively that tagging as a preprocessing step can increase parsing efficiency, with a twentyfold speedup for the Alpino system that they tested. Nevertheless, their approach, like most other approaches, fails to distinguish between the beneficial effects of tagging and the harmful effects of tagger errors.

A study which is close in some respects to the experiment reported here was conducted by Kaplan and King (2003), using the XLE parsing platform and ParGram English grammar, described below. Kaplan and King attempted to simulate the effect of a "perfect tagger" by using the preterminal category sequence from the Penn Treebank to tag the input string in parsing sentences in the Wall Street Journal corpus. One problem with this approach is that the Penn Treebank, like any manually annotated corpus, contains tagging errors; see, for example, Dickinson and Meurers (2003). Another problem is that the Penn Treebank preterminal categories and the preterminal categories of the ParGram English grammar are not compatible, which necessitated the introduction of a mapping function to mediate between the two tagsets. Kaplan and King concluded that parsing with input annotated with tags from the Penn Treebank speeds up parsing, but decreases parsing accuracy and coverage. Incompatibility between the Penn Treebank preterminals and the ParGram preterminals, with "tagging" errors introduced by errors in the mapping function, was a major source of difficulties for their approach.

Another closely related study was carried out by Toutanova et al. (2002), who investigated several techniques for disambiguation in parsing sentences from the Redwoods HPSG treebank (Oepen et al. 2002). One of the disambiguation techniques they investigated was adding a tagger trained on the gold standard treebank as a preprocessing step. They compared these results with the effects of a "perfect tagger" which, as in the Kaplan and King experiment, assigned the tags that appear as preterminals in the Redwoods treebank gold standard. They reported results which are very similar to the findings of the current study, as discussed in Section 6.4 below.

6.3 The Current Study: Methodology

This study is based on the result of parsing a corpus with a large-scale LFG-based English grammar (Riezler et al. 2002) running on the XLE grammar development platform (Maxwell and Kaplan 1993, 1996). The preterminal sequence for each parse was extracted — these are the "tags" that would be assigned by the perfect tagger — and sorted into equivalence class groups. This is useful in several respects. First, the number of tag sequence equivalence classes for each sentence in the corpus can be correlated with the number of parses of the sentence, allowing us to determine whether most cases of ambiguity are like the *Time flies* case, or like the *telescope* case. Second, the tags which tend to give rise to different tag sequence equivalence classes can be identified; these are the tags that can help the most in disambiguation by tagging.

6.3.1 The grammar and parser

This study uses the English grammar developed at the Palo Alto Research Center within the ParGram project, a large-scale multi-site LFG grammar development project (Butt et al. 2002). As of 2006, the ParGram project encompasses large-scale grammars of English, French, German, Japanese, Norwegian, Danish, Turkish, Welsh, Hungarian, Malagasy, Vietnamese, and Arabic, with smaller grammars of Korean and Urdu also under development. The version of the ParGram English grammar used in this experiment was released in 2002, and is described in Riezler et al. (2002); it is the result of about 9 person years of development, using the XLE grammar development and parsing platform for Lexical Functional Grammar.

Analysis of a string using the ParGram English grammar and the XLE parsing platform begins with the following steps:

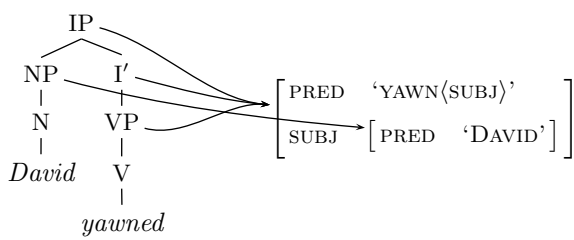
- Tokenization by finite-state transducer
- Finite-state morphological analysis, including part-of-speech information
- Guesser for forms not recognized by morphological analysis

The resulting input to syntactic rules is a chart consisting of all well-formed morphological analyses of all well-formed tokenizations. The 2002 version of the ParGram English grammar comprises 314 rules (left-hand side categories) with regular-expression right-hand sides. Lexical entries for most nouns and adjectives are constructed on the fly on the basis of the assigned syntactic category. The verb lexicon contains 9,652 stems and 23,525 subcategorization frame entries (Riezler et al. 2002).

6.3.2 Grammar output

The output of the XLE is a packed representation of well-formed pairs consisting of a *c-structure* or phrase structure tree and an *f-structure* or attribute-value structure:

(3)



The c-structure is a phrase structure tree representing surface phrasal relations and groupings; it appears on the left-hand side in (3). The f-structure is an attribute-value structure representing abstract functional syntactic relations like subject and object (Kaplan and Bresnan 1982, Dalrymple 2001); it appears on the right-hand side in (3). The mapping relating the two structures is represented by arrows from nodes of the c-structure to subparts of the f-structure. For the current study, f-structure information is not relevant and can be ignored, and the output can be treated simply as a packed parse forest.

Riezler et al. (2002) show that the coverage of the 2002 ParGram English grammar used in this study is very high, and the output is of very high quality. In a test of Section 23 of the Wall Street Journal corpus, 100% of the sentences received an analysis, though some analyses consisted of a set of well-formed fragments; 74.7% of sentences received a full (nonfragmentary) syntactic analysis. From Section 23, 700 sentences were randomly selected and a gold standard was hand-constructed (the PARC 700 Dependency Bank: King et al. 2003). These 700 sentences were then parsed using the English grammar, and the parse with the highest f-score[1] was chosen; these parses had an average f-score of 84.1% relative to the gold standard. The average f-score for a randomly-selected parse relative to the gold standard was 78.6%, still quite high.

[1] F-score (van Rijsbergen 1979) is an overall score representing a combination of precision and recall:

$$\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Precision is defined as the number of matches between the system output and the gold standard out of all items in the system output, and recall is the number of matches out of all items in the gold standard (Riezler et al. 2003).

6.3.3 ParGram tags

Table 1 contains the 115 preterminal node labels used in the ParGram English grammar, referred to as "tags" in the following. These have been divided into the following groups:

(4) a. Verbal, including auxiliaries
    b. Nominal
    c. Adverbial
    d. Prepositional
    e. Punctuation
    f. Other

With 115 tags, the ParGram tagset is more fine-grained than, for example, the Penn tagset, which has 48 tags (Marcus et al. 1994b). The relatively large size of the ParGram tagset does not in itself pose a problem for tagger training; as shown by Elworthy (1995), larger tagsets are not necessarily less accurate than smaller ones, at least for languages like English. Indeed, much larger tagsets have been used in tagger training: Toutanova et al. (2002) report reasonable results in training a tagger for English with the tagset for the HPSG Redwoods treebank. This tagset incorporates very detailed syntactic and semantic information, and assumes a tagset of 8,000 lexical tags.

Some of the ParGram English tag distinctions reflect subdivisions of standard phrase structure categories which are used in different syntactic contexts, such as the various verbal tags: V[fin] for finite verbs, V[perf] for perfect participles, V[prog] for progressive participles, and so on. Complex category labels such as these are used extensively in the ParGram grammars; see Butt et al. (1999:13.2.2) for discussion. Such morphologically-encoded differences should in principle be possible for a standard tagger to discriminate. Other tags directly reflect structural syntactic ambiguities, however: for example, the word *and* can be tagged either as CONJ (used in non-nominal conjunction) or as CONJnp (used in conjunction of noun phrases). Such distinctions may well be very difficult for a tagger to make.

On the other hand, the ParGram tagset is less fine-grained than tagsets such as those used by Toutanova et al. (2002) in their experiments with the HPSG Redwoods treebank or by Bangalore and Joshi (1999) and Clark and Curran (2004) for supertagging in Combinatory Categorial Grammar and Tree Adjoining Grammar. These tagsets contain much more detailed syntactic information than the ParGram tagset about the syntactic environment in which a word can appear.

When using a more fine-grained tagset which reflects syntactic ambiguities, we hypothesize that different syntactic analyses of a string

TABLE 1 ParGram English grammar preterminals

Verbal tags	AUX[fut,fin]	AUX[modal,fin]	
	AUX[pass,base]	AUX[pass,fin]	
	AUX[pass,perf]	AUX[pass,prog]	
	AUX[perf,base]	AUX[perf,fin]	
	AUX[perf,prog]	AUX[prog,base]	
	AUX[prog,fin]	AUX[prog,perf]	
	AUXdo[base]	AUXdo[fin]	
	AUXsubj[fin]	V[base]	
	V[fin]	V[pass]	
	V[perf]	V[prog]	
	Vcop[base]	Vcop[fin]	
	Vcop[perf]	Vcop[prog]	
Nominal tags	N	NAME	NP[int]
	NP[rel]	Ndate	Npart
	PRON	PRON[int]	PRON[rel]
	PRONemph	PRONfree	PRONheadless
	PRONposs	PRONpp	DAY
	HOUR	MN	MONTH
	TITLE		
Adverbial tags	ADV	ADVadj	
	ADVadj[post]	ADVadj[pre]	
	ADVcomp	ADVcompmod	
	ADVcoord	ADVdate[any]	
	ADVdate[fin]	ADVdet	
	ADVfoc	ADVinf	
	ADVint	ADVnum	
	ADVpmod	ADVtime	
Prepositional tags	P	Padj	Pnum
	Ppart	PP	PP[int]
	PP[rel]	PPcl	
Punctuation and non-word tags	CCOLON	CDASH	COLON
	COMMA	CSEMI-COLON	DASH
	ELLIPSIS	HYPHEN	INT-MARK
	L-CRL	L-PRN	L-QT
	LD-QT	PERIOD	R-CRL
	R-PRN	R-QT	RD-QT
	SEMI-COLON	U-QT	TOKEN
Other tags	A	Adate	Aquant
	CONJ	CONJcomp	CONJnp
	CONJsub	C[inf]	C[int]
	C[pred]	C[that]	D
	D[int]	Dcomp	INITIAL
	LETTER	NEG[con]	NEG[full]
	NUMBER	PA	PART
	PARTinf	POSS	PRE-N
	PRE-V	PRECONJ	PREDET
	PREint		

tend to be reflected in different tags for the words in the string; conversely, if a coarser-grained tagset is used, syntactic ambiguity tends not to be reflected in tags. If this is true, this study represents a middle ground for evaluation of syntactic disambiguation by tagging, since the ParGram tagset contains more syntactic information relevant for disambiguation than the Penn tagset, but less than is available from supertags.

6.3.4 The data

The current study examined sentences from two sections of the Wall Street Journal corpus (Marcus et al. 1994a). The first dataset consists of all of the sentences from Section 13; this was chosen as a large sample, but with no gold standard. The second dataset consists of 100 sentences hand-selected from Section 23 in which the correct parse was marked; this was chosen as a small gold-standard sample for comparison with the larger sample. These sentences were parsed using the XLE parsing platform and the ParGram English grammar.

The results for Section 13 were:

(5)	Total sentences in Section 13	2,481
	Full and fragmentary parses	2,429
	Nonfragmentary parses with extracted tag sequences	2,105

Of the 2,481 sentences in Section 13, 2,429 sentences obtained a complete (but possibly fragmentary) parse in the time allowed. Skimming was not used; in skimming mode, the parser may return a result containing only a subset of the parses licensed by the grammar (Riezler et al. 2002). From the sentences with full and complete parses, tags were extracted from 2,105 sentences. Tags were not extracted from sentences which obtained only a fragmentary parse, nor (because of resource limitations) from 5 sentences which had more than 80,000 full parses. For the sentences in Section 13, the correct parse was not marked, so there is no gold standard for evaluation of Section 13.

The 100 sentences chosen from Section 23 constitute a much smaller corpus. In creating this corpus, the sentences in Section 23 were parsed in sequence; sentences with a full and correct parse were banked, with the correct parse marked, until a 100-sentence treebank corpus had been produced. These 100 sentences all received a nonfragmentary parse, and tag sequences were extracted for each sentence.

(6)	Total sentences selected from Section 23	100
	Nonfragmentary parses with extracted tag sequences, correct parse marked	100

The 100-sentence Section 23 corpus tends to contain slightly shorter

and less ambiguous sentences:

(7)

	Num. sentences	Mean sentence length (words)	Mean num. parses	Median num. parses
Section 13	2,105	21.6	429	12
Section 23	100	18.2	63	8

Comparison of the two corpora is therefore difficult; the bulk of the study described below is conducted on the basis of the larger Section 13 corpus.

From the packed parse forest for each sentence in both corpora, a file was produced containing the tag sequence (the sequence of preterminal categories) for each parse. The contents of an example tagfile for the sentence *Hess declined to comment* is given in (8). The sentence has three parses. In the first parse, *to* has category P (preposition) and *comment* has category N (*decline* is intransitive in this case, and *to comment* is a prepositional phrase). In the second and third parses, *to* has category PARTinf (infinitival particle) and *comment* has category V[base] (one parse is the expected one, where *to comment* is an infinitival clause and an argument of *declined*; in the other parse, *declined* is intransitive and *to comment* is an infinitival purpose modifier).

(8)

NAME:Hess	V[fin]:declined	P:to	N:comment
NAME:Hess	V[fin]:declined	PARTinf:to	V[base]:comment
NAME:Hess	V[fin]:declined	PARTinf:to	V[base]:comment

Next, the tag sequences were grouped into equivalence classes. The tag sequences in (8) would be grouped into two equivalence classes, with the first parse in one class and the second and third parses in the other class. Example statistics for sentences 1000-1005 are given in (9).

(9)

Sentence	Number of words	Number of parses	Num. tag sequence equivalence classes
wsjS1000.tags	26	2	1
wsjS1001.tags	9	1	1
wsjS1002.tags	15	15	6
wsjS1003.tags	27	49	18
wsjS1004.tags	28	640	8
wsjS1005.tags	31	320	2

The results reported below are based on these equivalence class groupings.

6.4 Analysis

620 (29.45%) of the sentences in Section 13 had parses whose tag sequences all fell into one equivalence class; these sentences had between 1 and 320 parses, with 225 of the 620 sentences receiving only 1 parse. One sentence had parses falling into 600 equivalence classes (20 words, 7,336 parses); all the others had parses falling into 234 or fewer equivalence classes.

(10)

Num. tag seq. equiv. classes	Number of sentences	Number of parses	Cumulative percentage
1	620	1–320	29.45%
2	526	2–4,608	54.44%
3	102	3–4,758	59.29%
4-10	591	4–13,584	87.37%
11-20	143	12–30,576	94.16%
21-50	84	48–62,464	98.15%
51-100	21	176–82,704	99.15%
101-234	17	448–75,152	99.96%
600	1	7,336	100.00%
total	2,105	903,765	

In answering the question of whether tagging is useful in disambiguation, the most relevant statistic is the proportion of sentences whose tag sequences all fall into one equivalence class: tagging would not help to disambiguate 29.45% of the sentences in this corpus, while it would help with the remaining 70.53%.

The relation between degree of ambiguity and number of tag equivalence classes is given in (11):

(11)

Num. tag seq. equiv. classes	Number of parses	Mean num. of parses	Median num. of parses
1	1–320	7.18	2
2	2–4,608	46.04	8
3	3–4,758	103.96	12
4-10	4–13,584	198.38	36
11-20	12–30,576	943.30	179
21-50	48–62,464	2154.96	584
51-100	176–82,704	8672.48	2,484
101-234	448–75,152	12448.82	7,496
600	7,336	7,336	7,336

Somewhat surprisingly, the number of tag equivalence classes does not correlate well with either the length of the input string or with the number of parses of the sentence. Only 5% of the variation in tag

sequence classes is accounted for by sentence length, and only 16% by the number of parses:

(12)	Sentence length	Num. parses
Num. tag sequence equiv. classes	r=.2260	r=.3967
	r^2=.0510	r^2=.1574

This means that it is difficult to tell in advance whether tagging would be helpful: tagging only the longer sentences, for example, is not guaranteed to be the best strategy.

6.4.1 Ambiguity reduction from tagging

Our problem is to estimate the degree of ambiguity reduction that could be obtained if the correct tag sequence is specified for each sentence by the "perfect tagger". Specifying a tag sequence amounts to choosing a particular tag sequence equivalence class for a sentence; as a result of choosing an equivalence class as the correct one, the parses in other equivalence classes are ruled out. Therefore, the size of the equivalence class containing the correct parse represents the degree to which ambiguity can be reduced by tagging.

For the sentences in Section 13, the correct parse is not marked, and so it is not possible to determine which equivalence class contains the right parse. One way to guess how much ambiguity reduction is available by tagging is to compute the average size of the tag sequence equivalence classes. For Section 13, the tag sequence equivalence classes are distributed as follows:

(13)		
	Mean size of equivalence classes, Section 13:	14.31%
	Median:	4.16%

The equivalence classes have an average size of 14.31% of the total number of parses, with a median size of 4.16% of the parses, meaning that 85-95% of the parses can be ruled out by randomly choosing a tag sequence for an input string.

Again, however, it is not possible to determine for Section 13 which tag sequence equivalence class contains the correct parse. It may well be that the correct parse is usually contained in the *largest* tag sequence equivalence class (since most parses are contained in that class). This is the worst case for disambiguation by the tagger, since the smallest number of parses is ruled out if the largest tag sequence equivalence class turns out to be the correct one. For Section 13, the average size of the *largest* tag sequence equivalence class is:

(14)		
	Mean size of *largest* equivalence class, Section 13:	55.55%
	Median:	50.00%

If the largest tag sequence is taken to be the correct one, 45-50% of the potential parses for the sentence can be ruled out in the average case.

To decide which figure better reflects how much disambiguation can be expected from tagging, our corpus of 100 parses from Section 23 of the Wall Street Journal is relevant, since in this corpus the correct parse for each of the sentences has been hand-selected. For this corpus, the results are:

(15) Mean size of *correct* equivalence class, Section 23: 54.83%
Median: 50.00%

This result is surprisingly close to the result obtained by always choosing the largest equivalence class from the Section 13 corpus. If these data are representative, an average of 45-50% of the potential parses for a sentence can be ruled out by choosing the correct tag sequence for the sentence.

This result accords well with results obtained by Toutanova et al. (2002) in their work on disambiguation with the HPSG LinGO grammar. As outlined above, Toutanova et al. investigated several disambiguation techniques in parsing sentences from the HPSG Redwoods treebank, including tagging the input in a preprocessing step. Since the Redwood corpus represents a very large gold standard corpus, they can identify the correct tag sequence for each sentence, and determine how much disambiguation the "perfect tagger" would provide. They report a correct tag sequence equivalence class size of 54.59%, very close to the results found in the current study. This convergence of results is all the more surprising given the very different granularity of the tagsets: the ParGram English tagset contains 115 tags, while the HPSG Redwoods tagset contains 8,000 tags.

One difference between the two studies is that Toutanova et al. report results only for sentences that have more than one parse, and for which disambiguation is therefore an issue. Because it is impossible to know before parsing a sentence whether it is ambiguous or not, and because this study addresses the utility of a tagger as a preprocessor and not as a means of selecting the correct parse from the output of a parser, all sentences in our corpus have been included in the results reported above, not just the sentences that have more than one parse.

Our results are, of course, dependent on the grammar that was used in the experiment. Like the HPSG LinGO grammar, the ParGram grammar produces linguistically rich, detailed analyses in which subcategorization and other grammatical requirements must be satisfied; analyses which violate these requirements are ruled out by the grammar and do not appear in the output of the parser. It may be that tagging

would play a greater role in ambiguity reduction for looser, less constrained grammars which do not encode or enforce such grammatical requirements.

6.5 Ambiguous words

Identifying the tags that are most often involved in cases of ambiguity provides useful information for developers of taggers as well as for grammar writers to tune large-scale grammars and reduce unnecessary ambiguity.

Since the correct parse for the sentences in Section 13 is not marked, the correct parse cannot be compared to the rest of the parses to determine which words are most often tagged incorrectly. However, research conducted by Riezler et al. (2002) shows that the average quality of the analyses produced by the ParGram English grammar is quite high, with a randomly-selected parse getting an average f-score of 78.6%. It is possible, then, to perform the following experiment: first, an arbitrary parse is chosen as the standard to evaluate against. Then, the number of instances of disagreement relative to this arbitrarily-selected parse is recorded, where disagreement is defined as an instance of tag mismatch between a parse and the standard. Only parses with the same tokenization as the arbitrarily-chosen standard are considered, and parses with different tokenizations are discarded.

Table 2 contains the confusion matrix for disagreements representing at least 1% of the total disagreements in the data. Three entries in the table are worthy of note.

Tag disagreement between the category A (adjective) and N (noun) accounted for 29.63% of cases of disagreement between the arbitrarily-selected standard and the other parses (summing together the 21.86% of cases where the arbitrarily-chosen standard had category A and the other parses had N, and the 7.77% of cases where the standard had N and the other parses had A). The reason for this is that there are many words that can be used either as an adjective or as a noun, and it is difficult to allow only an adjective + noun parse for these cases and disallow a noun-noun compound parse. For example, the most obvious parse for a phrase like *green box* is the one where *green* is an adjective, but in order to parse examples like *Green is my favorite color*, *green* is also analyzed as a noun; thus, *green box* gets a noun-noun compound parse like the parse for *music box*.

Tag disagreement between the category CONJ (conjunction) and CONJnp (a special category for noun phrase conjunction) accounted for 7.21% (5.10% + 2.11%) of cases of disagreement. On the ParGram

	A	ADV	C[that]	CONJ	CONJnp	N	NAME	P	PA	PARTinf	V[base]	V[fin]	V[pass]	V[prog]
A		1.45				21.86								3.61
ADV	2.11													
CONJ					5.10									
CONJnp				2.11										
CONJsub								3.56						
D									2.23					
N	7.77						1.67				1.73	4.56		6.48
P										1.67				
PRON			3.17											
V[fin]						4.68							2.75	
V[prog]	1.20					1.83								

TABLE 2 Confusion matrix for tag mismatches >1%, Section 13. The preterminal symbols are defined as:

A	adjective	ADV	adverb
CONJ	conjunction	CONJnp	conjunction for noun phrases
CONJsub	subordinating conjunction	C[that]	*that* as complementizer
D	determiner	N	noun
NAME	proper name	P	preposition
PA	*a* in constructions like *5 times a day*	PARTinf	*to* as infinitival marker
PRON	pronoun	V[base]	base (citation) form of verb
V[fin]	finite verb	V[pass]	passive participle form of verb
V[prog]	progressive form of verb		

English grammar analysis of coordination, the lexical category of the conjunction reflects structural ambiguity arising from differences in scope of coordination. The grammar writer could alternatively have made a different choice: to use the same preterminal category for conjunctions used within noun phrases as in coordination more generally. If that had been done, the number of parses would have remained the same (the same structural ambiguities for scope of coordination would have been available), but these would not have been reflected in different tag sequences for the different possibilities: given such a grammar, choosing a tag sequence would not disambiguate between different coordination possibilities.

Tag disagreement between the category V[prog] and the category N accounted for 8.31% (1.83% + 6.48%) of cases of disagreement. This is because present participle forms like *swimming* can be analyzed either as progressive verbs (*He is swimming*) or as gerunds (*Swimming is fun*). Ambiguity can arise in a variety of situations: the most pernicious is in seemingly simple cases like *He is swimming*, which has besides the obvious present progressive reading, a reading where *is* is analyzed as a copula and *swimming* is a gerund, with a meaning something like *he is (the concept of) swimming*. Again, it is difficult to rule out such examples in a non-ad-hoc manner.

6.6 Conclusion

Examining the output of the ParGram English grammar allows us to assess the effect of incorporating a tagger into a large-scale processing system in the best case, abstracting away from errors that would inevitably be introduced by even the best tagger currently available. Results of the study show that a perfect tagger would reduce ambiguity by about 50%. Somewhat surprisingly, about 30% of the sentences in the corpus that was examined would not be disambiguated, even by the perfect tagger, since all of the parses for these sentences shared the same tag sequence. For at least some of the difficult cases, in particular ambiguity corresponding to scope of coordination, it is not at all clear whether a tagger could be expected to do better than a parser in any case.

Acknowledgments

A conversation with Ron Kaplan was the inspiration for this study, and I have benefited from discussion with him on every aspect of this work, particularly the statistical analysis. I've learned a lot from writing this paper, which is very different from most of the linguistic work I've done

before. Besides Ron, I'm grateful for very helpful discussion to Steve Clark, Ken Kahn, David Kahn, Tracy King, Steve Pulman, Heike Zinsmeister, two anonymous reviewers for *Natural Language Engineering*, and audiences at University of Brighton, University of Sheffield, and University of Oxford, and to Matty Dalrymple for help with Excel.

References

Alshawi, Hiyan, ed. 1992. *The Core Language Engine*. Cambridge, MA: The MIT Press.

Bangalore, Srinivas and Aravind Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics* 25(2):237–266.

Butt, Miriam, Tracy H. King, María-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. Stanford, CA: CSLI Publications.

Butt, Miriam, Helge Dyvik, Tracy H. King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), Workshop on Grammar Engineering and Evaluation*, pages 1–7. Taipei, ROC.

Charniak, Eugene, Glenn Carroll, John Adcock, Anthony Cassandra, Yoshihiko Gotoh, Jeremy Katz, Michael Littman, and John McCann. 1996. Taggers for parsers. *Artificial Intelligence* 85(1–2).

Clark, Stephen and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*, Geneva, Switzerland.

Copperman, Max and Frédérique Segond. 1996. Computational grammars and ambiguity: The bare bones of the situation. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 1996 (LFG'96)*. Grenoble, France: CSLI Online Publications.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell, III, and Annie Zaenen, eds. 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*, vol. 34 of *Syntax and Semantics*. New York, NY: Academic Press.

Dickinson, Markus and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 107–114. Budapest, Hungary.

Elworthy, David. 1995. Tagset design and inflected languages. In *From Texts to Tags: Issues in Multilingual Text Analysis: Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL'95)*. Dublin, Ireland.

Kaplan, Ronald M. and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, ed., *The Mental Representation of Grammatical Relations*, pages 173–281. Cambridge, MA: The MIT Press. Reprinted in Dalrymple et al. (1995:29–130).

Kaplan, Ronald M. and Tracy H. King. 2003. Low-level markup and large-scale LFG grammar processing. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2003 (LFG'03)*. Albany, NY: CSLI Online Publications.

King, Tracy H., Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 dependency bank. In A. Abeillé, S. Hansen-Schirra, and H. Uszkoreit, eds., *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), 4th Workshop on Linguistically Interpreted Corpora (LINC'03)*. Budapest, Hungary.

Marcus, Mitchell, Grace Kim, Mary A. Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994a. The Penn treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 114–119. San Francisco: Morgan Kaufmann.

Marcus, Mitchell P., Beatrice Santorini, and Mary A. Marcinkiewicz. 1994b. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics* 19(2):313–330.

Maxwell, John T., III and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics* 19(4):571–590.

Maxwell, John T., III and Ronald M. Kaplan. 1996. An efficient parser for LFG. In M. Butt and T. H. King, eds., *On-line Proceedings of the LFG96 Conference*.

Oepen, Stephan, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods treebank: Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, Taipei, ROC.

Prins, Robbert and Gertjan van Noord. 2001. Unsupervised POS-tagging improves parsing accuracy and parsing efficiency. In *Proceedings of the 7th International Workshop on Parsing Technologies (IWPT'02)*. Beijing, China.

Prins, Robbert and Gertjan van Noord. 2003. Reinforcing parser preferences through tagging. *Traitement Automatique des Langues* 44(3):121–139.

Pulman, Steve. 1992. Using tagging to improve analysis efficiency. In H. Thompson, ed., *SALT/ELSNET Workshop on Sub-Language Grammar and Lexicon Acquisition for Speech and Natural Language Processing*. Record of verbal presentation.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 271–278. Philadelphia, PA.

Riezler, Stefan, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for Lexical-Functional Grammar. In *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*.

Toutanova, Kristina, Christopher Manning, Stuart Shieber, Dan Flickinger, and Stephan Oepen. 2002. Parse disambiguation for a rich HPSG grammar. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT'02)*, pages 253–263. Sozopol, Bulgaria.

Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference and the 3rd Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*. Edmonton, Canada.

van Rijsbergen, C. J. 1979. *Information Retrieval*. London, United Kingdom: Butterworth-Hienemann, 2nd edn.

Voutilainen, Atro. 1998. Does tagging help parsing? A case study on finite-state parsing. In *Proceedings of Finite-State Methods in Natural Language Processing (FSMNLP'98)*, Ankara, Turkey.

Wauschkuhn, Oliver. 1995. The influence of tagging on the results of partial parsing in German corpora. In *Proceedings of the 4th International Workshop in Parsing Technologies (IWPT'95)*. Prague, Czech Republic.

7

Rapid Treebank-Based Acquisition of Multilingual LFG Resources

Josef van Genabith

7.1 Introduction

Treebank-based acquisition of "deep" grammar resources is motivated by the "knowledge acquisition bottleneck" familiar from other traditional, knowledge intensive, rule-based approaches in AI and NLP following the "rationalist" research paradigm. Deep grammatical resources have usually been hand-crafted. This is time consuming, expensive and difficult to scale to unrestricted text. Treebanks (parse-annotated corpora) have underpinned an alternative "empiricist" approach: wide-coverage, robust probabilistic grammatical resources are now routinely extracted (learned) from treebank resources (Charniak 1996, Collins 1997, Charniak 2000). Initially, however, these resources were "shallow". More recently, a considerable amount of research has emerged on treebank-based acquisition of deep grammatical resources in the TAG, HPSG, CCG and LFG grammar formalisms (Joshi 1987, Pollard and Sag 1994, Steedman 1996, Kaplan and Bresnan 1982). This paper provides an overview of research on rapid treebank-based acquisition of wide-coverage, robust, probabilistic, multilingual LFG resources. Section 7.2 draws a distinction between "deep" and "shallow" grammars. Section 7.3 outlines LFG. Section 7.4 summarises early proof-of-concept research on deriving LFG resources from treebanks. Section 7.5 surveys the current state-of-the-art in the acquisition of treebank-based LFG resources (including automatic f-structure annotation, lexicon extraction, parsing architectures and evaluation), provides the main re-

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

sults achieved and outlines ways of improving the resources. Most of the work presented in Section 7.5 is focused on English. Section 7.6 provides an overview of research on multilingual treebank-based LFG resources for German, Spanish and Chinese and describes work under way on the full-scale induction of Chinese, Japanese, Arabic, Spanish, French and German LFG resources. Section 7.7 presents work on robust probabilistic generation and Section 7.8 outlines ongoing research on the acquisition of resources for LFG- and transfer-based probabilistic machine translation. Section 7.9 concludes.

7.2 Shallow and Deep Grammars

Deep grammars relate strings to "information" (meaning representations) in the form of predicate-argument structures, dependencies or logical forms. Linguistic material is not always interpreted locally where it occurs in the string (or tree): "displaced" material often needs to be interpreted semantically elsewhere as, for example, an argument of a non-local predicate. In order to construct accurate and complete predicate-argument structures, deep grammars usually include a mechanism to resolve long distance dependencies (LDDs).

Like deep grammars, shallow grammars define languages as sets of strings and may associate syntactic structures with strings. Unlike deep grammars, shallow grammars do not associate strings with "information" and they do not usually involve LDD resolution.[1]

7.3 Lexical Functional Grammar (LFG)

LFG (Kaplan and Bresnan 1982, Bresnan 2001, Dalrymple 2001) is an early member of the family of constraint-based grammar formalisms including FUG, GPSG and HPSG (Kay 1985, Gazdar et al. 1985, Pollard and Sag 1994). Minimally, LFG involves two levels of representation: c(onstituent)-structure and f(unctional)-structure. C-structure represents word order, the grouping of words into phrases and the hierarchical structure among phrases in terms of context-free tree representations. F-structure abstracts away from surface representation and captures abstract grammatical functions (such as SUBJ(ect), OBJ(ect), OBL(ique)) in terms of attribute-value structures. C-structure and f-structure are related in terms of f-descriptions (attribute-value structure equations) associated with c-structure representations. Technically, f-structures are canonical representations of minimal models satis-

[1] Note that in other contexts the term "shallow grammar", or more often "shallow parser", refers to partial syntactic analyses, such as chunking, where typically the analysis does not result in a complete and connected hierarchical tree structure but rather in a flat, partial bracketing of an input string.

fying sets of equations. Although primarily an abstract syntactic representation, f-structures approximate to basic predicate-argument structures, deep dependencies or simple logical form (van Genabith and Crouch 1996, 1997).

7.4 Early Work in Treebank-Based Acquisition of LFG Resources

Lappin et al. (1989) is probably the first attempt to automatically identify LFG-like grammatical functions in context-free tree parser output (in order to facilitate the statement of transfer rules in an English-Hebrew machine translation system).

The earliest work on automatically acquiring LFG resources from treebanks is Kaplan (1996, p.c.). In order to generate a (training and evaluation) corpus for LFG-DOP (Bod and Kaplan 1998) parsing experiments, Kaplan implemented a procedure to recursively and destructively transform trees in the ATIS section (Hemphill et al. 1990) of the Penn-II treebank (Marcus et al. 1994) into f-structures.

Van Genabith et al. (1999) report on experiments on a non-destructive, annotation-based method to generate f-structures and LFG resources from treebank trees: they (i) extract CFG rule types from the publicly available subset of the AP treebank (Leech and Garside 1991), (ii) manually annotate the CFG rule types with LFG f-structure equations, (iii) automatically rematch the f-structure annotated CFG rules against the original treebank trees to obtain f-structure annotated treebank trees, (iv) for each tree automatically collect the f-structure equations and use a constraint solver to generate an f-structure for the treebank tree. Van Genabith et al. (1999) report on experiments on extracting lexical resources (subcategorisation frames) from the f-structures generated from the treebank trees. The research was small scale (the first 100 trees of the publically available subset of the AP treebank) and it became clear that it would be extremely difficult and time-consuming to scale the manual annotation part of the method to, for example, the more than 19,000 CFG rule types in the complete Penn-II treebank (Marcus et al. 1994).

Frank (2000) and Sadler et al. (2000) report research on replacing the manual f-structure annotation part of van Genabith, Way, and Sadler (1999) with automatic f-structure annotation methods. Sadler et al. (2000) present a regular expression-based method and software where f-structure annotations are expressed in terms of encoding annotation principles as regular expressions over CFG rule left-hand sides and right-hand sides. CFG rule sets are extracted from a treebank, automat-

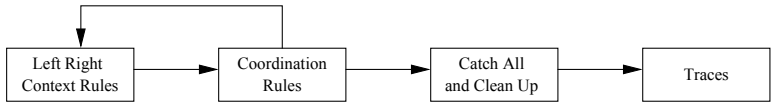


FIGURE 1 F-Structure Annotation Algorithm Components

ically annotated with f-structure equations and rematched against the original trees to produce trees with f-structure equations, from which f-structures are generated. Frank (2000) presents a method where treebank trees are automatically translated into a flat set of terms in a tree description language. F-structure annotation principles are stated in terms of a rewriting system originally designed for transfer-based machine translation, rewriting the flat tree descriptions into descriptions of f-structures. Both approaches (Sadler et al. 2000, Frank 2000) constitute principle-based LFG c-structure/f-structure interfaces and are further described and compared in detail in Frank et al. (2003). Both approaches are proof-of-concept and small scale — involving 100 AP (Leech and Garside 1991) and 166 SUSANNE (Sampson 1995) trees.

7.5 F-structure Annotation Algorithm, Subcategorisation Frame Extraction, Parsing Architectures and Evaluation

7.5.1 F-structure Annotation Algorithm

Cahill et al. (2002, 2004), McCarthy (2003) and Burke (2006) present a new automatic f-structure annotation architecture that scales to the full Penn-II (Marcus et al. 1994) treebank. The architecture is based on an f-structure annotation algorithm that traverses Penn-II trees and annotates phrasal nodes in the trees with LFG f-structure equations. Lexical information is provided in terms of macros associated with Penn-II POS classes.[2] The annotation algorithm is modular with 4 components (Figure 1).

The Left-Right Annotation component identifies the heads of Penn-II trees using a modified version of the head finding rules of Magerman (1994). This partitions each local subtree (of depth one) into a local head, a left context (left sisters) and a right context (right sisters). The contexts, together with information about the local mother and daughter categories and (if present) Penn-II functional tags, is used by the f-structure annotation algorithm. For each Penn-II mother (i.e. phrasal)

[2]To give an example, Penn-II POS-word sequences of the form `VBZ word` are automatically associated with the equations (↑PRED)=‘`word`’, (↑NUM)=sg, (↑PERS)=3 and (↑TENSE)=pres, where ‘`word`’ is the lemmatised `word`.

category an "annotation matrix" expresses generalisations about how to annotate immediate daughters dominated by the mother category relative to their location in relation to the local head. To give a (much simplified) example, the head finding rules for NPs state that the rightmost nominal (NN, NNS, NP, ...) not preceded by a comma or "–"[3] is likely to be the local head. The annotation matrix for NPs states (inter alia) that heads are annotated ↑=↓, that DTs (determiners) to the left of the head are annotated (↑ SPEC DET) = ↓, NPs to the right of the head as ↓∈ (↑ APP) (appositions). For each phrasal category, annotation matrices are constructed by inspecting the most frequent rule types expanding the category such that the token occurrences of these rule types cover more than 85% of all occurrences of expansions of the category in Penn-II. For NP rules, for example, this means that we analyse the most frequent 102 rule types expanding NP rather than the complete set of more than 6,500 Penn-II NP rule types in order to populate the NP annotation matrix. Annotation matrices generalise to unseen rule types as, in the case of NPs, these may also feature DTs to the left of the local head and NPs to the right, and similarly for rule types expanding other categories.

In order to support the modularity, maintainability and extendability of the annotation algorithm, the Left-Right Annotation Matrices apply only to local trees of depth one which do not feature coordination. This keeps the statement of Annotation Matrices perspicuous and compact.

The Penn-II treatment of coordination is (intentionally) flat. The annotation algorithm has modules for like- and unlike-constituent coordination. Coordinated constituents are elements of a COORD set and annotated ↓∈ (↑ COORD). The Coordination module reuses the Left-Right context annotation matrices to annotate any remaining nodes in a local subtree containing a coordinating conjunction.

The Catch-All and Clean-Up module provides defaults to capture remaining unannotated nodes (Catch-All) and corrects (Clean-Up) overgeneralisations resulting from the application of the Left-Right Context Annotation Matrices. The Left-Right Annotation Matrices are allowed a certain amount of overgeneralisation as this facilitates the perspicuous statement of generalisations and a separate statement of exceptions, supporting the modularity and maintainability of the annotation algorithm.

The Traces Module translates traces and coindexed material in

[3]If the rightmost nominal is preceded by a comma or "–", it is likely to be an apposition to the head.

TABLE 1 Quantitative F-Structure Evaluation Penn-II (Burke 2006)

# F-Str	# Trees	% Treebank
0	45	0.09
1	48329	99.80
2	50	0.10

Penn-II trees representing long-distance dependencies and control information into corresponding reentrancies at f-structure. Figures 2 and 3 show a Penn-II style tree before and after automatic annotation (lexical equations not given here) and the f-structure generated.

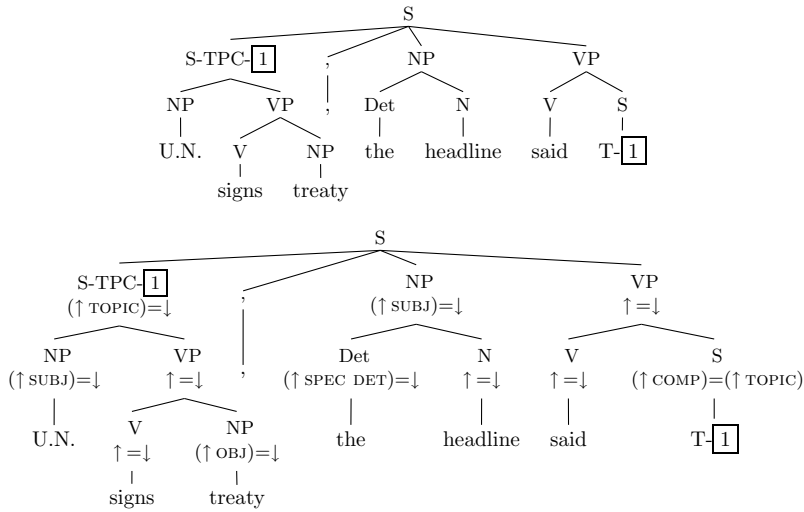


FIGURE 2 Penn-II style trees before/after automatic f-structure annotation.

The f-structure annotation algorithm is evaluated quantitatively and qualitatively. Quantitative evaluation measures annotation coverage in terms of the number of Penn-II trees that receive a covering and connected f-structure, no f-structure (for unresolvable f-descriptions), or a number of f-structure fragments (Table 1). The annotation algorithm achieves near complete coverage of the more than 48,000 Penn-II trees (without FRAG and X constituents), with 45 trees not receiving an f-structure (due to feature clashes) and 50 trees receiving 2 f-structure fragments each.

Qualitative evaluation measures annotation quality against gold-standards. To date, annotation quality has been evaluated against three

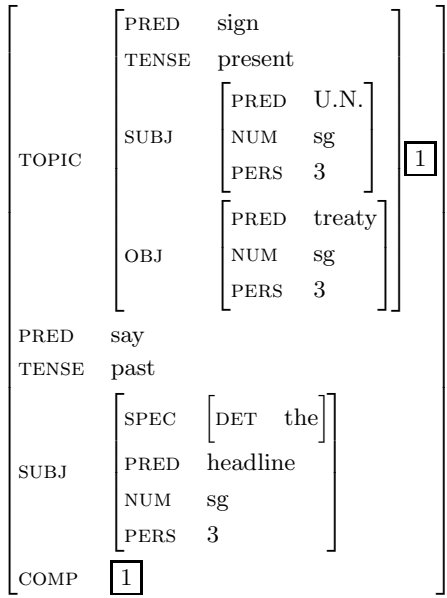


FIGURE 3 F-structure for annotated tree in Figure 2.

gold-standards: the DCU 105 F-Structure Bank, the PARC 700 Dependency Bank (King et al. 2003) and the semantic role-based PropBank (Kingsbury et al. 2002). In each case, we use the evaluation software of Crouch et al. (2002). All results reported below are from Burke (2006). Against the DCU 105, the annotation algorithm achieves an f-score of 96.93% for all grammatical functions and 94.28% for preds-only.[4] Against the PARC 700, the algorithm achieves 87.33% f-score for the feature set in Kaplan et al. (2004) and 84.45% for preds-only. Against PropBank, the algorithm achieves an f-score of 76.58%.[5] In contrast to the DCU 105 evaluation, the PARC 700 and the PropBank evaluations require extensive automatic mapping of our feature structures (dependencies) to the gold-standard dependencies due to systematic differences in feature nomenclatura and feature geometry in the analyses. The mappings are described extensively in Burke et al. (2004a), Burke et al. (2005) and Burke (2006).

[4]Preds-only only measures paths in the f-structure that end in a value for a PRED attribute, i.e. the predicate-argument-adjunct backbone of the f-structure.

[5]The PropBank evaluation results are preliminary and mostly based on simply associating an LFG grammatical function with a PropBank semantic role. More sophisticated mappings are possible (Miyao and Tsujii 2004).

Extraction of Lexical Resources

If the f-structures produced by the annotation algorithm are of good quality, then good quality subcategorisation frames (LFG semantic forms) can be extracted as follows: given an f-structure, for each level of embedding, determine the local PRED and the subcategorisable grammatical functions present at that level (van Genabith, Sadler, and Way 1999, O'Donovan et al. 2004, 2005a, O'Donovan 2006). The method can be applied to f-structures generated from treebank trees or parser output for raw text using the LFG parsing architectures presented in the next section. In general, treebank-based f-structures will yield high quality, but limited coverage f-structures and (hence) semantic forms, while parser-based f-structures and semantic forms can provide significantly broader coverage at a slightly reduced quality due to the noise introduced by the parser. Unlike in most other approaches, the method can extract grammatical function-based frames (RELY($\uparrow$SUBJ,$\uparrow$OBL$_{on}$)), CFG category-based frames (RELY(NP,PP$_{on}$)) and mixed grammatical function-CFG category based frames (RELY($\uparrow$SUBJ:NP,$\uparrow$OBL$_{on}$:PP$_{on}$)); the method differentiates between active and passive frames; the frames extracted fully reflect the long distance dependencies present in the source data-structures; and, finally, frames are associated with conditional (on lemma and/or voice) probabilities.

O'Donovan et al. (2004), O'Donovan et al. (2005a) and O'Donovan (2006) extract semantic forms from the Penn-II and Penn-III treebanks (Marcus et al. 1994) (with a total of approx. 1.3 million words) and evaluate the extracted resources against COMLEX (Macleod et al. 1994) and OALD (Hornby 1980). The results given below are from O'Donovan (2006). A total of 21,005 semantic form types (lemma-subcat frame pairs) for 4,362 verb lemmas are extracted from Penn-III. Using simple relative thresholds (1% and 5%)[6] and evaluating the frames for more than 3,450 active verb lemmas[7] against both COMLEX and the OALD, in each case the f-score baselines[8] are exceeded. To give one example, in evaluating full frames parameterised for prepositions (for oblique arguments) the induced resources achieve an f-score of 64.1% (baseline 56%) against OALD.

O'Donovan (2006) presents a parser-based method for the extraction of lexical resources from a 90 million word subset of the British National

[6] A relative threshold of 1% indicates that given a lemma, a frame that has less than 0.01 probability of occurring with that lemma is discarded.

[7] The number is different as the COMLEX and OALD lemmas intersect differently with the induced resource.

[8] The baseline is defined by assigning all lemmas both transitive and intransitive frames.

Corpus (BNC, Bernard 2002). The challenge in parser-based extraction is that the lexical resources extracted (subcategorisation frames) would, of course, help the parser to produce better parses (and hence improve the quality of the subcategorisation frames extracted) but, of course, those resources are not available to the parser (as they are being extracted), hence the resources that are extracted are subject to the limitations of the parser output. This is a classic chicken-and-egg problem. Fortunately, in the LFG parsing architectures presented in the next section, the only place where subcategorisation frames are used is in long-distance dependency (LDD) resolution. In order to address the chicken-and-egg problem, in the parser-based lexical extraction methodology, extraction of subcategorisation frames proceeds in two stages: first, subcategorisation frames are extracted from those (sub-)f-structures (generated from the parser output for the 90 million word section of the BNC) that do not involve or are not in the scope of LDDs. Second, these frames, together with the frames extracted from the Penn-II treebank, are then used to LDD-resolve the f-structures generated from parser output for the BNC. From the LDD-resolved f-structures, a complete set of subcategorisation frames is then extracted and evaluated against COMLEX and the OALD (as in the treebank-based extraction described above). Applying an absolute threshold of ≤ 5 (attested occurrences), semantic forms for about 7,000 verb lemmas are extracted.[9] Evaluating the more than 4,380 active lemmas[10] against COMLEX and OALD, in each case the baseline is exceeded. To give a single example, evaluating against COMLEX an f-score of 63.66% is achieved against a baseline of 51.25%.

O'Donovan (2006) also provides an evaluation against the state-of-the-art parser-based subcategorisation frame extraction system of Korhonen (2002) using the Korhonen evaluation software and her SUSANNE-based (Sampson 1995) gold standard and shows that subcategorisation frame extraction with the treebank-based LFG parsing system statistically significantly outperforms that of Korhonen with an f-score of 76.16% against 71.46%.

In an experiment to directly compare the treebank- with parser-based subcategorisation frame extraction architectures, O'Donovan (2006) uses the original treebank trees from WSJ sections 00, 01, 22, 23 and 24 with a total of 223,708 words to extract frames and then parses the strings in those same sections with a grammar trained on

[9]The exact number depends on the c-structure parsing engines (Collins 1999, Charniak 2000) used in the experiments.

[10]The exact number depends on the overlap of the extracted resources with COMLEX and OALD.

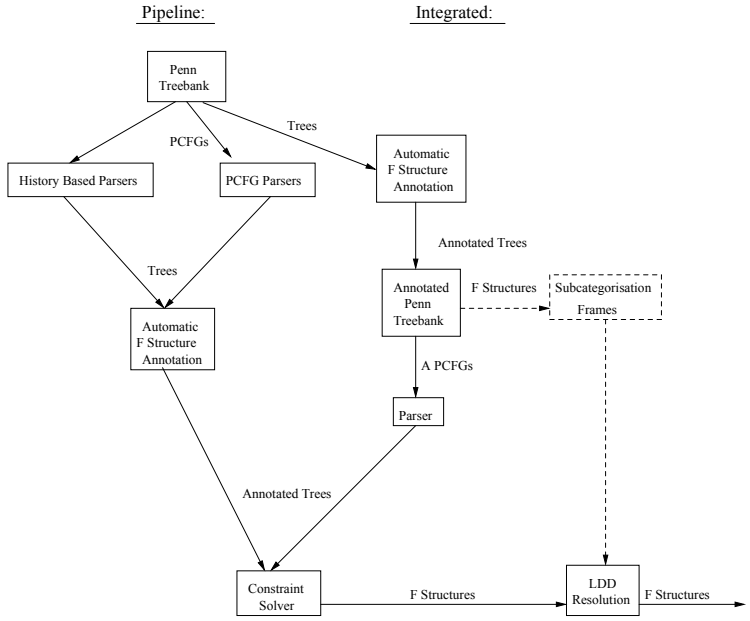


FIGURE 4 Grammar Acquisition and Parsing Architectures

sections 02-21 to extract frames from the f-structures generated from the parser output. Surprisingly, the results do not show any statistically significant drop in the quality of the frames extracted with the best parser-based LFG system against the frames extracted from the corresponding f-structure annotated gold-standard treebank trees.

Grammar Acquisition and Parsing

Cahill et al. (2002, 2004) present two flexible parsing architectures (Figure 4) based on the automatically extracted LFG resources: the pipeline architecture and the integrated architecture.

In the pipeline architecture, first PCFGs[11] extracted from WSJ sections 02-21 of the Penn-II treebank or history-based lexicalised generative parsers (Collins 1999, Charniak 2000, Bikel 2004) trained on the same sections are used to parse new text. The trees output by these parsers are then automatically annotated by the f-structure annotation algorithm, f-structures are generated and LDDs resolved (at the level of f-structure).

In the integrated architecture, the automatic f-structure annota-

[11]We use Schmid's (2004) BitPar as parsing engine for the extracted PCFGs.

tion algorithm first annotates Penn-II WSJ sections 02-21 and then *f-structure annotated* PCFGs are extracted from the f-structure annotated sections of Penn-II.[12] For grammar extraction and parsing, Penn-II CFG categories followed by f-structure annotations (equations) are treated as monadic categories. New text is parsed by the annotated PCFGs into trees with annotations, and f-structures are generated from these annotations and LDDs are resolved.

Unlike Penn-II treebank trees, most PCFG- and history-based approaches to parsing do not recover LDDs (exceptions include Collins's Model 3 (1999) and Johnson 2002), i.e. trees output by these parsers do not feature empty productions coindexed with material realised elsewhere in the tree. In LFG, LDDs are resolved at the level of f-structure using functional uncertainty equations (FUs: regular expressions over possibly unbounded f-structure paths), obviating the need for empty productions and coindexation in trees. Cahill et al. (2004) present an approach to resolving LDDs based on subcategorisation frames automatically extracted from the f-structure annotated Penn-II (O'Donovan et al. 2004), and automatically acquired finite approximations to FU equations from f-structure reentrancies in the f-structure annotated Penn-II treebank. In order to resolve a LDD, the relevant argument position must not already be filled and the local PRED must subcategorise for the "missing argument". LDD resolutions are ranked by multiplying FU path probabilities with the relevant subcategorisation frame probability, and the highest ranked solution is returned.

Cahill et al. (2006) present a detailed evaluation of the treebank-based LFG parsing resources against the PARC 700 (King et al. 2003) and the CBS 500 (Carroll et al. 1998) Dependency Banks and compare the treebank-based automatically acquired LFG resources with the hand-crafted XLE (Riezler et al. 2002, Kaplan et al. 2004) and RASP (Carroll and Briscoe 2002) parsing resources. Against the PARC 700, the treebank-based LFG resources achieve an f-score of 83.08%, a statistically significant improvement of 2.53% over the hand-crafted LFG grammar and the XLE (Kaplan et al. 2004); against the CBS 500, the treebank-based resources achieve an f-score of 80.23%, a statistically significant improvement of 3.66% over RASP (Carroll and Briscoe 2002).

7.5.2 Domain Variation

Treebank- and machine-learning based grammatical resources generally reflect the properties of their training sets. The question naturally arises

[12] We have not yet trained history-based lexicalised generative parsers in the integrated model.

as to how such resources perform on tasks that involve material that is substantially different from the training resources. A related question is how treebank-based resources can be adapted to a new domain. Gildea (2001) shows that parser performance drops under corpus variation. In order to test the impact of corpus variation on the treebank-based LFG resources, Judge et al. (2005) apply the Penn-II-induced LFG parsers to ATIS (Hemphill et al. 1990) material. ATIS is a corpus of transcribed spoken airline booking interactions across a phone line and constitutes an instance of strong domain variation for the Penn-II WSJ-based LFG resources. Judge et al. (2005) show that compared to held out Penn-II WSJ text, parser performance drops by about 11% (f-structure f-score) on ATIS material, but that retraining the c-structure engines (Bikel 2004) with a comparatively small amount of ATIS material improves performance by 9% to an all grammatical functions f-score of 83.33%. Importantly, no changes to the f-structure annotation algorithm are required for it to be applied to ATIS material, showing that the algorithm is already complete with respect to strong domain variation as exhibited by the Penn-II vs. ATIS material. In the treebank-based LFG acquisition paradigm, grammar development and adaptation takes the form of supplying small amounts of suitable training material.

7.5.3 Improvements and Future Work

Our best parser performance (currently around 83% f-structure f-score against the PARC 700) is fairly close to the upper bound (approx. 87% f-score for the f-structures derived from the original Penn-II treebank trees for the PARC 700). Our current scores can be improved in two ways: first, improvements of the f-structure annotation algorithm (to raise the upper bound) and, second, improvements to the parser output (to get closer to the upper bound).

The f-structure annotation algorithm can be improved in at least the following areas: recognition of passive voice, exploitation of configurational information to classify PPs inside NPs as adjuncts or complements, distribution of distributive grammatical functions into coordination sets, and typing of adjuncts according to Penn-II functional (-TMP, -LOC, -MAN, and so forth) tags. In addition, the algorithm can be tuned to directly produce PARC-style f-structures (with the PARC feature set and flat analyses of e.g. auxiliaries expressing tense and aspectual information). This obviates the need for lossy (both over- and under-generating) mapping software used in the PARC 700 evaluations.

Parser output can be improved in at least the following ways: currently our best results are achieved using Bikel's (2004) parser retrained to retain Penn-II functional tags. The presence of such tags is exploited

by the f-structure annotation algorithm to produce improved annotations. Following Blaheta and Charniak (2000), dedicated machine learning methods can be used to assign Penn-II functional tags to raw trees output by a parser. This might yield better results for Penn-II functional tag assignment than those achieved by retraining (Bikel 2004) (and hence improved performance of the f-structure annotation algorithm on the parser output) and, furthermore, would allow us to use a parser with higher labeled and unlabeled tree-based scores than Bikel (2004), such as Charniak (2000). Currently we have not integrated Named Entity (NE) or Multi-Word-Expression (MWE) recognition into our parsers. We would expect that integration of these would reduce parsing complexity and yield overall improved parsing results. We have not yet integrated history-based lexicalised generative parsers in the "integrated" parsing model (where parsers are trained on the f-structure annotated Penn-II treebank), but our experiments (Cahill et al. 2004) show that PCFG-based resources consistently perform best in the "integrated" model. Currently our probability models are strictly speaking mathematically inadequate, as they can leak probability mass: the f-structure equations generated for the highest ranked parse tree may not resolve to an f-structure and the probability mass associated with that tree may be lost (Abney 1997). Discriminative (log-linear) disambiguation models (e.g., Riezler et al. 2002) would provide more adequate models and may lead to improved parser output.

Unlike the CCG-based approach (and to a lesser extent the HPSG-based approach), our method does not involve a treebank clean-up phase. Hockenmaier and Steedman (2002) found that CCG grammar extraction and parsing results benefit considerably from a substantial clean-up. The result of this clean-up, the CCG-Bank, is now available (Linguistic Data Consortium) and adaptation and application of the f-structure annotation algorithm to and extraction of LFG grammatical resources from the CCG-Bank may well produce overall improved upper bounds and parsing results.

Currently there exists a curious asymmetry as regards the evaluation of treebank-based LFG, HPSG and CCG parsing resources. In CCG-style experiments, the LFG, HPSG and CCG parsers have been evaluated against automatically LFG-, HPSG- and CCG-annotated (converted) versions of the WSJ Section 23 of the Penn-II treebank. The LFG parsing resources have been evaluated against the PARC 700 and CBS 500, while the HPSG (Miyao and Tsujii 2004) and CCG (Gildea and Hockenmaier 2003) resources have been evaluated against PropBank (Kingsbury et al. 2002). Compared to PropBank, both the PARC 700 and CBS 500 are more fine-grained (they include a larger feature

set as well as information about NP internal dependencies) and come with publically available evaluation software. The PARC 700 and CBS 500 are Dependency Banks, whereas PropBank provides deep Semantic Role information.[13] In future work, we will evaluate our treebank-based LFG parsing resources against PropBank.

7.6 Rapid Acquisition of Multilingual LFG Resources

The treebank-based LFG acquisition paradigm described above was originally developed for English and the Penn-II treebank resource and data structures. English is strongly configurational and the Penn-II data structures consist of CFG trees with traces, empty productions and coindexation indicating non-local dependencies, and a number of tags that can be added to CFG categories to indicate text category, a limited number of grammatical functions (for example -SBJ) and "semantic roles" to classify adjuncts by type (for example -TMT, -LOC, -MNR). By contrast, many languages encode linguistic information morphologically and allow for much more varied word order. Currently a large number of new treebank resources are coming on line (Abeillé 2003), with a wide variety of data structures and encoding principles ranging from CFG-type tree encodings to dependency-style annotations with crossing branches. While many first generation treebanks provided only syntactic information (cf. the first version of the Penn treebank or the AP treebank), second (Penn-II) and third generation (TIGER: Brants et al. 2002) treebanks provide much additional information supporting the computation of semantic information.

The question naturally arises whether the treebank-based LFG acquisition paradigm described above can be applied to other languages, treebank encodings, data structures and treebank generations.

The basic underlying architecture of LFG with its partitioning of linguistic information into c-structure and f-structure is strongly conducive to multilingual grammar development. C-structure is the main locus of cross-linguistic variation whereas f-structure is a more abstract and cross-linguistically more stable level of representation.[14]

[13] In a sense, PropBank does not yet provide a single agreed upon gold standard: role information is provided indirectly and an evaluation gold-standard has to be computed from this. In doing so, choices have to be made as regards the representation of shared arguments and the analysis of coordinate structures (and so forth), and it is not clear that the same choices are currently made for evaluations carried out by different research groups.

[14] Of course, this is not to deny that there are important cross-linguistic differences between f-structures for sentences expressing the same proposition: argument-switching and head-switching are important examples of this.

In our multilingual treebank-based LFG acquisition paradigm, this is reflected in the fact that large sections of the "downstream" f-structure acquisition and processing components can be reused cross-linguistically. These components include the constraint solvers and LDD resolution and lexical resource extraction components. Significantly, even the generic architecture of the f-structure annotation algorithm originally developed for Penn-II is reused in our multilingual grammar acquisition research.

To date, in addition to our work on English, we have induced LFG grammars and lexical resources for three typologically strongly different languages: German (Cahill et al. 2003, 2005), Spanish (O'Donovan et al. 2005b) and Chinese (Burke et al. 2004b) from the TIGER (Brants et al. 2002), Cast3LB (Civit 2003) and CTB version 2 (Xue et al. 2004) treebank resources.

In each case, the generic architecture of the annotation algorithm carries over with head-finding rules, Left-Right Context Annotation principles/matrices, Coordination Annotation Principles, LDD resolution and a Catch-All and Clean-Up component. TIGER, Cast3LB and CTB version 2 provide significantly more semantically relevant information than Penn-II, usually in the form of labeled dependency tags or links. To capture this information, the annotation algorithms for German, Spanish and Chinese each contain an additional "Defaults" component (between the head-finding rules and the Left-Right context annotation principles), triggering f-structure annotations by explicit tags.

The basic data structures in Cast3LB and CTB version 2 are CFG trees, while TIGER adopts a dependency- and graph-based encoding, with crossing edges. In order to extract PCFG (and history-based lexicalised generative) parsers for our LFG parsing architectures, we automatically convert TIGER into a corresponding CFG format with empty productions, traces and coindexation to indicate non-local dependencies encoded in terms of crossing edges in the original TIGER dependency graphs.[15]

7.6.1 GramLab

Compared to our work on English, to date, our multilingual work (Cahill et al. 2003, 2005, O'Donovan et al. 2005b, Burke et al. 2004b) has been proof-of-concept: LDD resolution has yet to be integrated into the German, Spanish and Chinese grammars; with the exception of the German grammar (based on a version of TIGER with approx.

[15]Software due to Michael Schielen, IMS, University of Stuttgart.

40,000 trees), the Spanish and the Chinese grammars are relatively small (Cast3LB has approx. 3,500 trees and the CTB version 2 contains 4,183 trees); the f-structures for German, Spanish and Chinese are more coarse-grained than the ones we currently produce for English; with the exception of German (Forst 2003), the (hand-crafted and/or hand-corrected) gold-standards for parsing evaluation are small scale; the induced lexical resources have not yet been evaluated; and each of the German, Spanish and Chinese resources have been generated with a maximum of 3-4 person months development time.

Based on our previous multilingual work, a research project (GramLab 2004-8) has started for inducing substantial LFG grammar, parsing and lexical resources for Chinese, Japanese, Arabic, Spanish, French and German from the CTB version 5 (Linguistic Data Consortium), Kyoto University Corpus (Kurohashi and Nagao 1998), ATB version 4 (Linguistic Data Consortium), Cast3LB (Civit 2003), P7T (Abeillé et al. 2001) and TIGER (Brants et al. 2002) treebanks. First versions of the resources will become available by the end of 2006. While, for most of these languages, substantial treebank resources (> 20,000 trees) are already available, one research strand of the project will explore bootstrapping larger treebank resources: in particular for Spanish we will use the LFG resources induced from Cast3LB to iteratively parse new text, hand correct the output and add the corrected output to the training set for treebank-based LFG grammar acquisition and parsing of further text.

7.7 Robust Probabilistic Generation

Treebank-based grammatical resources are a cornerstone of state-of-the-art probabilistic parsing technology. It is surprising to note that, to date, treebank-based resources have not been used to the same extent in the complementary operation to parsing: generation. Most work on statistical generation has focused on hand-crafted grammars and on ranking or filtering generation output with language models (Langkilde 2000). The first paper using treebank-induced wide-coverage grammatical resources for generation is Nakanishi et al. (2005). Following Velldal and Oepen (2005), they present a maximum entropy-based discriminative realisation disambiguation method adapted to treebank-based HPSG grammars (Velldal and Oepen 2005 use hand-crafted HPSG grammars). Cahill and van Genabith (2006) present a new PCFG-based generation method. The method uses the f-structure annotated PCFGs from the integrated parsing architecture of Cahill et al. (2004) maximising the probability of a tree given an f-structure:

$$\text{argmax}_{Tree}\ P(Tree|F\text{-}Str)$$

using a grammatical function indexed generation chart and Viterbi-optimisation.[16] The probability of a tree given an f-structure is decomposed as the product of the probabilities of all f-structure annotated productions contributing to the tree, but where in addition to conditioning on the LHS of the production (as in the integrated parsing architecture of Cahill et al. (2004)), each production $X \rightarrow Y$ is now also conditioned on the set of features *Feats* ϕ-linked[17] to the LHS X of the rule:

$$P(Tree|F\text{-}Str) := \prod_{\substack{X \rightarrow Y\ in\ Tree \\ \phi(X) = Feats}} P(X \rightarrow Y|X, Feats)$$

$$P(X \rightarrow Y|X, Feats) = \frac{P(X \rightarrow Y, X, Feats)}{P(X, Feats)}$$

$$= \frac{P(X \rightarrow Y, Feats)}{P(X, Feats)} \approx \frac{\#(X \rightarrow Y, Feats)}{\#(X \rightarrow \ldots, Feats)}$$

and where probabilities are estimated using a simple Maximum Likelihood Estimator MLE and rule counts (#) from the automatically f-structure annotated training treebank resource of Cahill et al. (2004). Lexical rules (rules expanding preterminals) are conditioned on the full set of (atomic) feature-value pairs ϕ-linked to the LHS. The intuition for conditioning rules in this way is that local f-structure components of the input f-structure drive the generation process. This conditioning effectively turns the f-structure annotated PCFGs of Cahill et al. (2004) into a probabilistic generation grammar.

The generator is trained on WSJ Sections 02-21 of the automatically f-structure annotated Penn-II treebank. In order to evaluate the generator, we feed it automatically generated f-structures for the original Sec-

[16] Maximising the probability of the tree is only the first step towards a language model, which maximises the probability of the string generated by summing over trees with the same yield given the f-structure:

$$\text{argmax}_{String} \sum_{\{Tree|yield(Tree)=String\}} P(Tree|F\text{-}Str)$$

While potentially yielding better results than our simple method, the summation term is more difficult to implement efficiently.

[17] ϕ links LFG's c-structure to f-structure in terms of many-to-one functions from tree nodes into f-structure.

tion 23 trees and compare the strings generated from those f-structures with the original gold-standard strings in Section 23 using BLEU (Papineni et al. 2001) and GTM (Turian et al. 2003) scores.[18] The generator is robust as the automatically generated input f-structures are not guaranteed to be complete and coherent. Currently the generator achieves 96.52% coverage, 0.7282 BLEU and 0.8938 GTM scores on the 1034 sentences of 20 words or less from WSJ Section 23 of the Penn-II treebank.

String-based evaluations such as BLEU and GTM can be unduly harsh, as they do not reflect well-motivated and perfectly grammatical lexical and syntactic paraphrases. Consider again the example in Figure 3. Given the f-structure, the generator may generate any one of the following strings:

(1) `U.N. signed treaty , the headline said .`

(2) `U.N. signed treaty , said the headline .`

(3) `The headline said U.N. signed treaty .`

but only (1) will receive full scores in the BLEU and GTM evaluations against the reference string in Figure 3. Therefore, we also evaluate the output of our system in terms of dependency relations. Since our generation system has a CFG backbone, every string generated is also associated with an f-structure annotated CFG tree with the string as its yield. Using this tree and the LFG f-structure annotations already present on each node, we are able to automatically produce f-structures for the strings output by our generator, without reparsing the generated string. We evaluate these f-structures against the automatically generated f-structures for the original Section 23 trees using the triples format and software of Crouch et al. (2002). We convert f-structures into sets of dependency triples of the form `relation(head,dependent)`, for example `subj(sign~3,U.N.~4)` for the most embedded sub-f-structure in Figure 3. Testing on trees of length ≤ 20, we achieve a preds-only f-score of 86.94% and all grammatical functions f-score of 89.04%.

7.8 Machine Translation and other Applications

Within GramLab work has begun to induce the complete set of resources required for probabilistic transfer-based machine translation from parse-annotated sentence-aligned parallel corpora (bitexts — texts that are translations of each other). We use the treebank-based

[18] BLEU is the weighted average of n-gram precision against the gold standard sentences. GTM (General Text Matcher) measures similarity between texts in terms of the standard measures of precision, recall, and f-score.

multilingual LFG resources produced by the GramLab project to automatically parse-annotate the bitexts with c-structure and f-structure information to automatically learn transfer relations (and their probabilities) relating source and target f-structures. Once the transfer relations are established, in order to translate a source string into a target string, we first parse the source string into c- and f-structure representations using the source language treebank-based LFG grammar, apply the transfer component to generate a (possibly partial) target f-structure and finally use the target f-structure annotated treebank trained generator to generate the target string.

In addition to machine translation applications, work is under way to apply and evaluate the treebank-based LFG resources in multilingual information retrieval, extraction, text mining and question-answering tasks.

7.9 Conclusion

This paper has provided an overview of research on rapid treebank-based automatic acquisition of multilingual probabilistic and robust LFG resources undertaken at the National Centre for Language Technology (NCLT) in the School of Computing at Dublin City University. The advantages of the treebank-based acquisition paradigm, in particular the speed of acquisition and the quality, coverage and robustness of the resulting resources, make this an attractive alternative to and will in many cases replace the more traditional hand-crafting of deep LFG (and similar constraint-based) resources.

Acknowledgments

The research reported here owes a lot to Ron Kaplan, who, after strong initial scepticism about our approach, has supported our research by facilitating our participation in the ParGram research effort, by freely offering detailed constructive criticism and feedback and by supporting various project applications. Thank you, Ron! Over the last few years I have been extremely fortunate to lead an outstanding research team, including (at different times) Aoife Cahill, Mairead McCarthy, Michael Burke, Ruth O'Donovan, John Judge, and Andy Way in a number of projects at the National Centre for Language Technology (NCLT) in the School of Computing at Dublin City University. The research described by this paper has been carried out by this team. I am indebted to the two anonymous reviewers, whose insightful and extensive comments have improved the paper significantly. We gratefully acknowledge support from Enterprise Ireland Basic Research Grant SC/2001/0186

(2001–2004, Aoife Cahill, Mairead McCarthy, Michael Burke, Andy Way and Josef van Genabith), Science Foundation Ireland Basic Research Grant 04/BR/CS0370 (2004–2007, Aoife Cahill and Josef van Genabith), Irish Research Council for Science, Technology and Engineering PhD Grant (2002-2005, Ruth O'Donovan), Irish Research Council for Science, Technology and Engineering PhD Grant (2002–2005, John Judge) and an IBM Ph.D. fellowship 2004-2005 (Michael Burke).

References

Abeillé, Anne, ed. 2003. *Treebanks*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Abeillé, Anne, Lionel Clement, Alexandra Kinyon, and François Toussenel. 2001. A Treebank for French: some experimental results. In *Proceedings of the International Conference on Corpus Linguistics (CL'2001)*. Lancaster, United Kingdom.

Abney, Stephen. 1997. Stochastic attribute-value grammars. *Computational Linguistics* 23(4):597–618.

Bernard, L. 2002. User Reference Guide for the British National Corpus. Technical report, Oxford University Computing Services.

Bikel, Daniel. 2004. Intricacies of Collins parsing model. *Computational Linguistics* 30(4):479–511.

Blaheta, Don and Eugene Charniak. 2000. Assigning function tags to parsed text. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, pages 234–240. Seattle, WA.

Bod, Rens and Ron Kaplan. 1998. A probabilistic corpus-driven model for Lexical Functional Analysis. In C. Boitet and P. Whitelock, eds., *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL'98) and 17th International Conference on Computational Linguistics (COLING'98)*, pages 145–151. Montreal, Canada: Morgan Kaufmann.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In E. Hinrichs and K. Simov, eds., *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT'02)*, pages 24–41. Sozopol, Bulgaria.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Oxford, United Kingdom: Blackwell.

Burke, Michael. 2006. *Automatic Treebank Annotation for the Acquisition of LFG Resources*. Ph.D. thesis, School of Computing, Dublin City University, Ireland.

Burke, Michael, Aoife Cahill, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004a. Evaluation of an Automatic Annotation Algorithm

against the PARC 700 Dependency Bank. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2004 (LFG'04)*, pages 101–121. Christchurch, New Zealand: CSLI Online Publications.

Burke, Michael, Olivia Lam, Rowena Chan, Aoife Cahill, Ruth O'Donovan, Adams Bodomo, Josef van Genabith, and Andy Way. 2004b. Treebank-based acquisition of a Chinese Lexical-Functional Grammar. In H. Masuichi, T. Ohkuma, K. Ishikawa, Y. Harada, and K. Yoshimoto, eds., *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation (PACLIC18)*, pages 161–172. Tokyo, Japan: The Logico-Linguistic Society of Japan.

Burke, Michael, Aoife Cahill, Josef van Genabith, and Andy Way. 2005. Evaluating automatically acquired f-structures against PropBank. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2005 (LFG'05)*, pages 84–99. Bergen, Norway: CSLI Online Publications.

Cahill, Aoife, Mairéad McCarthy, Josef van Genabith, and Andy Way. 2002. Parsing with PCFGs and Automatic F-Structure Annotation. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2002 (LFG'02)*, pages 76–95. Athens, Greece: CSLI Online Publications.

Cahill, Aoife, Martin Forst, Mairéad McCarthy, Ruth O'Donovan, Christian Rohrer, Josef van Genabith, and Andy Way. 2003. Treebank-based multilingual unification-grammar development. In *Proceedings of the 15th European Summer School in Logic Language and Information (ESSLLI'03), Workshop on Ideas and Strategies for Multilingual Grammar Development*, pages 17–24. Vienna, Austria.

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2004. Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, pages 320–327. Barcelona, Spain.

Cahill, Aoife, Michael Burke, Martin Forst, Ruth O'Donovan, Christian Rohrer, Josef van Genabith, and Andy Way. 2005. Treebank-Based Acquisition of Multilingual Unification Grammar Resources. *Research on Language and Computation* **3**(2-3):247–279.

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Stefan Riezler, Josef van Genabith, and Andy Way. 2006. Shallow and deep parser comparison with automatically generated dependency relations. Under Review.

Cahill, Aoife and Josef van Genabith. 2006. Robust PCFG-based generation using automatically acquired treebank-based LFG approximations. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL'06) and 21st International Conference on Computational Linguistics (COLING'06)*. Sydney, Australia. To appear.

Carroll, John, Edward Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: A survey and new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC'98)*, pages 447–454. Granada, Spain.

Carroll, John and Edward Briscoe. 2002. High precision extraction of grammatical relations. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'03)*, pages 134–140. Taipei, ROC.

Charniak, Eugene. 1996. Tree-Bank Grammars. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, pages 1031–1036. Menlo Park, CA: The MIT Press/AAAI Press.

Charniak, Eugene. 2000. A maximum entropy inspired parser. In *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, pages 132–139. Seattle, WA.

Civit, Montserrat. 2003. *Criterios de etiquetación y desambiguación morfosintáctica de corpus en español*. Ph.D. thesis, Universitat de Barcelona, Spain.

Collins, Michael. 1997. Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*, pages 16–23. Madrid, Spain.

Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Crouch, Richard, Ronald M. Kaplan, Tracy H. King, and Stefan Riezler. 2002. A comparison of evaluation metrics for a broad-coverage parser. In J. Carroll, A. Frank, D. Lin, D. Prescher, and H. Uszkoreit, eds., *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02), Workshop 'Beyond PARSEVAL: Towards improved evaluation mesures for parsing systems'*, pages 67–74. Las Palmas, Spain.

Dalrymple, Mary. 2001. *Lexical-Functional Grammar*. San Diego, CA: Academic Press.

Forst, Martin. 2003. Treebank Conversion — establishing a test suite for a broad-coverage LFG from the TIGER treebank. In A. Abeillé, S. Hansen-Schirra, and H. Uszkoreit, eds., *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), 4th Workshop on Linguistically Interpreted Corpora (LINC'03)*, pages 25–32. Budapest, Hungary.

Frank, Anette. 2000. Automatic f-structure annotation of treebank trees. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2000 (LFG'00)*, pages 139–160. Berkeley, CA: CSLI Online Publications.

Frank, Anette, Louisa Sadler, Josef van Genabith, and Andy Way. 2003. From treebank resources to LFG f-structures. In A. Abeillé, ed., *Treebanks: Building and Using Syntactically Annotated Corpora*, pages 367–389. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar*. Oxford, United Kingdom: Blackwell.

Gildea, Daniel. 2001. Corpus variation and parser performance. In L. Lee and D. Harman, eds., *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP'01)*, pages 167–202. Pittsburgh, PA.

Gildea, Daniel and Julia Hockenmaier. 2003. Identifying semantic roles using combinatory categorial grammar. In M. Collins and M. Steedman, eds., *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP'03)*, pages 57–64. Sapporo, Japan.

Hemphill, Charles T., John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 96–101. Hidden Valley, PA.

Hockenmaier, Julia and Mark Steedman. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02)*. Las Palmas, Spain.

Hornby, Albert S., ed. 1980. *Oxford Advanced Learner's Dictionary of Current English*. Oxford, United Kingdom: Oxford University Press.

Johnson, Mark. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 136–143. Philadelphia, PA.

Joshi, Aravind K. 1987. An introduction to Tree Adjoining Grammar. In A. Manaster-Ramer, ed., *The Mathematics of Language*, pages 87–114. Amsterdam, the Netherlands: John Benjamins.

Judge, John, Michael Burke, Aoife Cahill, Ruth O'Donovan, Josef van Genabith, and Andy Way. 2005. Strong domain variation and treebank-induced LFG resources. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2005 (LFG'05)*, pages 186–204. Bergen, Norway: CSLI Online Publications.

Kaplan, Ron and Joan Bresnan. 1982. Lexical Functional Grammar, a formal system for grammatical representation. In J. Bresnan, ed., *The Mental Representation of Grammatical Relations*, pages 173–281. Cambridge, MA: The MIT Press.

Kaplan, Ron and Jürgen Wedekind. 2000. LFG generation produces context-free languages. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'00)*, pages 141–148. Saarbrücken, Germany.

Kaplan, Ron, Stefan Riezler, Tracy Holloway King, John T. Maxwell, Alexander Vasserman, and Richard Crouch. 2004. Speed and Accuracy in Shallow and Deep Stochastic Parsing. In *Proceedings of the Human Language Technology Conference and the 4th Annual Meeting of the North*

American Chapter of the Association for Computational Linguistics (HLT-NAACL'04), pages 97–104. Boston, MA.

Kay, Martin. 1985. Parsing in Functional Unification Grammar. In L. K. David R. Dowty and A. M. Zwicky, eds., *Natural Language Parsing*, pages 251–278. Cambridge, United Kingdom: Cambridge University Press.

King, Tracy H., Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 Dependency Bank. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), 4th International Workshop on Linguistically Interpreted Corpora (LINC'03)*, pages 1–8. Budapest, Hungary.

Kingsbury, Paul, Martha Palmer, and Mitch Marcus. 2002. Adding semantic annotation to the Penn TreeBank. In *Proceedings of the Human Language Technology Conference (HLT'02)*. San Diego, CA.

Korhonen, Anna. 2002. *Subcategorization Acquisition*. Ph.D. thesis, Computer Laboratory, University of Cambridge, United Kingdom.

Kurohashi, Sadao and Makoto Nagao. 1998. Building a Japanese parsed corpus while improving the parsing system. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC'98)*, pages 719–724. Granada, Spain.

Langkilde, Irene. 2000. Forest-based statistical sentence generation. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, pages 170–177. Seattle, WA.

Lappin, Shalom, Igal Golan, and Mori Rimon. 1989. Computing grammatical functions from configurational parse trees. Technical report 88.268, IBM Israel, Haifa, Israel.

Leech, Geoffrey and Roger Garside. 1991. Running a grammar factory: On the compilation of parsed corpora, or 'treebanks'. In S. Johansson and A.-B. Stenström, eds., *English Computer Corpora: selected papers*, pages 15–32. Berlin, Germany: Mouton de Gruyter.

Macleod, Catherine, Adam Meyers, and Ralph Grishman. 1994. The COMLEX syntax project: The first year. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 669–703. Princeton, NJ.

Magerman, David. 1994. *Natural Language Parsing as Statistical Pattern Recognition*. Ph.D. thesis, Department of Computer Science, Stanford University, CA.

Marcus, Mitchell, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 110–115. Princeton, NJ.

McCarthy, Mairéad. 2003. *Design and Evaluation of the Linguistic Basis of an Automatic F-Structure Annotation Algorithm for the Penn-II Treebank*. Master's thesis, School of Computing, Dublin City University, Ireland.

Miyao, Yusuke and Jun'ichi Tsujii. 2004. Deep linguistic analysis for the accurate identification of predicate-argument relations. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*, pages 1392–1397. Geneva, Switzerland.

Nakanishi, Hiroko, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic models for disambiguation of an HPSG-based chart generator. In *Proceedings of the International Workshop on Parsing Technology (IWPT 2005)*. Vancouver, Canada.

O'Donovan, Ruth. 2006. *Automatic Extraction of Large-Scale Multilingual Lexical Resources*. Ph.D. thesis, School of Computing, Dublin City University, Ireland.

O'Donovan, Ruth, Michael Burke, Aoife Cahill, Josef van Genabith, and Andy Way. 2004. Large-scale induction and evaluation of lexical resources from the Penn-II Treebank. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, pages 368–375. Barcelona, Spain.

O'Donovan, Ruth, Michael Burke, Aoife Cahill, Josef van Genabith, and Andy Way. 2005a. Large-scale induction and evaluation of lexical resources from the Penn-II and Penn-III treebanks. *Computational Linguistics* 31(3):329–365.

O'Donovan, Ruth, Aoife Cahill, Josef van Genabith, and Andy Way. 2005b. Automatic acquisition of Spanish LFG resources from the Cast3LB Treebank. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2005 (LFG'05)*, pages 334–352. Bergen, Norway: CSLI Online Publications.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Tech. Rep. IBM Research Division Technical Report, RC22176 (W0190-022), Yorktown Heights, NY.

Pollard, Carl and Ivan Sag. 1994. *Head-driven Phrase Structure Grammar*. Stanford, CA: CSLI Publications.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 271–278. Philadelphia, PA.

Sadler, Louisa, Josef van Genabith, and Andy Way. 2000. Automatic f-structure annotation from the AP Treebank. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2000 (LFG'00)*, pages 226–243. Berkeley, CA: CSLI Online Publications.

Sampson, Geoffrey. 1995. *English for the Computer: The SUSANNE Corpus and analytic scheme*. Oxford, United Kingdom: Clarendon Press.

Schmid, Helmut. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*, pages 162–168. Geneva, Switzerland.

Steedman, Mark. 1996. *Surface Structure and Interpretation*. Cambridge, MA: The MIT Press.

Turian, Joseph P., Luke Shen, and I. Dan Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proceedings of the 9th Machine Translation Summit (MTS IX)*, pages 23–28. New Orleans, LA.

van Genabith, Josef and Richard Crouch. 1996. Direct and Underspecified Interpretations of LFG f-Structures. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, pages 262–267. Copenhagen, Denmark.

van Genabith, Josef and Richard Crouch. 1997. On interpreting f-structures as UDRSs. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97) and the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL'97)*, pages 402–409. Madrid, Spain.

van Genabith, Josef, Louisa Sadler, and Andy Way. 1999. Data-driven compilation of LFG semantic forms. In *Proceedings of the 9th Conference of the European Chapter of the of the Association for Computational Linguistics (EACL'99), Workshop on Linguistically Interpreted Corpora (LINC'99)*, pages 69–76. Bergen, Norway.

van Genabith, Josef, Andy Way, and Louisa Sadler. 1999. Semi-automatic generation of f-structures from treebanks. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 1999 (LFG'99)*. Manchester, United Kingdom: CSLI Online Publications.

Velldal, Erik and Stephan Oepen. 2005. Maximum entropy models for realization ranking. In *Proceedings of the 10th Machine Translation Summit (MTS X)*, pages 109–116. Phuket, Thailand.

Xue, Nianwen, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2004. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering* 10(4):1–30.

8

Hand-crafted Grammar Development – How Far Can It Go?

Christian Rohrer and Martin Forst

8.1 Introduction

Hand-crafted 'deep' computational grammars have rarely been scaled to free text. The only hand-crafted 'deep' grammar for English that has been evaluated on a test set from the Wall Street Journal part of the Penn Treebank is the English *ParGram* LFG (Riezler et al. 2002). As to 'deep' grammars for German, we do not know of any applications to newspaper corpora apart from the experiments reported in Rohrer and Forst (2006). However, broad-coverage 'deep' grammars that can also be used for generation are important for a variety of applications, among which we would just like to name question-answering and grammar-based (statistical) machine translation (see, e.g., Riezler and Maxwell 2006 and Riezler and Maxwell, this volume).

Of course, there are a variety of parsers for German that achieve very good results on newspaper text. But their analyses are often considerably shallower than LFG parses, i.e. they contain significantly less detailed information, and they are not reversible, i.e. they cannot be used for generation. The majority of these resources are variations of context-free grammars that were induced from treebanks. A notable exception among the treebank-induced German parsers with respect to depth of analysis and reversibility is the LFG for German by Cahill et al. (2005), which was induced from the TIGER Treebank (Brants et al. 2002), a syntactically annotated corpus of German newspaper texts comprising about 50,000 sentences. As, for the time being, the

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

parsing quality that is achieved with the induced LFG resources is noticeably less good for German than for English, however, it remains to be seen whether the methodology of inducing LFG grammars from treebanks is as successful for languages with more variation in word order as it is for English (see also van Genabith, this volume).

Hand-crafted grammar development therefore remains an interesting option in building a 'deep' and reversible grammar for a language like German. In this paper we present the current state of the German *ParGram* LFG, and we discuss the development steps that were taken to scale it from about 50% coverage in terms of full parses (Dipper 2003) to the current stage of more than 80%. These steps are of three main kinds: revision of the finite-state transducers that are used in a preprocessing stage for tokenization and morphological analysis, grammar extension, and restriction of computationally expensive rules. These are discussed in Section 8.2, Section 8.3 and Section 8.4 respectively. Section 8.5 presents and discusses the results of the revised grammar on a manually validated dependency-based gold standard, and Section 8.6 concludes.

8.2 Revising the Transducers

Before the input sentences are parsed by means of the broad-coverage grammar proper, they are preprocessed by a cascade of finite-state transducers (FSTs) that take care of tokenization, morphological analysis and, to a certain, and still rather limited, extent, named-entity recognition. The coverage and accuracy of these transducers have a major effect on the quality of the overall coverage and parse quality. We will present here the modifications of the transducers that helped to improve the parsing results of the grammar.

8.2.1 Tokenization

When parsing the TIGER Corpus for the first time (instead of much smaller corpora or linguistic examples) with the original grammar, we noticed that a considerable number of sentences could not be analyzed due to inappropriate tokenization. The phenomena not handled sufficiently can be classified as non-trivial tokenization issues, text normalization and the interpretation of quotes as potential 'markup' for foreign material.

Non-trivial tokenization issues not handled by the original tokenizer

The original tokenizer performed a very basic segmentation of the input sentences into tokens. For instance, all periods, except the ones at the

end of a short list of common abbreviations and decimal/numerical points, were treated as separate tokens, which is clearly not intended in strings like the following. (In addition to the gloss and instead of a fluent translation, we indicate the intended tokenization in the third line of each example, where *TB* stands for 'token boundary'.)

(1) eine "K.o.-Tropfen-Bande"
a "k.o. drops gang"
eine TB " TB K.o.-Tropfen-Bande TB "

(2) in der Dominikus-Zimmermann-Str. 9
in the Dominikus Zimmermann street 9
in TB der TB Dominikus-Zimmermann-Str. TB 9

Similarly, the original tokenizer treated basically all commas, except for decimal commas in pure numbers, as separate tokens. For example, the comma in (3) was incorrectly considered a token on its own.

(3) die 1,63-Meter-Frau
the 1.63-metre-woman
die TB 1,63-Meter-Frau

Apostrophes were also systematically treated as separate tokens, which would be problematic in (4), where the apostrophe replaces the elided *e* of the expletive pronoun *es*, in (5), where it is part of the genitive of *Stiegl*,[1] and in (6), where it marks the genitive of a proper name ending in *s*.

(4) Gibt's wieder Freikarten?
Are there again free tickets?
gibt TB 's TB wieder TB Freikarten TB ?

(5) Veranstaltungsort ist Stiegl's Brauerei.
Event place is Stiegl's brewery.
Veranstaltungsort TB ist TB Stiegl's TB Brauerei TB .

(6) Karamanlis' Politik
Karamanlis's policy
Karamanlis' TB Politik

Finally, parentheses, quotes and blanks were always treated as separate tokens and token boundaries respectively, which posed problems for strings like these:

(7) an zivilem (Verwaltungs-)Personal
of civil (administration) personnel
an TB zivilem TB (Verwaltungs-)Personal

[1]This spelling is not standard German, but due to the influence of English, it is becoming more and more common.

(8) das “Soldaten sind Mörder”-Zitat
the “soldiers are murderers” citation
das TB “Soldaten sind Mörder”-Zitat

(9) rund 300 000 Quadratmeter
approximately 300,000 square metres
rund TB 300000 TB Quadratmeter

(10) ihre New York-Reise
their New York trip
ihre TB New York-Reise

Text normalization

Another inadequacy of the original tokenizer was its insufficient ability to normalize text.

Decapitalization

The most important normalization when parsing free text is decapitalization, not only at the beginning of a sentence, but also after opening quotes, brackets, colons and hyphens. The original tokenizer failed to decapitalize words after hyphens as in (11) or after opening parentheses as in (12).

(11) im Buch “Mystik Mythos Metaphysik – Die
in the book “mystics myth metaphysics – The
. . . Metaphysik TB – TB die TB
Spur des vermißten Gottes”
trace of the missed God”
Spur TB . . .

(12) (Siehe auch Wirtschaft)
(See also economics)
(TB siehe TB auch TB Wirtschaft TB)

Normalization of variation in numbers

German exhibits considerable variation in the notation of numbers: both the comma and the point appear as digital separators, and large numbers expressed in digits occur without any segmentation, segmented by a point, a blank or even an apostrophe.

(13) Insgesamt 130000 DM
In total 130,000 DM
insgesamt TB 130000 TB DM

(14) Die rund 18.400 Ladiner
The approximately 18,400 Ladinians
die TB rund TB 18400 TB Ladiner

(15) 35 000 Menschen protestierten
35,000 people protested
35000 TB Menschen TB protestierten

(16)	300'000	Anleger	wollen	Swisscom-Aktien
	300,000	investors	want	Swisscom shares
	300000 TB	Anleger TB	wollen TB	Swisscom-Aktien

Since applications that make use of the grammar do not want to deal with this type of variation, it is reasonable to map all the segmented variants onto their unsegmented counterpart and to use the comma as the 'normal' digital separator. Moreover, all of the resulting strings can then be analyzed adequately by our FST morphology, whereas most of the segmented variants could not.

Haplology

Since the grammar is punctuation-sensitive, the tokenizer needs to provide additional punctuation marks, if we want to keep punctuation-related rules within reasonable complexity. In German, as in many languages, certain punctuation marks that would show up sentence-internally are merged into the following punctuation mark (haplology).

The original tokenizer inserted two optional commas before visible commas, periods, question marks and exclamation marks. However, this solution was too simplistic for two reasons: first, other punctuation marks take part in this type of interaction, e.g. hyphens. Second, it is desirable to distinguish commas that are present in the surface string from the additional ones provided by the tokenizer.

(17)	Sie	werden	enteignet	–	was
	They	are	expropriated	–	which
	sie TB	werden TB	enteignet TB	– TB	was TB
	manche	ablehnen.			
	some	reject.			
	manche TB	ablehnen TB _, TB .			

(17) illustrates the insertion of a an additional comma (_,). This insertion makes it possible to write a rule that allows a relative clause to be opened by a hyphen and closed by such a comma.

Interpreting quotes as potential 'markup' for foreign material

One major obstacle for full coverage of German newspaper texts is the common occurrence of material from foreign languages in the text. However, this material is often 'marked' by quotes for the human reader, and it seems wise to take advantage of these quotes for parsing. One way this can be done is the following: all strings between balanced quotes are optionally considered as one single token and marked as *+Quoted-String*. (18) illustrates this solution. The output is then 'collected' by a special grammar rule which treats the quoted material as a name or whatever the foreign material proves to be.

(18) des zweiten Gegenkongresses
of the second anti-congress
des TB zweiten TB Gegenkongresses TB
"The other Economic Summit"
"The other Economic Summit"
The other Economic Summit TB +QuotedString

8.2.2 Morphological analysis and guessing

Apart from additions and corrections in our lists of open class stems, several systematic extensions had to be made to our morphological analyzers. These concern, e.g., Roman numbers as in (19) and (20), genitive forms of proper names ending in *-s'* as in (21), and person designations that refer explicitly to both male and female persons as in (22), all of which were not covered by the morphology before. (In addition to the gloss and instead of a fluent translation, we give the intended morphological analysis in the third line of each example, where + introduces part-of-speech tags, ^ marks tags that convey derivational information, and . introduces all other tags.)

(19) auf dem XIV. Parteitag
on the XIVth party convention
... XIV. +ORD .Roman ...

(20) in die Phase II
in the phase II
... II +CARD .Roman

(21) Karamanlis' Politik
Karamanlis's policy
Karamanlis +NPROP .NoGend .Gen .Sg ...

(22) die KollegInnen
the (explicitly both male and female) colleagues
... Kollege ^MF +NN .Fem .NGDA .Pl

After the changes to the tokenizer that help to preserve derivations and compounds involving parentheses, quotes and spaces as tokens instead of splitting them, the morphological analyzers had to be adapted so as to handle these forms.

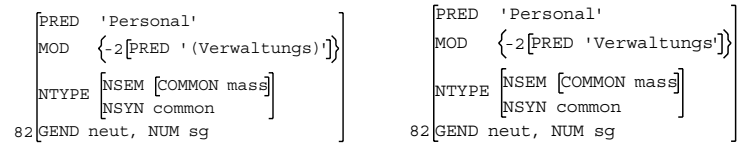
(23) die Kolleg(inn)en
the (explicitly both male and female) colleagues
... Kollege ^MF +NN .Fem .NGDA .Pl

(24) an (Verwaltungs-)Personal
of (administration) personnel
... ^HYP (Verwaltungs) +CMPD Personal +NN .Neut .NDA .Sg

(25) "Soldaten sind Mörder"-Zitat
"soldiers are murderers" citation
^HYP "Soldaten sind Mörder" +CMPD Zitat +NN .Neut .NDA .Sg

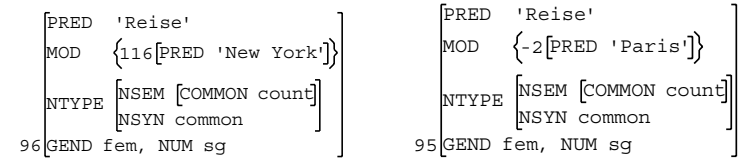
(26) ihre New York-Reise
their New York trip
... ^HYP New York +CMPD Reise +NN .Fem .NGDA .Sg

The morphological analyses of the 'special' compounds in (24) through (26) allow us to project f-structures parallel to those of 'standard' compounds. Hence, the morphological analysis of *(Verwaltungs-) Personal* is associated with an f-structure parallel to the f-structure projected from the morphological analysis of *Verwaltungspersonal* (see Figure 1), and *New York-Reise* receives an f-structure parallel to the one of *Paris-Reise* (see Figure 2). In all of these, the compound non-head PREDs are projected into the set-valued feature MOD.



(a) ^HYP (Verwaltungs) +CMPD Personal ... (b) Verwaltungs +CMPD Personal ...

FIGURE 1 F-structures associated with *(Verwaltungs-)Personal* and *Verwaltungspersonal*



(a) ^HYP New York +CMPD Reise ... (b) ^HYP Paris +CMPD Reise ...

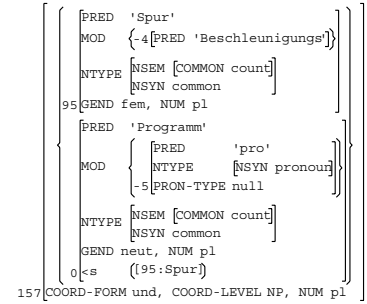
FIGURE 2 F-structures associated with *New York-Reise* and *Paris-Reise*

Finally, another category of compound forms that was not treated by the original morphology are truncated forms like *-programme* in (27). Truncated forms like *Bildungs-* in (28), however, were analyzed. This situation was remedied by an extension of the morphological transducer that now allows for an analysis of these forms along the lines of the existing analysis of compounds. The f-structures projected from forms

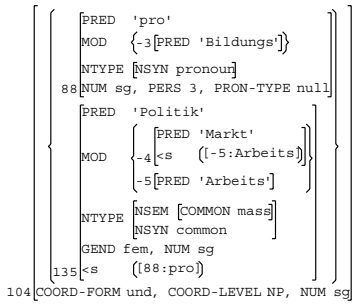
like *-programme* and *Bildungs-* are illustrated in Figure 3. They have in common that the elided part of the truncated compound shows up in the f-structure as a null pronoun.[2] In Figure 3(a), this null pronoun is projected into the Mod set, since it is a non-head that is elided here; in Figure 3(b), the null pronoun is the head Pred because it is the head which is elided.

(27) Beschleunigungsspuren und
acceleration lanes and
Beschleunigungs +CMPD Spur ... und +CONJ .Coord
-programme
programs
+CMPD Programm +NN .Neut .NGA .Pl

(28) Bildungs- und
education and
Bildungs +CMPD +TRUNC und +CONJ .Coord
Arbeitsmarktpolitik
labour market policy
Arbeits +CMPD Markt +CMPD +Politik ...



(a) ... +CMPD Programm +NN .Neut .NGA .Pl



(b) Bildungs +CMPD +TRUNC ...

FIGURE 3 F-structures associated with *Beschleunigungsspuren und -programme* and *Bildungs- und Arbeitsmarktpolitik*

8.2.3 The importance of precise preprocessing

Forst and Kaplan (2006) have shown that precise tokenizing is of considerable importance for the coverage of a large-scale LFG. They report a gain in coverage in terms of full parses of 5.1 points (or 7.5%) due to

[2] From a semantic point of view, it would be desirable to project *Beschleunigungs* as the Mod Pred of *Programm* and *Politik* as the head Pred of *Bildungs*, but we have not found an efficient way to do this at f-structure level.

the revision of just the finite-state tokenizer described above. It is reasonable to conjecture that a similar percentage of the corpus sentences owe their full analyses to the revision and extension of the finite-state morphology and the guesser. In conclusion, although few of the tokenization and morphology issues discussed above are interesting from a strictly linguistic perspective, they have to be considered and accounted for when scaling a computational grammar to free text.

8.3 Corpus-based Enlargement of Grammar Coverage

The initial grammar used for the work presented in this paper produced full parses for slightly more than 50% of the sentences in the TIGER Corpus. Even with robustness techniques that produce partial parses for the remaining sentences, this percentage is too low to allow the grammar to produce analyses that can compete in quality with other state-of-the-art parsers. In addition to correcting and completing the FSTs used for preprocessing, extending the grammar proper was thus necessary to scale the grammar to free text.

In the initial stage of the *DLFG* project on corpus-based grammar development, we tried to identify gaps in the grammar by automatically extracting testsuites of relatively small constituents, e.g. DPs or PPs, from the syntactically annotated TIGER Corpus. The motivation for this was to separate issues of grammar coverage proper from issues of efficiency, which often enter into consideration when parsing entire corpus sentences, and to limit the size of the units to be inspected, so that the manual error analysis would be facilitated.[3]

However, this methodology proved to be relatively inefficient, as the vast majority of instances of these small constituents were covered by the original grammar already. What really posed problem for the original grammar was to analyze certain constituents **in context**. For example, many sentences did not receive a full parse because the specific subcategorization frame of the verb was not recorded in the original grammar's lexicon, whereas all DPs and PPs taken in isolation could be parsed. Further examples are temporal or modal Adjunct DPs as in (29) and (30), which could be analyzed on their own by the original grammar, but not as part of the clause, since the original grammar allowed only very few types of DPs as Adjuncts.

(29) Er schläft lieber **den ganzen Tag**.
He sleeps preferably the whole day.
'He prefers sleeping the whole day.'

[3]Indeed, it is a non-trivial task to find out what caused a sentence not to be analyzed, and a thorough knowledge of both the grammar and the FSTs used for preprocessing is needed to perform this type of error analysis.

(30) Sie marschieren **Schulter an Schulter** gegen die Faschisten.
They march shoulder at shoulder against the fascists.
'Shoulder to shoulder they march against the fascists.'

As the testsuites of small constituents were of limited usefulness, we reverted back to the testsuite made up of all TIGER Corpus sentences except for the ones from the test section. Sentences that received no analysis were inspected manually and the cause of failure was identified and, if possible, classified. With this classification, it is possible to use corpus query tools such as TIGERSearch (Lezius 2002) or CQP (Christ 1994) in order to estimate the importance of the phenomenon under consideration in terms of frequency and to get a better understanding of the generalizations that apply (or do not apply) to it.

For all phenomena that occur with some minimal frequency in our corpora, we developed rules for the revised version of the grammar that allow the parser to analyze them. However, there were rules for certain phenomena which were then deactivated because the gain in coverage achieved through including them did not compensate for their negative impact on efficiency. One such phenomenon are coordinated VPs that are separated only by commas, without any coordinating conjunction. In the following, we will present some selected phenomena for which rules were introduced into the revised grammar according to this methodology.

8.3.1 Coordination

Kaplan and Maxwell (1988) presented a theory that accounts for like constituent coordination. Initially, they ignored cases of non-constituent coordination. A few years later, Maxwell and Manning (1996) proposed an account for certain types of non-constituent coordination that are notoriously difficult to describe, such as 'conjunction reduction' and 'right-node-raising', based on finite-state rules. However, this proposal has not been implemented, since it would pose serious efficiency problems.

When scaling a grammar to free text, non-standard coordinated structures can no longer be left completely untreated. In our revised grammar, we therefore try to account for two additional types of non-constituent coordination with the help of special rules: coordination of the functionally similar categories ADVP and PP and a subclass of subject gap in finite constructions (SGF). Furthermore, we propose an analysis for same-constituent coordinations in which an ADVP or a PP intervenes between the coordinating conjunction and the following conjunct.

Coordination of ADVPs and PPs

The original grammar can handle one case of unlike constituent coordination with a special rule, namely the coordination of DPs, APs and PPs, which is possible when these are used predicatively. (31) is an example of such a coordination.

(31) Sie ist **Lehrerin, katholisch und aus Bayern**.
She is teacher, Catholic and from Bavaria.
'She is a teacher, Catholic and from Bavaria.'

By analogy with this special coordination rule, a newly added rule accounts for the coordination of ADVPs and PPs as in (32) in the revised grammar version.

(32) dort und in zahlreichen Zeitungen
there and in numerous newspapers
'there and in numerous newspapers'

This rule basically takes the following form:[4]

```
ADVP[_type] -->
            ADVP[_type]: ! $ ^;
            CONJco: ^ = !;
            PP[_type]: ! $ ^.
```

Apart from the fact that unlike constituents are combined, it is parallel to rules for like constituent coordination.

The entire coordinated construction is classified as an ADVP and not as a PP, since the attachment possibilities of ADVPs are more restricted than those of PPs. This helps to keep the influence of the additional rule on parsing efficiency negligible.

Subject gap in finite constructions (SGF)

Following Höhle (1983), this phenomenon has been widely discussed in German linguistics. Höhle (1983) provides the following example:

(33) In den Wald ging der Jäger und schoß einen Hasen.
Into the forest went the hunter and shot a rabbit.
'Into the forest, the hunter went, and shot a rabbit.'

In such constructions, we seem to be in the presence of Cbar coordination, but the shared subject is in the Mittelfeld of the first conjunct

[4]The rule shown here is a much simplified version of the actual ADVP rule in the grammar and it illustrates only the conjunct that captures the coordination of ADVPs and PPs. All rules in the following will be simplified in a similar way. The notation of the rules follows the XLE conventions, where f-annotations are found after the corresponding category, enclosed by a colon and a semicolon, and where ^ stands for ↑, ! stands for ↓ and $ stands for ∈. For more detailed information, please refer to Crouch et al. (2006).

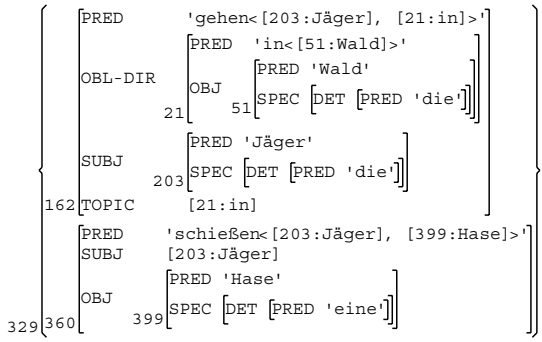


FIGURE 4 F-structure associated with (33)

instead of being in the Vorfeld. This means that it is not distributed into the second conjunct by the standard Cbar coordination rule.

We have implemented an analysis following Frank (2002, this volume), who treats SGF coordination as a marked case of CP coordination that can only occur given a very particular information structure and where the Topic position of the second conjunct is empty. Unlike Frank (this volume), we formulate the rule as a coordination of a CP and a Cbar, but this is a detail motivated by efficiency considerations. The crucial features common to Frank (this volume)'s and our analyses is the additional annotation `(^SUBJ) = (!SUBJ)` on the first CP conjunct **and** the fact that the Vorfeld constituent of the first conjunct is **not** distributed into the subsequent conjuncts, as it is only part of the first conjunct. As a consequence, the conjuncts share their Subject, but no Topic is found in the f-structure of the second conjunct. Consider the following rule as well as the f-structure that it gives rise to (see Figure 4):

```
CProot[std] -->
            CProot[std]: ! $ ^
                         (^SUBJ) = (!SUBJ);
            CONJco
            Cbar: ! $ ^.
```

Sentences with SGF coordination are quite frequent. Despite our restricted treatment of SGF coordination, we account for most occurrences.

Adverbs and/or PPs between conjunction and last conjunct
In contrast to SGF coordination, the phenomenon illustrated by the following sentence does not seem to have attracted the attention of

theoretical linguists.

(34) Der Deutsche Fußball-Bund und **mit ihm** eine ganze
The German Football Federation and with it a whole
Reihe Experten waren damals ganz anderer Ansicht.
series experts were then wholly different opinion.
'The German Football Federation and, with them, a whole series of experts had a totally different opinion then.'

The phenomenon under consideration is the occurrence of an ADVP or a PP to the left of the last conjunct of a coordinated structure. It is crucial to note that this kind of ADVP/PP can only appear in this position within a constituent if the constituent involves coordination. **mit ihm eine ganze Reihe Experten*, i.e. the second conjunct of the coordination in (34), is not a well-formed DP on its own.

As a source of inspiration for an adequate description of these constructions, we considered the conventions adopted for them in the annotation schemes of two treebanks, namely the TIGER Corpus and the Penn Treebank. It turned out that the annotators of the two treebanks had opted for different solutions: whereas the TIGER annotation scheme goes for a more semantically motivated analysis, namely the attachment of the ADVP/PP to the following conjunct, the Penn annotation scheme adopts a more syntactically motivated analysis, namely high attachment, i.e. attachment to the coordinated NP node. The TIGER convention, on the one hand, leads to the problem that the treebank contains a context-free NP rule that should be restricted to the context of coordination, but is not, due to its context-freeness; this is undesirable, in particular for grammar induction. On the other hand, the Penn convention produces a tree representation that is problematic from a semantic point of view, since the ADVP/PP under consideration clearly does not modify the entire coordination.

LFG, with its two levels of representation, allows us to restrict the occurrence of this kind of ADVP/PP to coordinated constituents and to project a semantically plausible dependency structure at the same time. This is achieved with the following extension of the DP rule:[5]

```
DP[_type] -->
          DP[_type]: ! $ ^;
          CONJco: ^ = !;
          ( { ADVP[std]: ! $ (f::RS* ADJUNCT)
                         (*WEIGHT) < 3;
            | PP[std]: ! $ (f::RS* ADJUNCT)
                       (*WEIGHT) < 5; } )
          DP[_type]: ! $ ^.
```

[5] A parallel extension was introduced in the PP, PREDP and CPdep rules.

At the level of c-structure, the ADVP/PP is attached high, to the coordinated node. At the level of f-structure, however, the notation `! $ (f::RS* ADJUNCT)` guarantees that the sub-f-structure of the ADVP/PP is analyzed as an Adjunct of the sub-f-structure of its right sister, i.e. of the following conjunct, as illustrated in Figure 5. The additional restrictions on the 'weight' of the intervening ADVP/PP ensure that only an ADVP of maximally three tokens or a PP of maximally five tokens is admitted here, which is what we observe in the TIGER Corpus and other corpora.

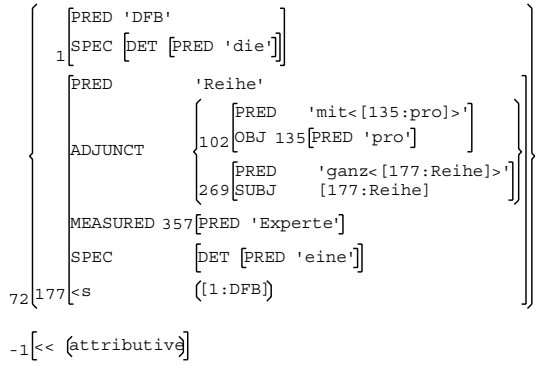


FIGURE 5 F-structure of the coordinated DP in (34)

8.3.2 Parentheticals

Apart from coordinated constructions, the treatment of parenthetical constructions plays an important role for grammar coverage, since these constructions are frequent. About 4-5% of the sentences in the TIGER Corpus contain constituents labeled as parentheticals (PAR), and there are certain types of parentheticals, in particular reportive parentheticals, that are not labeled as PARs, so that the above number is even an underestimate. In order to make the revised grammar cover parentheticals or at least most of them, we implemented rules along the lines of Fortmann (2005).

Reportive parentheticals without 'real' verbum dicendi

Reportive parentheticals are a particular challenge for grammatical description because c-structurally, the parenthetical is embedded in the host clause, whereas f-structurally the host clause is an argument of the head verb of the parenthetical. The following sentence exemplifies reportive parentheticals with a verbum dicendi:

(35) Perot, sagte ein Manager, sei ein autoritärer Macher.
Perot, said a manager, is an authoritarian doer.
'Perot, said a manager, was an authoritarian doer.'

Sentences of this kind were already captured in the original grammar by a rule that allows verb-initial reportive parentheticals to appear anywhere within a matrix clause and where the following optional equation is part of the f-annotation of the parenthetical: `(!COMP) = ^`.

In German newspaper text, a further type of reportive parenthetical occurs relatively frequently. These have the same distribution as the type just discussed, but they do not contain any 'real' verbum dicendi, i.e. a reportive verb that subcategorizes for a Comp, which means that the annotation just presented does not work for them. The following sentence is an example of this other type of reportive parenthetical:

(36) Perot sei, beschreibt ihn ein Manager, ein autoritärer Macher.
Perot is, describes him a manager, an authoritarian doer.
'Perot is, says a manager describing him, an authoritarian doer.'

The host clause represents reported speech, but the reportive parenthetical which introduces the reported speech does not contain a verb of saying. *Beschreiben* does not subcategorize for a sentential Complement; it takes a Subject and an Object. Other verbs or verbal constructions introducing this type of reported speech in the TIGER Corpus are: *sich ärgern* 'be annoyed', *resignieren* 'resign', *die Hoffnungen dämpfen* 'reduce the hope', *eine Erklärung versuchen* 'give a tentative explanation'. This short list gives an impression of how creative journalists are in choosing a verb(al construction) for this type of parenthetical. Although semantically, all of them express an activity of saying or thinking, they take diverse syntactic forms, which makes it impossible to list them, e.g. in one of our subcategorization lexicons.

Despite this difference with respect to their verbal head, reportive parentheticals with a verbum dicendi and their counterparts without a 'real' verb of saying behave largely alike, both with respect to their distribution and semantically. In our new analysis of the latter, we tried to capture this parallelism, by introducing an alternative to the `(!COMP) = ^` annotation presented above, namely `(!REPORTEDSPEECH) = ^`. ReportedSpeech is defined as a semantic function, similar to an Adjunct, whereas Comp is, of course, a grammatical function. Apart from the fact that no element subcategorizes for the ReportedSpeech sub-f-structure, the analyses of sentences containing a verb-initial reportive parenthetical look alike, regardless of whether or not the parenthetical is headed by a verb of saying. This is illustrated in Figure 6:

By restricting the semantic function ReportedSpeech to verb-initial

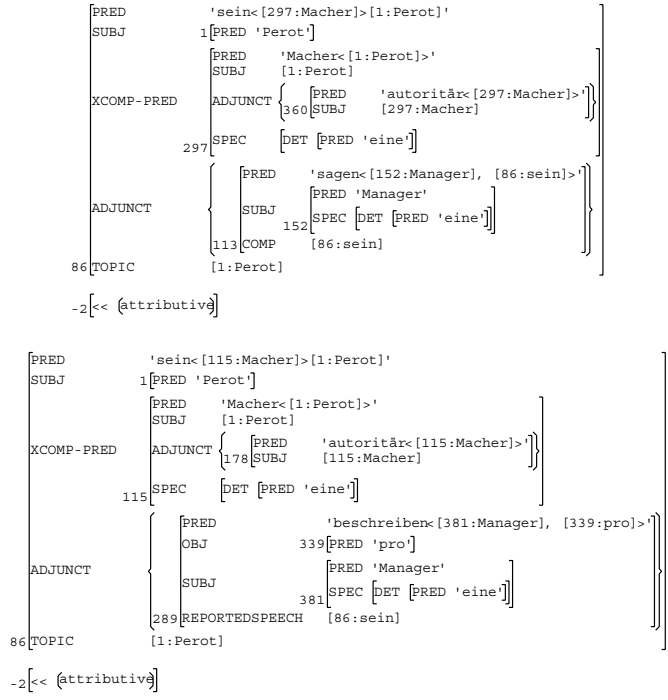


FIGURE 6 F-structures associated with (36) and (35)

reportive parentheticals, we can largely avoid overapplication of this additional rule. There are very few cases where our parser erroneously analyses a clause as ReportedSpeech.

8.3.3 Elliptical constructions

Newspaper texts contain many elliptical constructions like (37), which is an elliptical version of (38).

(37) Schade, dass Beton nicht brennt.
Pity, that concrete not burns.
'Pity that concrete does not burn.'

(38) Es ist schade, dass Beton nicht brennt.
It is pity, that concrete not burns.
'It is a pity that concrete does not burn.'

For many elliptical constructions, it is not realistic to define a rule that does not seriously affect efficiency. For these, fragment parses will be all that can be achieved.

For certain recurrent elliptical constructions, however, it is worth

defining special rules so that they receive a full analysis. This is the case of the above example. For this type of construction, the missing copula is 'reconstructed' in the f-structure, which can be done without a negative impact on efficiency, because the distribution of this construction is fairly restricted: the PREDP must be in sentence-initial position, as the corresponding rule illustrates.

```
CProot[std] -->
            PREDP[std]: (^XCOMP-PRED) = !
                        (^SUBJ) = (^XCOMP-PRED SUBJ)
                        (^PRED) = 'sein <(^XCOMP-PRED)>(^SUBJ)';
            COMMA: ^ = !;
            CPdep[std]: (^SUBJ) = !;
            PERIOD: ^ = !.
```

8.3.4 Lexically restricted rules

The larger the coverage of the grammar gets, the more lexically restricted are the constructions which are still not covered. In order to avoid unwanted interactions due to overgeneration, these lexical restrictions must appear in the corresponding rules.

For instance, DPs that occur as predicative arguments as in (39) have nominative case in more than 99% of the cases. Nevertheless, there are also genitive DPs that function as predicative arguments, as can be seen in (40) and (41). These, however, always have a lexical head from a relatively short list of nouns that can be enumerated.

(39) Sie ist **eine gute Lehrerin**.
She is a-NOM good-NOM female teacher-NOM.
'She is a good teacher.'

(40) Deutscher Großhandel ist **guter Dinge**
German wholesale is good-GEN things-GEN
'German wholesale is optimistic'

(41) Er ist **der Meinung, dass die Aktien steigen**.
He is the-GEN opinion-GEN, that the equities rise.
'He is of the opinion that equities will rise.'

In order to cover these predicative arguments in genitive case, the original PREDP rule is modified as follows:

```
PREDP[_type] -->
             DP[_type]: ^ = !
                        { (!CASE) = nom
                        | (!CASE) = gen
                          (!PRED FN) $c { Ding Mut Ansicht
                                          Auffassung Meinung...}
                        }.
```

Note the constraint on the Pred of the predicative DP that holds when it is in genitive case.

The fact that our efforts to enhance grammar coverage forced us introduce rules that only apply to a very limited list of lexemes is not only interesting for the sake of the constructions under consideration, but it signals something more general about our grammar development efforts: we seem to level out. In other words, the effort that we have to put into the definition of additional rules is in an increasingly unfavorable relation to the gain in coverage that is achieved. The law of diminishing returns seems to be at play.

A further observation with respect to these lexically restricted constructions is their link to multi-word expressions (MWEs). MWEs present a largely unsolved problem for NLP, and this is even more true for those MWEs that exhibit a large degree of variation. The 'constructions' presented above seem to be somewhere in between syntax and the lexicon; just as they can be considered lexically restricted 'constructions', they can also be considered very variable MWEs. Apart from the lexical restrictions that apply to them, a further argument in favour of this view is that these constructions are often also restricted formally, e.g. with respect to number and possible modifiers. Hence, it does not come as a surprise that they were not treated at all in the original grammar and that their treatment in the revised grammar is still tentative.

8.4 Corpus-based Methods to Restrict Grammar Rules

Apart from gaps in the grammar, which can cause sentences not to receive a full parse, processing efficiency is an important issue when scaling a hand-crafted grammar to free text. In fact, for a considerable percentage of the sentences of the TIGER Corpus (about 5%), the original grammar could not produce an analysis within the given time limit of 300 seconds or the memory limit of 2 GB on a Linux PC with two Intel Xeon 2.8 GHz CPUs and 4 GB of RAM.[6] For these sentences, we do not know whether or not they are covered by the grammar rules, but we do know that the parser builds a lot of structures for them which are expensive to process.

To identify the rules that cause efficiency problems, we inspected sentences that, despite their relatively short length, timed out or caused a storage overflow. The causes of inefficiency that we identified can be classified as follows: overly general rules; complex functional uncer-

[6] On less powerful machines, this percentage is, of course, even higher.

tainty equations; inside-out function application; application to very long strings of rules that typically cover short spans. In the following, we present each of these types of inefficiencies and what can be done to remedy them.

This said, it has to be kept in mind that there sometimes is a real tension between linguistic adequacy and generality on the one side and processing efficiency on the other side. As the original grammar was mainly developed in a linguistically inspired fashion, it is not surprising that its efficiency needed improvement. Nevertheless, linguistic generalizations implemented in a grammar may also turn out to be overkill or even linguistically inadequate when tested on corpus data.

8.4.1 Rule specialization

In the original version of our grammar, the aim was to write rules as generally as possible. However, when taking a closer look at corpus data and their processing, we observed that these generalizations can lead to efficiency problems and even turn out not to reflect corpus data adequately. The example of participial VPs will help us to illustrate the importance of avoiding generalizations that go too far and, hence, the justification for rule specialization. Specialized rules in the same spirit were also developed for nominalized APs and non-deverbal APs in adverbial function.

Participial VPs in adverbial function

Uninflected participial VPs have the distribution of uninflected APs, which, in German, can function as Adjuncts of a clause. In the original grammar, this state of affairs is expressed in the rules `AP[std,-infl]` → `VP[v,-infl]` and `ADVP[std]` → `AP[std,-infl]`, which express the very strong generalizations that any uninflected participial VP has the distribution of an uninflected AP and, more problematically, that any uninflected AP has the distribution of an ADVP.

With respect to efficiency, this double generalization is problematic because it allows for the construction of a lot of undesired c-structures. Let us consider the following sentence:

(42) Weil er die Belegschaft im letzten Jahr um 6775
Because he the workforce in the last year by 6,775
Arbeitskräfte auf 277 353 Beschäftigte aufgestockt hat ...
workers to 277,353 employees increased has ...
'Because, during the last year, he has increased the workforce by 6,775 workers to 277,353 employees ...

Among the suboptimal readings of this sentence produced by the original grammar, there are analyses with *hat* as a main verb (*er hat die*

Belegschaft 'he has the workforce'). Strings (a) through (d) can then be interpreted as an ADVP that functions as an adjunct of this main verb:

(a) *aufgestockt*
(b) *auf 277 353 Beschäftigte aufgestockt*
(c) *um 6775 Arbeitskräfte auf 277 353 Beschäftigte aufgestockt*
(d) *im letzten Jahr um 6775 Arbeitskräfte auf 277353 Beschäftigte aufgestockt*

At the stage of context-free parsing, i.e. before the solution of f-structural constraints, even the strings in (e) and (f) are parsed as ADVPs.

(e) *die Belegschaft im letzten Jahr um 6775 Arbeitskräfte auf 277353 Beschäftigte aufgestockt*
(f) *er die Belegschaft im letzten Jahr um 6775 Arbeitskräfte auf 277353 Beschäftigte aufgestockt*

Of course, the larger such a potential ADVP, the more attachment ambiguities arise in it and the more serious is the impact on efficiency.

A further concern is that the vast majority, if not all, of the additional readings that the generalization encoded in the original grammar gives rise to are linguistically inadequate. In (42), the treatment of *haben* as a main verb is clearly a misinterpretation. The same holds in similar sentences with a past participle followed by a form of the auxiliary *haben (to have)* and potentially preceded by PPs and ADVPs, and similar misinterpretations occur in sentences with the auxiliary *sein (to be)* like (43), where the original rule produces a reading with *ist* as a form of the copula and *in London gewesen* as an ADVP.

(43) In London gewesen ist er gestern.
In London been has he yesterday.
'He was in London yesterday.'

But what can be done about these problems with efficiency and linguistically inadequate readings? A query on the TIGER Corpus searching for past participle VPs that function as modifiers (MO) shows that only a small fraction of them function truly adverbially, and if they do, they are just modified by one or two ADVPs or PPs. Most instances, especially longer and more complex ones, are secondary predications in the Vorfeld or Nachfeld position, often separated by commas. This means that the generalization stating that any arbitrarily complex participial VP can function as an Adjunct of a clause is not confirmed by the corpus data. The best solution to the problems discussed above thus seems to be the definition of a specialized rule for adverbially used

participial VPs. By drastically reducing its possible expansions, this specialized rule addresses efficiency and unnecessary ambiguity at the same time and thus kills two birds with one stone. This is the rule used in the current version of the grammar:

```
ADVPvp -->
       { ( PP[std]: { ! $ (^ADJUNCT)
                    | (^OBL) = !
                    } )
         ( ADVP[std]: ! $ (^ADJUNCT) )
         V[v,-infl]: ^ = !
       | V[v,-infl]: ^ = !
         PP[std]: { ! $ (^ADJUNCT)
                  | (^OBL) = !
                  }
       }.
```

Apart from restricting the number of PPs and ADVPs in adverbially used participial VPs, this rule solves a further problem that existed with the original implementation: it prevents ADVPs and PPs from appearing simultaneously both to the left and to the right of the participial head of the VP; according to the original rule, this was possible, since adverbially used participial VPs were analyzed via the general VP rule. The data in (44), (45) and (46) support this restriction, and so do the findings from the TIGER Corpus, which does not contain a single occurrence of a participial VP that is labeled as MO and has ADVPs/PPs both to the left **and** right of the participle.

(44) gemessen an den Konzerneinnahmen
measured at the group income
'measured with respect to the group's income'

(45) am DJS gemessen
at the DJS measured
'measured with respect to the DJS'

(46) *gestern in Frankfurt gemessen am Dax
yesterday in Frankfurt measured at the Dax
'measured yesterday in Frankfurt with respect to the Dax'

Participial VPs as attributive APs

With inflected participial VPs, similar problems arise as with the uninflected participial VPs, although there are also some differences in their respective behaviour. One difference is that in inflected participial VPs, nothing can follow the head participle, whereas this is possible in adverbially used participial VPs, as we have just seen. A further difference is that the number and type of constituents that occur before the head

participle are considerably less restricted in inflected attributive APs than in their adverbially used counterparts, although they are considerably more restricted than in finite VPs. This motivated the definition of a specialized rule for participial VPs that function as attributive APs.

One important feature that both specialized rules, the one for adverbially used participial APs and the one for attributively used ones, have in common is that they exclude recursion in the verbal complex, i.e. the embedding of an infinitival VP as an argument in the participial VP. This is motivated by the fact that, in the TIGER Corpus, there is not a single occurrence of an AP whose head is a participle and which dominates a VP, nor are there VPs labeled as MO that are headed by a participle and dominate another VP.

The exclusion of recursion in deverbal APs has a very positive impact on the efficiency of the grammar because there are numerous forms that can be both an inflected or uninflected past participle and a finite verb form. Consider the following subordinate clause:

(47) Weil er die Frau die Aktien zu verkaufen überredete, ...
Because he the woman the shares to sell convinced, ...
'Because he convinced the woman to sell the shares ...'

The form *überredete* can be both a past tense form and an inflected past participle. As the original grammar allows infinitival VPs to be embedded in attributive deverbal APs, it can, at least temporarily, analyze the string *die Aktien zu verkaufen überredete* as an inflected AP, and since the grammar contains a rule that allows inflected adjectives to be analyzed as the head of an NP, this inflected AP can then be analyzed as an NP without a head noun and, finally, even as a DP. With forms that can be both a finite verb form and an uninflected past participle, the problems are similar. Of course, no NP and DP trees can be build on top of them, but they can still be interpreted as ADVPs in the chart. In both cases, this means that a large number of undesired local c-structures are built which are only ruled out higher up in the chart or even during the solution of the f-structure constraints. Of course, with respect to efficiency, it is a very attractive feature of the revised grammar that these erroneous c-structures are not built in the first place.

8.4.2 Restricting functional uncertainty equations

Solving the equations which account for long distance dependencies can be very expensive in terms of time and computational resources. We therefore simplified these equations based on a corpus study. They concern mainly extraposed relative clauses as in (48), comparative arguments as in (49) and extraposed Comp CPs and VComp VPs like the

daß clause in (50).

(48) Er hat [Deponien [von Firmen]] untersucht, die giftige
He has dumps of companies examined, which poisonous
Chemikalien herstellen.
chemicals produce.
'He has examined dumps of companies that produce poisonous chemicals.'

(49) Er hat [Deponien [von mehr Firmen]] untersucht als die Behörde.
He has dumps of more companies examined than the authority.
'He has examined dumps of more companies than the authority.'

(50) Er hat [den Wahrheitsgehalt [von Gerüchten]] untersucht, dass
He has the truth of rumours examined that
die Firmen giftige Chemikalien herstellen.
the companies poisonous chemicals produce
'He has examined the truth of rumours that the companies produce poisonous chemicals.'

All of these extraposed constituents may be dependents of a noun (or an adjective or a quantifier in the case of comparative arguments) which is embedded at arbitrary depth inside the Mittelfeld VP. These long distance dependencies are accounted for by means of functional uncertainty equations that contain the templates VP-PATH and DP-PATH. In the original grammar, they were defined very generally:

```
VP-PATH =  XCOMP* (VCOMP) XCOMP*.

DP-PATH = { SUBJ
          | VP-PATH
            { OBJ
            | OBJ2
            | OBL OBJ
            | OBLsem OBJ
            | ADJUNCT $ OBJ
            | ADJ-GEN
            | XCOMP-PRED (OBJ) }
          }
          { ADJUNCT $ OBJ | ADJ-GEN }*.
```

A corpus study showed, however, that the deep embeddings made possible by the Kleene stars in both templates occur only rarely in free text. We therefore changed VP-PATH as follows:

```
VP-PATH = ((XCOMP) { VCOMP | XCOMP }).
```

In DP-PATH, the final disjunction that, due to the Kleene star, could be iterated arbitrarily many times was replaced by a single optional

occurrence of this disjunction:

```
DP-PATH = ...
          ({ ADJUNCT $ OBJ | ADJ-GEN }).
```

The simplification of VP-PATH had practically no effect on performance. The simplification of DP-PATH, however, improved performance. On a testsuite of 5,000 sentences, it reduced the number of skimmed sentences by 21. This simplification also had a positive effect on the overall quality of the analyses. This is due to the fact that the parsing quality of skimmed sentences with partial analysis is always worse than the parsing quality of sentences which get a full parse (see Section 8.5). So even if individual extraposed relative clauses, comparative arguments or *daß* clauses are no longer attached to their correct antecedents in some sentences due to the simplification, the overall parse quality improves.

8.4.3 Inside-out function application

Inside-out function applications are formally very similar to functional uncertainty equations. Another characteristic that they share with the latter is that they are sometimes time-consuming to process. The original German grammar, for example, contained a template which, for every DP argument, checked via inside-out function application whether it constituted the Topic of the sentence. This check was used to disprefer objects in topic position with the help of an OT mark. The check, however, proved to be very costly. As, moreover, experiments on the corpus-based learning of the OT mark ranking (Forst et al. 2005) showed that the concerned dispreference mark is not reliable enough for a pre-filter like the OT-inspired disambiguation module, the check was deactivated altogether.

We also deactivated the template which forces agreement in person and number between the subject and the corresponding reflexive pronoun. Although this constraint makes perfect sense linguistically and its deactivation leads to overgeneration, this helps to improve the overall parse quality. Again, this is due to the fact that improved efficiency prevents a number of sentences from being skimmed and that full parses are systematically of better quality than skimmed partial parses.

8.5 Evaluation

As is now common practice with hand-crafted grammars, we evaluated the quality of the parses produced by the revised LFG on manually validated dependency annotations for 1,602 randomly selected sentences from the TiGer Dependency Bank (Forst et al. 2004). This dependency

bank encodes the same type of dependency triples as the PARC 700 Dependency Bank (King et al. 2003). At the same time, the grammatical relations and morphosyntactic features are the same as in the TIGER Treebank, except for systematic changes meant to make the TiGer DB more suitable for parser evaluation.

For evaluation, the f-structures produced by the grammar are converted into the same type of dependencies, so that precision and recall of the parses with respect to the gold standard can be computed. As a measure that combines precision and recall, we indicate F-score, which is defined as the harmonic mean of the two. We use the triple encoding and evaluation software of Crouch et al. (2002).

8.5.1 Robustness techniques

In order to collect as much information as possible in cases where a sentence does not get a full parse, we augmented the standard grammar with a FRAGMENT grammar. The parser returns well-formed chunks like DPs, PPs, VPs, CPs, etc. The grammar has a fewest-chunk method for determining the least fragmented parse. It turned out that the quality of fragment parses can be improved by restricting complex rules (e.g. the CP rule) in the fragment grammar with respect to the standard grammar.

In order to cope with timeouts and memory problems, we use the skimming technique (see Riezler et al. 2002). When skimming, we use a restricted version of our grammar. This is achieved with the help of special OT marks (Frank et al. 2001), so-called SKIMMING_NOGOOD marks, which turn off expensive rules like headless NPs, 'free' datives, etc. during skimming.

8.5.2 Results

Since for the test set, we simulated a perfect named-entity recognizer by manual marking, the coverage is very high: 88.6% of the sentences receive a full parse. This figure drops noticeably on corpora where the named-entities are not marked, but for no corpus that we parsed did it drop below 80%. All the remaining 11.4% of the sentences receive a fragment parse, out of which a bit less than a third are 'skimmed'. See Table 1 for the exact figures. Thanks to the robustness techniques implemented in XLE, we achieve an overall coverage of 100%.

Symbolic grammars like the LFG presented here often output highly ambiguous analyses. As our disambiguation module is not yet completed, we give the results of two types of parse selection that indicate the range within which the results of the final automatic parse selection will be: (1) lower bound: In the lower bound a parse from the set of

parses is chosen randomly. (2) upper bound: In the case of the upper bound, the best F-score according to the annotation scheme is chosen.

	all	full	full and non-sk. fragments	frag.	non-skimmed frag.	skimmed frag.
% of test set	100	88.6	96.6	11.4	8.0	3.4
upper bound	87.3	88.8	87.9	76.0	78.7	69.7
lower bound	81.7	83.6	82.9	72.2	74.2	66.3
avg. sentence length	16.2	14.9	15.2	24.6	17.6	41.7
avg. parse time in sec.	3.91	1.52	2.64	18.56	6.00	56.78

TABLE 1 F-scores for grammatical relations and morphosyntactic features in the 1602 TiGer DB examples broken down according to parse quality

As Table 1 shows, the overall F-score is between 81.7% (lower bound) and 87.3% (upper bound). Parsing quality is best for full parses and worst for skimmed fragment parses, in terms of both upper bound and lower bound. The average length of the sentences that receive full parses is shortest (14.9 words) and the length of those that receive a skimmed fragment analysis is longest (41.7 words). The same obervation holds for the average parse time, which is 1.52 seconds for sentences that receive a full parse and 56.78 seconds for skimmed fragment parses. This is to be expected, as the skimming mechanism handles sentences that pose efficiency problems, and long sentences are, of course, more likely to pose such problems than short sentences.

For F-score, precision and recall figures broken down according to grammatical functions and morphosyntactic features, see Rohrer and Forst (2006). That paper also contains preliminary figures for automatically disambiguated parses.

8.5.3 Discussion

In order to get a full parse, the input sentence has to be well-formed. At least 1% of the sentences in the testsuite contain spelling mistakes, punctuation errors or grammatical errors.

Among the well-formed sentences which receive a partial parse we have to distinguish three types: (1) constructions for which our grammar contains rules which are turned off for efficiency reasons (e.g. coordination without an explicit conjunction, floating quantifiers, very complex adverb phrases, etc.) (2) constructions for which we do not have rules (e.g., special types of non-constituent coordination, certain parenthetical constructions, heavy ellipsis), (3) sentences which contain lexical material that is not in the lexicon and which our guesser

cannot handle (e.g., problems of subcategorization, idioms and collocations which do not follow 'normal' syntactic rules, lexical material from other languages which is not enclosed in quotation marks).

Our upper bound for full parses is roughly identical to the result that Riezler et al. (2002) achieve for English. Our values for the complete test set are better than their results (87.3% vs. 84.1%) because more sentences of our testsuite receive a full parse. If we subtract the 55 sentences with an average length of 41.7 words that get a partial parse after skimming, we obtain for 96.6% of our testsuite an upper bound of 87.9% and a lower bound of 82.9%.

8.6 Conclusion and Outlook

We have presented the three main domains in which work needed to be done to scale to newspaper text a hand-crafted LFG grammar that was initially developed in a principally phenomena-driven fashion: preprocessing via FSTs, grammar extension and performance tuning via the restriction of grammar rules. We have shown that the resulting revised LFG grammar achieves good results when evaluated against a dependency-based gold standard. The results can compete with other broad-coverage parsers for German, although a few other parsers achieve even higher F-scores. However, to our knowledge the LFG grammar described in this article is the only 'deep' and reversible grammar for German that has been scaled to free text. This will make it an obvious candidate for application contexts requiring great depth of analysis and the capacity to generate, such as question answering or machine translation, as soon as our disambiguation module is completed.

In the process of scaling-up, insights from linguistic theory can be integrated fruitfully into the grammar, but there are also phenomena which, despite their frequency in corpora, have not received much linguistic attention. For still others, linguistically valid generalizations have a negative effect on efficiency so that, in terms of F-score achieved on dependency triples, the rules for these phenomena do more harm than good. Syntactically annotated corpora such as the TIGER Corpus, coupled with powerful query tools such as TIGERSearch, make it possible to estimate the frequency of constructions and, hence, the importance of grammar coverage for them. It may be better to deactivate rules for constructions that turn out to be expensive to process, if the constructions occur only rarely in corpora of the genre that is to be parsed.

As to further possibilities of improving the coverage and the parsing

quality of the grammar, i.e. of raising the upper bound, we believe that more work can be done, in particular with respect to the modelling of multi-word expressions (MWEs), but as far as general grammar rules are concerned, we believe that we have almost leveled out. Adding a new rule for some rare construction, at least if it is not strictly restricted to certain lexical items, now almost always affects efficiency, with the result that what is won in terms of F-score in the newly parsed sentences is lost in other sentences that do not receive a full parse any longer because of timeouts or storage overflows. Apart from MWEs, our main focus in ongoing and future work is thus disambiguation, since a good stochastic disambiguation component will help us to maximally exploit the potential of the symbolic LFG grammar, i.e. to perform an automatic parse selection that comes as close as possible to the upper bound.

Acknowledgments

The work described in this paper was financed by the DFG (Deutsche Forschungsgemeinschaft — German Research Foundation) as part of the *DLFG* project (Disambiguierung einer Lexikalisch-Funktionalen Grammatik für das Deutsche — Disambiguation of a Lexical Functional Grammar for German, grant *Ro 245/18-1*) . The development of the initial grammar was also funded in part by the DFG. The grammar writers (and, hence, the grammar) have greatly benefited from the extremely fruitful cooperation and exchange within the *ParGram* initiative, of which Ron Kaplan is one of the founding members. We would like to thank him and his colleagues that have contributed ideas, technical improvements in XLE and lots of other kinds of support.

References

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In E. Hinrichs and K. Simov, eds., *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT'02)*, pages 24–41. Sozopol, Bulgaria.

Cahill, Aoife, Michael Burke, Martin Forst, Ruth O'Donovan, Christian Rohrer, Josef van Genabith, and Andy Way. 2005. Treebank-based multilingual unification-grammar resources. *Research in Language and Computation* 3(2):247–279.

Christ, Oliver. 1994. A modular and flexible architecture for an integrated corpus query system. In *Proceedings of the 3rd Conference on Computational Lexicography and Text Research (COMPLEX'94)*, pages 23–32. Budapest, Hungary.

Crouch, Richard, Ronald M. Kaplan, Tracy H. King, and Stefan Riezler. 2002. A comparison of evaluation metrics for a broad-coverage parser. In J. Carroll, A. Frank, D. Lin, D. Prescher, and H. Uszkoreit, eds., *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02), Workshop 'Beyond PARSEVAL: Towards improved evaluation mesures for parsing systems'*, pages 67–74. Las Palmas, Spain.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, John T. Maxwell, III, and Paula Newman. 2006. XLE Documentation. Palo Alto Research Center.

Dipper, Stefanie. 2003. *Implementing and Documenting Large-scale Grammars – German LFG*. Ph.D. thesis, IMS, University of Stuttgart. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), vol. 9, no. 1.

Forst, Martin, Núria Bertomeu, Berthold Crysmann, Frederik Fouvry, Silvia Hansen-Schirra, and Valia Kordoni. 2004. Towards a dependency-based gold standard for German parsers – The TiGer Dependency Bank. In S. Hansen-Schirra, S. Oepen, and H. Uszkoreit, eds., *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), 5th Workshop on Linguistically Interpreted Corpora (LINC'04)*, pages 31–38. Geneva, Switzerland.

Forst, Martin, Jonas Kuhn, and Christian Rohrer. 2005. Corpus-based learning of OT constraint rankings for large-scale LFG grammars. In T. H. King and M. Butt, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2005 (LFG'05)*, pages 154–165. Bergen, Norway: CSLI Online Publications.

Forst, Martin and Ronald M. Kaplan. 2006. The importance of precise tokenizing for deep grammars. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy.

Fortmann, Christian. 2005. On parentheticals (in German). In T. H. King and M. Butt, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2005 (LFG'05)*, pages 166–185. Bergen, Norway: CSLI Online Publications.

Frank, Anette, Tracy H. King, Jonas Kuhn, and John T. Maxwell, III. 2001. Optimality Theory style constraint ranking in large-scale LFG grammars. In P. Sells, ed., *Formal and Empirical Issues in Optimality Theoretic Syntax*, pages 367–397. Stanford, CA: CSLI Publications.

Frank, Anette. 2002. A (discourse) functional analysis of asymmetric coordination. In T. H. King and M. Butt, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2002 (LFG'02)*, pages 174–196. Athens, Greece: CSLI Online Publications.

Höhle, Tilmann. 1983. *Topologische Felder*. Ph.D. thesis, University of Cologne.

Kaplan, Ronald M. and John T. Maxwell, III. 1988. Constituent coordination in Lexical-Functional Grammar. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING'88)*. Budapest,

Hungary. Reprinted in Dalrymple et al. (eds.), *Formal Issues in Lexical-Functional Grammar*. CSLI, 1995.

King, Tracy H., Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 Dependency Bank. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), 4th International Workshop on Linguistically Interpreted Corpora (LINC'03)*, pages 1–8. Budapest, Hungary.

Lezius, Wolfgang. 2002. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora*. Ph.D. thesis, IMS, University of Stuttgart. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), vol. 8, no. 4.

Maxwell, John T., III and Christopher Manning. 1996. A theory of non-constituent coordination based on finite state rules. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 1996 (LFG'96)*. Grenoble, France: CSLI Online Publications.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 271–278. Philadelphia, PA.

Riezler, Stefan and John T. Maxwell, III. 2006. Grammatical machine translation. In *Proceedings of the Human Language Technology Conference and the 6th Annual Meeting of the North American Chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL'06)*. New York, NY.

Rohrer, Christian and Martin Forst. 2006. Improving coverage and parsing quality of a large-scale LFG for German. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy.

9

Using a Large, External Dictionary in an LFG Grammar: The STO Experiments

Beau Sheil and Bjarne Ørsnes

9.1 Introduction

Many practical language processing applications (e.g., technical translation) require access to very wide coverage lexicons (e.g., of technical terminology). Even general purpose systems occasionally need to look up uncommon terms. The major ParGram LFG grammar projects (Butt et al. 2002) have therefore devoted significant effort to compile large lexicons, using a combination of techniques, including corpus analysis and automatic and semi-automatic conversion of existing lexical resources, as well as extensive manual coding (Butt et al. 1999).

However, few grammar development projects have the resources to invest in lexical development on this scale. Grammars for some languages, such as English, can avoid some of this effort by relying on a 'guessing' strategy and relatively regular morphology. But there are other languages, such as Danish, where some amount of information, e.g. gender, is required for a great many words.

Further, lexical development for a language is not a 'one-time' cost. Languages evolve, sometimes slowly and organically, and at other times in bursts and torrents. In particular, the language of both popular culture and technical literature, especially in cases such as Danish, where much of it is being imported from other languages, often with varying orthography, can grow at an alarming rate. Maintaining a realistic ap-

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

plication lexicon is thus a daunting challenge, even for those with the resources to develop one in the first place.

Fortunately, in response to this problem, many countries now have nationally funded projects to assemble and maintain definitive, wide coverage dictionaries of their national language(s). Increasingly, some of these are being developed with an explicit goal of directly supporting natural language applications, and thus they attempt to include significant amounts of syntactic information. This raises the intriguing possibility of building a large lexicon for an LFG grammar entirely by mechanical transformation of such a public, evergreen resource. Provided that the descriptions for each item in the source dictionary are sufficiently detailed, i.e. that the distinctions which the grammar writer needs are present in or derivable from the dictionary's entries, this approach might allow even a small development team to have a lexicon of a size previously available only to large projects.

Fully realizing this goal depends essentially on the import of the external dictionary being entirely mechanical. Languages change, and these external dictionaries will change with them, often quite frequently. If we are to profit from their ongoing maintenance and updates, we will have to import this material frequently. A semi-automatic porting process which requires extensive manual processing of the imported material simply makes this impractical.

Of course, there will be places (most notably in the closed word classes of the language) where a general dictionary's data are unlikely to be sufficient. There will be places where the grammar writer will wish to override the dictionary's data. And developers working within a specialized context, say translating tractor manuals, may still need to develop lexical entries for some specialized terms, and fine-tune the interpretations for others. But none of this undoes the potentially huge advantage of not having to develop or maintain entries for general vocabulary, or having to track a host of new terms like 'IPod', which are outside one's immediate application but which might appear from time to time in one's input.

All of the large (i.e., 40,000 stems or more) lexicons developed for LFG grammars that we know of have used external dictionaries as one source (among several) of lexical material. However, although some of this material might have been imported mechanically (e.g., Brazil 1997, cited in Butt et al. 1999:197), only two projects, IMSLex (Lezius et al. 2000) and the Norwegian ParGram group (Rosén 2001), suggest a commitment to deriving their lexicon primarily from an external lexical data source, without subsequent manual editing, and neither of them gives much detail about how it was done. In our experience, it is not

quite as simple a process as this might suggest. So, one goal of this paper is simply to provide one reasonably detailed account of the issues that arise in such a project.

There are four broad issues involved in using a big, external dictionary:

- Accessing the material from the dictionary from within the LFG grammar,
- Mapping the information from the dictionary into LFG terms,
- Dealing, in the grammar, with the onslaught of ambiguity that using a really large lexicon produces, and
- Software engineering issues concerned with managing the size of the very large lexicons that result.

We will discuss each of these in the context of experiments that we have carried out to use the STO (SprogTeknologisk Ordbase) dictionary of Danish from within the Danish LFG grammar. Some of our observations are, of course, specific to this dictionary-grammar pair, although the issues we faced would probably be of interest to anyone embarking on a similar project, or on a project to produce a 'multi-purpose' lexical resource. But, in addition, we have developed some engineering techniques that may be generally applicable to using very large scale lexicons within the XLE implementation of LFG (Crouch et al. 2006).

9.2 A Broad Coverage LFG Grammar for Danish

Until recently there was no broad-coverage computational grammar of Danish in a contemporary linguistic framework, and indeed there are few descriptions of Danish in any formal syntactic framework. This was the main motivation for launching a project, funded by the Nordic Research Council, to start developing such a grammar as a linguistic resource for a range of natural language applications.

The Danish grammar is an LFG grammar (Kaplan and Bresnan 1982), implemented in XLE (Maxwell and Kaplan 1993). Throughout its development, this project has been affiliated to the ParGram project (Butt et al. 2002). The Norwegian ParGram grammar (Dyvik 1999), in particular, has been a major source of inspiration. However, instead of porting the Norwegian grammar into Danish (a process described by Kim et al. 2003), the Danish grammar was developed from scratch in a traditional fashion, covering the basic constructions and phrase types, and incrementally extended to cover additional constructions found in real-life texts, especially newspaper texts. The goal has been to produce linguistically sophisticated analyses of a wide range of frequent constructions based on existing LFG analyses, and, in addition,

to provide analyses of phenomena that are hardly covered at all in the prior literature on Danish, exploiting the key insights of existing LFG theory. Examples of such phenomena are cleft-sentences with non-core grammatical functions as the clefted element, pronouns in unbounded dependency constructions (Ørsnes 2002), VP-topicalization (Wedekind and Ørsnes 2004), and passivization of verbs with propositional complements (Ørsnes and Wedekind 2006). The grammar thus not only serves as a resource for natural language applications, but also as a test-bed for descriptions of Danish in a contemporary constraint-based linguistic framework.

The focus has been on syntax. No attempts have been made until now to integrate a morphological module, or to make use of projections other than c- and f-structure. In parallel, sub-projects have explored new implementation techniques, e.g. of the verbal complex using the restriction operator (e.g., Wedekind and Ørsnes 2003).

At an early stage, it was decided not to develop a large lexicon but instead to rely on eventually importing lexical material from external sources. An initial lexicon was developed using a set of templates designed to capture the syntactic and morphological distinctions anticipated by the grammar. It was not clear that these templates would be reused directly when importing external lexical resources (and in fact they were not in this project) because external resources have their own structures. Often it is easier to provide definitions for these structures in terms of primitive operations than it is to construct a mapping between them and an existing template set. However, the templates did provide a specification for the lexical representation needed by the grammar and, in this way, laid a foundation for the later import of external lexical resources. The grammar itself was developed using a typical 'linguist's workbench' lexicon of a few hundred stems entered by hand. Accessing the enormously wider coverage of a large scale dictionary therefore had both obvious practical appeal and significant theoretical interest. How would the existing grammar have to change to handle the range of phenomena that such a large dictionary would contain?

9.3 The STO Dictionary of Danish

The STO is an orthographic, computational lexicon developed by the Center for Language Technology in collaboration with the University of Copenhagen, Copenhagen Business School and the University of Southern Denmark. Its development was initiated by the Danish Ministry of Research with the goal of developing a flexible, i.e. theory-neutral, broad-coverage, lexical resource for language technology applications.

STO is intended for use not only in research, but also in NLP applications, such as machine translation, tools for machine-aided translation, speech synthesis, speech recognition and computer-aided language instruction. It contains both general vocabulary and also specialized vocabulary from domains such as computing, environment, health care, business, finance and public administration. The lemma selection and the lexical descriptions themselves were developed using corpora. For the specialized vocabulary, corpora were compiled from the internet especially for use in the development of STO (Jørgensen et al. 2003).

As of May 2005, STO contained some 81,000 lemmas, some 68,000 from common vocabulary and some 13,000 from language for specialized purposes from six different domains (Braasch and Olsen 2004). Most of these lemmas are annotated with morphological, syntactic, and semantic information, coded on the basis of extensive use of corpora. About 45,000 lemmas have associated syntactic information, using some 1,400 syntactic descriptors. The rest are mainly common nouns or instances of word classes such as prepositions, for which STO does not yet provide explicit syntactic descriptions. About 10,000 lemmas also have semantic annotations. Almost all lemmas are also annotated with inflectional and compound morphology, producing some 694,000 inflected forms.

The considerable size of the STO dictionary, in terms of stem forms, is largely the result of two things. First, the dictionary includes a large number of technical terms, as mentioned above, including a range of borrowed words from other languages (sometimes rendered with varying orthography). Second, Danish, like many Northern European languages, compounds freely, and the dictionary contains many compounded forms. Early in this project, an attempt was made to reduce the number of stems by representing these compounds productively. However, despite the presence of a substantial amount of compounding information in the dictionary entries, it proved difficult to develop a compound analyzer/generator that would reduce the stem set significantly without introducing an unacceptable number of invalid forms. Even an attempt to decompose compounds in the STO dictionary solely on the basis of stems already available in the dictionary led to numerous nonsensical segmentations and ambiguous structures for compounds. So, it was decided to proceed with the stem set as given. If an accurate treatment of Danish compounds were available, some of our implementation choices for Danish might be reconsidered. But stem sets of a comparable size are bound to arise in other situations, e.g. in the treatment of technical terminology, so techniques for handling very large stems sets are of interest in any case.

Physically, STO takes the form of an Oracle database which is hosted at the Center for Language Technology. In database form, it occupies approximately 50MB; as dumped text files, approximately 25MB. A Web-based tool (`http://www.cst.dk/sto/webinterface/`) is provided for browsing the data, and extracts from the database can be prepared on request for particular applications.

9.4 Accessing the STO Dictionary from within the Danish Grammar

The XLE environment in which the Danish grammar is implemented had no facilities for accessing grammatical material other than by reading text files when this work was started.[1] Therefore, rather than access the STO database directly, e.g. to look up the syntactic properties for some word, it was necessary to extract this information, and process and reformat it into text files of the correct format for XLE.

Most database systems, including the Oracle database in which STO resides, have some kind of built-in programming facilities which would allow this. However, the code described here was written in pure Java, for several reasons. First, although the source database for the STO material is Oracle, there was good reason not to tie this code to that platform. For example, writing the code in a platform independent way made it possible to recreate the STO database on another platform and use that for development when the source database could not be accessed directly. Second, it was not clear in advance how difficult the transformations required might be, so having a fully functioned programming system in which to do the development might be of great value. This proved to be a good choice at a later stage when, among other things, code to transform LFG equations symbolically had to be written. Finally, as will be discussed later, it might be very interesting to configure code like this as a server that could be consulted by an XLE based process, perhaps remotely over a network. The choice of Java would make it easy to deploy this code in that fashion.

For now, the code operates by accessing the STO database using SQL queries, processing the retrieved material and then writing out text files that are input to XLE and its tools. The code contains four major sub-systems, comprising a total of about 6,000 lines of Java code.

[1] In 2005, 'grammar libraries' were added to XLE to allow an external program to provide morphological analysis. These can be used to provide a 'server' interface for dictionary data to XLE in ways that will be discussed later.

9.5 Using the STO Dictionary Material in the Danish LFG Grammar

An XLE grammar needs at least two different kinds of information about the word forms it deals with: morphological, which relates each surface form to a base form and a set of inflectionally marked properties (e.g. singular, possessive, etc.), and syntactic, which provides for each base form its part of speech and other information about the syntactic patterns it can appear in. Traditionally, in XLE, the morphology is encoded as a finite state transducer (FST) that maps each inflected form to a base form and a set of 'tags' that encode the inflectionally marked properties. The syntactic information is usually represented as one or more text lexicon files which give, for each base form, its part of speech and a specification of the syntactic patterns it can appear in. (For a more detailed description, see Butt et al. 1999, 11.3 ff.)

Another approach, which might actually be practical for large vocabularies if one could use a database to hold and access the information, would be to use a full-form lexicon (i.e. one that has a full, separate lexicon entry for each inflected form). However, without using grammar libraries, it is not possible to access lexical material in an external database directly from within XLE, and the sheer number of inflected forms in STO (nearly 700,000) argued against building text file lexicons of this size. We therefore began by mapping the STO data into a classical XLE stem lexicon, using a separate FST to represent the morphology.

9.5.1 Morphological information

STO represents morphological information in a very convenient way for building a FST morphology. Each stem in STO is associated with one or more 'inflectional bundles' which specify a string to be added to the stem, one to be removed, and a set of inflectional tags that apply to the result. Thus, the 'bundle' `MFG0257`, which includes

```
-el +lerne NEUTER PLURAL DEFINITE
```

maps (among others) the stem form 'hængsel' (Danish)/'hinge' (English) into the surface form 'hængslerne'/'the hinges'. Generating a FST from this data is very straightforward. Every stem form in the dictionary can be expanded into all of its surface forms, with the inflectional information for each surface form appended as tags. Thus, the example above adds a single line

```
hængslerne : /HÆNGSEL/ +Neuter +Plural +Definite
```

to the morphology input file. A finite state morphology transducer built from a set of lines like this will map an input token like 'hængslerne'/

'the hinges' into the set of tokens that appear to the right of the colon above. In that set of tokens, HÆNGSEL links to the stem

```
HÆNGSEL N XLE @Dn0.
```

where `N` gives the part of speech (Noun) and the template `@Dn0` defines its syntax properties, as described below. The rest of the tags, e.g. `+Neuter` etc., are used by a sub-lexical rule in the grammar to add XLE equations to assert the corresponding properties of the inflected form, e.g.

```
+Neuter VTAG XLE (↑ AGR GEND)=neut.
```

The tag `+Neuter` adds the feature specification that the lexical item has the value neut(er) as value of the gend(er) feature among the agr(eement) features.

In this way, a complete lexical entry can be assembled by combining the information from the inflectional tags and their associated meanings with the information associated with the stem form in the lexicon. And this is indeed the classical XLE approach.

This was easy to implement using the STO data. Two simple database queries and a small amount of text manipulation (e.g. to map STO's inflectional values into unique sub-lexical tags) produced a 41MB text file of inflectional data. XLE's built-in FST utility `create-transducer` reduced this to a comfortably sized 499KB FST morphological analyzer. Definitions for each of the 81,000 stems were written into a 2.8MB lexicon file.

9.5.2 Syntactic information

Accessing the syntactic information in the STO dictionary was always going to be more challenging than accessing the morphological information. Any dictionary must have some implicit syntactic theory, even if all it does is discriminate between transitive and intransitive verbs. Dictionaries that are intended as a base for computational analysis, such as STO, have a lot more. But, unless they have been designed within the same syntactic theory as one's grammar, there will be issues of how well the two descriptions match. These are either of practical inconvenience or of theoretical interest, depending on one's point of view.

Most of the 81,000 stems listed in the lexicon file just described are associated in STO with one or more of about 1,400 syntactic 'descriptors', and virtually all the syntactic information is keyed off those descriptors. The problem is to map each of these descriptors into LFG terms. We begin with a brief description of what an LFG grammar expects in its lexical syntactic descriptions, then a slightly longer one of

what we found in STO, and then a discussion of how we mapped from one to the other.

Lexical syntactic description in LFG

In LFG, lexical syntactic information is expressed by assigning to each lexical stem a part of speech and (optionally) one or more predicate expressions and constraining functional equations. Predicates are expressed as relations on syntactic functions (e.g., subject, object, obliques) rather than on syntactic categories, like NP or PP. The functions of a predicate are listed in a lexical item's definition as its PRED-value. The entry below shows that the verb 'arrangere'/ 'to arrange' combines with a subject and an object, i.e. a well-formed f-structure containing the predicate must contain both a subject and an object.

```
(↑ PRED)='arrangere<(↑ SUBJ)(↑ OBJ)>'
```

Constraints on the realization of each syntactic function can be stated as part of the lexical entry. For example, restrictions such as case, number, or the preposition heading a prepositional object can be stated as constraints on the appropriate syntactic function. If a function must be instantiated as a particular syntactic category (e.g., some verbs only allow adjectival phrases as subject complements), the lexical entry may specify that. The expression, in LFG notation,

```
(↑ PRED)='sige<(↑ SUBJ)(↑ OBJ)>'
CAT((↑ OBJ), {CP})
```

for the verb 'sige'/ 'to say', states that the constituent structure that realizes OBJ must contain a node with the category CP, i.e. the OBJ must be a clause.

In LFG, as in other lexicalist theories, one can also express generalizations across lexical entries. Lexical rewriting rules, for example, allow a single lexical entry to represent a set of related lexical expressions. For example, alternations involving alternative subcategorization frames, e.g. passive, dative-shift and alternations between DP-complements and oblique complements (which are very frequent in Danish: Durst-Andersen and Herslund 1996), are generally expressed through lexical rules. The following lexical rule (in LFG notation) relates a transitive verb to a verb selecting a prepositional object:

```
OBJtoOBL: P<(↑ SUBJ)(↑ OBJ)> → P<(↑ SUBJ)(↑ OBL-på)>
```

as in:

(1) a. Peter skriver en roman
'Peter is writing a novel'

b. Peter skriver på en roman
'Peter is writing on a novel'

We will consider some of the issues involved in establishing and using lexical rules in the context of importing lexical resources below.

Another important technique in LFG grammars implemented in XLE is to structure the lexicon by the use of templates, which allow bundles of lexical annotations to be grouped under a specified name which can then be invoked in the lexical entries. In this way, a template can effectively define a lexical class of items that share a cluster of properties. Templates may invoke other templates, thus giving the effects of an inheritance hierarchy defining relations between structural descriptions (Dalrymple et al. 2004). Hierarchically organized templates are a useful way to express abstractions, and thus to help organize the development of a lexicon, and their use is standard practice when doing lexical development in the XLE environment. However, they may be less useful when importing externally developed lexical material if the external material is not already expressed in (or cannot be mechanically re-structured into) a similar hierarchy. As already mentioned, the existing Danish grammar already contained an extensive template set. One open question was the extent to which this template set could be used during the import.

Syntactic description in the STO-dictionary

Syntactic information is expressed in STO by assigning one or more syntax descriptors to a stem. For example, the descriptor `Dv2xvPa0-dir-loc` describes a verb, with two governing relations, a directional particle and an optional locative prepositional phrase. In the first version of STO, these descriptors were defined by some 50 pages of documentation that explained how descriptor names were composed out of a set of syntax element codes. Later, this documentation was superseded with explicit definitions for most descriptors, as a set of tables within the database.

Initial attempts to relate these syntax descriptors to the existing grammar's template hierarchy were not promising. To begin with, there was a gross mismatch of scale — the grammar had on the order of a hundred templates that could be thought of as defining lexical classes; STO had 1,400 descriptors. Clearly, STO was making much finer (in some sense) discriminations than the grammar was. Further, the STO descriptors were 'flat', i.e. they did not reference each other to form a hierarchy. Instead, each descriptor was a complete, self-contained description. The cross-description abstractions that loomed so large in the existing template hierarchy were at best latent in this description

set. So an attempt to construct a direct mapping from descriptors to templates was abandoned in favor of a close consideration of the descriptors themselves, in the expectation that some abstractions would emerge.

As noted above, the descriptor names had a regular structure, and the accompanying documentation purported to explain how this structure was to be interpreted. An attempt to construct an interpreter for these names was not successful. Not only was there no formal definition (e.g. as a BNF grammar) of the descriptors' syntax, but some of them seemed genuinely open to multiple interpretations.

So instead an attempt was made to process these descriptors 'semi-automatically'. Writing LFG definitions for all 1,400 descriptors would have been extremely time-consuming, but the frequency distribution of these descriptors is, as one would expect, very sharply skewed. Only a couple of hundred of them (out of 1,400) cover the vast majority of stems. So, some tools were developed that allowed for parameterized definitions of subsets of the descriptors. For example, some descriptors varied only in the prepositional marker for some relation. Descriptors like `Dv2xn-efter` and `Dv2xn-fra` correspond to much the same set of LFG equations, except for the particle marker. However, even using pattern matching tools, encoding such a large number of descriptors manually proved to be very laborious. It also missed significant generalizations, since substructures shared by different templates often went undiscovered and were redundantly specified. (An account of this phase of our work can be found in Ørsnes 2005.)

In 2005 a new version of STO was released that provided explicit definitions within the database of most of the descriptors (1,355/1,400). In this newer version of STO, each descriptor has a set of attributes (which vary by part of speech) and one or more 'constructions' each of which defines one or more 'frames', each frame consisting of an ordered set of valency specifications. Using a utility developed to aggregate and format this information in more or less human-readable terms, a descriptor for a verb with a single frame consisting of a subject and two optional prepositional arguments might be shown as:

```
Dv3fP0P0-fra-til e.g. omvende
{PASSIVE=NO, REFLEXIVE=YES, MODAL=NO, PARTICLE=NO,
 AUXILIARY=have}
    SUBJECT NP NOM
    [PREPOSITIONAL_OBJECT PP FRA_NP_NOC_NOC]
    [PREPOSITIONAL_OBJECT PP TIL_NP_NOC_NOC]
```

The first line gives some properties of the verb in an attribute-value

format. The following three lines give the valency information for each of the three elements of the single frame: the verb takes an obligatory subject which is a nominative NP, followed by two optional prepositional phrases (indicated with brackets), headed by the prepositions 'fra'/'from and 'til'/'to'. The complement of each preposition is an NP with no control relationships.

This new representation allowed a more straightforward automatic conversion of the descriptions into a form that the Danish LFG-grammar could read. The descriptions are very rich in detail. They specify arity, optional and obligatory complements, valency bound prepositions and particles, thematic or non-thematic arguments, restrictions on specific lexical items, i.e. reflexive pronouns, syntactic categories for complements (also category of objects of prepositions in prepositional complements) and syntactic function. However, richness does not necessarily imply fit. In some cases, STO makes distinctions that the Danish LFG grammar does not choose to make, so data has to be winnowed down. In other cases, distinctions are made, but in awkward ways, and the data has to be recast. And, of course, sometimes the information one wants just is not there at all.

The easy case is when one has too much detail and one can discard what one does not need. For example, STO routinely provides syntactic category restrictions on the functions selected by a predicate, e.g. subjects specified as NPs. LFG grammars usually do not make such restrictions, so it is an option simply to ignore this data. However, category restrictions can sometimes be idiosyncratic, and since there is no way to tell automatically whether a particular STO restriction is idiosyncratic, it was decided to import all of them. Information on category restrictions occasionally helps and rarely interferes. Information was only discarded from the STO data where it could be anticipated that it might cause problems. For example, in STO, subjects and objects are specified to be nominative or accusative. However, in Danish, case is only marked for personal pronouns, so the grammar does not use case features for general NPs. Rather than make changes to the grammar to include them, it was decided to ignore the case information, and to handle case marked personal pronouns in a separate, static core lexicon for the closed word classes (Ørsnes 2002).

Among the things that mapped well, but not exactly, was the set of syntactic functions used in STO. In the following table, the 15 STO valency functions are mapped into LFG relations. Most of them map directly into familiar LFG relations.

STO function	LFG relation	Comments
SUBJECT	SUBJ	
OBJECT	OBJ	
ACOMP	OBJ2	
ADVERBIAL	Varies	OBL-x mainly
CLAUSCOMP	COMP	
EXTERN_COMP	SUBJ	
FORMAL_COMP	SUBJ	non-Thematic SUBJ
FORMAL_SUBJECT	SUBJ	non-Thematic SUBJ, remaps whole frame
INDIRECT_OBJECT	Varies	
OBJECT_PREDICATE	XCOMP	
PREPOSITIONAL_OBJECT	OBL-prep	
REL_GEN	POSS	
SOM_PP	OBL-som	
SPEC_N	None	
SUBJECT_PREDICATE	XCOMP	

Others need additional context to be mapped correctly. But some, like FORMAL_SUBJECT, require a remapping of the whole frame, i.e. almost all of the LFG relations must be changed. Other problems occur with valency-bound adverbials (which are treated as semantically differentiated OBLiques in the grammar) and the syntactic function of embedded clauses. The latter problem stems from the fact that STO does not make a distinction between clauses as (X)COMPs and clauses that fill core syntactic functions (Dalrymple and Lødrup 2000), and that there appears to be no systematic treatment of extraposed complement clauses occurring with expletive subjects, such as:

(2) a. at han kommer overrasker mig
'that he comes surprises me'
b. det overrasker mig at han kommer
'it surprises me that he comes'

This last problem is an instance of something that is likely to occur in any project to adapt externally developed lexical resources: the existence of different views about what should be treated as a productive valency alternation and what should be considered as (more or less idiosyncratic) alternate frames for individual lexical items. For example, most verbs and adjectives that take clausal subjects in Danish allow these complement clauses to be extraposed with an expletive occurring as the subject. In an LFG grammar, this is an obvious candidate for a lexical rule stating this as a productive alternation, rather than stipulating it in the individual lexical entries. The lexical rule relates

a verb requiring a clausal subject with a verb requiring an expletive subject and an OBJ (treating this construction similarly to the 'there'-construction). But some dictionary builders may be loath to require that their users have such a mechanism, and may list the alternate forms explicitly. In STO, there appears to be no systematic principle; we find both approaches in different places. So, dative alternation is represented through alternate frames for lexical items, while passivization is assumed to be handled by a rule. Raising verbs are marked as such, while raising adjectives are not, and different frames are given for the various realizations.

From an LFG perspective, a lexicon which fails to capture productive cross-lexical item generalizations is not very insightful. One way to take full advantage of the lexical material would be to identify frequent clusters of frames and to extract the abstract rule that related them and thus represent these clusters in a succinct manner. However, as we will see later, it turned out not to be a major performance concern, and so, using the previously stated tactic of not throwing away information when it was not doing any harm, all the STO frames were kept, i.e. all the explicit structures which otherwise could have been generated by means of productive lexical rules. The only lexical rules needed to supplement the imported STO material were the passive, because STO provides no treatment of productive passivization, and a lexical rule for raising verbs.

This issue raises a basic question in the design of multi-purpose lexical resources. STO appears to have taken the stance (except for the passive) that individual items should be associated with a set of descriptors that capture all of its variation. In this approach, little or no attempt is made to state generalizations across the lexical items. There are probably some users who would appreciate such an explicit enumeration of the possibilities. However, for users working with in a syntactic framework which encourages the statement and productive use of lexical generalizations, one wonders why the generalizations could not also be stated, even if an explicit realization of them is also provided. For instance, simply stating that an adjective is a raising adjective would allow those who are using syntax formalisms that permit this to be used productively to do so. Raising may be given different formulations in different syntax formalisms, but there appears to be a fair consensus on the phenomenon as such. A basic question in the design of multi-purpose lexical resources is how much syntactic behavior should be given as explicit description for individual items and how much should just be specified as underlying syntactic properties, leaving the exact formulation of what it means to be, for example, a 'raising

verb' to the syntax used for a particular application.

Mostly, the information required by the existing grammar was found in the STO dictionary. But often it was in unexpected places. For example, STO often relies on the morphology to carry information that one would otherwise expect to find elsewhere. So, the gender of a noun is not given as an inherent property of the noun, but is attached (unchanging) to all its inflected forms. Verbs do not indicate whether they can passivize.[2] Instead, the ability to passivize is read off the morphology — a verb can passivize only if it has a passive form in its morphology. This means that one has to either make a pass over the morphology to determine whether a verb should be subject to the passive lexical rule, or generate passive alternative frames which will never succeed in cases where there are no passive verb forms to match them. All minor problems, but the kind of thing that should be expected when importing a large, external dictionary.

Finally, as one might expect, the really major differences between the STO dictionary and the Danish grammar were found among the closed word classes. Pronouns, articles, auxiliaries and copular verbs are best thought of as part of the syntax: for any syntactic theory, these words appear to have far too much syntactic structure associated with them to expect them to be adequately described in a general purpose dictionary. The solution adopted here was to maintain a separate 'core' lexicon of closed-class items for which we could 'hand-craft' exactly the coding we wanted. This lexicon is given higher priority than anything derived from STO, so its coding takes precedence. Even for things like particles and case marking prepositions, their generic STO dictionary entry was less interesting than the fact that they had been used in various verb frames. So their generic description was ignored in favor of another high-priority static lexicon which was generated from the verb frame data, plus lists of 'directional' and 'locative' items extracted from the accompanying documentation. Less than 100 word forms had to be treated this way, out of the 81,000 covered by the dictionary.

Assembling LFG syntax descriptions from the STO data

Having discussed some of the issues of linguistic congruence between the STO and LFG description frameworks, we now turn to an example of a typical translation. An essential point is that all of the lexical descriptions in the grammar are generated automatically from the STO database, instead of mapping the STO descriptors to hand-crafted tem-

[2]The `PASSIVE=NO` shown in the frame for OMVENDE above does not indicate whether the verb passivizes, but indicates instead whether this frame is an irregular frame for a passive form.

plates providing essentially the same information as contained in the database. Here the STO description is converted into a template providing XLE definitions of the STO syntactic description. The template is referenced by a name identical to the STO syntactic descriptor, and this name in turn is associated with the stem in the stem lexicon. Consider the example from above:

```
Dv3fP0P0-fra-til e.g. omvende
 {PASSIVE=NO, REFLEXIVE=YES, MODAL=NO, PARTICLE=NO,
  AUXILIARY=have}
    SUBJECT NP NOM
    [PREPOSITIONAL_OBJECT PP FRA_NP_NOC_NOC]
    [PREPOSITIONAL_OBJECT PP TIL_NP_NOC_NOC]
```

To assemble the LFG PRED form, the STO valency functions are mapped into LFG governing relations, `SUBJECT` to SUBJ and the two `PREPOSITIONAL_OBJECT`s to OBL-fra and OBL-til, respectively. Note that the prepositional marker serves to individuate the oblique phrases. Since XLE does not support optional PRED elements, these forms are expanded into four separate disjunctive clauses.

```
{   (↑ PRED)='%stem<(↑ SUBJ)>' @(NP (↑ SUBJ))
  | (↑ PRED)='%stem<(↑ SUBJ)(↑ OBL-til)>'
    @(NP (↑ OBL-til)) @(NP (↑ SUBJ))
  | (↑ PRED)='%stem<(↑ SUBJ)(↑ OBL-fra)>'
    @(NP (↑ OBL-fra)) @(NP (↑ SUBJ))
  | (↑ PRED)='%stem<(↑ SUBJ)(↑ OBL-fra)(↑ OBL-til)>'
    @(NP (↑ OBL-fra)) @(NP (↑ OBL-til)) @(NP (↑ SUBJ)) }
```

Each PRED form is followed by one or more template calls which apply category restrictions from the STO frame elements, as discussed above. For example, the oblique phrases of these verbs require NP complements, i.e. the prepositionally marked obliques must have an NP complement in their c-structure. The grammatical relation is passed on to the template as a parameter, saying e.g. that the nodes mapping to the function OBL-fra must contain an NP.

As for the attributes, the property `REFLEXIVE=YES` indicates that the verb takes a non-thematic object in the form of a reflexive pronoun. So, a non-thematic OBJ is added along with the restricting equation (↑ OBJ PRON-TYPE)=c refl for it. The `PARTICLE=NO` requires the lexical entry to block unneeded particles, which is the verb default; had there been a particle (or a particle type) specified, a (↑ PART-FORM) or (↑ PART-TYPE) constraint would have been added. The modal and auxiliary could also have added equations (↑ AUXILIARY)=have and

(↑ MODAL)=−, but since these are true for most verbs, they also are set as defaults and only add non-default equations in the exceptional case.

Finally, the `PASSIVE=NO` indicates that this frame is not itself an irregular frame for a passive form. STO provides explicit definitions for passive frames which are irregular, but regular passive frames are not explicitly represented and must be derived — which is done by a lexical rule here. This frame has a thematic SUBJ, so it might passivize[3], and so the passive lexical rule should be applied.

One form of the Danish passive (the 'impersonal passive') requires the addition of a non-thematic subject to the PRED form. In XLE, lexical rules can delete grammatical functions or map them into other grammatical functions, but they cannot add new ones. So, in order for the passive lexical rule to be able to add a non-thematic subject for impersonal passives, we have to reserve a place for this new function in the lexical form. We therefore add a place-holder (↑ X) into the non-thematic functions of the PRED (outside of the angle brackets). Where a non-thematic subject is needed, the passive lexical rule will map this (↑ X) into a (↑ SUBJ); when no non-thematic subject is needed, this (↑ X) will be deleted, i.e. mapped to NULL. After this addition has been made, an invocation of the passive rule is wrapped around the whole expression.

The final result, after we simplify and factor out common elements (just to make it more readable), is

```
@(PASSIVE
 [{ (↑ PRED)='%stem<(↑ SUBJ)>(↑ X)(↑ OBJ)'
   |(↑ PRED)='%stem<(↑ SUBJ)(↑ OBL-til)>(↑ X)(↑ OBJ)'
    @(NP (↑ OBL-til))
   |(↑ PRED)='%stem<(↑ SUBJ)(↑ OBL-fra)>(↑ X)(↑ OBJ)'
    @(NP (↑ OBL-fra))
   |(↑ PRED)='%stem<(↑ SUBJ)(↑ OBL-fra)(↑ OBL-til)>(↑ X)(↑ OBJ)'
    @(NP (↑ OBL-fra)) @(NP (↑ OBL-til)) }
  @(NP (↑ SUBJ)) (↑ OBJ PRON-TYPE)=c refl ] )
```

If `@Dv3fP0P0-fra-til` is given this as a template definition, then we are now able to define stems that are mapped to this syntax description e.g. OMVENDE/'convert' as follows:

```
OMVENDE V XLE @Dv3fP0P0-fra-til.
```

or, in the case of stems assigned multiple descriptors, something like:

[3]A specific verb that has, possibly among others, this frame may not have any passive inflected forms, in which case it will not passivize, even though this frame implies that it might.

```
OPKVALIFICERE V XLE { @Dv3NPnio0-til | @Dv3refNPnis0-til }.
```

This approach follows the classic XLE style of implementation: the morphology maps surface forms to stems and inflections, and a stem lexicon provides a part of speech and a syntax description for each stem. Here, the syntax descriptions are automatically generated templates, i.e. they are direct encodings of the STO descriptors, rather than the usual carefully constructed quasi-type hierarchy that one would construct in a manual lexicon — but they work the same way. Using this approach, we were able to import and use much of the lexical material from the STO dictionary within the Danish LFG grammar.

9.5.3 Ambiguity

As any grammar writer knows, the downside of increased coverage is increased ambiguity, since new material added to provide more coverage often creates many new, unanticipated, readings for previously unambiguous cases. The scale of this when one goes from a desk lexicon of a few hundred forms to a massive dictionary is simply staggering. Some previously unambiguous sentences suddenly had literally hundreds of readings!

Although our work to cope with this problem is by no means complete, a few principles are already clear. First, a large dictionary will expose problems in the grammar: distinctions that should have been made, but were not because no test cases that required them were ever encountered. Finding and fixing these is a grammar development opportunity that a large-scale lexicon makes possible. Second, some of the word forms that produce new ambiguity will be rare or obsolete uses. A good dictionary will have many of these, and will have them so marked, as STO does. It is very tempting to ignore these markings in one's first attempts to import the material. But in the end, one has to go back and map the dictionary markings into optimality marks (see Butt et al. 1999 and Frank et al. 2001 for details of this mechanism) to filter these readings to the back of the queue. Third, dual to the problem of not capturing distinctions from the dictionary that do matter, is capturing distinctions that do not. For example, STO provides adverbial readings of many adjectives. Many of these have multiple adjectival subcat frames, which are not used in their adverbial reading. But if one (inadvertently?) preserves these, the result is multiple readings for any instance of that adverb. Lastly, there are cases where the dictionary entry does not provide information on which the grammar depends to enable or block some interpretation. For example, the Danish grammar distinguishes sentence adverbials ('fortunately', 'not', etc.) from other kinds of adverbials. This distinction is not made in the current version

of STO, so these words require static definition. In some cases, one can use the lexicon override capability of XLE (Kaplan and Newman 1997) to alter or replace a definition from the source dictionary. However, outside the closed word classes, this kind of manual override is only practical to a limited extent. In particular, although not as damaging as hand editing the imported material, an extensive set of overrides is likely to interact with new releases of the source dictionary and make it difficult to move forward. It is far better, if one can, to find ways to generate the changes mechanically.

9.5.4 Software engineering issues

Although the system we have described so far (the classic XLE stem lexicon plus morphology, hereafter the STEM-approach) does work, it has some undesirable properties. Specifically, it is far too big. Looking at the size of various components of this system, we see

LFG grammar, templates, etc.	44K
Morphology (FST)	499K
Lexicon	2,848K
Lexicon index files	2,164K
Lexicon template files	254K
Other index files	382K

for a total of just over 6MB. That seems excessive. What's worse, the size of the lexicon files and the number of templates (more than 1,400, since there is one for every syntax descriptor) broke some of the Unix systems we first tried to run the system on. While these could be fixed by changing the system configurations, it is awkward in a shared organizational context and gives some indication of the high computational demands that this version places on its environment.

Representing syntax in morphology tags (the TAG approach)

When considering how to reduce the size of the STEM version, the lexicon files were an immediate focus of concern. Although the number of lexical templates can be significantly reduced by looking for shared sub-structure across them and factoring it out (essentially recreating the template abstraction hierarchy that would exist in a classical XLE system), that only addresses a small portion of the size cost. Also, since the templates are generated from an externally provided database that is subject to intermittent update, any such factoring has to be entirely mechanical; if one noticed shared structure and factored it manually, the next database update would destroy that work.

Looking at the lexicon, using nearly 5MB to represent 81,000 stems seems out of proportion to the less than 0.5MB that the morphology

uses to represent nearly ten times as many entries, many of which are more complex than the relatively simple stem lexicon entries. We were naturally drawn to consider whether we could use the morphology tools to construct a more compact representation of this data.

One way to do this is to merge the two sets of information together and have the morphology represent them both. So, rather than have a stem entry like

```
SKINNE V XLE { @Dv1 | @Dv1xv-op }.
```

and a morphological entry for each inflected form like

```
skinner : /SKINNE/ +Active +Indic +Present
```

the stem's part of speech and syntax tag could be added to the morphological entry for each inflected form. Thus, for 'skinner', the present tense of SKINNE ('shine'), the morphology would contain

```
skinner : /SKINNE/ +Verb +Dv1 +Active +Indic +Present
skinner : /SKINNE/ +Verb +Dv1xv-op +Active +Indic +Present
```

and the syntax descriptors (`Dv1` and `Dv1xv-op`) would each have definitions as morphological tags, rather than as templates. That is, rather than

```
Dv1 =  @(PASSIVE [  (↑ PRED)='%stem<(↑ SUBJ)>(↑ X)'
                    @(NP (↑ SUBJ)) ])
```

`Dv1` would be defined as

```
+Dv1 STO_TAG XLE @(PASSIVE [ (↑ PRED)='%xxx<(↑ SUBJ)>(↑ X)'
                             @(NP (↑ SUBJ)) ])
```

i.e., as a morphology tag (the + prefix is an XLE convention for such tags) which would add these syntax constraints to any lexical item built from it.

Both the template and the morphological tag contain PRED forms for which the head is not known, but which will be provided later when the definition is paired with a stem form. In template definitions invoked by a lexical entry, this is done using the special name '%stem'. Morphology tags cannot use this mechanism: at the time they are read, %stem will not be bound to the eventual stem. Instead, they use a dummy name, '%xxx', as a place-holder for the head. The stem is set as the head of a partial PRED form in the lexical entry used for unknown words:

```
-unknown X XLE (↑ PRED FN)=%stem. "sets the head of PRED"
```

Then, when the morphology tag and the incomplete stem are combined using a sub-lexical rule such as

```
V → X_BASE V_SFX_BASE STO_TAG_BASE VTAG_BASE*.
```

which takes a stem, a part of speech tag (here, +Verb), a syntax tag (here, `+Dv1`) and zero or more inflectional tags (+Active, etc.), the head from the stem entry will be merged into the headless PRED form to make a complete frame.

Replacing template definitions with equivalent lexicon entries does not in itself save any space. But, by allowing part of speech and syntax descriptor(s) to be expressed in the morphology, it completely eliminates the need for a stem lexicon! All of the information that was in the stem lexicon can now be in the morphology, with a much smaller lexicon providing definitions for the 1,400 syntax descriptors. The definition of each inflected form will be slightly larger, and stems with multiple syntax tags will have multiple definitions for each inflection, so the morphology input file will grow in size to 58MB, from 40MB. But after conversion to a FST, it takes up only 876KB, compared to the previous 499KB. A side by side comparison shows

	STEM system	TAG system
LFG grammar, templates, etc.	44K	45K
Morphology (FST)	499K	876K
Lexicon	2,848K	267K
Lexicon index files	2,164K	127K
Lexicon template files	254K	–
Other index files	382K	357K
Total size	6.0M	1.6M

Essentially, the definitions from the template file have been moved into the lexicon and the stems that were in the lexicon have shrunk from 5MB of lexicon text file and indices to an incremental 377K of morphology FST.

This is a truly remarkable result. To give credit where credit is due, this is entirely due to the remarkable efficiency of the finite state machinery in both compressing out common substructure and indexing large text sets. One could approximate the same result in space by zip compressing the lexicon file, but then it would be neither indexed nor indexable. The finite state machinery does both.

Factored syntactic information in morphology tags (the FRAGment approach)

Moving the static lexicon into the morphology was such a dramatic improvement that it is tempting to see if one could repeat the success by using the same tools on the lexicon of syntax descriptors. Consider a tag like `Dv1x-efter`. It is defined (ignoring the passive for a moment) as

```
+Dv1x-efter STO_TAG XLE  (↑PRED)='%xxx<(↑SUBJ)>'
                         @(NP (↑SUBJ))
                         (↑PART-FORM)=c efter.
```

and a completely unrelated tag, `Dv2Na0`, is defined as

```
+Dv2Na0 STO_TAG XLE
              { (↑PRED)='%xxx<(↑SUBJ)>'
               |(↑PRED)='%xxx<(↑SUBJ)(↑OBL-measure)>'
                @(NP (↑OBL-measure)) }
              @(NP (↑SUBJ)).
```

Even in these two unrelated tags, there are two common sub-expressions: the SUBJ-only PRED and the NP SUBJ restriction. Looking across the whole set of tags, the same sub-expressions occur over and over. This is hardly surprising. Each sub-expression is a result of a pattern in the STO syntax descriptions. They are composed in many different combinations, but there are not all that many distinct elements. In fact, it turns out that only 497 primitive sub-expressions make up the entire vocabulary of syntax descriptions for the STO dictionary! (Of course, there are far more, over 9,000, instances of these expressions in the syntax tag lexicon).

So, one could replace each syntax descriptor with a series of tags, each of which is defined as one of the sub-expressions found in the descriptor's definition, and let the finite state machinery do what it can to optimize the result. For example, rather than defining 'giver'/ 'give +Present' in terms of a syntax descriptor `DV1x-efter`, whose definition is

```
+Dv1x-efter STO_TAG XLE  (↑PRED)='%xxx<(↑SUBJ)>'
                         @(NP (↑SUBJ))
                         (↑PART-FORM)=c efter.
```

it could be defined as

```
giver: /GIVE/ +Verb +P0 +S0 +S102 +Active +Indic +Present
```

if there were definitions

```
+P0 VTAG XLE (↑PRED)='%xxx<(↑SUBJ)>'.
```

```
+S0 VTAG XLE @(NP (↑SUBJ)).
```

```
+S102 VTAG XLE (↑PART-FORM)=c efter.
```

with no definition for `+Dv1x-efter` itself at all!

However, the above discussion ignores the passive. `Dv1x-efter` verbs (such as GIVE) passivize, and the passive lexical rewrite rule will change the sub-expressions of `Dv1x-efter` in systematic ways that cross

sub-expression boundaries, e.g. if SUBJ is mapped to OBJ, the restrictions stated on SUBJ must be applied to OBJ, etc. XLE provides no way to apply a lexical rewrite rule to a set of sub-expressions derived from the morphology — it has to be done at the time an expression is read from the lexicon.

So, in order to factor the sub-expressions of a syntax descriptor to which a passive lexical rule might apply, the passive rule has to be applied to the descriptor first (using Java code that simulates what XLE would do using the Danish passive lexical rule) and the resulting expanded expressions factored.

However, the Danish passive is unusually productive: a Danish verb can have as many as four different passive frames. Intransitive verbs passivize and acquire a non-thematic subject 'there' (impersonal passive). Transitive and ditransitive verbs passivize by promoting the active object to subject (personal passive), but they also allow the impersonal passive with a non-thematic 'there' and an indefinite object. So, a transitive verb, such as 'arrangere'/ 'arrange', in the sentence

(3) Peter arranger festen
'Peter arranges the.party'

has four different passive variants:

(4) a. festen arrangeres
'the.party is.arranged'
b. festen arrangeres af Peter
'the.party is.arranged by Peter'
c. der arrangeres en fest
'there is.arranged a party'
d. der arrangeres en fest af Peter
'there is.arranged a party by Peter'

The cost of all these alternatives is not easily visible in a classic XLE implementation, since the expansions only take place on a stem by stem basis, as needed. However, in order to factor these expansions for the whole set of descriptors, they all have to be done in statically in advance.

It turns out that the new sub-expressions generated by passive expansion are relatively regular in sub-structure, so they do not increase the size of the sub-expression set all that much — only to 783 from 497. But they do provide up to four additional alternative expression sets for each verb descriptor. It was not clear which effect (the compression that the FST would achieve on the expansions, or the inflation of the number of expansions by the passive) would prevail.

A transducer that maps syntax descriptor names from the TAG morphology into codes for their component sub-expressions can be applied in two ways. It can either be run with the TAG morphology transducer in series, applying to the output of the TAG transducer to map the syntax tags therein into their sub-expression tags, or the two transducers can be statically composed, either by using the Xerox FST tools or by substituting the appropriate set of sub-expression tags for each syntax tag and then remaking it as a single transducer.

One might expect the composed transducer to be larger and faster than the combined size of the two transducers run in series, but in fact it is both smaller and faster. One reason for this is that a static composition of transducers can statically eliminate internally inconsistent expansions. In particular, an expansion that combines an active inflected verb form with a passive syntax frame (and vice versa) can simply be discarded at composition time. Another advantage of static composition is that it allows us to optimize the token ordering in different alternatives to maximize compression. A transducer can only collapse paths efficiently if it finds the same sequence of tokens on many paths. In this case, where we are just using tokens to AND together a logical expression, the order in which they appear is of no interest to us: we just want all of them to be output eventually. But the FST machinery is unaware of this and will preserve distinct paths for each order in which it finds a set of tokens in its input. So, by reordering tokens, e.g. by frequency of occurrence within a set of paths (e.g. all paths for an inflected form), so that low frequency tokens will not disrupt common sequences, the achieved compression can be significantly improved.

Using a statically composed transducer which incorporates these optimizations, we get

	STEM system	TAG system	FRAG system
Grammar, templates, etc.	44K	45K	46K
Morphology (FST)	499K	876K	958K
Lexicon	2,848K	267K	42K
Lexicon index files	2,164K	127K	105K
Lexicon template files	254K	–	–
Other index files	382K	357K	357K
Total size	6.0M	1.6M	1.5M

for a reduction of about 160K, or 10%. In this configuration, the entire morphology, stem set and syntactic descriptions of the STO database (a very large chunk of Danish), plus the Danish LFG grammar, have

been packed into less than 1.5MB!

Considerations other than space for morphology lexicons

Collapsing the lexicon into the morphology (the TAG and FRAG approaches) was principally motivated by the high space costs of large static lexicon files, compared to the very space efficient finite state machinery. However, space is not the only metric of evaluation. In this section, we look briefly at some other implications of this implementation choice.

Performance In the abstract, one might expect the morphology lexicons to be slightly faster than the classic text file ones. In the limit, the large text lexicon files should impose a cost in terms of increased paging traffic, whether they are loaded into memory or paged in on demand. However, this would also be affected by many other aspects of the run-time situation, beginning with the amount of memory that is actually available and the other demands on it. Further, the XLE system makes no commitments as to exactly how different steps are carried out and different implementation choices (e.g., of how sub-lexical rules are applied) could well dominate. Finally, the entire process of morphological analysis and chart population (the steps which a morphology lexicon impacts most directly) are usually only a small proportion of the total run-time, so it is quite possible that any differences will be relatively insignificant.

Empirically, for this one grammar/lexicon pair, the overall parse times for the relatively small test sets we have run so far show TAG to be about 5% faster than STEM, with FRAG a few percentage points faster again. But there is also still a great deal of variability within and across the three implementations, too much to permit any clear claims of run-time advantage.

Generation One thing that is clear is that the morphology based lexicons do not currently support generation. Morphology lexicons depend essentially on dynamically binding the stem that matches the -Unknown morphology entry to the placeholder (the %xxx local name) used instead of a literal stem in the partial predicate(s). The use of %xxx is similar in spirit to that of %stem: a specially interpreted place holder that permits a more compact set of lexical entries. However, unlike %stem, the machinery is not now in place to build the inverse map for generation.

Whether this is a deep problem, or whether machinery could be added to support it is unclear (J. Maxwell and R. Kaplan, personal communications). It would seem reasonable to treat this like other new notations, e.g. restriction, that have been added to XLE to increase

expressive power: wait to see how useful it actually is before investing in implementing its generation inverse. One possibility, which would be anathema for a manually developed lexicon, but which is quite reasonable in the context of mechanically generated lexicons, would be to provide this inverse mapping to XLE as a separate input for the generation process, rather than have the generator build it by re-indexing the lexicon, as is now the practice.

Grammar library servers We earlier made passing reference to a relatively new feature of XLE which allows the grammar writer to specify a code library to be called to provide morphological analysis. This does not solve the problem which originally motivated morphology lexicons (the large static text lexicon files) but in conjunction with a morphology lexicon it does allow the construction of a remote lexical server.

Consider an application that occasionally needs to consult a lexicon of esoteric terminology, perhaps housed at some remote institute of esoteric studies. This application could be configured with a primary (USEFIRST) local morphology for everyday language, and a secondary morphology, implemented as a grammar lib, which would send any word form not recognized by the first to this remote server. If the remote server responds in terms of tags from a common morphology based lexicon, the client application would be able to interpret these, even though the two systems do not have a common stem lexicon (common tag sets, to be sure).

A remote lexicon server cannot currently be implemented in XLE using a conventional text lexicon, because there is no way to provide a text lexicon entry on demand; they must all be presented statically when the grammar is loaded. The combination of morphology based lexicons and XLE's grammar library facility makes it possible for the first time to create remote, layered servers for lexical material. This is of major architectural importance, as we shall argue shortly.

Conceptual issues Having stem forms and their properties embedded in the morphology is a category error from the point of view of classical linguistic theory. And, on the face of it, having part of speech and syntax information for a stem replicated in every one of its surface forms is not very elegant. On the other hand, neither is it very elegant to have a productive inflection pattern replicated in every one of its surface forms, although this is now common practice in a modern finite state morphology.

Further, having the part of speech and the syntax encoding colocated with the inflectional information permits some elegant sim-

plifications. We have already seen that, in FRAG, surface verb forms marked as active do not have passive frames. They do not get proposed from the lexicon entry for the stem and then fail when the attributes mismatch — they are just not there. On a slightly deeper level, consider trans-categorized forms: nominal forms of verbs and the like. STO represents these as inflected forms of their base stem. So, from the verb GIVE/'give' we might have

```
given : /GIVE/ +Noun +Dv3NPind +Common +Singular +Gerund
```

which would make it very straightforward, were one so inclined, to allow the gerund to inherit the verbal subcat frame for GIVE in sentences like

(5) The giving of the book to Mary by John was the last straw.

Conversely, in a morphology based lexicon it is equally easy to suppress this interpretation and turn 'given' into a noun with a default frame (which is what is done here, following ParGram practice: Butt et al. 1999:46). But in a stem lexicon, one has to create a separate lexicon entry for each trans-categorized form in order to assign it the correct part of speech. This adds over 20,000 lexicon entries in the case of STO.

Representing syntactic information within the morphology certainly does blur the classical distinction between syntax and morphology. However, given the practical advantages of doing so, it is not clear that the structure of an implementation must necessarily follow such distinctions.

9.6 Future Directions

Wide coverage lexicons are an inherently good thing for a grammar. Not only is a wide vocabulary required for many applications, but dealing with the issues raised by that diversity is very therapeutic for the grammar and its writers. Traditionally, wide coverage lexicons for XLE grammars have been developed, semi-manually, in tandem with the grammar by the grammar writers themselves. But this is a price that only a few can pay. For many, the only practical way to acquire a wide coverage lexicon will be to mechanically generate it from the resources of an existing public project. Although neither cheap nor easy, this method at least bounds the otherwise prohibitive costs of both acquisition and maintenance.

But doing this changes one's outlook on both the material and one's tools. The tools of XLE evolved as components of a 'linguist's workbench'. Thus, they emphasize individual control, flexibility and low adoption cost. However, if one adopts a million word form dictionary, other issues emerge.

It is not an accident that the dictionary we chose to import was in the form of a database. Databases have many advantages as the data repository for projects involving many people distributed over multiple locations — and this is exactly the kind of project that is required to build very wide coverage lexicons. Databases provide efficient, fine grained, multiple asynchronous access for both developers and consumers of data, transactional integrity, audit trails, data integrity checks, etc. It is hard to imagine launching a project to assemble a big dictionary using anything else as a substrate. The big dictionaries for most languages, and sub-languages for specialized communities, are going to be developed and reside in databases.

In that context, copying a database into a series of indexed text files seems like a strange choice, however cleverly one might do it. Databases already provide that functionality and the data we want is already there. If we are going to take full advantage of that, it would be far better to be able to access these databases from within LFG where they are and as they are — to become clients to them as data sources, taking advantage in real-time of both their ongoing development and their ability to handle many clients and very large data sets.

Of course, the data in any such database is highly unlikely to be in a form we can use directly. As we have seen with STO, the data will need transforming to match it to our grammar and to the formats needed by the XLE interpreter. For example, at the simplest level, databases tend to store invertible transformations (such as morphological ones) in only one form. (This is almost an article of faith among database designers.) STO, for example, stores the stem and an add and remove list. This makes it easy to find the inflections of a root, but harder to find the analysis of an inflected form. To do that efficiently, another index must be added. More complex transformations will also be required, similar to the syntax code mappings we have discussed for STO.

Since the grammar writing community will not in general control these dictionaries, we cannot expect them to incorporate the data transformations that we need. Nor would we wish to see XLE cluttered up with a variety of ad hoc data base accessing code. Instead, we hope to see the development of network resident services that will mediate these translations, either for a single dictionary or possibly for several. We have already seen one way (using grammar libraries and morphology based lexicons) that such a service could be implemented.

The great advantage of accessing lexical data via a service is that it detaches the grammar from an overly tight binding to one particular representation of it, and so opens up the possibility of accessing multiple sources within the same framework. This is important because, over

time, even these comprehensive national lexical monuments are bound to split into families of dictionaries (hopefully sharing some common structure), each owned by their relevant community. In the long term, it is hard to see the national vocabulary of, say, micro-biology being developed by anyone except the micro-biologists. In time, they will come to own it. When that happens, it would be nice if we had an architecture so that the XLE applications that want to can access it without it being copied in by hand.

In fact, we can see an evolution here from the idea of a single, static lexicon to a layered cascade of lexicons: a 'core' lexicon of closed word classes (auxiliaries, pronouns, prepositions and the like) which are really part of the basic language model, and are very tightly bound to the LFG interpreter. Then a series of dictionaries being consulted, over networks, in progressive remove; much as a person switches from their working vocabulary to a desk dictionary and on to specialized reference works for technical terms, as they need them.

9.7 Conclusions

We have reported on some experiments that we have carried out to use a large database resident dictionary of Danish as the central lexicon for an XLE based Danish grammar. We have had a reasonable amount of success in doing so, and have radically expanded the vocabulary of the Danish grammar. We have also encountered areas where the two systems' views are not easily reconciled, and the process of exploring the differences and working with and around them will continue, as a linguistic endeavor, for some time. We also discovered some ways in which the existing text file based XLE tools are not well adapted to such large lexicons and developed some ways to circumvent these problems using the remarkably efficient finite state tools of XLE. In the long term, we feel that building mechanisms for engaging these large lexical databases more directly would be beneficial — both for the LFG community and also, perhaps, even for the lexical databases themselves.

Acknowledgments

This work would not have been possible without the technical advice, support, encouragement, and friendship of Ron Kaplan. The key technical insight that enables the morphology to build PRED forms that contain the stem, and thus express its syntax, is directly due to Ron, and the dramatic performance that results is a testament to the power of the finite state technology that he pioneered.

Beyond that, Ron, Tracy Holloway King and John Maxwell, all of PARC, have all helped us with advice, solutions to technical problems and answers for many questions over the years. We are very grateful to have such friends.

We would like to thank Anna Braasch, Costanza Navaretta, Lene Offersgaard and Sussi Olsen for help with several clarifications (technical as well as conceptual) of the STO database, and also our reviewers and editors for their helpful comments and suggestions.

Finally, the first author would like to thank Copenhagen Business School, and in particular its Department of Computational Linguistics, for hosting him in Copenhagen during a sabbatical visit in Spring of 2004, when this work was begun.

References

Braasch, Anna and Sussi Olsen. 2004. STO: A Danish Lexicon Resource - Ready for Applications. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*, vol. IV, pages 1079–1082. Lisbon, Portugal.

Butt, Miriam, Tracy H. King, Maria-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. Stanford, CA: CSLI Publications.

Butt, Miriam, Helge Dyvik, Tracy H. King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), Workshop on Grammar Engineering and Evaluation*, pages 1–7. Taipei, ROC.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, John T. Maxwell, III, and Paula S. Newman. 2006. XLE Documentation. Palo Alto Research Center.

Dalrymple, Mary and Helge Lødrup. 2000. The grammatical functions of complement clauses. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2000 (LFG'00)*, pages 104–121. Berkeley, CA: CSLI Online Publications.

Dalrymple, Mary, Ronald M. Kaplan, and Tracy H. King. 2004. Linguistic generalizations over descriptions. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2004 (LFG'04)*, pages 199–208. Christchurch, New Zealand: CSLI Online Publications.

Durst-Andersen, Per and Michael Herslund. 1996. Prepositional objects in Danish. In L. Heltoft and H. Haberland, eds., *Proceedings of the 13th Scandinavian Conference of Linguistics (SCL 13)*, pages 93–108. Roskilde, Denmark.

Frank, Anette, Tracy H. King, Jonas Kuhn, and John T. Maxwell, III. 2001. Optimality theory style constraint ranking in large-scale LFG grammars.

In P. Sells, ed., *Formal and Empirical Issues in Optimality Theoretic Syntax*, pages 367–397. Stanford, CA: CSLI Publications.

Jørgensen, Stig W., Carsten Hansen, Jette Drost, Dorte Haltrup, Anna Braasch, and Sussi Olsen. 2003. Domain specific corpus building and lemma selection in a computational lexicon. In *Corpus Linguistics 2003 Proceedings*, pages 374–383. Lancaster, United Kingdom.

Kaplan, Ronald M. and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, ed., *The Mental Representation of Grammatical Relations*, pages 173–281. Cambridge, MA: The MIT Press.

Kaplan, Ronald M. and Paula S. Newman. 1997. Lexical resource reconciliation in the Xerox Linguistic Environment. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97) and the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL'97), Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, pages 54–61. Madrid, Spain.

Kaplan, Ronald M. and Tracy H. King. 2003. Low-level mark-up and large-scale LFG grammar processing. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2003 (LFG'03)*, pages 238–249. Albany, NY: CSLI Online Publications.

Kaplan, Ronald, John T. Maxwell, III, Tracy H. King, and Richard Crouch. 2004. Integrating finite-state technology with deep LFG grammars. In *Proceedings of the European Summer School in Logic Language and Information 2004 (ESSLII'04), Workshop on Combining Shallow and Deep Processing for NLP*, pages 11–20. Nancy, France.

Kim, Roger, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, Hiroshi Masuichi, and Tomoko Ohkuma. 2003. Multilingual grammar development via grammar porting. In *Proceedings of the European Summer School in Logic Language and Information 2003 (ESSLII'03), Workshop on Ideas and Strategies for Multilingual Grammar Development*, pages 49–56. Wien, Austria.

Lezius, Wolfgang, Stefanie Dipper, and Arne Fitschen. 2000. IMSLex – representing morphological and syntactical information in a relational database. In S. Evert, E. Lehmann, and C. Rohrer, eds., *Proceedings of the 9th European Association of Lexicographers (EURALEX) International Congress*, pages 133–139. Stuttgart, Germany.

Maxwell, John T., III and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics* 19:571–589.

Rosén, Victoria. 2001. Fra Bokmålsordboka via NorKompLeks til et LFG-leksikon for norsk. In I. Moen, H. G. Simonsen, A. Torp, and K. I. Vannebo, eds., *MONS 9: Utvalgte artikler fra Det Niende Møtet om norsk språk i Oslo*. Oslo, Norway: Novus.

SprogTeknologisk Ordbase for Dansk. 2004. *SprogTeknologisk Ordbase for Dansk. Lingvistiske Specifikationer*. Center for Sprogteknologi. Manual.

Wedekind, Jürgen and Bjarne Ørsnes. 2003. Restriction and verbal complexes in LFG - a case study for Danish. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2003 (LFG'03)*, pages 424–450. Albany, NY: CSLI Online Publications.

Wedekind, Jürgen and Bjarne Ørsnes. 2004. An LFG account of the Danish verbal complex and its topicalization. *Acta Linguistica Hafniensia* 36:35–64.

Ørsnes, Bjarne. 2002. Case marking and subject extraction in Danish. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2002 (LFG'02)*, pages 333–353. Athens, Greece: CSLI Online Publications.

Ørsnes, Bjarne and Jürgen Wedekind. 2003. Parallelle datamatiske grammatikker for Norsk og Dansk. In H. Holmboe, ed., *Nordisk Sprogteknologi 2002*, pages 113–130. Copenhagen, Denmark: Museum Tusculanums Forlag.

Ørsnes, Bjarne and Jürgen Wedekind. 2004. Parallelle datamatiske grammatikker for Norsk og Dansk: Analyse og disambiguering af modalverber. In H. Holmboe, ed., *Nordisk Sprogteknologi 2003*, pages 165–182. Copenhagen, Denmark: Museum Tusculanums Forlag.

Ørsnes, Bjarne. 2005. Automatisk opbygning af et LFG-baseret datamatisk leksikon for dansk. In H. Holmboe, ed., *Nordisk Sprogteknologi 2004*, pages 211–236. Copenhagen, Denmark: Museum Tusculanums Forlag.

Ørsnes, Bjarne and Jürgen Wedekind. 2006. Datamatiske grammatikker og lingvistisk teori - med en leksikalsk-funktionel analyse af den danske komplekse passiv som illustration. In A. Braasch et al., ed., *Sprogteknologi i dansk perspektiv*. Copenhagen, Denmark: Reitzel. To appear.

Part III

Constraints on Syntax and Morphology

10

Agentive Nominalizations in Gĩkũyũ and the Theory of Mixed Categories

Joan Bresnan and John Mugane

10.1 Introduction

Mixed categories[1] are constructions which combine the syntactic and morphological properties of two distinct categories, such as noun and verb, while being headed by a single word. These constructions challenge two basic principles of syntax — endocentricity and lexical integrity:

(1) a. **Endocentricity:** every phrasal projection has a unique lexical head which determines its categorial properties.

[1]In order to make examples morphologically transparent, we have transcribed certain vowels separately between morpheme boundaries even though some of these vowels are elided or coalesced in speech. Our segmental transcriptions follow Mugane (1996, 1997). The seven vowels transcribed *i, ĩ, e, a, o, ũ, u* have the approximate respective values [i, e, ɛ, a, ɔ, o, u], each of which can be short or long. Long vowels are indicated by doubling. *m, n, ny, ng'* represent labial, coronal, palatal, and velar nasals, respectively. *mb, nd, nj, ng* are the corresponding prenasalized stops, and *b, th, c, g* the corresponding fricatives. The glosses use Arabic numerals to represent noun classes and small Roman numerals to represent values in the category of person. Abbreviations are NOM/nominalizer, DEM/demonstrative, ASSOC/associative, APPLIC/applicative, RECIP/reciprocal, NEG/negation, REL/relativizer, INT/intensifier, FUT/future, PERF/perfect, SUBJ/subject, OBJ/object, SG/singular, REFL/reflexive, INTJ/interjection, PERF.PART/perfect participle, HAB.PART/habitual participle, FV/final vowel.

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

b. **Lexical integrity:** every lexical head is a morphologically complete word formed out of different elements and by different principles from syntactic phrases.

The paradigm examples of mixed categories have often been taken to be verbal forms such as gerunds and infinitives. One example is the English construction in (2).

(2) a politician's reportedly not telling the public the truth

In (2) the genitive NP, otherwise exclusively a constituent of noun phrases in English, cooccurs with constituents otherwise exclusively found in verb phrases — the double NP complements.[2] The standard sentential negation *not* and the preverbal adverb in this example are also typical VP constituents. Thus the construction mixes together properties found exclusively in verb-headed structures like (3) with those found in purely nominal structures like (4):

(3) He is reportedly not telling the public the truth.

(4) a politician's reported (non-)telling(s) of the truth to the public[3]

Another example is the Italian *infinito sostantivato* (what Zucchi 1993 calls "VP-infinitival NPs"). In (5–6) determiners, possessives, and qualifying adjectives — NP/DP constituents — appear before the infinitive, while direct object NPs and adverbs — VP constituents — appear after the infinitive (Zucchi 1993):

(5) il suo continuo momorare parole dolci
the his/her continual whisper.INF words sweet
'his continual whispering of soft words' (Zucchi 1993:239)

(6) il suo scribere quella lettera improvvisamente
the his/her write.INF that letter suddenly
'his suddenly writing that letter' (Zucchi 1993:54)

The purely nominal infinitives (what Zucchi 1993 calls "N-infinitival NPs") take adjectives and not adverbs as modifiers; compare the nominalization in (7):

[2]Double NP complements also occur in ellipsis constructions having VP antecedents, such as the sequence of NPs [*their cats*] [*cabbage*] following *than* in *The survey revealed that more of the children gave their dogs spaghetti than their cats cabbage.*

[3]The nominal gerund differs morphologically from both the verbal gerund and the participle in allowing negative prefixation by *non-* and plural suffixation, which are impossible with participles and verbal gerunds. In addition its syntactic complement type (the *of* phrase) is also characteristic of relational nouns underived from verbs; the latter take PPs, like the *of* phrase in *her picture of Mary, *her picture Mary.*

(7) la cessazione improvvisa/*improvvisamente delle ostilitá
the cessation sudden/suddenly of the hostilities
'the sudden cessation of the hostilities' (Zucchi 1993:223)

The *infinito sostantivato* shows its mixed properties in being able to take both adjectives and adverbs at the same time:

(8) il suo continuo eseguire la canzone impeccabilimente
the his/her continual perform.INF the song impeccably
(Zucchi 1993:55)
'his continually performing the song impeccably'

However, the mixed verbal constructions that we see in English and Italian have a special property which is not true of mixed category constructions in Gĩkũyũ: their meanings belong to the same semantic type as those of clauses headed by verbs. Zucchi (1993:251ff) argues that the mixed Italian constructions (such as (5), (6), and (8)) denote proposition-like entities, while N-infinitival NPs like (7) denote events. He argues that a similar difference appears with English verbal and nominal gerundive constructions (*his performing the song* vs. *his performing of the song*) (Zucchi 1993:67–71).

Because the meanings of infinitives and gerunds are of the same semantic types as those of verbs, these kinds of mixed categories can be regarded simply as inflectional subtypes of the base lexical category. The problem of mixed categories then appears to be primarily a syntactic one of correlating the morphology of the head with the categorially mixed syntax. However, Gĩkũyũ shows us that mixed categories can be derived not only by inflectional morphology, but by morphology which fundamentally changes the category of lexical meaning type — that is, by what is considered to be classically lexical derivational morphology. Gĩkũyũ has deverbal agentive nominalizations analogous to the English example in (9), except that they are fully grammatical. An example is given in (10).[4]

(9) *the driver a rusty truck to Arizona reluctantly

(10) ũyũ mũ-thĩĩnj-i mbũri ũũru
1.DEM 1-slaughter-NOM 10.goat badly
'this bad goat slaughterer'; lit.: 'this slaughterer goats badly'

The important properties of these Gĩkũyũ constructions have been developed in Mugane (1996, 2003).[5] In what follows we review both

[4]Example (9) is based on Ackema and Neeleman (2001), and its implications are discussed further in Sections 10.5 and 10.6 below.

[5]Mugane (2003) also provides tonal transcriptions, which unfortunately could not be included in the present study.

the morphosyntax of such agentive nominalizations in Gĩkũyũ and the range of available analyses, in order to develop an explanation within the LFG framework which adheres to principles (1a,b).

10.2 Gĩkũyũ Agentive Nominalizations

Although prototypical referents of these nominalizations are agents (e.g., *mũ-in-i*, 1-sing-NOM, 'singer'), they may also have other roles, such as instrument *gĩ-thĩĩnj-i*, 7-slaughter-NOM, 'something to slaughter with', *i-thĩĩnj-i*, 8-slaughter-NOM, 'things to slaughter with' (plural).[6] We use the term 'agentive' nominalization with the understanding that agents are only the typical and not the exclusive referents of these nominals.

10.2.1 Morphology

Gĩkũyũ agentive nominalizations are illustrated in (11):

(11) a. mũ-in-i
1-sing-NOM
'singer'

b. mũ-thĩĩnj-i
1-slaughter-NOM
'slaughterer'

c. a-ndĩk-i
2-write-NOM
'writers'

Note that these nominalizations bear noun class markers, which are the prefixes glossed by Arabic numerals in (11). These clearly mark (11a-c) as belonging to the inflectional class of nouns in Bantu. Other categories have concordial class marking prefixes, but they differ in shape in some classes (Mugane 1997:26–27). For example, the class 8 prefix *ci-,i-* does not appear on adjectives and adnominal verbs, which instead mark class 8 by prenasalizing the initial consonant of the base. The subject prefixes of finite verbs differ from nouns in classes 1, 3, 4, 5, 9, and 10. In (12) the same stem is shown with four different noun class markers:

(12) a. mũ-thuur-i
1-select-NOM
'selector (human)'

[6] To account for this fact in English, Rappaport Hovav and Levin (1992) use the term '*-er* nominal', but it obviously lacks crosslinguistic identifiability.

b. (gĩ)-thuur-i
7-select-NOM
'selector (augmentative/derogatory)'
c. (ma)-thuur-i
6-select-NOM
'selectors (collective)'
d. (tũ)-thuur-i
13-select-NOM
'selectors (diminutive/ameliorative)'

The noun class markers indicate the class with which all nominal modifiers and predicates must agree. For example, if a noun phrase headed by an 'augmentative/derogatory' class noun (12b) appears as the subject of a verb, the verb must show class 7 agreement with its subject marking prefix; a quantifier phrase modifying the subject nominal must also show class 7 agreement.

The noun class marker is prefixed to a verb stem to which a nominalizing suffix has been attached. The agentive suffix *-i* in these nominalizations is one of a series of suffixal nominalizers (Mugane 1997:88–89). Several are illustrated below with the verb stem for 'slaughter'. (Many other deverbal nominalizing suffixes occur; for example, distinct suffixes exist for deverbal occasions, abstract concepts, and states.)

(13) Nominalizations of the verb *thĩĩnja* 'slaughter':

Nominalization	Gloss	Type
mũ-thĩĩnj-i	'slaughterer' (class 1)	agentive
mũ-thĩĩnj-ĩre	'manner of slaughter' (class 3)	manner
gĩ-thĩĩnj-ĩro	'slaughter location' (class 7)	location

The forms of the nominalizing suffixes are generally not possible as verb desinences in Gĩkũyũ, further evidence that the derived forms do not belong to the inflectional class of verbs.[7] The agentive suffix requires that the verb stem have an agentive role semantically, accounting for the fact that nonagentive verbs 'be' and 'have' are semantically incompatible with agentive nominalization:

(14) a. *mũ-korw-i
1-be-NOM
'a "be-er," one who is'

[7]However, *-i* can be used to turn a verb into a habitual participle which is used with the adjective class prefixes as an adnominal modifier, as discussed below in section 10.3.

b. *mũ-rĩ-i
1-have-NOM
'a "hav-er," one who has'

The verbal base of the nominalization may undergo various stem derivation processes exclusive to verbs, including the reduplication, applicativization, and reciprocalization illustrated in (15a–c):[8]

(15) a. mũ-rũgarũg-i
1-jump.jump-NOM
'one who jumps repeatedly halfheartedly'
b. a-ndĩk-ir-i
2-write-APPLIC-NOM
'those who write for/to (others)'
c. a-ndĩk-án-i
2-write-RECIP-NOM
'those who write each other'

In sum, these agentive nominalizations consist of a verbal base which is nominalized by an agentive suffix and prefixed by a noun class marker. The base undergoes a subset of verbal morphological processes, including verbal extension by suffixation, reduplication, and reflexive prefixing. The meaning, inflectional class, lexical category, and morphological type of the nominalization tell us that it is a deverbal noun.

10.2.2 NP constructions

Agentive nominalizations may head purely nominal syntactic phrases, as in (16):

[8]The verbal base of the nominalization cannot be inflected for negation, tense, or aspect (although unnominalized verbs may be):

(i) *mũ-ti-on-i
1-NEG-see-NOM
'one who does not see'
(ii) *mũ-ka-on-i
1-FUT-see-NOM
'one who will see (habitual)'

This restriction cannot be attributed to a ban on prefixal verbal morphology on the verbal base, however, because an aspectual suffix of verb stems is also excluded:

(iii) *mũ-on-ag-i
1-see-HAB-NOM
'one who sees (habitual)'

and because a reflexive prefix may appear:

(iv) mũ-ĩ-on-i
1-10.REFL-see-NOM
'one who sees himself/herself (a braggart)'

(16) a. mũ-in-i w-a i-tũũra
1-sing-NOM 1-ASSOC 5-settlement
'singer of the settlement'
b. mũ-in-i w-a nyĩmbo
1-sing-NOM 1-ASSOC 10.song
'singer of songs'

While the verb 'sing' takes a direct NP object, the agentive nominalization 'singer' in (16b) takes an associative phrase expressing the semantic role of the object. An associative phrase is an adnominal phrase headed by a particle *-a* 'of' which bears a concordial prefix agreeing in noun class with the nominal it modifies. The same associative marker marks nominal complements and nominal adjuncts, as illustrated in (16a,b). Complement associative phrases (16b) appear in exactly the same positions as the adjunct phrases (16a) relative to other adnominal constituents (Mugane 1996). For example, in the unmarked order of NP-constituents shown abstractly in (19), both are separated from the head by the demonstrative:

(17) a. mũ-in-i ũyũ w-a i-tũũra
1-sing-NOM 1.DEM 1-ASSOC 5-settlement
'this singer of the settlement'
b. mũ-in-i ũyũ w-a nyĩmbo
1-sing-NOM 1.DEM 1-ASSOC 10.song
'this singer of songs'

Hence the associative phrase interpreted as a complement to the head nominal is probably an argument adjunct — an optional adjunct that is interpreted with respect to a specific argument role of the head nominal, much as the passive agentive phrase has been analyzed (e.g., by Alsina 1996).

Demonstratives, possessive pronouns, adjectives, and relative clauses may also modify an NP headed by an agentive nominalization. The head N is NP-initial, preceding all of these constituents except for the determiner, which may optionally appear in initial position when focused:

(18) a. mũ-in-i ũyũ, ũyũ mũ-in-i
1-sing-NOM 1.DEM 1.DEM 1-sing-NOM
'this singer'
b. mũ-in-i w-itũ
1-sing-NOM 1-our
'our singer'

c. a-in-i a-nene
2-sing-NOM 2-big
'big singers'

d. a-in-i a-rĩa ũ-ĩ
2-sing-NOM 2-REL 2.SG.SUBJ-know
'the singers whom you know'

The pragmatically unmarked order of nominal dependents is shown in (19) (Mugane 1996:88).[9]

(19) `N < Dem < PossPron < QP < AP < AssocP`

Other word orders are possible, but are marked with pauses. These word order generalizations are exactly the same in NP constructions headed by underived nouns. Compare the unmarked orders in (20a,b):

(20) a. [mũ-end-i]$_N$ uyu w-a a-ndũ
1-love-NOM 1.DEM 1-ASSOC 2-person
'this lover of people'

b. [nyũngũ]$_N$ ĩno y-a u-cũrũ
9.pot 9.DEM 9-ASSOC 14-porridge
'this pot of porridge'

The same nominal modifiers may occur when the nominalized verb bears a reflexive prefix or a verbal extension, as exemplified by (21) and (22), respectively:[10]

(21) mũ-ĩ-rut-i ũyũ
1-10.REFL-see-NOM 1.DEM
'this one who sees himself/herself'

(22) mũ-hũr-an-i ũyũ
1-fight-RECIP-NOM 1.DEM
'this one who fights with others'

Just as the internal structure of these agentive phrases is typical of NPs, so is their external distribution. They may be subjects or objects of verbs or prepositional objects, they may induce noun class concord

[9] All of these nominal constituents, including the head N, are optional. Omission of the head results in a null anaphoric interpretation.

[10] As noted by Mugane (1996:104–5), a pronominal object marker can be prefixed to the verb stem, but cannot cooccur with nominal modifiers. This could be explained if pronominal object prefixation turns out to be a property of the habitual participle, an adnominal form of the verb which resembles the agentive nominalization (Section 10.3). However, we have not yet been able to find clear evidence for or against this hypothesis because the class 8 concordial morphological distinction between nouns and participles appears to be neutralized when the participle bears an object marker, as discussed below.

with their matrix verbs, and they may be clefted and relativized — all properties of NPs, but not of nonnominal categories such as VPs or CPs in Gĩkũyũ. (See Mugane 1996 for detailed exemplification.)

10.2.3 Mixed NP/VP constructions

While the preceding examples of agentive nominalization constructions reveal an NP headed by a deverbal noun, the same agentive nominalizations also appear in mixed category constructions, as shown in (23):

(23) a. [mũ-thĩĩnj-i]$_{N}$ [mbũri]$_{NP}$ [wega]$_{ADV}$ w-a Nairobi
1-slaughter-NOM 10.goat 1.well 1-ASSOC N.
'a good goat slaughterer from Nairobi'
Lit.:'(a) slaughterer goats well from Nairobi'

b. [mũ-in-ĩr-i]$_{N}$ [a-ndũ]$_{NP}$ [nyĩmbo]$_{NP}$ ũyũ
1-sing-APPLIC-NOM 2-person 10.song 1.DEM
'this singer of songs for people'
Lit.: 'this singer people songs'

c. [mũ-in-i]$_{N}$ [wega]$_{ADV}$ ũ-rĩa mũ-nene
1-sing-NOM well 1-REL 1-big
'the one who sings well who is big'
Lit.: '(the) singer well who is big'

The Gĩkũyũ constructions in (23a-c) consist of the head, which is an agentive nominalization, immediately followed by a sequence of verbal dependents — a direct object and adverb in (23a), two NP objects in (23b), and an adverb in (23c) — followed in turn by nominal dependents — the associative ('of' phrase) adnominal modifier in (23a), the demonstrative in (23b), and a relative clause in (23c). Elsewhere in Gĩkũyũ, double NP complements occur exclusively in VPs and certain adverbial adjuncts are not found as the immediate constituents of NPs or DPs.

The semantic types of adverbial modifiers include manner ('skillfully', 'cleverly', 'quickly', 'slowly', 'carefully'), duration ('for a long time'), temporal ('early'), evaluation ('badly', 'well'), and intensity ('very', 'totally').[11] They can be expressed by PPs ('with knowledge'), verbal phrases ('caring for them'), a small closed class of adverbs, and interjective particles. Examples of an intensifier and emphatic interjection are given in (24):

[11]Thus they are not semantically limited to the types that have fallen in the domain of verbal case assignment in case-marking languages (cf. Wechsler and Lee 1996, Przepiórkowski 1999, Lee 1999a,b).

(24) a. mũ-kir-i mũno
1-quiet-NOM very
'one who is very silent'
b. mũ-kir-i ki
1-quiet-NOM INTJ
'one who is totally silent'

Not only do constituents of the VP occur in these mixed categories, they must occur in exactly the same order as in sentence VPs. The order of VP constituents in a simple sentence is shown in (25) (Mugane 1996:142):

(25) `Verb < Indirect Object < Direct Object < Target Locative < Manner Adverbial < Setting Locative < Temporal Adverb`

For example, an adverb must follow any NP objects in the mixed category construction:

(26) a. $[$mũ-in-ĩr-i$]_{N}$ $[$a-ndũ$]_{NP}$ $[$nyĩmbo$]_{NP}$ $[$wega$]_{ADV}$
1-sing-APPLIC-NOM 2-person 10.song well
'one who sings songs for people well'
b. *$[$mũ-in-ĩr-i$]_{N}$ $[$a-ndũ$]_{NP}$ $[$wega$]_{ADV}$ $[$nyĩmbo$]_{NP}$
1-sing-APPLIC-NOM 2-person well 10.song
c. *$[$mũ-in-ĩr-i$]_{N}$ $[$wega$]_{ADV}$ $[$a-ndũ$]_{NP}$ $[$nyĩmbo$]_{NP}$
1-sing-APPLIC-NOM well 2-person 10.song

The same is true in the corresponding sentence VPs:

(27) a. nĩ-a-a-in-ĩr-a
FOC-III.SG.SUBJ-PERF-sing-APPLIC-FV
$[$a-ndũ$]_{NP}$ $[$nyĩmbo$]_{NP}$ $[$wega$]_{ADV}$
2-person 10.song well
'she/he has sung songs for people well'
b. *nĩ-a-a-in-ĩr-a $[$a-ndũ$]_{NP}$ $[$wega$]_{ADV}$ $[$nyĩmbo$]_{NP}$
...sing... 2-person well 10.song
c. *nĩ-a-a-in-ĩr-a $[$wega$]_{ADV}$ $[$a-ndũ$]_{NP}$ $[$nyĩmbo$]_{NP}$
...sing... well 2-person 10.song

Similarly, the applied beneficiary object must precede the theme object in the mixed category construction (28a), and the same holds in the VP of a sentence (28b):

(28) a. *$[$mũ-in-ĩr-i$]_{N}$ $[$nyĩmbo$]_{NP}$ $[$a-ndũ$]_{NP}$ $[$wega$]_{ADV}$
1-sing-APPLIC-NOM 10.song 2-person well
'one who sings songs for people well'
Lit.: 'one who sings songs people well'

b. *[nĩ-a-a-in-ĩr-a]$_V$
foc-iii.sg.s-perf-sing-applic-fv
[nyĩmbo]$_{NP}$ [a-ndũ]$_{NP}$ [wega]$_{ADV}$
10.song 2-person well
'one who sings songs for people well'
Lit.: 'she/he has sung songs people well'

Moreover, an object NP within the agentive nominalization phrase is in complementary distribution with the reflexive prefix (29), just as it is in sentence VPs (30):

(29) a. mũ-rut-i ci-ana Gĩ-thweri
1-teach-nom 8-child 7-Swahili
'one who teaches children Swahili'

b. mũ-ĩ-rut-i Gĩ-thweri
1-refl-teach-nom 7-Swahili
'one who teaches himself/herself Swahili'

c. *mũ-ĩ-rut-i ci-ana Gĩ-thweri
1-refl-teach-nom 8-child 7-Swahili
Lit.: 'one who teaches himself/herself children Swahili'

(30) a. nĩ-a-a-rut-a ci-ana Gĩ-thweri
foc-iii.sg.subj-perf-teach-fv 8-child 7-Swahili
'She/he has taught children Swahili'

b. nĩ-a-a-ĩ-rut-a Gĩ-thweri
foc-iii.sg.subj-perf-refl-teach-fv 7-Swahili
'She/he has taught herself/himself Swahili'

c. *nĩ-a-a-ĩ-rut-a ci-ana Gĩ-thweri
foc-iii.sg.subj-perf-refl-teach-fv 8-child 7-Swahili
Lit.: 'She/he has taught herself/himself children Swahili.'

In general, then, all and only the post-head immediate constituents of VPs are possible post-head constituents of the mixed agentive nominalization phrase, and all and only the possible orderings of these VP constituents are possible orderings of the same constituents in the mixed agentive phrase.

Let us now turn from the VP-like portion of the structure to the NP-like portion. Note first that the full set of nominal modifiers is possible in the presence of the VP-style constituents:

(31) a. mũ-thĩĩnj-i mbũri ũyũ
1-slaughter-nom 10.goat 1.dem
'this goat slaughterer'

b. mũ-thĩĩnj-i mbũri w-itũ
1-slaughter-NOM 10.goat 1-our
'our goat slaughterer'

c. a-thĩĩnj-i mbũri othe
2-slaughter-NOM 10.goat 2.all
'all goat slaughterers'

d. a-thĩĩnj-i mbũri a-nene
2-slaughter-NOM 10.goat 2-big
'big goat slaughterers'

e. mũ-thĩĩnj-i mbũri w-a gĩ-cagi
1-slaughter-NOM 10.goat 1-ASSOC 7-village
'goat slaughterer of the village'

These nominal elements occur in the normal unmarked order (19) as well as the marked orders. For example, it is unmarked for a quantifier (QP) to precede an adjective phrase (AP) in a pure NP construction, and the same is true in the mixed NP construction:

(32) a. a-thĩĩnj-i mbũri othe a-nene
2-slaughter-NOM 10.goat 2.all 2-big
'all big goat slaughterers' (unmarked)

b. a-thĩĩnj-i mbũri a-nene, othe
2-slaughter-NOM 10.goat 2-big 2.all
'all big goat slaughterers' (marked)

Likewise, the demonstrative precedes other nominal modifiers in the unmarked order.[12]

A further point of interest is that all complements selected by the head must be of the same type: either verbal or nominal. The split complements in (33a) illustrate this; the beneficiary argument is a verbal complement type (applied object NP), while the patient argument is a nominal complement type (associative phrase). Because applied NPs cannot be expressed by associative phrases (Mugane 1997:106), (33b) is also bad. Consequently, ditransitive nominalization is only possible with verbal-type (direct NP) complements, as in (33c):

[12]One restriction, however, is that it is unacceptable to have the determiner follow an adverb (i); the NP-initial order of the determiner is preferred in this case (ii):

(i) ??mũ-thĩĩnj-i mbũri ũũru ũyũ
1-slaughter-NOM 10.goat badly 1.DEM
'this bad goat slaughterer'

(ii) ũyũ mũ-thĩĩnj-i mbũri ũũru
1.DEM 1-slaughter-NOM 10.goat badly
'this bad goat slaughterer'

(33) a. *mũ-thĩĩnj-ĩr-i a-ndũ w-a mbũri
1-slaughter-APPLIC-NOM 2-person 1-ASSOC 10.goat
'one who slaughters goats for people'

b. *mũ-thĩĩnj-ĩr-i w-a a-ndũ w-a mbũri
1-slaughter-APPLIC-NOM 1-ASSOC 2-person 1-ASSOC 10.goat
'one who slaughters goats for people'

c. mũ-thĩĩnj-ĩr-i a-ndũ mbũri
1-slaughter-APPLIC-NOM 2-person 10.goat
'one who slaughters goats for people'

This homogeneity of selected complement types is a kind of 'lexical coherence' (Malouf 1998, 2000).

Gĩkũyũ mixed categories manifest not only lexical coherence (selecting complements of uniform type) but also phrasal coherence: the verbal constituents, regardless of whether they are lexically selected by the head, cohere with each other as a constituent. Thus, while in an NP it is possible to reorder all of the nominal dependents to produce marked orders, in the mixed construction all of the VP-type constituents must precede all of the NP-type constituents. Any ordering that interleaves the two types of constituents is disallowed. In (34a-c) we see that an associative phrase cannot interrupt a sequence of an object NP followed by Adverb:

(34) a. [mũ-thĩĩnj-i]$_{N}$ [mbũri]$_{NP}$ [wega]$_{ADV}$ [w-a Nairobi]
1-slaughter-NOM 10.goat 1.well 1-ASSOC N.
'a good goat slaughterer from Nairobi'

b. *[mũ-thĩĩnj-i]$_{N}$ [mbũri]$_{NP}$ [w-a Nairobi] [wega]$_{ADV}$
1-slaughter-NOM 10.goat 1-ASSOC N. 1.well

c. *[mũ-thĩĩnj-i]$_{N}$ [w-a Nairobi] [mbũri]$_{NP}$ [wega]$_{ADV}$
1-slaughter-NOM 1-ASSOC N. 10.goat 1.well

In (35a-c) a demonstrative cannot interrupt an object NP Adverb sequence:

(35) a. [mũ-end-i]$_{N}$ [a-ndũ]$_{NP}$ [mũno]$_{ADV}$ ũyũ
1-love-NOM 2-person a lot 1.DEM
'this one who loves people a lot'

b. *[mũ-end-i]$_{N}$ [a-ndũ]$_{NP}$ ũyũ [mũno]$_{ADV}$
1-love-NOM 2-person 1.DEM a lot

c. *[mũ-end-i]$_{N}$ ũyũ [a-ndũ]$_{NP}$ [mũno]$_{ADV}$
1-love-NOM 1.DEM 2-person a lot

(36a-b) shows that a relative clause cannot precede an adverb:

(36) a. [mũ-in-i]$_{N}$ [wega]$_{ADV}$ [ũ-rĩa mũ-nene]
1-sing-NOM well 1-REL 1-big
'the one who sings well who is big'
Lit.: 'singer well who is big'
b. *[mũ-in-i]$_{N}$ [ũ-rĩa mũ-nene] [wega]$_{ADV}$
1-sing-NOM 1-REL 1-big well

The impossibility of interleaving the two types of constituents holds in a specific region of structure immediately following the head N. Thus, while the demonstrative can precede the head in both the pure NP and the mixed NP/VP constructions —

(37) a. ũyũ [mũ-end-i]$_{N}$ w-a a-ndũ
1.DEM 1-love-NOM 1-ASSOC 2-person
'this lover of people'
b. ũyũ [mũ-end-i]$_{N}$ [a-ndũ]$_{NP}$
1.DEM 1-love-NOM 2-person
'this lover of people'

— following the head a choice must be made: either complements and modifiers will be uniformly nominal (in the permitted orders of NP-type constituents (19)), or they will be uniformly verbal (in the permitted orders of VP-type constituents (25)) until the verbal sequence is exhausted and the nominal sequence begins. In (38) the post-head demonstrative ũyũ marks the nominal choice-point:

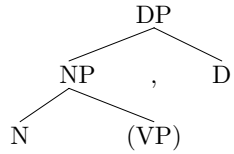
(38) a. [mũ-end-i]$_{N}$ ũyũ w-a a-ndũ
1-love-NOM 1.DEM 1-ASSOC 2-person
'this lover of people'
b. *[mũ-end-i]$_{N}$ ũyũ [a-ndũ]$_{NP}$
1-love-NOM 1.DEM 2-person
'this lover of people'
c. [mũ-end-i]$_{N}$ [a-ndũ]$_{NP}$ ũyũ
1-love-NOM 2-person 1.DEM
'this lover of people'

In general, then, the VP-like constituents and the NP-like constituents — regardless of whether they are selected or unselected by the head — belong to two separate, coherent regions of the structure, each subject to its own ordering constraints.

These generalizations can be explained by adding to the lexical coherence of selection for verbal or nominal complement types a requirement of phrasal coherence: the VP-style constituents within the mixed

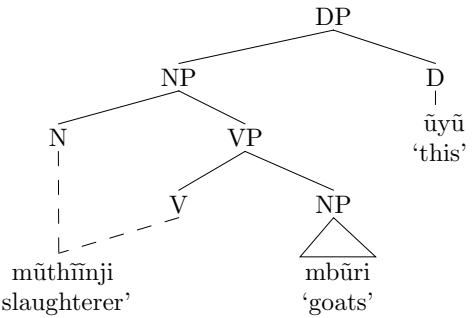
category cluster together as a unit, preventing higher nominal elements from interrupting them (Mugane 1996, Bresnan 1997):

(39)



If the VP complement in (39) is omitted, the resulting structure is that of the pure NP construction.[13] This structure can also explain without any further assumptions why the agentive nominalization in mixed categories must precede all of the other complements and modifiers (except for the optional preposing of a focused demonstrative): it occupies the typical head-initial position of Nouns in their nominal projections.[14] It remains to be answered how the two projections can share the same head, as depicted informally in (40):

(40) Gĩkũyũ mixed category:



We address this question in Section 10.6.

We see, then, that the internal syntax of agentive phrases seems to be grafted together from two different categorial projections sharing a single head, thereby displaying a combination of the properties displayed by VPs and NPs. Mugane (1996) shows that they also have the external syntax of nominal phrases (NPs/DPs), inducing subject or object agreement with a matrix verb and allowing extraction by clefting and relativization. These are properties not shared by VPs and CPs.

[13]Following Mugane (1996), we assume that all of the concordial NP modifiers following the demonstrative are adjoined to DP.

[14]It is not possible to conjoin two clusters of VP-style constituents under the same head; this may be because it is not possible to conjoin two VPs in general in Gĩkũyũ.

10.3 Alternative Analyses

Two interesting alternative analyses of the mixed category facts suggest themselves. By reinterpreting the data of mixed categories as either not truly phrasal or not truly mixed, they remove the challenge of Gĩkũyũ action nominalizations to the single projection theory. The counter-analyses also bring further properties of Gĩkũyũ and Bantu into the picture.

10.3.1 Synthetic compounds?

The first analysis is based on the fact that compound words in Bantu are head-initial (Mchombo 1978, Myers 1987). This fact suggests an analysis of these mixed constructions as synthetic compounds (dominated by a lexical rather than a phrasal category). After all, agentive nominalizations in English take phrasal PPs rather than direct NPs (41a), but in synthetic compounds they take a bare nominal complement unmediated by a preposition (41b):

(41) a. an eater of pumpkins, *an eater pumpkins
b. a pumpkin eater

In English the compounds can be easily distinguished from the phrases: complements appear before the head in compounds (41b), and after the head in phrases (41a). In Bantu, in contrast, complements follow the head in both compounds and phrases; one might be easily mistaken for the other.

Mugane (1996: Ch. 5) argues in detail that mixed category constructions are not synthetic compounds. The complements of the agentive nominalizations may be freely modified, allowing both pre- and post-head determiners (Mugane 1996:137–138) as in (42), and relative clauses (Mugane 1996:154) as in (43):

(42) mũ-end-i [aya a-ndũ], mũ-end-i [a-ndũ aya]
1-love-NOM 2.DEM 2-person 1-love-NOM 2-person 2.DEM
'a lover of these people'

(43) mũ-thĩĩnj-i [ĩno mbũri njeke]
1-slaughter-NOM 9.DEM 9.goat 9.thin
'a slaughterer of this thin goat'

The complements may also be freely coordinated (Mugane 1996:146):

(44) mũ-thĩĩnj-i [mbũri na [ngũkũ ici]]
1-slaughter-NOM 10.goat and 10.chicken 10.DEM
'a slaughterer of goats and these chickens'

A pure indexical such as a deictic pronoun may be the complement (Mugane 1996:155):

(45) mũ-thĩĩnj-i ĩyo
1-slaughter-NOM 9.DEM
'a slaughterer of that (class 9)'

These properties are not shared by lexical compounds in general (Bresnan and Mchombo 1995).[15] Thus, the mixed categories headed by agentive nominalizations in Gĩkũyũ are not synthetic compounds but phrasal constructions of syntax.

10.3.2 Adnominal participles?

A second alternative analysis of the mixed category facts is based on the existence of the class of adnominal participles in Gĩkũyũ. These are verbal forms which are used as modifiers of nominal heads. The participle is formed when the verb base (absent subject agreement and tense/aspect prefixes) is given a suffix (*-u* for perfect, *-e* for passive, and *-i* for habitual) and prefixed by the adjectival class markers to form an adnominal modifier. Examples are given in (46):

(46) a. (a-ndu) a-thĩĩnj-u mbũri
2-person 2-slaughter-PERF.PART 10.goat
'people who have slaughtered goats'

b. (a-ndu) a-thĩĩnj-e
2-person 2-slaughter-PASS.PART
'people who are operated on (by a surgical procedure)'

c. (mbũri) thĩĩnj-e ni a-ndũ
10.goat 10.slaughter-PASS.PART by 2-person
'goats that are slaughtered by people'

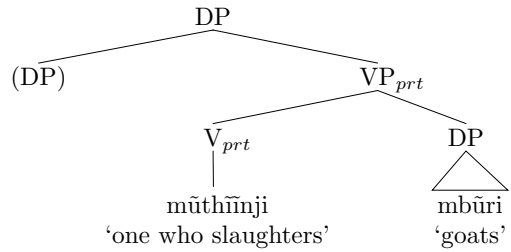
d. (a-ndu) a-thĩĩnj-i mbũri
2-person 2-slaughter-HAB.PART 10.goat
'people who slaughter goats'

The habitual form in (46d) resembles the agentive nominalization. Moreover, as happens with all types of NPs, the head nouns of these constructions can be dropped, yielding a definite or indefinite null anaphoric interpretation (see fn. 9). In other words, the modifying

[15] All of the above properties contrast with those of a compound in Gĩkũyũ called the '[mũ-...-a] compound' by Mugane (1996: Ch. 5). However, the Gĩkũyũ [mũ-...-a] forms can be loosely compounded with a single uncoordinated syntactic NP which does not begin with a determiner. This is in contrast to the much stricter constraints on phrasal recursivity shown by cognate compounds in Chichewa (Mchombo 1978, Bresnan and Mchombo 1995). See Mugane 1996 for details.

phrase can be used by itself (or in conjunction with other modifiers) as a referential expression. The structure of these examples is shown in (47):[16]

(47) Adnominal participle construction:



Could these adnominal participles be the solution to our problematic mixed category constructions? The answer is no. First, the participles bear the adjectival series of prefixes. These are formally identical to the noun prefix series except in class 8, where nouns have a prefix *ci-, i-* (depending on whether the stem begins with a vowel or not) and adjectives/participles have prenasalization of a verb-stem initial unaspirated consonant as in classes 9/10. This difference is illustrated in (48):

(48) a. ci-ana ndoot-i (< N-rot-i)
8-child 8.dream-PERF.PART
'children who dream (habitually)'
b. ndoot-i
8.dream-PERF.PART
'ones who dream (habitually)'
c. i-rot-i
8-dream-NOM
'dreamers (class 8)'

Only the adjectival prefixes may be used with adnominal verbal modifiers. Now class 8 mixed category constructions exist, headed by nominals bearing the noun prefix for class 8:

(49) i-mũrĩk-ĩr-i a-ndũ njira wega
8-shine-APPLIC-NOM 2-person 9.path well
'ones that illuminate paths for people well'

[16]The head position is shown in parentheses in (47). We assume that the null pronominal is not represented by a empty phrase structure category, but is functionally incorporated into the verbal morphology, where its f-structure value preempts the expression of a phrasal head in c-structure. See Bresnan and Mchombo (1987), Andrews (1990), Mugane (1996), Austin and Bresnan (1996), Bresnan (2001), and the works cited therein.

This fact clearly indicates that our mixed category agentive nominals are not simply adnominal participles in headless (null anaphora) constructions.

Secondly, the word order of the mixed category nominalizations differs from that of adnominal participles. Within the DP, adnominal participles occupy the same word order position as adjectives, designated 'AP' in the unmarked word order (19), repeated here:

(50) `N < Dem < PossPron < QP < AP < AssocP`

As such, they follow (in their unmarked order) all of the other types of adnominal modifiers except for associative phrases:

(51) mũ-ndũ w-a-kwa ũ-mwe mũ-thĩĩnj-i mbũri
1-person 1-my 1-one 1-slaughter-HAB.PART 10.goat
'my one person who is a slaughterer of goats'

They may precede or follow other APs, such as the adjective in (52):

(52) a. mũ-ndũ mw-ega mũ-thĩĩnj-i mbũri
1-person 1-good 1-slaughter-HAB.PART 10.goat
'a good person who slaughters goats'
b. mũ-ndũ mũ-thĩĩnj-i mbũri mw-ega
1-person 1-slaughter-HAB.PART 10.goat 1-good

But when they precede a number expression (which is an instance of QP in (50)), for example, they are separated by a pause, showing this to be a marked order:

(53) a. mũ-ndũ ũ-mwe mũ-thĩĩnj-i mbũri
1-person 1-one 1-slaughter-HAB.PART 10.goat
'one person who is a slaughterer of goats'
b. mũ-ndũ mũ-thĩĩnj-i mbũri, ũ-mwe
1-person 1-slaughter-HAB.PART 10.goat, 1-one

The agentive nominalizations clearly contrast in their word order possibilities, as we saw in (31). Compare, for example, the unmarked order of (51) with that of (54):

(54) mũ-thĩĩnj-i mbũri w-a-kwa ũ-mwe
1-slaughter-NOM 10.goat 1-ASSOC-my 1-one
'my one goat slaughterer'

In short, the agentive nominal occurs not in the position of an AP, following the head and other modifiers, but in the position of the head itself, preceding all other modifiers except for focused demonstratives.

We see, then, that the agentive nominalization has both the morphology and the syntactic positioning of the head of an NP. In these

respects it behaves like a pure noun, which can never be used adnominally (without an associative particle). The agentive nominal head of the pure NP construction shares this pure nominal property:

(55) *mũ-ndũ mũ-thĩĩnj-i w-a mbũri
1-person 1-slaughter-NOM 1-ASSOC 10.goat
Lit.: 'a person slaughterer of goats'

In sum, the mixed categories in question are both truly phrasal and truly mixed, in the sense that they consist of a VP embedded within an NP whose head position is occupied by the agentive nominalization.

10.4 Haspelmath's Generalization

So far we have found evidence for the following two conclusions about the mixed agentive nominalization constructions in Gĩkũyũ:

(56) a. The agentive nominal heads of mixed categories in Gĩkũyũ are deverbal nouns occupying the phrase-initial head position of a nominal projection (NP).
b. These mixed category constructions in Gĩkũyũ consist syntactically of components of a verbal projection (VP) embedded within a nominal projection (NP).

(56a,b) have been established in Sections 10.2 and 10.3. We now observe that there is a relation between these two conclusions. The morphological structure of the head reflects the syntactic structure of the phrasal construction: the construction consists syntactically of a verbal phrase embedded within a nominal phrase, as we saw in (39) and (40), and the head contains a verbal base embedded within nominal morphology, as we see in (57). (The v, n subscripts indicate the categorial type of the stems as respectively verbal or nominal.)

(57) a. [mũ-[[thĩĩnj]$_v$-i]$_n$]$_{\text{N}}$
1-slaughter-NOM
b. [mũ-[[in-ĩr]$_v$-i]$_n$]$_{\text{N}}$
1-sing-APPLIC-NOM

This relationship is not to be dismissed as an accident or a purely language-particular phenomenon. The existence of similar morphology-syntax relations in mixed categories is widespread crosslinguistically, and has been generalized by Haspelmath (1995), who specifically relates the syntactic structure of a mixed category to the morphological structure of the head (Haspelmath 1995:56–58):

(58) **Haspelmath's generalization:**
(a) In words derived by *inflectional* word-class-changing morphology, the internal syntax of the base tends to be preserved.
(b) In words derived by *derivational* word-class-changing morphology, the internal syntax of the base tends to be altered and assimilated to the internal syntax of primitive members of the derived word-class.

Haspelmath defines 'inflectional' morphology as productive morphology. By 'internal syntax' Haspelmath refers to the combination of the head with its dependents inside its phrase; the 'external syntax' — how the head combines with elements outside its phrase — is determined by the derived word class (Haspelmath 1995:52).

Nikitina (2005, 2006) shows that constructions with mixed-category syntax occur in some Mande languages which lack formal word-class changing morphology. Thus the morphology-syntax relationship formulated by Haspelmath is only a one-way implication: productive word-class changing morphology is associated with mixed-category syntax, but mixed-category syntax can also arise independently. With this understanding of its limitations, we reformulate the generalization in our terms as in (59):

(59) The productive morphological derivation of a word of one category $\mathcal{C}_1$ from a base of another category $\mathcal{C}_2$ will tend to preserve the syntactic structure of $\mathcal{CP}_2$ within the syntactic context of $\mathcal{CP}_1$, while less productive category-changing morphology will tend to alter the syntactic context of the base category $\mathcal{CP}_2$ to that of $\mathcal{CP}_1$.

For Gĩkũyũ $\mathcal{C}_1 = \text{N}$ and $\mathcal{C}_2 = \text{V}$. Thus, the agentive nominalization is a nominal word of category N productively derived from a verbal base of category V (the verb stem), and VP structure is preserved within the syntactic context of NP.

10.5 Implications for Theories of Mixed Categories

Haspelmath's generalization and its particular instantiation in Gĩkũyũ are highly problematic for one previous approach to mixed categories, which we call 'the single projection' hypothesis:

(60) **The single-projection hypothesis:**
A mixed category is the single phrasal projection of a morphologically 'mixed' (underspecified, indeterminate, bivalent) head.

In precisely what way the head is morphologically 'mixed' under the

single-projection hypothesis (60) varies with the particular version of the approach. The feature-neutralization version assumes that the head of the mixed construction is lexically underspecified for its category and so projects a categorially indeterminate phrasal structure which may contain constituents of mixed types (Aoun 1981, van Riemsdijk 1983, Grimshaw 1991). A consequence of this approach is that mixed category constructions must have underspecified heads which are formally ambiguous as to category type — just as we see in English gerundive verb constructions (2) or the Italian *infinito sostantivato* (5), (6), (8). In contrast, a type-hierarchical version of the single projection hypothesis assumes that such mixed categories belong to a distinct fine-grained category that inherits some typical properties from nouns and some from verbs (Malouf 1998:89, 163). Under the former (feature-neutralization) version the mixed category is thus underspecified or neutral in category, while on the latter (type-hierarchical) version it is multiply-specified or bivalent. Under both versions of this approach it heads a single endocentric projection and the lexical integrity of the head is preserved.

A basic problem for the feature-neutralization version of (60) is that category neutrality of the head is *not* a universal characteristic of mixed category constructions, as Gĩkũyũ shows. Categorially unambiguous heads also appear in Quechua nominalization-headed clauses (Lefebvre and Muysken 1988), Arabic deverbal process nominals or *maṣdars* (Fassi Fehri 1993), Hebrew action nominalizations (Hazout 1995, Falk 2006), and many other examples (Haspelmath 1995).

A second problem, which applies to all varieties of the single projection hypothesis, is that phrasal coherence constrains the mixing of categories. That is, mixed category constructions (in configurational languages, at least) do not freely mix or interleave constituents of the different category types, but instead cohere within distinct regions which can be bounded by distinct phrase structure brackets. For example, the VP-style constituents within the Gĩkũyũ mixed category cluster together as a unit, preventing higher NP-style elements from interrupting them. We have observed this property in Gĩkũyũ in Sections 10.2.3 and 10.3.1, and represented it by the tree structure (39).

Phrasal coherence appears to be a general property of mixed category constructions across languages (Bresnan 1997). With the Italian *infinito sostantivato*, for example, the constituents preceding the infinitive are always nominal (determiners and adjectives) and can cooccur with either post-infinitive VP constituents (such as direct objects and adverbs) or NP constituents (such as postnominal adjectives and *di* phrases). However, the post-infinitive constituents of different category

types cannot cooccur, but must be uniformly of the VP type or the NP type. As shown in the following examples from Zucchi (1993:222) and Bresnan (1997), a post-infinitive adjective permits only nominal constituents (e.g., other adjectives and *di* phrases) to follow, while a post-infinitive adverb permits only verbal constituents (e.g., a direct object or other verbal complement) to follow.

(61) a. il mormorare sommesso/*sommessamente del mare
the whisper.INF soft/softly of.the sea
'the soft whispering of the sea' (Zucchi 1993:220)

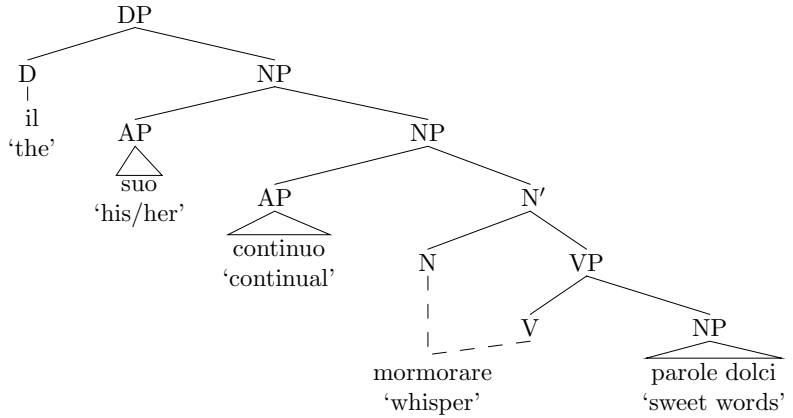
b. il suo mormorare sommessamente
the his/her whisper.INF softly
'his/her whispering softly' (Zucchi 1993:226)

c. il suo momorare continuamente parole dolci
the his/her whisper.INF continually words sweet

d. *il suo momorare continuo parole dolci
the his/her whisper.INF continual words sweet
[compare to (5)] (Zucchi 1993:245)

This phrasal organization suggests that the infinitival head may take a VP complement, which prevents a postnominal adjective (required to appear in postnominal position adjacent to the head) from appearing. Bresnan (1997) depicts the syntactic structure of the Italian mixed category construction in the following diagram:[17]

(62) Italian *infinito sostantivato*:



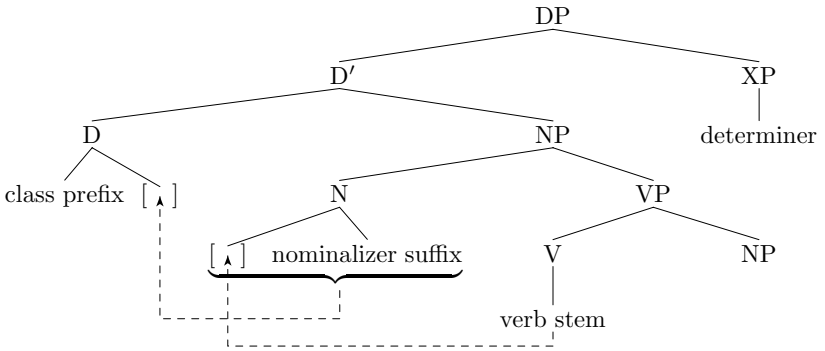
These considerations motivate the dual-projection hypothesis:

[17]Zucchi (1993:251) analyzes the infinitive as dominated by V, but all of the evidence he cites is consistent with its being dominated by N.

(63) **The dual-projection hypothesis:**
Mixed categories consist of not one, but two projections that differ in category type in a way reflected in the morphology of the head.

The most widely known version of the dual-projection hypothesis assumes that the verbal base of a deverbal nominal mixed category starts out as a verb heading the VP and then is moved into the N position (or to the position of a nominal functional projection), as illustrated in (64) (cf. Fassi Fehri 1993, Hazout 1995, Borsley and Kornfilt 2000). It is schematically applied to the Gĩkũyũ construction in (64):

(64) Syntactic word-formation by head movement:



By further assuming that the derived X^0 category is moved into the next higher X^0 category, this theory can also explain Haspelmath's generalization (59): the morphemic structure of the agentive nominalization, under these assumptions, must reflect the syntactic embedding relations of the projections.

This approach to mixed categories preserves the principle of endocentricity, explaining how two different categories of syntactic projections can arise from a single word: the categories are separately projected from different heads which are subsequently joined by syntactic movement into a single word, and it captures the systematic relation between the morphological composition of the head and the syntactic structure of the mixed category.

The weakness of the approach is in failing to explain the relations between lexically and syntactically derived words. Lexically derived words do not give evidence of phrasal sources for their morphological components. This point is made by Ackema and Neeleman (2001) for English, using the following example:[18]

[18]The same point is made by Bresnan and Mchombo (1995) for Bantu noun class prefixal morphology.

(65) *the truck driver reluctantly a rusty to Arizona
'the one who reluctantly drives a rusty truck to Arizona'

If the synthetic compound *truck driver* were syntactically derived from phrasal syntactic sources by head-movement, then the specifiers, complements, and modifiers of the phrasal sources of the morphological components should be visible, exactly as is hypothesized in the case of the mixed category construction (64). This point is illustrated by (66):

(66) the -er [$_{VP}$ reluctantly drive a rusty truck to Arizona] ⇒
the -er [$_{VP}$ reluctantly truck-drive a rusty ___ to Arizona] ⇒
the truck drive-er [$_{VP}$ reluctantly ___ a rusty ___ to Arizona]

In Gĩkũyũ, too, there is evidence that agentive nominals heading the pure NP constructions cannot be syntactically derived. If these were derived in the syntax in the way hypothesized for the heads of the mixed NP/VP constructions (64), then we would expect to find evidence of a syntactic VP, such as stranded adverbials or the like, even in the unmixed NP construction. But nothing of this sort can appear:

(67) a. *mũ-in-i wega w-a nyĩmbo
1-sing-NOM well 1-ASSOC 10.song
Lit.: 'a singer well of songs'
b. *mũ-in-i w-a nyĩmbo wega
1-sing-NOM 1-ASSOC 10.song well
Lit.: 'a singer of songs well'

In (67) the selection of the associative phrase *wa nyĩmbo* to express the nominal complement (perhaps as an argument adjunct) clearly marks the construction as unmixed. This inference follows from the uniformity of selection of complement type, described as 'lexical coherence' in Section 1.3. As such, it cannot take an adverbial modifier. Contrast the construction in (68), where the adjunct 'of the settlement' is not selected by the nominalization and hence is consistent with the structure of a mixed NP/VP construction. Here an adverb is possible (68a), but only in the coherent VP portion of the structure, which precedes the higher adnominal adjunct (68b):

(68) a. mũ-in-i wega w-a i-tũũra
1-sing-NOM well 1-ASSOC 5-settlement
Lit.: 'a singer well of the settlement'
b. *mũ-in-i w-a i-tũũra wega
1-sing-NOM 1-ASSOC 5-settlement well

Lit.: 'a singer of the settlement well'

The inability of the pure NP agentive nominals to take manner adverbs holds for both Gĩkũyũ and English, as the literal translations of (67a,b) show. Rappaport Hovav and Levin (1992) propose an account of why English *-er* nominals, although showing some eventive properties, nevertheless prohibit certain types of adverbs. They suggest that such adverbs are modifiers of an open event variable in the argument structure of a base verb, but that in the case of agentive nominals, this event variable is lexically quantified prior to syntactic argument linking (Rappaport Hovav and Levin 1992:143). On this account, the possibility or impossibility of such adverbial modifiers follows from a *lexical* property of the nominalized forms, and is expressed in their lexical argument structures.

Given that there are lexically derived agentive nominals, the problem for syntactic word formation is twofold. First, words hypothesized to be syntactically derived do not differ in morphological structure from those lexically formed (see Bresnan and Mchombo (1995) for a review of evidence). While this fact can be captured by various stipulations, it remains fundamentally unexplained by the syntactic word-formation approach, because the opposite state of affairs could be captured just as readily and would in fact be seen as confirmation of the theory. Second, the question of which words are lexically and which syntactically derived — or to put it more neutrally, which words head unmixed and which head mixed category constructions — needs to be answered by the syntactic word formation approach just as much as by other approaches.

10.6 An Analysis Within LFG

Within LFG there is a simple solution to these problems posed by Gĩkũyũ. Suppose that each lexeme carries a categorization constraint which is preserved under productive morphological processes. Such a constraint is easily formalized via inside-out function application using the 'CAT' function (Halvorsen and Kaplan 1988, Kaplan 1995, Nordlinger 1998, Crouch et al. 2006). For example, a verb lexeme would carry a constraint like that in (69a), and a noun lexeme would carry one like that in (69b):

(69) a. VP ε CAT((PRED $\uparrow$))
b. NP ε CAT((PRED $\uparrow$))

Such constraints categorize the c-structure domain in which a lexical head (providing the PRED attribute) must be found. Technically, the

constraints require that a VP (respectively NP) be among the c-structure categories of the nodes in the inverse image of the ϕ mapping from the f-structure containing the PRED.

Productive morphological processes such as tense-marking or number inflection will preserve categorization constraints. If the English verb *slaughter*, for example, carries the constraint (69a), so will its present tense form *slaughters*. In contrast, derivational morphology usually does not preserve the categorization information of the base lexeme. For example, the argument structure of the English deverbal noun *slaughterer* is derived from its verbal base *slaughter*. The lexical relation of *slaughterer* to *slaughter* is relatively transparent, as illustrated in (70):[19]

(70) slaughter: 'slaughter$< x, y >_v$'
slaughterer: 'agent-of $< x$, slaughter$< x, y >>_n$'

The notations '$\langle \ldots \rangle_v$' and '$\langle \ldots \rangle_n$' represent the categorization of the predicators as verbal or nominal, respectively. Note that the categorization of the base verb is not retained in the nominalization of the verb. These features of the argument structures will flag the presence of the categorization constraints in (69) in the lexical entries for these predicators (which are presumably derived by some version of the lexical mapping theory):[20]

(71) a. slaughter: V: ($\uparrow$ PRED) = 'slaughter$<$($\uparrow$ SUBJ)($\uparrow$ OBJ)$>_v$'
v: VP ε CAT((PRED$\uparrow$))
b. slaughterer: N: ($\uparrow$ PRED) = 'slaughterer$<$($\uparrow$ OBL$_\theta$)$>_n$'
n: NP ε CAT((PRED$\uparrow$))

For Gĩkũyũ we simply assume that mixed categories are productively formed words which retain the categorization constraints of their bases, as in (72):

(72) mũthĩĩnji: 'agent-of $< x$, slaughter$< x, y >_v >_n$'

The argument structure of the Gĩkũyũ agentive nominalization in (72), unlike the English (70), has the categorization information '$\langle \ldots \rangle_v$' embedded within it.

Predicators of the type in (72) are formed in the component of grammar which produces argument structures, in this case the lexical morphology. The verbal argument structure is transparently embed-

[19]Recall that extracting the agent role is only the most typical function of the agentive nominalizing suffix, as mentioned at the outset of Section 10.2.

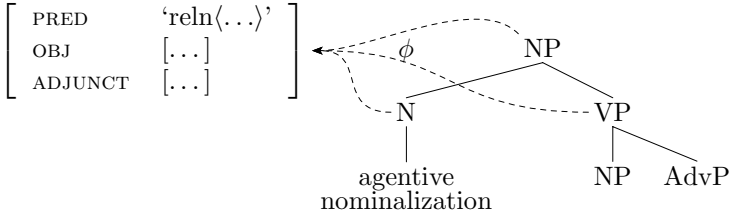
[20]The lexical entry forms show only the grammatical functions required for completeness and coherence, abstracting away from the argument relations among base and derivative shown in (70).

ded within a nominally categorizing argument structure (contributed by the nominalizing morphology and designated '$\langle \ldots \rangle_n$'). From this information a lexical entry such as (73) can be derived:[21]

(73) mũthĩĩnji: N: (↑ PRED) = 'slaughterer<<(↑ OBJ)>$_v$>$_n$'
v: VP ε CAT((PRED↑))
n: NP ε CAT((PRED↑))

This analysis permits a complete and coherent f-structure for the entire construction. To see this, consider the following. The lexical entry (73) requires that the f-structure of the PRED must be the image under ϕ of VP as well as NP. In other words, the lexical entry licenses the presence of both a VP and an NP. The head N contributes to f-structure both the noun class required of every NP in Gĩkũyũ and the attributes of predicator, while the VP allows an object and adverbial adjunct, which are characteristic of VP f-structures, as illustrated in (74).

(74)



The question arises, given (73), why must VP be inside NP rather than the other way around? An answer is provided by extended head theory (Jar 1993; Zaenen and Kaplan 1995:221–2; Bresnan 2001): an extended head by definition cannot appear lower in the tree than the phrase(s) which it heads. Hence the nominalization's NP projection must dominate the VP.[22]

Of course, not every nominalized verb will be able to serve simultaneously as a VP and NP predicator. In Gĩkũyũ, we find that agentive and other nominalizations can head mixed categories, while infinitive nouns cannot (Mugane 1996). In Italian it is the reverse (Zucchi 1993).

[21] Again, the lexical entry forms show only the grammatical functions required for completeness and coherence, abstracting away from the argument relations among base and derivative shown in (72).

[22] To see this, note that in (74) the agentive nominalization is a noun and is the c-structure head (as well as the extended head) of the NP which dominates it. It is also the extended head of its VP sister, which is annotated by the principle permitting lexical categories to have co-heads as an option (Bresnan 2001: Ch. 6). Hence in (74) the f-structures of the N and VP are identified through unification as permitted by the extended head theory.

In English, agentive nominalizations are unmixed nominals, which take only nominal complements and modifiers (75a), and reject the objects and adverbs of verbal constructions (75b,c):

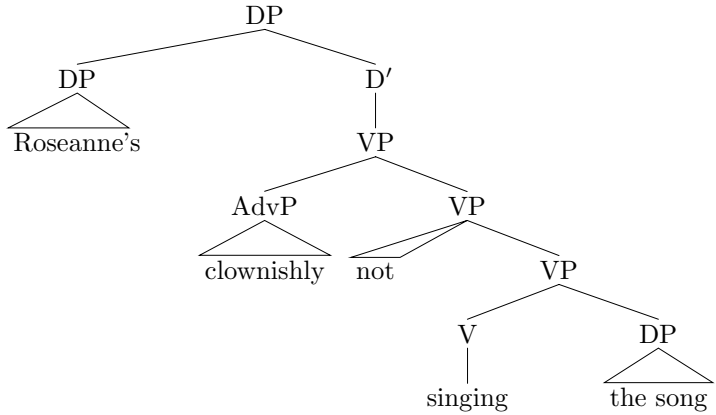
(75) a. this unthinking slaughterer of goats
b. *this slaughterer goats unthinkingly
c. Don't slaughter goats unthinkingly!

As illustrated above (71), the PRED value of the English nominalization simply lacks a transparently embedded verbally categorizing argument structure corresponding to its verbal base.

Thus, the lexical morphology of a language must provide the resources to support mixed categories in the syntax by licensing appropriate f-structure attributes. But argument structure alone will not suffice to solve the syntactic problems of phrasal coherence, endocentricity, and head positioning presented by mixed categories. For these, the theory of structure-function mapping appears essential.

The extended head theory of mixed categories makes an interesting prediction about the syntactic positioning of the heads in their phrasal structures. Suppose that a mixed category involves a lexical category such as VP embedded in a functional category such as DP. In this case, the head may be positioned in VP without violating endocentricity. Every lexical category must have an extended head, but a functional category need not, because functional categories are headed by recoverable classes of elements (Bresnan 2001: Ch. 7). This gives us a natural structure for the English gerundive construction (Bresnan 2001: Ch. 13) illustrated in (76).

(76)



The Gĩkũyũ-style analysis given above (74) would be inappropriate for the verbal English construction because it would have the gerun-

dive verb in the head N position of the mixed category, where we would expect the possibility of prenominal AP modifiers (such as are found in the Italian infinitive noun construction) and nominal negative prefixation cooccurring with the VP properties:

(77) Roseanne's clownish non-singing *(of) the national anthem.

There is also positive evidence for the presence of DP in (76). The DP explains why no nominal head is required in the construction: the nominal functional category DP need not have a head. The DP also explains why the subject of the verbal gerundive construction takes the genitive form, because this is the syntactic attribute of the specifier of DP.[23]

There is in fact interesting evidence from quantifier scope that the genitive NP has the scope properties of possessive NPs of nouns, and not of subjects of embedded Ss. Observe the contrast in (78) (Zucchi 1993:50):

(78) a. John resents everyone's taking a day off.
b. John resents that everyone takes a day off.

The quantifier phrase in (78a) may have wide scope, exactly as in (79):

(79) John resents everyone's absence.

Both (78a) and (79) are ambiguous: John may resent only the universal absence of other employees, leaving him stuck with all the work (the wide scope reading); or for each absent employee, John may resent that person's individual absence (the narrow scope reading). But (78b) differs in preferring the narrow scope reading.

Finally, in the context of the theory of extended heads, the DP in (76) can explain why the verbal gerund shares some properties of deverbal nouns: a nominally categorizing argument structure is needed to support a possessor. Thus the two types of gerundive verb forms in English can be represented in our theory as in (80a,b):

(80) a. singing: N: $(\uparrow \textsc{pred}) =$ 'singing$<(\uparrow \textsc{obl}_\theta)>_n$'
b. singing: V: $(\uparrow \textsc{pred}) =$ 'singing$<<(\uparrow \textsc{subj})(\uparrow \textsc{obj})>_v>_n$

Note that the outer nominal typing of the verbal argument structure in (80b) does not prevent the categorization of the gerundive as a V in c-structure. This is evidence that the category identity properties of heads can be distinct from their f-structure licensing properties, as we have assumed. If, however, the verbal gerund in (80b) were categorized

[23] We may assume that the Specifier of DP is the most prominent argument function for verbs or nouns: POSS or SUBJ, depending on the a-structure requirements (Laczkó 1995, 1997).

as N instead of V, a mixed lexical category construction would result, allowing examples such as (77). Interestingly, examples of this type did occur in historically earlier stages of English (Tajima 1985).

We see, then, that when category mixing involves a lexical and a functional category, the head may appear in the lower, lexical category. But when category mixing involves two lexical categories sharing the same head, it is predicted that the head must appear in the upper lexical category. This follows because by endocentricity every lexical category must have an extended head, and an extended head by definition cannot appear lower in the tree than the phrase(s) which it heads.

Finally, we observe a limitation of our solution. We have reconciled the conflict between the principles of endocentricity and lexical integrity by exploiting the fact that words and phrases talk to each other through their common functional structure. Thus a single lexical word such as a denominal agentive nominalization can constrain the category types of the regions of tree structure that correspond to its functional domain. At the same time, a single lexical head in constituent structure can serve as the extended head of a cascade of phrases in the tree structure above or below it through the many-to-one correspondence of tree structure nodes to functional structures. This mapping between expressions and functional structure is intentionally imperfect: by flattening trees, it loses information. This property of the correspondence architecture is considered a feature, not a bug, because many languages in fact make far less use of hierarchical constituent structure than do highly endocentric languages like English (Bresnan 2001 and references). However, it follows from this property that only a minimal amount of matching between the word-structure and the constituent structure can be explained by the analysis offered here. To extend the matching between more than two levels of morphological derivation and syntactic tree structure would require that word derivation define hierarchical functional structures (Simpson 1991, Nordlinger 1998), but that is beyond the scope of the present study.

Acknowledgments

The analysis of mixed categories we have presented draws heavily on the flexibility and power of the LFG architecture, and in particular on the central conception of the ϕ mapping between categorial structures and feature structures (as well as some of its specific applications) — which are due to Ron Kaplan. Thanks, Ron!

References

Ackema, Peter and Ad Neeleman. 2001. Competition between syntax and morphology. In G. Legendre, J. Grimshaw, and S. Vikner, eds., *Optimality Theoretic Syntax*, pages 29–60. Cambridge, MA: The MIT Press.

Alsina, Alex. 1996. *The Role of Argument Structure in Grammar. Evidence from Romance*. Stanford, CA: CSLI Publications.

Andrews, Avery D. 1990. Unification and morphological blocking. *Natural Language and Linguistic Theory* 8:507–557.

Aoun, Yosef. 1981. Parts of speech: a case of redistribution. In A. Belletti, L. Brandi, and L. Rizzi, eds., *Theory of Markedness in Generative Grammar*, pages 3–24. Pisa, Italy: Scuola Normale Superiore di Pisa.

Austin, Peter and Joan Bresnan. 1996. Non-configurationality in Australian aboriginal languages. *Natural Language and Linguistic Theory* 14:215–268.

Borsley, Robert D. and Jaklin Kornfilt. 2000. Mixed extended projections. In R. D. Borsley, ed., *The Nature and Function of Syntactic Categories*, pages 101–131. New York, NY: Academic Press.

Bresnan, Joan and Sam A. Mchombo. 1987. Topic, pronoun, and agreement in Chicheŵa. *Language* 63(4):741–782.

Bresnan, Joan and Sam A. Mchombo. 1995. The lexical integrity principle: Evidence from Bantu. *Natural Language and Linguistic Theory* 13:181–252.

Bresnan, Joan. 1997. Mixed categories as head-sharing constructions. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 1997 (LFG'97)*. San Diego, CA: CSLI Online Publications.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Oxford, United Kingdom: Blackwell.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, John T. Maxwell, III, and Paula S. Newman. 2006. XLE Documentation. Palo Alto Research Center.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III, and Annie Zaenen, eds. 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Falk, Yehuda N. 2006. Constituent structure and grammatical functions in the Hebrew nominal phrase. In J. Grimshaw, J. Maling, C. Manning, J. Simpson, and A. Zaenen, eds., *Architectures, Rules and Preferences*. Stanford, CA: CSLI Publications. To appear.

Fassi Fehri, Abdelkader. 1993. *Issues in the Structure of Arabic Clauses and Words*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Grimshaw, Jane. 1991. Extended projection. Brandeis University Linguistics and Cognitive Science Program, Waltham, MA.

Halvorsen, Per-Kristian and Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*

(FGCS'88), pages 1116–1122. Tokyo, Japan. Reprinted in Dalrymple et al. (1995:279–292).

Haspelmath, Martin. 1995. Word-class-changing inflection and morphological theory. In G. Booij and J. van Marle, eds., *Yearbook of Morphology 1995*, pages 43–66. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Hazout, Ilan. 1995. Action nominalizations and the lexicalist hypothesis. *Natural Language and Linguistic Theory* 13:355–404.

Jar, M. [John T. Maxwell III, Annie Zaenen, Ronald M. Kaplan and Mary Dalrymple]. 1993. Reconstituted X′ constituents in LFG. Palo Alto, CA: Xerox Palo Alto Research Center, duplicated MS.

Kaplan, Ronald M. 1995. The formal architecture of Lexical-Functional Grammar. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell, III, and A. Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, pages 7–27. Stanford, CA: CSLI Publications.

Laczkó, Tibor. 1995. *The Syntax of Hungarian Noun Phrases — A Lexical-Functional Approach*. Frankfurt am Main, Germany: Peter Lang.

Laczkó, Tibor. 1997. Action nominalization and the possessor function. *Acta Linguistica Hungarica* 44(3–4):413–475.

Lee, Hanjung. 1999a. Aspectual and thematic licensing of grammatical case. In S. J. Billings, J. P. Boyle, and A. M. Griffith, eds., *Papers from the 35th Regional Meeting of the Chicago Linguistic Society (CLS)*, pages 203–222. Chicago, IL.

Lee, Hanjung. 1999b. The domain of grammatical case in Lexical-Functional Grammar. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 1999 (LFG'99)*. Manchester, United Kingdom: CSLI Online Publications.

Lefebvre, Claire and Pieter Muysken. 1988. *Mixed Categories: Nominalizations in Quechua*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Malouf, Robert P. 1998. *Mixed Categories in the Hierarchical Lexicon*. Ph.D. thesis, Stanford University.

Malouf, Robert P. 2000. *Mixed Categories in the Hierarchical Lexicon*. Stanford, CA: CSLI Publications.

Mchombo, Sam A. 1978. *A Critical Appraisal of the Place of Derivational Morphology within transformational Grammar, Considered with Primary Reference to Chichewa and Swahili*. Ph.D. thesis, School of Oriental and African Studies, University of London, London, United Kingdom.

Mugane, John. 1996. *Bantu Nominalization Structures*. Ph.D. thesis, University of Arizona Department of Linguistics, Tucson, AZ.

Mugane, John. 1997. *A Paradigmatic Grammar of Gĩkũyũ*. Stanford Monographs on African Languages. Stanford, CA: CSLI Publications.

Mugane, John. 2003. Hybrid constructions in Gĩkũyũ: Agentive nominalizations and infinitive-gerund constructions. In M. Butt and T. H. King, eds., *Nominals Inside and Out*, pages 235–265. Stanford, CA: CSLI Publications.

Myers, Scott. 1987. *Tone and the Structure of Words in Shona*. Ph.D. thesis, University of Massachusetts, Amherst, MA.

Nikitina, Tatiana. 2005. Mixed category constructions and word order change in Niger-Congo. Presented at the 17th International Conference on Historical Linguistics (ICHL'05), Workshop on Constructions in Language Change, Madison, WI.

Nikitina, Tatiana. 2006. Morphological exponent of mixed category constructions: Embedded clauses with mixed syntactic properties in Wan. Unpublished Manuscript, Stanford University.

Nordlinger, Rachel. 1998. *Constructive Case: Evidence from Australian Languages*. Stanford, CA: CSLI Publications.

Przepiórkowski, Adam. 1999. On case assignment and "adjuncts as complements". In G. Webelhuth, J.-P. Koenig, and A. Kathol, eds., *Lexical and Constructional Aspects of Linguistic Explanation*, pages 231–245. Stanford, CA: CSLI Publications.

Rappaport Hovav, Malka and Beth Levin. 1992. *-er* nominals: implications for the theory of argument structure. In T. Stowell and E. Wehrli, eds., *Syntax and the Lexicon. Syntax and Semantics*, vol. 26, pages 127–153. San Diego, CA: Academic Press.

Simpson, Jane. 1991. *Warlpiri Morpho-Syntax: A Lexicalist Approach*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Tajima, Matsuji. 1985. *The Syntactic Development of the Gerund in Middle English*. Tokyo, Japan: Nan'un-do.

van Riemsdijk, Henk. 1983. The case of German adjectives. In F. Heny and B. Richards, eds., *Linguistic Categories: Auxiliaries and Related Puzzles*, pages 223–52. Dordrecht, The Netherlands: Reidel.

Wechsler, Stephen and Yae-Sheik Lee. 1996. The domain of direct case assignment. *Natural Language and Linguistic Theory* 14(3):629–664.

Zaenen, Annie and Ronald M. Kaplan. 1995. Formal devices for linguistic generalizations: West Germanic word order in LFG. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell III, and A. Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, pages 215–239. Stanford, CA: CSLI Publications.

Zucchi, Alessandro. 1993. *The Language of Propositions and Events*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

11

Restriction for Morphological Valency Alternations: The Urdu Causative

Miriam Butt and Tracy Holloway King

11.1 Introduction

This paper explores the use of the *Restriction Operator* (Kaplan and Wedekind 1993) for a computational treatment of complex predication. The Restriction Operator has already been applied to a treatment of syntactically formed complex predicates (Butt, King and Maxwell 2003). It has not, however, so far been applied to morphologically formed complex predicates. In this paper, we present an implementation that uses restriction for dealing with both Urdu causatives (morphologically formed complex predicates) and Urdu permissives (syntactically formed complex predicates). The finite-state realizational model (Karttunen 2003) standardly used within the ParGram project (Butt et al. 1999, Butt et al. 2002) serves as the morphology-syntax interface. We also examine the interaction of the different types of complex predicates with one another and with periphrastic passive formation. As will be seen in the course of the paper, the use of the Restriction Operator raises some interesting architectural and theoretical issues, which we discuss in the concluding section (section 11.7). The structure of the paper is as follows. In section 11.2, we briefly discuss the challenges presented by complex predicates, and we contrast theoretical and computational perspectives. In section 11.3, we introduce the Restriction Operator and illustrate how it has previously been applied

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

to Urdu permissives in the syntax. Section 11.4 provides some theoretical background to the analysis of causatives. In section 11.5, with respect to Urdu morphological causatives, we show how the Restriction Operator can operate within the morphological component. In section 11.6, we examine the interactions between syntactic and morphological complex predicates and passives and then explore the theoretical and computational implications.

11.2 Complex Predicates: Theoretical vs. Computational Perspectives

Complex predicate formation is akin to valency changing operations in that two clearly identifiable heads each contribute to a joint, complex argument structure. Some examples which have been dealt with extensively from an LFG perspective are shown in (1).[1]

(1) a. yassIn=nE nAdyA=kO gHar banA-n-E
Yassin=Erg Nadya=Dat house.M.Nom make-Inf-Obl
dI-yA
give-Perf.M.Sg
'Yassin let Nadya make a house.' [Urdu Permissive, Butt 1995]

b. nAdyA=nE xat likH lI-yA
Nadya=Erg letter.M.Nom write take-Perf.M.Sg
'Nadya wrote a letter (completely).' [Urdu V-V, Butt 1995]

c. nAdyA=nE kahAnI yAd k=I
Nadya=Erg story.F.Nom memory do-Perf.F.Sg
'Nadya remembered a/the story. [Hindi/Urdu N-V, Mohanan 1994]

d. L'elefant fa riure les hienes.
the elephant makes laugh the hyenas
'The elephant makes the hyenas laugh.' [Catalan causative, Alsina 1997]

It has been shown conclusively for all of these constructions by a variety of tests that the f(unctional)-structure must be monoclausal (i.e., there is no embedded subject) even though the a(rgument)-structure is complex (e.g., the discussions in Mohanan 1994, Butt 1995, Alsina 1996, 1997).

From a theoretical linguistic perspective, morphological valency changing operations have always been regarded as easy: they are gen-

[1]The transcription of the Urdu examples here follows the simple ASCII based transcription used within the Urdu ParGram grammar. Capital letters stand for long vowels or retroflex consonants, capital H indicates aspiration and capital N shows nasalization of the preceding vowel.

erally accounted for by lexical rules or by different realizations in argument structure. Syntactic valency changing operations are also easy if a syntactic element can be treated as an operator which triggers the *addition* or *deletion* of an argument, as is the case in applicatives (addition) or passives (deletion). However, they are more complicated to account for if the subcategorization frame is *jointly* determined by different pieces of the syntax (verbs, nouns, or adjectives). As shown conclusively within LFG, this type of joint argument structure determination is exactly what occurs with true complex predicate formation as illustrated in (1) (again, see Mohanan 1994, Butt 1995, Alsina 1996).

The analysis developed for complex predicate formation in Romance and Urdu/Hindi entails that argument structure composition cannot be confined to the lexicon, as had until then been assumed by Lexical Mapping Theory,[2] but must also be able to take place in the syntax. Alsina's and Butt's LFG analyses led to a complication of LFG's architecture, but none that went beyond the possibilities of LFG's relatively powerful projection model, whereby any one linguistic projection (e.g., a-structure, c(onstituent)-structure, f-structure) can be related (directly, indirectly, or via an inverse relation) to another projection.

The core idea behind Linking Theory is attractively elegant and simple to implement. However, almost every new paper dealing with linking involves some form of "tinkering" with the standard theory as articulated in Bresnan and Zaenen (1990). That is, the discussion and introduction of new data generally also entail a proposal for a different version of the linking algorithm. Alsina (1996), for example, argues for a version of linking theory which is more subject-oriented than object-oriented (as instantiated by the [±o] feature). His version also integrates the notion of Proto-Roles (Dowty 1991, Van Valin 1977) into Linking Theory. Zaenen (1993) similarly proposes an incorporation of Proto-Roles into Linking Theory, but in a manner that is very different from Alsina's. Zaenen's (1993) proposal has been taken up by several researchers, especially those looking at linking in nominal domains. These papers, as well as ones which propose incorporating Optimality Theory constraints into Linking Theory, are too numerous to mention here (see any paper on linking in the LFG On-Line Proceedings).

Computational accounts have generally shied away from implementing the complex architecture demanded by Alsina's and Butt's original analysis of syntactically formed complex predicates. The general perception among LFG computational linguists is that a-structure and its relation to f-structure and c-structure are not theoretically well enough

[2]See Butt (2006) for an overview of the development of Mapping Theory.

understood to warrant the effort of maintaining an extra projection, since extra projections are computationally expensive and are complex to maintain from the point of view of grammar engineering (Butt et al. 1999). Analyses of complex predicates thus reveal an interesting tension between computational and theoretical approaches. As discussed in the next section, the introduction of the Restriction Operator represents an attempt to resolve this tension.

11.3 The Restriction Operator

Eschewing the use of a separate projection for a-structure, Kaplan and Wedekind (1993) introduced an account of V-V complex predicates that employed the *Restriction Operator*, which manipulates f-structure representations and operates within the lexicon. However, Butt (1994) showed that this initial solution requires a large amount of undesirable lexical stipulation and cannot account for the full combinatorial power of complex predicate formation, which is a major drawback. Subsequent developments then allowed the Restriction Operator to operate within the syntax as well as the lexicon, thus avoiding the disadvantages brought up by Butt (1994). In particular, Butt, King, and Maxwell (2003) show that it is possible to implement the restriction analysis of complex predicates for Urdu in a way that seems to capture the original observations of Alsina and Butt satisfactorily.[3] In this section, we briefly present and discuss their solution in order to provide the necessary background for the discussion of the Restriction Operator as applied to morphological causatives.

As already mentioned, complex predicate formation involves the composition of two separate argument structures, those of a main predicate and a so-called "light" verb (see Butt 2003 for a discussion of this syntactic category). This complex a-structure corresponds to a single monoclausal f-structure, and this many-to-one correspondence is a hallmark of complex predicate formation. From the theoretical linking perspective, this means that an analysis must be formulated which maps a complex argument structure to a simplex f-structure. From the computational Restriction Operator perspective, the problem can be restated as one by which the f-structural subcategorization frame of the main verb needs to be manipulated in order to take the contribution of the light verb into account. From both perspectives, the essential problem is how to form a complex PRED value.

An example illustrating the problem is shown for Urdu in (2b) with

[3]Wedekind and Ørsnes (2003) show that this version of restriction can also be used to analyze analytic passive constructions in Danish.

the intransitive, unergative main verb 'cough' and the light verb 'give'. (2a) shows the simple, non-complex-predicate use of the verb 'cough'. In (2b) the main verb 'cough' combines with the light verb 'give' to form a complex argument structure by which 'Yassin' is the permitter of the action (agent), and 'Nadya' is the permissee who is allowed to perform a certain action. This means that 'Nadya' plays a dual role: that of matrix recipient/goal and that of embedded agent. In terms of the f-structure, 'Nadya' is a dative marked $\textsc{obj}_\theta$.

(2) a. nAdyA kHANs-I
Nadya.Nom cough-Perf.F.Sg
'Nadya coughed.'

b. yassIn=nE nAdyA=kO kHANs-n-E dI-yA
Yassin=Erg Nadya=Dat cough-Inf-Obl give-Perf.M.Sg
'Yassin let Nadya cough.'

An example of an Urdu permissive formed with a transitive main verb ('make') is shown in (3b) with the non-complex-predicate version in (3a). As in (2b), the permissee is in the dative.

(3) a. nAdyA=nE gHar banA-yA
Nadya=Erg house.Nom make-Perf.M.Sg
'Nadya made a house.'

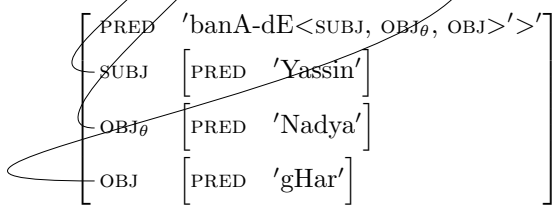
b. yassIn=nE nAdyA=kO gHar banA-n-E
Yassin=Erg Nadya=Dat house.Nom make-Inf-Obl
dI-yA
give-Perf.M.Sg
'Yassin let Nadya make a house.'

As already mentioned, Butt (1995) proposes a theoretical analysis of the permissive using a(rgument)-structure and Linking Theory. Under this analysis, which is in line with Alsina's proposals for causatives, the permissive ('give') is a light verb with three arguments. One of these arguments is an event which must be filled by the a-structure of a main verb or a complex predicate which itself is formed with a main verb. The full analysis for (3b), in which a biclausal a-structure links to a monoclausal f-structure, is shown in (4) and (5).[4]

[4]'Nadya' again plays a dual role: that of matrix recipient/goal and that of embedded agent. In terms of the f-structure, however, 'Nadya' is realized as just one grammatical function: a dative marked $\textsc{obj}_\theta$. This characteristic is one that sets complex predication apart from simple valency changing operations such as the mere deletion (i.e., passives) or addition of an argument (i.e., applicatives). Simple valency changing involves no argument merger or "fusion". From a more broadly semantic perspective, the difference is that constructions identified as complex predicates involve modification of the primary event semantics, while simple addition/deletion

(4) banA/give < ag go dE/make < ag th >>

(5)



Now consider the Urdu permissive from the perspective of a restriction analysis. From this f-structure oriented perspective, the effect of the permissive light verb is to "add" a new subject to the predication and to "demote" the main verb's subject to a dative-marked indirect object. The sample lexical entries for the light verb 'give' and the main verb 'make' from this perspective are given in (6) and (7), respectively.

(6) (↑ PRED) = 'dE<(↑ SUBJ), %PRED2>'

(7) (↑ PRED) = 'banA<(↑ SUBJ), (↑ OBJ)>'

Rather than being analyzed as a three-place predicate, the permissive *dE* 'give' is now rendered as a two-place predicate, in which the second argument is a local variable, %PRED2, which will be filled by the main verb predicate by the c-structure annotations, as discussed below. This approach avoids a complex merger of arguments (as assumed in the a-structure/linking approach) and is quite similar to Minimalist analyses of complex predicates (e.g., Butt and Ramchand 2005), which are geared towards the combination of binary structures.

Restriction allows f-structures and predicates to be manipulated in a controlled and detailed fashion. The Restriction Operator, represented as '\', can be applied to an f-structure with respect to a feature in order to arrive at a restricted f-structure which does not contain that feature (see Kaplan and Wedekind 1993 for a formal definition). In the case of the permissive, it is used to restrict out the embedded subject so that a different grammatical function can be assigned to that argument.

In order to achieve this, restriction is used as part of the f-structure annotations on phrase structure rules. The rule in (8) shows the Restriction Operator within the c-structure rule for a complex predicate. In particular, the restriction on the V node is what allows the composition of the new PRED. The annotation states that the up node (↑) comprising the complex predicate is the same as the down node (↓) comprising the main verb, except that the SUBJ of the main verb is restricted out,

operators maintain the same event semantics, but differ in the perspective on the event and their information-structural content.

as are the SUBJ and thematic object (OBJ-GO) of the complex predicate. This allows the former subject of 'make' to be identified as an OBJ-GO, via the (↑ OBJ-GO)=(↓ SUBJ) equation in (8) (cf. (10)).[5]

(8)

	(*banAnE*)	(*dIyA*)
V ⟶	V	Vlight
	↓\SUBJ\PRED=↑\SUBJ\OBJ-GO\PRED	↑=↓
	(↑ PRED ARG2)=(↓ PRED)	
	(↓ VFORM) =c inf	
	(↑ OBJ-GO)=(↓ SUBJ)	

Similarly, as the PRED is restricted out, a PRED can be constructed that is different from either of the PREDs stored in the lexicon (cf. (6) and (7)). With the permissive in (8), this is achieved via the equation on the main verb (↑ PRED ARG2)=(↓ PRED), which builds a complex PRED by assigning the main verb's (↓) PRED to the second argument of the complex predicate's PRED. ARG# provides a way of referring to specific argument positions within a PRED in the f-structure annotation and lexical rules (Crouch et al. 2006).

The restricted out f-structure of the main verb *banA* 'make' in (3b) is shown in (9). This is similar to the f-structure for the non-complex predicate in (3a) except that the case marking on the arguments and the tense and aspect information are those of the complex predicate, whose f-structure is shown in (10).

(9)
$$\begin{bmatrix} \text{PRED} & \text{'banA<SUBJ, OBJ>'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Nadya'} \\ \text{CASE} & \text{dat} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'gHar'} \\ \text{CASE} & \text{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{ASP} & \text{perf} \\ \text{TENSE} & \text{pres} \end{bmatrix} \end{bmatrix}$$

In the final complex f-structure, the predicates *dE* 'give' and *banA* 'make' have been composed. The "embedded" SUBJ 'Nadya' has been restricted out as part of the composition. This is shown in (10).

[5]The restriction of more than one feature is represented notationally by multiple instances of the restriction operator. So, the annotation in (8) indicates that both the PRED and the SUBJ are restricted out from the daughter f-structure, while the PRED, SUBJ, and OBJ-GO are restricted out from the mother f-structure. If the light verb can have a VFORM feature different from the infinitival feature required on the main verb by (8), then VFORM would also need to be restricted out. Here we show just the restriction of the grammatical functions to simplify the rule slightly.

(10)
$$\begin{bmatrix} \text{PRED} & \text{'dE<SUBJ, 'banA<OBJ-GO, OBJ>'>'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Yassin'} \\ \text{CASE} & \text{erg} \end{bmatrix} \\ \text{OBJ-GO} & \begin{bmatrix} \text{PRED} & \text{'Nadya'} \\ \text{CASE} & \text{dat} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'gHar'} \\ \text{CASE} & \text{nom} \end{bmatrix} \\ \text{TNS-ASP} & \begin{bmatrix} \text{ASP} & \text{perf} \\ \text{TENSE} & \text{pres} \end{bmatrix} \end{bmatrix}$$

Restriction thus allows f-structures and predicates to be manipulated in a controlled and detailed fashion. Given an f-structure, the Restriction Operator can be applied to the current f-structure with respect to a feature in order to arrive at a restricted f-structure which does not contain that feature. The resulting f-structure is exactly the f-structure representation argued for from a theoretical perspective by Butt (1995) and Alsina (1996), even though no representation of a-structure has been integrated into the implementation. This result eases the tension between the computational and the theoretical perspectives.

Furthermore, the analysis of the permissive complex predicate uses restriction as part of the f-structure annotations on phrase structure rules. This means that there must be a c-structure node on which to put the restriction annotation that composes the valency of the verb and creates the final f-structure. Again, this mirrors Alsina's and Butt's arguments that the complex a-structure of a complex predicate which consists of two different lexical items (i.e., N-V, V-V) has to be put together in the syntax, not the lexicon.

11.4 Causatives: Theoretical vs. Computational Perspectives

One of the very interesting aspects of Alsina's (1996, 1997) account of causatives is that he demonstrates that complex argument structure composition and the linking of thematic arguments to grammatical functions follows the same analysis regardless of whether the complex predicate is formed syntactically, as in the French examples in (11), or morphologically, as in the Chichewa examples in (12). The two languages even show the same semantic alternation with respect to causatives, even though one forms causatives morphologically and the other syntactically and even though the grammatical functions are

realized differently in both of the languages.

(11) a. Jean a fait manger les gâteaux aux enfants.
Jean has made eat the cakes to the children
'Jean made the children eat the cakes.' [French]

b. Jean a fait manger les gâteaux par les enfants.
Jean has made eat the cakes by the children
'Jean had the cakes eaten by the children.' [French]

(12) a. Nŭngu i-na-phík-ítsa kadzīdzi maûngu
porcupine SUBJ-PAST-cook-CAUS owl pumpkins
'The porcupine made the owl cook the pumpkins.' [Chichewa]

b. Nŭngu i-na-phík-ítsa maûngu kwá kádzīdzi
porcupine SUBJ-PAST-cook-CAUS pumpkins by owl
'The porcupine had the pumpkins cooked by the owl.' [Chichewa]

The alternation illustrated in (12) and (13) as been analyzed in terms of *affectedness* (illustrated in (18) and (19)); see Alsina 1996, as well as Saksena 1980, who analyzes a similar alternation in Hindi. When the animate causee is the direct object in Bantu (placed right next to the verb), or marked with *à* 'to' in French, then the causee is directly affected by the caused action, i.e., undergoes some change of state as well as being the agent that performs the caused action. In contrast, when the animate causee is marked by a 'by'-phrase, then the causee is just the agent/instrument by which the caused action took place, but no relevant change of state is assumed to have taken place.[6]

[6]This semantic distinction in terms of affectedness is one that should be explored more deeply from a semantic perspective, but this goes beyond the scope of our paper. That a difference in affectedness is involved seems to be an intuition that goes rather deep. Consider Speijer's (1886) description of a similar alternation between an accusative and an instrumental causee in Classical Sanskrit, (i) and (ii).

i. mantrapūtam carum **rājñīm** prāśayat
consecrated.Acc porridge.Acc queen.Sg.**Acc** eat.Caus.Impf.3.Sg
munisattamaḥ
best-of-ascetic.Nom
'The best of ascetics made the queen eat a consecrated porridge.'
(Kathaāsaritsāgar 9.10)

ii. tām **śvabhiḥ** khādayet rājā
Demon.F.Sg.Acc dog.Pl.**Inst** eat.Caus.Opt.3.Sg king.Nom
'Her the king should order to be devoured by dogs.'
(Mahābhārata 8.371)

If one wants to say *he causes me to do something, it is by his impulse I act*, there is room for the type [accusative causee], but if it be meant *he gets something done by me, I am only the agent or instrument through which he acts*, the instrumental is on its place. [Speijer (1886, §49)]

Important for this paper is that causatives crosslinguistically display the same a-structure and semantic properties, regardless of whether they are expressed morphologically or syntactically. Within LFG, this is expected as morphology and syntax are treated as equals in terms of the information provided to the f-structure analysis of the clause. A sample a-structure analysis of both morphological and syntactic causatives is shown in (13) (essentially Alsina's 1996 analysis).

(13) $'\text{Cause} <$ ag th $'\text{pred} <$ ag ... $>'>'$
SUBJ OBJ ...

However, LFG's linking theory as originally formulated was situated within the lexicon and so could deal with the complex a-structures of morphological causatives, but had to be modified to allow for the complex combination of a-structures within the syntax for syntactic causatives (Alsina 1996, Butt 1995; see the discussion in section 11.2).

From a computational linguistic perspective, *anything* involving complex argument composition is difficult. This is because information specified by the PRED is used to check Coherence and Completeness. Thus, operations which change the information specified by the PRED are difficult. There is, of course, a standard method of manipulating PRED values within LFG: lexical rules. Lexical rules are standardly used for simple argument deletions (passives) and renaming of grammatical functions (passives, dative shift), but our experiments with the grammar development platform XLE have shown that they are not powerful enough to deal with complex predication. Lexical rules provide ways of deleting, renaming, and adding simple arguments to a predicate, but not complex ways of merging them. Furthermore, even the addition of simple arguments to a predicate is complicated in that there must be a way of stating which argument slot is to be added and what happens to the existing arguments (for example, should the new argument be the first argument, thereby forcing all the other arguments one lower, or the last argument or the second?).

In light of the theoretical work showing the parallels between syntactic and morphological causatives, the question which arises with respect to the Restriction Operator is whether our proposals for syntactically formed complex predicates such as the Urdu permissive can also be applied to morphologically formed complex predicates, such as the Urdu causative. Morphological causatives are usually assumed to comprise a single lexical item and hence a single c-structure node. In the next section, we first present the basic data with respect to Urdu

causatives and then show that our application of the Restriction Operator can be extended straightforwardly to morphological causatives. The key lies in the structure of the sublexical component and in the morphology-syntax interface assumed in the ParGram grammars.

That our analysis can be applied to both syntactic and morphological domains is encouraging, because our analysis remains true to Alsina's original insight that syntactically and morphologically formed complex predicates essentially work the same way with respect to complex predication. The difference between analyses like Alsina's and the one outlined here lies in the fact that the Restriction Operator analysis eschews a separate a-structure projection. Some consequences of the restriction analysis will be discussed in section 11.7.

11.5 The Urdu Causative and Restriction

As in the Chichewa examples in (12), Urdu causatives are formed morphologically by affixation. Unlike in Chichewa, in Urdu there are two causatives: the *-vA* causative is usually associated with indirect causation, the *-A* causative with direct causation (Saksena 1982). In addition, there are two ways to realize the causee. Some verb classes allow only an instrumental (*=sE*) causee, some only a dative/accusative one (*=kO*), and some both. The surface realization is determined by the "affectedness" of the causee (Saksena 1982, Butt 1998). In the next section, we first present some of the basic Urdu causative data and then show how our Restriction analysis applies to them.

11.5.1 The Urdu Causative Data

There are very few basic transitive verbs in Urdu. Most transitive verbs are causatives of intransitives. Both unergatives like 'laugh' in (14) and unaccusatives like 'burn' in (15) realize the causee either as a *=kO* marked accusative if the object is specific or as unmarked nominative if the object is non-specific, as in the alternation in (15b).[7] Both of the examples are instances of the *-A* causative; using the *-vA* causative would indicate a more indirect causation.

(14) a. yassIn has-A
Yassin.M.Nom laugh-Perf.M.Sg
'Yassin laughed.'

b. nAdyA=nE yassIn=kO has-**A**-yA
Nadya=Erg Yassin=Acc laugh-**Caus**-Perf.M.Sg
'Nadya made Yassin laugh.'

[7]For details on the Urdu case marking system in general and the nominative/accusative alternation on objects in particular, see Butt and King (2005).

(15) a. jangal jal-A
jungle.M.Nom burn-Perf.M.Sg
'The jungle burned.'
b. fauj=nE jangal(=kO) jal-**A**-yA
army.F=Erg jungle.M.Nom(=Acc) burn-**Caus**-Perf.M.Sg
'The army burned (the) jungle.'

Example (16) shows the causativization of an agentive transitive.[8] Causativization of a typical agentive transitive licenses an instrumental causee, as shown in (16b). In contrast to the intransitive pattern in (14) and (15), a *kO* marked causee is ungrammatical.

(16) a. yassin=nE paodA kAT-A
Yassin=Erg plant.M.Nom cut-Perf.M.Sg
'Yassin cut the plant.'
b. nAdyA=nE yassIn=sE/*kO paoda
Nadya=Erg Yassin=Inst/Dat plant.M.Nom
kaT-**A**-yA
cut-**Caus**-Perf.M.Sg
'Nadya had the plant cut by Yassin.'

In Urdu, the case clitic *kO* functions both as an accusative and a dative (homophony). For an extensive discussion of the patterns and distribution of dative and accusative case, see Butt and King (2005) and Mohanan (1994). With causatives, the distribution works as follows. When *kO* marks the only object in the clause, it functions as an accusative and participates in the specificity alternation, i.e., its realization is optional and marks specificity as in (15b). When there is another object in the clause, it marks an OBJ_θ and functions as a dative (i.e., it does not participate in the specificity alternation). Another way to tell the difference between accusative *kO* and dative *kO* is that accusatives can be passivized while datives cannot.

Not all transitives work as in (16). For example, with the class of *ingestive* verbs (e.g., *drink*, *eat*, *learn*, *read*) the agent is always seen as being affected by the action and so only a *kO* marked causee is allowed as in (17).

(17) a. yassin=nE kHAnA kHa-yA
Yassin=Erg food.Nom eat-Perf.M.Sg
'Yassin ate food.'

[8] As already mentioned, transitives are usually related to an intransitive verb root. In (16), the transitive *kAT* 'cut' is related to an intransitive verb root *kaT* 'be cut' via "vowel strengthening". The causativized version of (16a) is shown in (16b). The *-A/-vA* causative is added to the intransitive form of the root. The precise morphophonological factors involved in causation remain a subject of investigation.

b. nAdyA=nE yassIn=kO/*sE kHAnA kHil-**A**-yA
Nadya=Erg Yassin=Dat/Inst food.Nom eat-**Caus**-Perf.M.Sg
'Nadya had Yassin eat (fed Yassin).'

With some other verbs, both *kO* and *sE* marked causees are allowed. An example is shown in (18). (Other members of this class are *read, write, sing.*) These verbs allow a semantic alternation that is similar to the one discussed with respect to the French and Chichewa examples in (11) and (12). When the causee is marked with *kO*, the causee is interpreted as affected, as in (18); when the causee is instrumental, as in (19), it is interpreted as an agentive, non-affected causee.[9]

(18) anjum=nE saddaf=kO masAlA
Anjum.F=Erg Saddaf.F=Dat spice.M.Nom
cakH-**vA**-yA
taste-**Caus**-Perf.M.Sg
'Anjum had Saddaf taste the seasoning.'

(19) anjum=nE saddaf=sE masAlA(=kO)
Anjum.F=Erg Saddaf.F=Inst spice.M.Nom(=Acc)
cakH-**vA**-yA
taste-**Caus**-Perf.M.Sg
'Anjum had the seasoning tasted by Saddaf.'

[9]A reviewer asks why one could not interpret 'Saddaf' as causee in (18) (caused Saddaf to taste the seasoning), but the 'spice/seasoning' as causee in (19) (caused the seasoning to be tasted by Saddaf). This is exactly the analysis proposed by Alsina, as illustrated in (i) and (ii) for Chichewa (Alsina and Joshi 1991).

i. phik-itsa 'cause < ag pt 'cook< ag pt >'> (OBJ CAUSEE)
cook-CAUS

ii. phik-itsa 'cause < ag pt 'cook< ag pt >'> (OBL CAUSEE)
cook-CAUS

This is a possible analysis, as is the idea coming out of Relational Grammar and Government-Binding that the oblique causee is derived by first demoting the embedded agent via passivization and then combining the argument structures (so that one gets an instrumental, oblique causee). The passivization idea is not satisfactory, given that there is nothing passive about the causative. Alsina's alternative in terms of Parameters on argument fusion raises the question whether there are any constraints on argument fusion: can any thematic argument in the matrix a-structure potentially combine with any argument in the embedded a-structure? Observations about complex predicates crosslinguistically indicate that argument merger/fusion acts much like control: the lowest item in the matrix structure is generally identified with the highest argument of the embedded structure. If one adheres to this generalization, then the causee is always 'Saddaf'. Semantically, this would make more sense since it is difficult to act upon the seasoning to get the caused action of 'tasting' done. See Butt (1998) for discussion.

The data here cover the basic patterns found in Urdu. We have not discussed differences between *-A* and *-vA* causatives, which tend to signal "direct" vs. "indirect" causation, but the differences are subtle. Furthermore, not all verbs allow an *-A* causative and not all verbs allow a *-vA* causative. See Saksena (1982) and Butt (1998) for more details on patterns and analyses. However, these further complexities are not germane to the point addressed in this paper: can the Restriction Operator in principle be used to analyze morphological causation?

11.5.2 F-structures for Causatives

Let us take the most complex example in (18)–(19). The basic f-structure for the non-causative version (20a) is shown in (20b). The f-structures in (21) and (22) give the representations for the causatives in (18) and (19), respectively.

(20) a. saddaf=nE masAlA cakH-A
Saddaf.F=Erg spice.M.Nom taste-Perf.M.Sg
'Saddaf tasted the seasoning.'

b. $\begin{bmatrix} \text{PRED} & \text{'taste<SUBJ, OBJ>'} \\ \text{SUBJ} & [\ \text{PRED 'Saddaf'}\ ] \\ \text{OBJ} & [\ \text{PRED 'seasoning'}\ ] \end{bmatrix}$

(21) $\begin{bmatrix} \text{PRED} & \text{'Cause<SUBJ, 'taste<OBJ-GO, OBJ>'>'} \\ \text{SUBJ} & [\ \text{PRED 'Anjum'}\ ] \\ \text{OBJ-GO} & [\ \text{PRED 'Saddaf'}\ ] \\ \text{OBJ} & [\ \text{PRED 'seasoning'}\ ] \end{bmatrix}$

(22) $\begin{bmatrix} \text{PRED} & \text{'Cause<SUBJ, 'taste<OBL-AG, OBJ>'>'} \\ \text{SUBJ} & [\ \text{PRED 'Anjum'}\ ] \\ \text{OBL-AG} & [\ \text{PRED 'Saddaf'}\ ] \\ \text{OBJ} & [\ \text{PRED 'seasoning'}\ ] \end{bmatrix}$

From the perspective of the Restriction Operator what is needed is something which "adds" a subject argument and "demotes" the argument of the main (embedded) verb to an OBL-AG, OBJ-GO or an OBJ in the case of intransitives, as illustrated in (23) and (24) for the examples in (14).

(23) $\begin{bmatrix} \text{PRED} & \text{'laugh<SUBJ>'} \\ \text{SUBJ} & [\ \text{PRED 'Yassin'}\ ] \end{bmatrix}$

(24) $\begin{bmatrix} \text{PRED} & \text{'Cause<SUBJ, 'laugh<OBJ>'>'} \\ \text{SUBJ} & [\ \text{PRED 'Nadya'}\ ] \\ \text{OBJ} & [\ \text{PRED 'Yassin'}\ ] \end{bmatrix}$

So, in parallel to the analysis for the permissive, one would like to postulate a lexical entry like the one in (25) for the causative morphemes.

(25) (↑ PRED) = 'CAUSE<(↑ SUBJ), %PRED2>'

However, given that the causative morphemes are part of the morphology, it may at first not be clear how this can be done (or whether this *should* be done). In the next section we therefore turn to a brief discussion of the morphology-syntax interface as implemented within the ParGram grammars and then show how the writing of lexical entries as in (25) is straightforward and unproblematic.

11.5.3 Causative Morphology and Morphology-Syntax Interface

Morphological analysis is integrated within the ParGram grammars via the finite-state methods described in Beesley and Karttunen (2003). In finite-state morphologies, morphemes are represented more or less abstractly (depending on the needs of the grammar)[10] and are arranged into finite-state continuation classes of the type shown in (26) for the Urdu verb *has* 'laugh'.

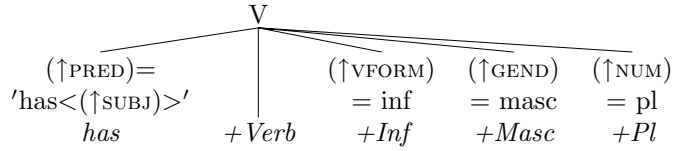
(26)
```
LEXICON Verbs
has+Verb:has          TnsAsp;

LEXICON TnsAsp
                      Imperfect;
                      Perfect;
                      Infinitive;

LEXICON Infinitive
+Inf:n                GendInf;

LEXICON GendInf
+Fem+Sg:I             #;
+Masc+Sg:A            #;
+Masc+Sg+Obl:E        #;
+Masc+Pl:E            #;
```

[10]In the Urdu grammar, we have so far represented the morphemes quite concretely. However, it is possible to posit more abstract representations in order to deal with allomorphy and phonological processes such as vowel harmony or assimilation. It is also possible to deal with complex morphology such as that of the Arabic templatic stem realization; see Beesley and Karttunen (2003) and Karttunen (2003) for an in-depth discussion.

The extract from the finite-state morphology in (26) shows the association of morphemes with abstract "tags". The surface form *has* is associated with a stem *has* that is marked as a verb (+Verb). The finite-state morphology specifies that verbs must have Tense/Aspect morphology, as indicated in the definition of Verbs which refers to the continuation class TnsAsp. This can take several forms, e.g. Imperfect, Perfect or Infinitive. To construct or parse an infinitive form such as *hasnE*, we follow the continuation class that points to "Infinitive" through TnsAsp, where the infinitive marker *-n-* is found. This is associated with an abstract +Inf tag. From here the finite-state morphology points to the paradigm for gender and number marking that is appropriate for infinitives. As indicated by the *E* entries under GendInf, the form *hasnE* could be masculine singular oblique or it could be masculine plural.

The interface to the syntax uses abstract tags associated with the surface morphemes (see Butt et al. 1999, Butt and Sadler 2003, Kaplan et al. 2004). Essentially, the abstract tags are parsed via sublexical phrase structure rules. As part of this module, the abstract tags are also annotated with f-structure information, thus allowing information to flow into the syntactic analysis. As a concrete example, one of the possible sublexical trees for *hasnE* is shown in (27).[11]

(27)



The information necessary for a Restriction analysis can be associated straightforwardly with the causative morphology. The morphemes *-A* and *-vA* can be associated with an abstract +Caus tag. In the Urdu grammar, we have assigned the tags +Caus1 and +Caus2 to *-A* and *-vA*, respectively, in order to capture the differing semantic/pragmatic information associated with the two different morphemes.

The morphological analysis of the causativized perfect version of the verb *cakH* 'taste', for example, looks as in (28b,c) (cf. (18) and (19)).

(28) a. cakHA ⇔ cakH +Verb +Perf +Masc +Sg
b. cakHAyA ⇔ cakH +Verb +Caus1 +Perf +Masc +Sg

[11] Urdu allows subject, object and default (=no) agreement. Infinitives represent a special case because they agree only if they are acting as an object or a subject of a verb (analyzed as verbal nouns in this case: Butt 1995). As such the agreement statements are quite complex and we have left them out of the representation in (27) for ease of exposition.

c. cakHvAyA ⇔ cakH +Verb +Caus2 +Perf +Masc +Sg

The lexical entry in (25) can be associated with the +Caus tags, as shown in (29) for +Caus1.

(29) +Caus1 (↑ PRED) = 'Cause<(↑ SUBJ), %PRED2>'

The lemma and morphological tags in (28) can be parsed by sublexical c-structure rules (Kaplan et al. 2004), as illustrated in (27). The sublexical rules are formally identical to standard c-structure rules and hence can be annotated in the same way as more traditional c-structure rules, such as those used in the formation of the Urdu permissive.[12]

In the morphological causative, the +Caus tags thus provide a phrase-structure locus for the Restriction Operator. The causative annotated sublexical c-structure rule is shown in (30). This rule states that the main verb (↓) is identical to that of the causative verb (↑) except that the SUBJ and the original PRED are restricted out (↓\SUBJ\PRED=↑\SUBJ\PRED). The subject of the main verb is identified with the OBJ-GO, OBJ, or OBL of the causative verb ((↓SUBJ)={ (↑OBJ-GO) | (↑OBJ) | (↑OBL) }). Which of these grammatical functions is chosen depends on the affectedness of the causee, the type of causative, and the lexical semantics of the main verb.[13] Finally, the PRED of the main verb is assigned to the second argument of the causative predicate ((↑ PRED ARG2)=(↓ PRED)), just as in the permissive.

(30) V → Vstem CauseMorph
↓\SUBJ\PRED=↑\SUBJ\PRED ↑=↓
(↓SUBJ)= { (↑OBJ-GO)
| (↑OBJ)
| (↑OBL) }
(↑ PRED ARG2)=(↓ PRED)

The restriction analysis thus treats morphologically and syntactically formed complex predicates the same, as was the case in Alsina's (1997) analysis. In addition, the morphology-syntax interface is well-understood and cleanly formulated.[14] In the next section, we further

[12]These rules do not violate lexical integrity since they are located in the sublexical domain. The sublexical rules shown in (27) are flat. However, if necessary, one can have a more configurational sublexical tree (i.e., to indicate hierarchical relations between morphemes).

[13]In the current implementation, simple lexical semantics such as unaccusative vs. unergative verbs are encoded in f-structure (they could be encoded at s(emantic)-structure, but the current implementation does not include this projection). The causative rule can then refer to this feature.

[14]Karttunen (2003) shows that Realizational Morphology (Stump 2001), which has been extensively argued to be suitable for LFG (LFG02 workshop on morphology, Sadler and Spencer 2005) is finite-state equivalent.

explore the effects of our analysis by examining interactions between morphologically and syntactically formed complex predicates and periphrastic passives.

11.6 Interactions with the Restriction Analysis of Causatives

Urdu allows productive interactions between different types of complex predicates and between complex predicates and passives. In this section, we first look at some of the interactions between different types of complex predicates (permissives and causatives) in order to see whether the application of the Restriction Operator in different parts of the grammar (syntax and morphology) causes problems. As section 11.6.1 shows, this interaction works robustly and unproblematically. In section 11.6.2 we examine the interaction between the Restriction Operator and lexical rules by looking at passive causatives.

11.6.1 Interaction of Causatives and Complex Predicates

One syntactically formed complex predicate can interact with another one, as illustrated by (31) in which the causative version of 'laugh' acts as the main verb in a permissive complex predicate. The f-structure representation of (31) is shown in (32).

(31) anjum nE nAdyA kO yassIn kO has-A-n-E
Anjum=Erg Nadya=Dat Yassin=Acc laugh-Caus-Inf-Obl
dI-yA
give-Perf.M.Sg
'Anjum let Nadya make Yassin laugh.'

(32)
$$\begin{bmatrix} \textsc{pred} & \text{'give}\langle \textsc{subj}, \text{'Cause}\langle \textsc{obj-go}, \text{'laugh}\langle \textsc{obj}\rangle'\rangle'\rangle' \\ \textsc{subj} & [\ \textsc{pred}\ \text{'Hassan'}\ ] \\ \textsc{obj-go} & [\ \textsc{pred}\ \text{'Nadya'}\ ] \\ \textsc{obj} & [\ \textsc{pred}\ \text{'Yassin'}\ ] \end{bmatrix}$$

In this case, one complex PRED is built within the morphological component, namely the combination of *Cause* and *has* 'laugh'. This combination results in the complex PRED 'Cause<SUBJ,'laugh<OBJ>'>', which is then combined with the permissive light verb *dE* 'give' to yield the PRED shown in (32). This interaction between Restriction Operators situated in different parts of the grammar is completely unproblematic.

11.6.2 Interaction of Causative and Passive

Next we consider the interaction of the passive with the causative. In the Urdu ParGram grammar, passives are treated via a standard pas-

sive lexical rule by which the active subject is identified as the passive OBL-AG and the active object is identified as the passive SUBJ. This passive lexical rule is triggered by a periphrastic construction formed with an auxiliary based on the verb 'go'. The main verb must carry "perfect" morphology. An example of a passive causative is shown in (33b). Note that the causative applies first, creating a transitive verb from the intransitive *has* 'laugh', and then the passive applies.

(33) a. nAdyA=nE yassIn=kO has-A-yA
Nadya=Erg Yassin=Acc laugh-Caus-Perf.M.Sg
'Nadya made Yassin laugh.'

b. yassIn (nAdyA=sE) has-A-ya ga-yA
Yassin.Nom Nadya=Inst laugh-Caus-Perf.M.Sg go-Perf.M.Sg
'Yassin was made to laugh (by Nadya).'

The f-structure representation for (33a) is shown in (34a). The complex PRED is a combination of the main verb *has* 'laugh' and the causative. The f-structure representation for (33b) is shown in (34b). In (34b) the causative verb undergoes passive just as an underlyingly transitive verb would have.

(34) a. Causative:

$$\begin{bmatrix} \text{PRED} & \text{'Cause<SUBJ,'laugh<OBJ>'>'} \\ \text{SUBJ} & [\ \text{PRED 'Nadya'}\ ] \\ \text{OBJ} & [\ \text{PRED 'Yassin'}\ ] \end{bmatrix}$$

b. Causative + Passive:

$$\begin{bmatrix} \text{PRED} & \text{'Cause<OBL-AG,'laugh<SUBJ>'>'} \\ \text{OBL-AG} & [\ \text{PRED 'Nadya'}\ ] \\ \text{SUBJ} & [\ \text{PRED 'Yassin'}\ ] \\ \text{PASSIVE} & + \end{bmatrix}$$

From a theoretical standpoint, applying the lexical rule based passive to the causative is straightforward. However, implementing this interaction using a combination of the restriction operator on the sublexical rules for the causative and a lexical rule for the passive proved challenging and highlights some interesting issues that would otherwise have remained unexplored. For example, at one stage in the development of the Urdu grammar, in the analysis for (33b), the subject of the causative had been correctly realized as the OBL-AG but the object had not been realized as the SUBJ in the final, restricted f-structure. This type of structure results from not sufficiently constraining the lexical rules to apply only to the final, non-restricted structure. This is particularly apparent in that there was no SUBJ for the final f-structure. Although Urdu obeys the Subject Condition, the XLE implementation

of LFG does not universally impose a Subject Condition in order to allow for languages which have truly subjectless constructions (see Babby 1993 on Russian adversity impersonals). The subjectless structures incorrectly obtained for Urdu during the development of the causative analysis highlight the need to carefully state the Subject Condition and its interaction with the Restriction Operator in such a way as to avoid producing subjectless constructions from ones with subjects.

Currently, we are trying to fully understand the interaction between lexical rules and the Restriction Operator. We are exploring whether this is a grammar engineering issue in the sense that we have not found a robust enough statement of the interaction, or whether this is a fundamental implementational and theoretical issue in that the formal underpinnings of the interaction between restriction and lexical rules need to be better understood and reimplemented. Understanding the interaction is not trivial because the Restriction Operator is a complex and powerful method of manipulating PRED values. Because of the possibility of building complex PREDs via the Restriction Operator, the checks for Completeness, Coherence and the Subject Condition have to be done differently. Indeed, Alsina (1996) addresses this issue at length from an a-structure perspective. Alsina has to address this issue because his (and Butt's 1995) analysis of complex predicates assumed a complicated projection architecture involving a-structure. Interestingly, it seems that even when an overt use of a-structure representations is avoided, i.e., by allowing a composition of PRED values within the f-structure, deep architectural questions arise. This is because the essential problem, namely the composition of PREDs, has not gone away: it has simply been moved to a different part of the grammar.

11.7 Conclusions: Morphology and LFG Architecture

This paper has discussed different methods of dealing with complex predication by looking at the interaction between morphologically and syntactically formed complex predicates and passives. We have shown that complex morphological valency changing operations such as the morphological causative can be analyzed using the Restriction Operator. This allows for the seamless integration of the causatives with complex valency changing operations in Urdu that are situated in the syntax. The key to the formal integration of this analysis is the interaction of the morphology with the syntax, in particular in the domain of the annotated phrase-structure rules.

However, while the Restriction Operator allows the formation of complex PREDs according to the analyses presented in Alsina (1996),

Butt (1995) and Mohanan (1994), it also raises further theoretical and implementational issues. Beyond the issue of how to check for Coherence, Completeness and the Subject Condition, questions about the status of the Principle of Direct Syntactic Encoding are also raised.[15] That is, one of the principles of LFG was to avoid the overly powerful transformation architecture of Transformational Grammar (and its successors) and to not allow for the change of grammatical functions in the syntax (lexical rules operate on lexical representations).

The introduction of the Restriction Operator as first proposed by Kaplan and Wedekind (1993) respected this principle: the Restriction Operator was only applied within the lexical domain. However, as Butt (1994) showed, this domain of application could not do justice to the syntactically productive nature of complex predicate formation. With Butt, King and Maxwell's (2003) application of the Restriction Operator within the syntax, violations of the Principle of Direct Syntactic Encoding become eminently possible.

The advantage of the a-structure approach therefore seems to be that it assembles pieces of the complex predicate at the level of a-structure and only maps this information to f-structure in a very final step, thus avoiding the direct manipulation of grammatical functions. From a computational point of view, however, this means that precise well-formedness conditions for a-structure must be formulated and implemented. It is not the case that all a-structures can be combined in all ways: a-structure composition is governed by strict constraints. However, our theoretical understanding of these constraints remains limited and therefore the computational rendering of them is difficult. In addition, some well-formedness checks, like Coherence and Completeness, have to be performed both at f-structure and at a-structure, thus duplicating the efforts at well-formedness checking (see also Dalrymple 2001 on how glue semantics accounts for Completeness and Coherence). Alsina (1996) therefore proposes to abandon checking at f-structure and to perform well-formedness checks only at a-structure (see also Alsina, Mohanan and Mohanan 2005).

The crux of the matter is therefore how to deal with complex valency changing phenomena that go beyond the simple addition (applicatives), deletion (passives) or renaming (dative shift) of arguments/grammatical functions. The proper treatment of derivational morphology within LFG is a related issue. As with valency changing phenomena, the metaphors linguists use when talking about derivational morphology

[15]We would like to thank Joan Bresnan for pointing this out and engaging in on-going discussions on this issue with us.

are ones in which some original information (e.g., the lexical information associated with a verb) is changed into some other kind of information via the addition of derivational morphology (e.g., a nominalizer like *-ion*). Again, these are transformational metaphors and given the close interaction between derivational morphology and syntactic encoding, some of the same issues arise as with complex predicates.

Here we see the integration of finite-state morphologies into LFG grammars as an advantage. Conceptually, the finite-state approach provides a clean and well-defined interface to larger grammatical processes. However, little has been done until now to model a theoretically interesting approach to derivational morphology within the LFG grammars (note that Stump's 2001 theory of morphology, which has been advocated for adoption within LFG, is confined to inflectional morphology). As morphological causatives represent a type of derivational morphology, we feel that this paper is taking a first step in that direction and is already uncovering interesting architectural issues. In particular, it seems crucial to us that any further exploration of these issues take into account both theoretical and computational perspectives.

Acknowledgments

Over the years since the completion of our dissertations in the early nineties, it has been a real pleasure to work together with Ron Kaplan and to profit from his endless patience and his formal insights and instincts. His linguistic instincts are often in complete opposition to ours, which means that our exposure to his sceptical perspective has served to hone our theoretical argumentation in a way that very few other challenges have been able to do. The result has been an extremely productive give-and-take between theory and computation, a give-and-take that we hope is reflected in this paper.

References

Alsina, Alex and Smita Joshi. 1991. Parameters in causative constructions. In L. M. Dobrin, L. Nichols, and R. M. Rodriguez, eds., *Papers from the 27th Regional Meeting of the Chicago Linguistic Society (CLS)*, pages 1–16. Chicago, IL.

Alsina, Alex. 1993. *Predicate Composition: A Theory of Syntactic Function Alternations*. Ph.D. thesis, Stanford University.

Alsina, Alex. 1996. *The Role of Argument Structure in Grammar: Evidence from Romance*. Stanford, CA: CSLI Publications.

Alsina, Alex. 1997. A theory of complex predicates: evidence from causatives in Bantu and Romance. In A. Alsina, J. Bresnan, and P. Sells, eds., *Complex Predicates*, pages 203–246. Stanford, CA: CSLI Publications.

Alsina, Alex, Tara Mohanan, and KP Mohanan. 2005. How to get rid of the COMP. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2005 (LFG'05)*, pages 21–41. Bergen, Norway: CSLI Online Publications.

Babby, Leonard. 1993. Adversity impersonal sentences in Russian. In S. Avrutin, S. Franks, and L. Progovac, eds., *Formal Approaches to Slavic Linguistics: The MIT Meeting*, pages 25–67. Ann Arbor, MI: Michigan Slavic Publications.

Beesley, Kenneth and Lauri Karttunen. 2003. *Finite State Morphology*. Stanford, CA: CSLI Publications.

Bresnan, Joan and Annie Zaenen. 1990. Deep unaccusativity in LFG. In K. Dziwirek, P. Farrell, and E. Mejías-Bikandi, eds., *Grammatical Relations: A Cross-Theoretical Perspective*, pages 45–57. Stanford, CA: CSLI Publications.

Butt, Miriam. 1994. Machine translation and complex predicates. In H. Trost, ed., *Proceedings of the Conference Verarbeitung natürlicher Sprache (KONVENS'94)*, pages 62–71. Vienna, Austria: Springer Verlag.

Butt, Miriam. 1995. *The Structure of Complex Predicates in Urdu*. Stanford, CA: CSLI Publications.

Butt, Miriam. 1998. Constraining argument merger through aspect. In E. Hinrichs, A. Kathol, and T. Nakazawa, eds., *Complex Predicates in Nonderivational Syntax*, pages 73–113. New York, NY: Academic Press.

Butt, Miriam, Tracy H. King, María-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. Stanford, CA: CSLI Publications.

Butt, Miriam, Helge Dyvik, Tracy H. King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar project. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), Workshop on Grammar Engineering and Evaluation*, pages 1–7. Taipei, ROC.

Butt, Miriam. 2003. The light verb jungle. In G. Aygen, C. Bowern, and C. Quinn, eds., *Harvard Working Papers in Linguistics: Papers from the GSAS/Dudley House Workshop on Light Verbs*, vol. 9, pages 1–49. Cambridge, MA: Harvard University.

Butt, Miriam, Tracy H. King, and John T. Maxwell, III. 2003. Complex predicates via restriction. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2003 (LFG'03)*, pages 92–104. Albany, NY: CSLI Online Publications.

Butt, Miriam and Louisa Sadler. 2003. Verbal morphology and agreement in Urdu. In U. Junghans and L. Szucsich, eds., *Syntactic Structures and Morphological Information*, pages 57–100. Berlin, Germany: Mouton de Gruyter.

Butt, Miriam and Tracy H. King. 2005. The status of case. In V. Dayal and A. Mahajan, eds., *Clause Structure in South Asian Languages*, pages 153–198. Berlin, Germany: Springer Verlag.

Butt, Miriam and Gillian Ramchand. 2005. Complex aspectual structure in Hindi/Urdu. In N. Ertischik-Shir and T. Rapoport, eds., *The Syntax of Aspect*, pages 117–153. Oxford, United Kingdom: Oxford University Press.

Butt, Miriam. 2006. *Theories of Case*. Cambridge, United Kingdom: Cambridge University Press.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, John T. Maxwell, III, and Paula S. Newman. 2006. XLE Documentation. Palo Alto Research Center.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. San Diego, CA: Academic Press.

Dowty, David. 1991. Thematic proto-roles and argument selection. *Language* 67(3):547–619.

Kaplan, Ronald M. and Jürgen Wedekind. 1993. Restriction and correspondence-based translation. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL'93)*, pages 193–202. Utrecht, The Netherlands.

Kaplan, Ronald M., John T. Maxwell, III, Tracy Holloway King, and Richard Crouch. 2004. Integrating finite-state technology with deep LFG grammars. In *Proceedings of the 16th European Summer School on Logic, Language and Information (ESSLLI'04), Workshop on Combining Shallow and Deep Processing for NLP*. Nancy, France.

Karttunen, Lauri. 2003. Computing with realizational morphology. In A. Gelbukh, ed., *Computational Linguistics and Intelligent Text Processing*, pages 205–216. Berlin, Germany: Springer Verlag.

Mohanan, Tara. 1994. *Argument Structure in Hindi*. Stanford, CA: CSLI Publications.

Sadler, Louisa and Andrew Spencer, eds. 2005. *Projecting Morphology*. Stanford, CA: CSLI Publications.

Saksena, Anuradha. 1980. The affected agent. *Language* 56(4):812–826.

Saksena, Anuradha. 1982. *Topics in the Analysis of Causatives with an Account of Hindi Paradigms*. Berkeley, CA: University of California Press.

Speijer, J. S. 1886. *Sanskrit Syntax*. Delhi, India: Motilal Banarsidass. Republished 1973.

Stump, Gregory. 2001. *Inflectional Morphology*. Cambridge, United Kingdom: Cambridge University Press.

Van Valin, Robert D. 1977. *Aspects of Lakhota Syntax*. Ph.D. thesis, University of California, Berkeley.

Wedekind, Jürgen and Bjarne Ørsnes. 2003. Restriction and verbal complexes in LFG: A case study for Danish. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2003 (LFG'03)*, pages 424–450. Albany, NY: CSLI Online Publications.

Zaenen, Annie. 1993. Unaccusativity in Dutch: Integrating syntax and lexical semantics. In J. Pustejovsky, ed., *Semantics and the Lexicon*, pages 129–161. Dordrecht, The Netherlands: Kluwer Academic Publishers.

12

A (Discourse-)Functional Analysis of Asymmetric Coordination

Anette Frank

12.1 Introduction

12.1.1 Coordination for economic linguistic realisation

Coordination is an excellent syntactic means of efficient and economic linguistic realisation. The contrasts in (1) and (2) exemplify the successful avoidance of redundancy in overt linguistic expression by use of an appropriate coordination construction.

(1) a. The hunter went into the forest and *he* caught a rabbit.
    b. The hunter went into the forest and caught a rabbit.

(2) a. Fred knows Rome and *Fred* loves *Rome*.
    b. Fred knows and loves Rome.

(1b) and (2b) are instances of standard constituent coordination — VP and V coordination. As illustrated in (3), the subcategorisation requirements of the coordinated heads are not fulfilled within the individual conjuncts. Instead, the *unique* arguments realised outside the coordinate structure need to be *distributed* over the conjuncts, to satisfy the subcategorisation requirements of the individual coordinated heads.

(3) a. The hunter [[$_{VP}$ went into the forest] and [$_{VP}$ caught a rabbit]].

    b. Fred [[$_{V}$ knows] and [$_{V}$ loves]] Rome.

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

Thus, redundancies that are avoided in coordination constructions lead — prima facie — to violations of basic syntactic principles, most prominently, agreement and subcategorisation. Modern syntactic theories provide special mechanisms to apply in coordination constructions to account for their 'reductionist' properties, while excluding ungrammatical constructs. Constituent coordination is in this sense well understood, and successfully handled by all major syntactic frameworks.[1]

12.1.2 The challenge of asymmetric coordination

Asymmetric coordination is a long-standing puzzle in syntax. First discussed by Höhle (1983a) and Wunderlich (1988), the famous *hunter (Jäger)* and *bailiff (Gerichtsvollzieher) sentences* have engendered a wealth of analyses in nearly all grammatical frameworks.

Asymmetric coordination is challenging for any type of syntactic theory, as it seems to violate basic principles of accessibility (or distribution) as established for cases of regular constituent coordination: the subject of the non-initial conjunct is not overtly realised, but interpreted as bound by the subject of the initial conjunct (hence 'subject gap'). The latter, however, is realised in a middle field position, and is thus — under standard analyses of constituent coordination — not accessible from within the second conjunct.

Two types of asymmetric coordination are illustrated below: The so-called *SGF coordination (Subject Gap in Finite/Fronted constructions)* introduced by Höhle (4), and the related, special case of verb-last and verb-first (VL/VF) coordination discussed by Wunderlich (5).[2] Both coordination types are very frequent and not restricted to specific registers or style.[3]

(4) a. In den Wald ging *der Jäger* und fing einen Hasen.
Into the forest went the hunter and caught a rabbit
'The hunter went into the forest and caught a rabbit.'

b. Nimmt *man* den Deckel ab und rührt die Füllung um,
Takes one the lid off and stirs the contents Prt,
steigen Dämpfe auf.
rise fumes Prt
'If one takes the lid off and stirs the contents, fumes will rise.'

[1] This does not hold for the wide variety of so-called non-constituent coordinations, including gapping, conjunction reduction, ellipsis, etc. (cf. Crysmann 2006).

[2] We will use classical examples from previous work in Höhle (1983a), Wunderlich (1988), Büring and Hartmann (1998), Kathol (1999), and avoid repeated glossing.

[3] See Frank (2001) for a corpus study on asymmetic coordination.

(5) Wenn *Du* in ein Kaufhaus kommst und (Du) hast kein Geld,
If you into a shop come and you have no money
kannst Du nichts kaufen.
can you nothing buy
'If you enter a shop and (you) don't have any money, you can't buy anything.'

The second conjunct in these constructions is always verb second (V2). Since German is verb final, this is most evident in (4b), with a separable prefix in the VP head position, and also holds for (4a) and (5), where the finite verb in V2 position precedes the VP internal constituents. Examples like (5) are thus coordinations of a verb last and a V2 construction, with an optional subject gap in the second conjunct.

SGF constructions have been analysed as asymmetrically embedded constituents (Wunderlich 1988, Höhle 1990, Heycock and Kroch 1993, Büring and Hartmann 1998, among others) or symmetric conjuncts (Steedman 1990, Kathol 1995, 1999). Asymmetric analyses are problematic as they involve extraction asymmetries, or an analysis of coordination as adjunction. Symmetric analyses assume special licensing conditions that are not independently motivated. The word order conditions of Kathol (1999) are especially lacking independent motivation.

12.1.3 A multi-factorial analysis of asymmetric coordination

We develop a multi-factorial LFG analysis of asymmetric coordination constructions — SGF and VL/VF coordination (cf. (4) and (5)) — building on independently motivated principles of correspondence between c–structure, f–structure, and semantic representation. Asymmetric coordination is analysed as symmetric coordination in c–structure. Binding of the (prima facie) inaccessible subject of the first conjunct is enabled, at the level of f–structure, by asymmetric projection of a *grammaticalised discourse function* (GDF), a TOPIC, FOCUS or SUBJ function (see Bresnan 2001). Asymmetric GDF projection is motivated by relating the discourse-functional properties of asymmetric coordination to well-known discourse subordination effects of modal subordination (cf. Roberts 1989, Frank 1997). Our analysis is in accordance with the *Principle of Economy of Expression* (Bresnan 2001), explains the mysterious word order constraints of asymmetric coordination, and accounts for some puzzling properties of scope.

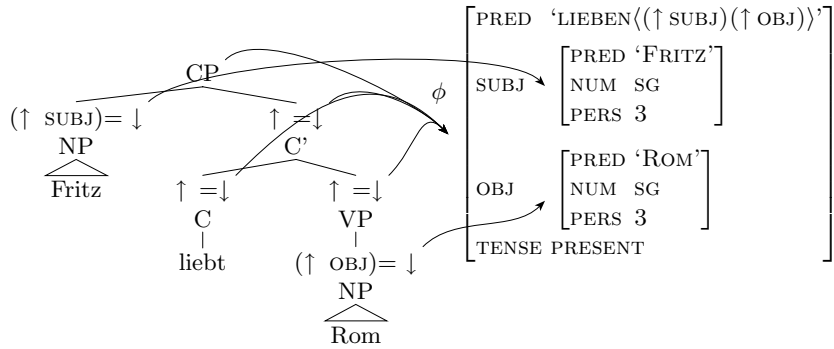


FIGURE 1 C– and f–structure for *Fritz liebt Rom (Fritz loves Rome)*

12.1.4 Overview

The remainder of this paper is structured as follows. Section 2 introduces the basic analysis of constituent coordination in LFG. Section 3 characterises the challenge of asymmetric coordination constructions within the LFG treatment of coordination. We give an overview of the syntactic and semantic properties of SGF and VL/VF coordinations, to be accounted for by any successful analysis of asymmetric coordination. Section 4 reviews some typical approaches to SGF coordination, focusing in particular on the symmetric analysis of Kathol (1999). In Section 5 we develop our multi-factorial analysis of asymmetric coordination, which accounts for the special discourse-functional properties of asymmetric coordination. Section 6 presents some conclusions.

12.2 Coordination in LFG

12.2.1 Multi-level syntactic representation

Lexical-Functional Grammar provides two levels of syntactic representation: c– and f–structure. C–structure is a tree representation that encodes constituency and word order, while f–structure is an attribute-value representation that encodes functional-syntactic properties, in particular grammatical functions and morphosyntactic information.

C– and f–structure are set into correspondence by way of a functional mapping, the ϕ–correspondence. It is encoded by functional annotations on c–structure nodes, which define the correspondence between c–structure nodes and their associated representation in the f–structure. In Figure 1, the annotation (↑ OBJ)=↓ on the VP-internal NP node requires that the f–structure projected by the NP node — which contains the feature-value pair PRED='ROM' — plays the role of

OBJ in the f-structure that corresponds to its mother VP node.

Both representation levels are subject to principles of wellformedness. The c–structure must obey principles of X-bar theory for lexical and functional categories (Bresnan 2001). Grammatical functions in f–structure are classified as argument vs. non-argument functions. Argument functions need to be subcategorised by their local predicator (PRED) (*Coherence Principle*), and vice versa, all argument functions subcategorised by a predicator need to be realised (*Completeness Principle*). Finally, the *Principle of Economy of Expression* states that of all valid c–/f–structure representations only those that are maximally economic are considered *optimal*, and hence grammatical. In Bresnan (2001) Economy of Expression is measured in terms of the number of syntactic c–structure nodes.

12.2.2 Set-valued f–structures and distribution

A special c–structure rule schema defines coordinated phrases of like constituents (6). In the associated f–structure, the coordinated phrase is represented as a set-valued f-structure. Each of the conjuncts is defined as an element within this set, by the functional annotations $\downarrow \in \uparrow$.

(6)
$$\begin{array}{lccc} \text{XP} \rightarrow & \text{XP} & \text{Conj} & \text{XP} \\ & \downarrow \in \uparrow & \uparrow = \downarrow & \downarrow \in \uparrow \end{array}$$

Figure 2 displays the resulting c–/f–structure pair for a coordination of C' constituents with a shared SUBJ outside the coordinated phrase. Without further assumptions, the f–structure is incomplete regarding the elements of the set, which are both missing a SUBJ.

To account for shared arguments in coordinate structures (as in (3)), the operation of **distribution** is automatically applied to all features that are declared *distributive*. In particular, all grammatical functions are distributive features.

> **Distribution of features into set elements**
> If a is a distributive feature and s is a set of f–structures, then $(s\ a) = v$ holds if and only if $(f\ a) = v$ for all f–structures f that are members of the set s. (Dalrymple 2001:158)

As a result, we obtain the wellformed f–structure in Figure 3. The distributed SUBJ f–structure satisfies Completeness in both conjuncts.

12.3 Asymmetric Coordination

12.3.1 Problems for the standard coordination analysis

Let us now consider the problem that SGF coordination presents for the standard LFG coordination analysis.

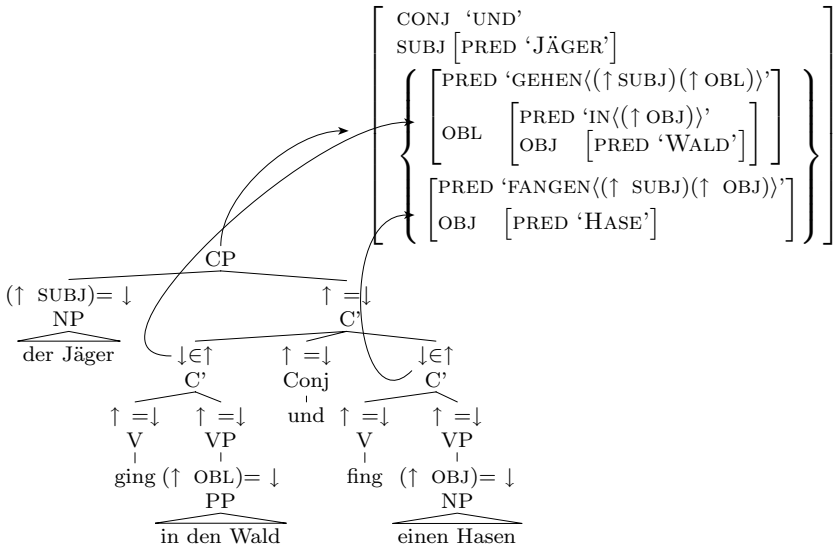


FIGURE 2 C'-coordination and f-structure without distribution.

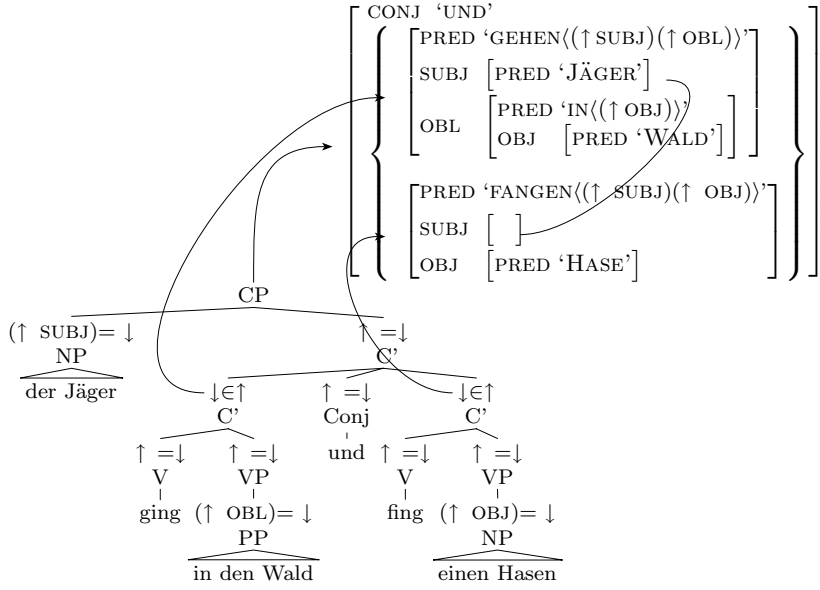


FIGURE 3 C'-coordination and f-structure with distributed SUBJ function.

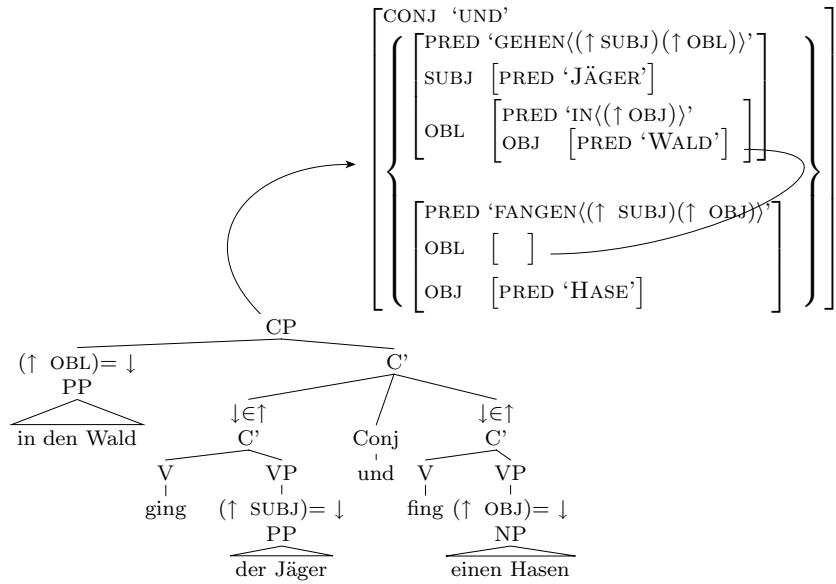


FIGURE 4 Asymmetric (SGF) coordination as C'-coordination

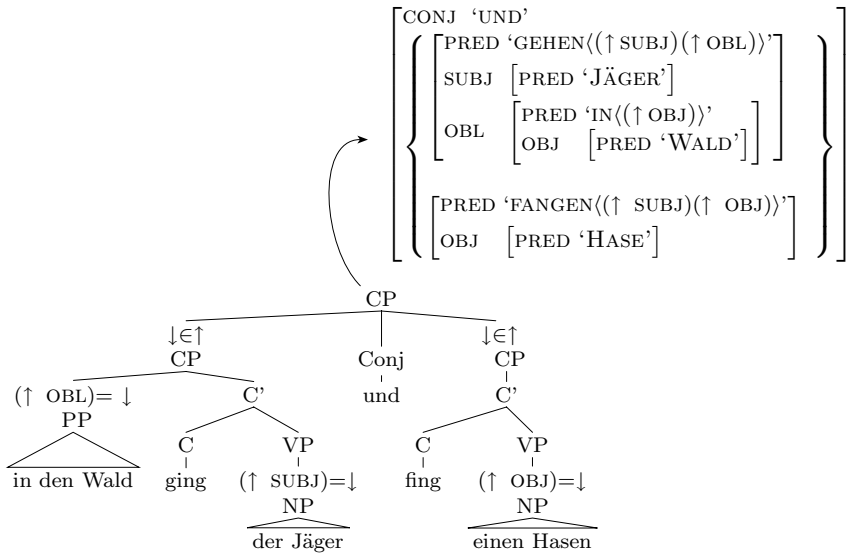


FIGURE 5 Asymmetric (SGF) coordination as CP-coordination

If we analyse (4a) as a coordination of C' constituents, as in Figure 4, distribution applies to the topicalised oblique PP *in den Wald*. While this yields a wellformed f–structure for the first conjunct, distribution into the second conjunct violates Coherence: *fangen* does not subcategorise for an oblique argument. Moreover, since the subject is realised within the first conjunct, it is not subject to distribution. The second conjunct is missing a subject, violating Completeness.

If we analyse SGF coordination as involving symmetric CP coordination as in Figure 5, we avoid illicit distribution of the topicalised phrase, but still encounter the problem of a conjunct internal subject that cannot be distributed — the notorious 'subject gap' problem.

12.3.2 Syntactic properties of asymmetric coordination

Having illustrated the problems we encounter when applying established principles of regular constituent coordination to SGF coordination constructions, we now review the major syntactic and semantic characteristics of SGF coordination that need to be accounted for by any successful analysis (see Kathol 1999).

Number and Type of Gaps. Example (7) illustrates the fact that SGF coordination does not license additional gaps in the right conjunct(s), besides the characteristic subject gap.

(7) * Einen Wagen$_j$ kaufte Hans$_i$ und meldete e$_i$ e$_j$ an.
A car bought Hans and registered Prt
'A car bought Hans and registered.'

Only subjects can be "gapped" in asymmetric coordination. Equivalent examples with a non-subject (here: object) gap are ungrammatical.

(8) * Gestern kaufte Hans den Wagen$_i$ und meldete Max e$_i$ an.
Yesterday bought Hans the car and registered Max Prt
'Yesterday Hans bought the car and Max registered.'

Word Order Properties. SGF coordination shows a peculiar word order restriction, preventing the specifier position of CP in the right conjunct from being overtly realised: whereas (9a) with an object in SpecCP is a perfectly grammatical sentence, the specifier position cannot be occupied in (9b). Only the serialisation in (9c) is acceptable.

(9) a. Einen Hasen schoss der Jäger an.
A rabbit shot the hunter Prt
'A rabbit, the hunter wounded.'

b. * In den Wald ging der Jäger und einen Hasen schoss an.
Into the forest went the hunter and a rabbit shot Prt
'Into the forest went the hunter and a rabbit wounded.'

c. In den Wald ging der Jäger und schoss einen Hasen an.
Into the forest went the hunter and shot a rabbit Prt
'Into the forest went the hunter and wounded a rabbit.'

Quantifier Scope. As observed by Büring and Hartmann (1998) and Kathol (1999), the same interpretation is obtained for (10a) and (10b), irrespective of the position of the quantified subject. In both examples the subject takes scope over both conjuncts: the interpretation is that for almost no one is it the case that he or she both buys a car and takes the bus. This is surprising for the SGF construction (10b), as the quantified subject occupies the middle field *within* the first conjunct, from where it does not structurally outscope the second conjunct.

(10c), on the other hand, is problematic for analyses that assume an empty PRO subject in SGF constructions: (10c) with an overt (repeated) quantified subject only allows for a narrow scope reading, where the quantifiers take scope only over the individual conjuncts, meaning that almost no one buys a car and almost no one takes the bus.

(10) a. Die wenigsten Leute kaufen ein Auto und fahren mit dem Bus.
Almost no one buys a car and drives with the bus
'Almost no one buys a car and takes the bus.'

b. Daher kaufen die wenigsten Leute ein Auto und fahren
Therefore buys almost no one a car and drives
mit dem Bus.
with the bus
'Therefore almost no one buys a car and takes the bus.'

c. Daher kaufen die wenigsten Leute ein Auto und fahren
Therefore buys almost no one a car and drives
die wenigsten Leute mit dem Bus.
almost no one with the bus
'Therefore almost no one buys a car and almost no one takes the bus.'

Asymmetric verb-last/verb-fronted (VL/VF) coordination. Coordination of verb-last/verb-first sentences is only supported in the order VL/VF (cf. (11b)).

(11) a. [$_{CP}$ Wenn Du in ein Kaufhaus kommst] und
If you into a shop come and
[$_{CP}$ (Du) hast kein Geld], ...
you have no money, ...
'If you enter a shop and (you) don't have any money, ...'

b. * [$_{CP}$ In ein Kaufhaus kommst Du und
Into a shop come you and

[$_{CP}$ wenn (Du) kein Geld hast], ...
if you no money have, ...
'A shop you enter and if (you) don't have any money, ...'

Asymmetric VL/VF coordinations are closely related to SGF constructions: the subject of the right conjunct can be omitted, in which case we find a similar accessibility paradox since the subject within the first conjunct cannot be distributed to the second conjunct, which is always V2 (12a). A gap in the second conjunct is only licensed for subjects (12b), and as with SGF coordination, we cannot have multiple gaps (12c). Finally, similar to SGF coordinations, the second conjunct's SpecCP position cannot be filled by a non-subject constituent (12d).

(12) a. Wenn Du in ein Kaufhaus kommst und (Du) hast kein Geld
If you into a shop come and you have no money
'If you enter a shop and (you) don't have any money, ...'

b. * Wenn Du einen Kunden$_j$ hast und Du beleidigst e$_j$
If you a customer have and you offend
'If you have a customer and you offend, ...'

c. * Wenn Du$_i$ ein Stück$_j$ übst und (Du$_i$) führst e$_j$ auf
If you a play practice and (you) perform Prt
'If you practice a play and (you) perform, ...'

d. * Wenn Du in ein Kaufhaus kommst und kein Geld hast (Du)
If you into a shop come and no money have (you)
'If you enter a shop and no money (you) have, ...'

12.4 Previous Approaches

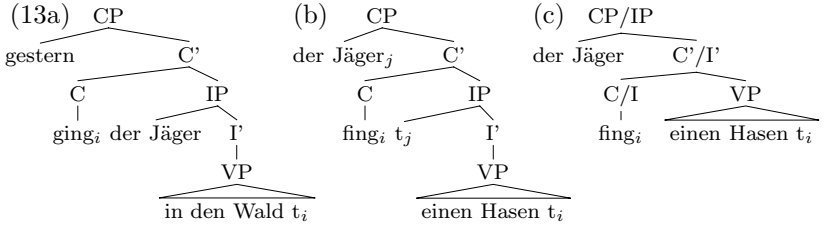
Before developing our own analysis, we review two types of approaches that have been explored in previous work: analysis by asymmetrically embedded constituents, and coordination of symmetric conjuncts.

12.4.1 Asymmetric analyses

Heycock and Kroch (1993) proposed an analysis in the P&P model that is similar, at a conceptual level, to the early analyses of Wunderlich (1988) and Höhle (1990). It will be discussed here as representative of the class of analyses that admit coordination of unlike constituents to account for the observed asymmetry of SGF coordinations.

The analysis builds on independent assumptions about the phrase structure of verb second (V2) languages like German. V2 is analysed as I-to-C movement. The specifier of CP can be filled by a non-subject phrase, as in (13a). In subject initial V2 sentences, the subject must move from SpecIP to SpecCP, leaving behind an empty I projection

(13b). Similar to Haider (1988), the empty I projection and the structurally isomorphic C projection are "folded" into a *matching projection* of a complex category C/I in (13c).

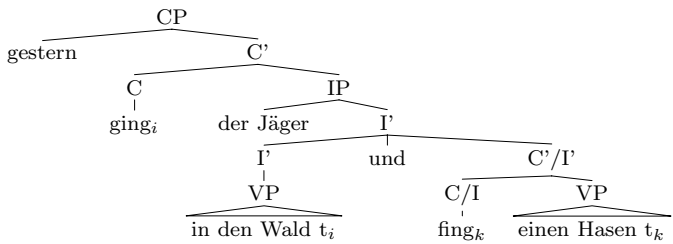


Heycock and Kroch's analysis of SGF coordination naturally emerges from this *matching projection* analysis of subject-initial V2 sentences: An SGF coordination can be constructed from (13a) and (13c) by coordination of I' and C'/I' constituents, which are unlike, but share the categorial features of I.

The resulting SGF coordination structure is displayed in (14). Due to low coordination at the level of I', the shared subject governs both conjuncts, accounting for the main syntactic properties of SGF constructions: restriction to subject gaps and wide scope of quantified subjects.

However, the analysis necessarily involves extraction asymmetries that are otherwise ungrammatical. It is well-known that extraction from coordinated phrases is only possible "across-the-board". The structure (14), by contrast, involves head movement out of the first conjunct only. Similarly, (15), with a topic argument, violates the ATB extraction constraint and results in a fully evacuated first conjunct. Finally, the analysis needs to explain why a topicalised adjunct does not necessarily take scope over the second conjunct (as discussed by Höhle).

(14)



(15) In den Wald$_j$ ging$_i$ der Jäger [[e$_i$ e$_j$] und [fing einen Hasen]].

Büring and Hartmann (1998) present an analysis of SGF coordination that avoids extraction asymmetries by considering it as an instance of adjunction, rather than coordination. Their analysis accounts

for new data on scope, but nevertheless suffers from two problems. First, as opposed to classical adjunction, SGF coordination does not admit topicalisation of the adjoined material (16) (cf. Kathol 1999).

(16) a. [Ohne sie anzuschauen]$_i$ hat Fritz Maria geküsst e$_i$.
Without her to.look.at has Fritz Maria kissed
'Fritz kissed Maria without looking at her'
b. $*$ [(Und) fing einen Hasen]$_i$ ging der Jäger in den Wald e$_i$.

More importantly, while Büring and Hartmann motivate their analysis by special binding and scoping phenomena that can be found in SGF constructions, they must concede that the same type of data is also found in uncontroversial VP coordination structures.[4] Our conclusion is therefore that instead of reanalysing classical VP coordination as adjunction, we need to account for the observed special scoping and binding asymmetries in a different way.

12.4.2 Symmetric analyses

There are few symmetric analyses of asymmetric coordination.

Steedman (1990) accounts for SGF coordination within his theory of gapping. He proposes special functional application rules for coordination in the CCG framework that operate in gapping and SGF coordination constructions alike. While the analysis is very general, it fails to explain important restrictions of the SGF construction, such as the restriction to applying to a unique grammatical function, the subject.

Kathol (1995, 1999) developed a linearization-based model of German syntax that is extended to account for SGF coordination. In Kathol (1999) special licensing conditions are defined to account for word order constraints of SGF coordination. We discuss his analysis and the problems it encounters in more detail.

Kathol starts from the observation that the two coordinations in (17) are merely linearisation variants of a unique underlying predicate coordination structure, with a shared subject.

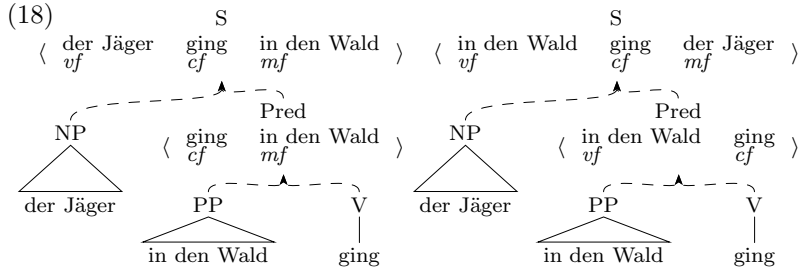
(17) a. *Der Jäger* {*ging in den Wald*} *und* {*fing einen Hasen*}.
b. {*In den Wald ging*} *der Jäger und* {*fing einen Hasen*}.

This intuition, however, cannot be formalised in a phrase structure tree, which encodes constituency *and* word order at the same time. He therefore develops a "linearisation-based model of syntax" that provides a modular representation of constituency and (variable) linearisation.

[4] The analysis of Büring and Hartmann (1998) is extremely interesting and thoroughly worked out, in particular for the new data on scoping facts it accommodates. It cannot be discussed here in more detail for reasons of space.

An illustration is given in (18), where the same constituent tree (represented by dotted arcs) is associated with different word orders (displayed in angle brackets). Restrictions on possible linearisations are defined by topological constraints (19) that need to be met by the assigned topological labels *vf, cf, mf, vc, (nf)* in a sentential clause.[5] Both linearisations in (18) satisfy the topological linearisation constraints (19) that require, in particular, *vf* to precede *cf*, and *cf* to precede *mf*.

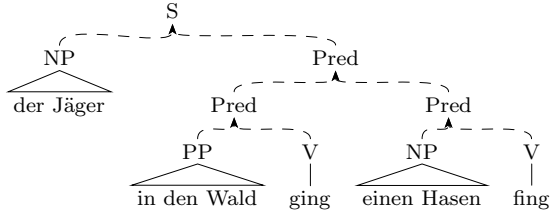
(18)



(19) $vf < cf < mf < vc$

To account for the reductionist properties of coordination, Kathol introduces the notion of a *combinatorial factor* for phrases that are shared among coordinated phrases. A combinatorial factor needs to be "linearised" to the second conjunct's tier. If this happens, it is called a *linear factor*. (20) gives an example for symmetric V2 coordination.

(20) Der Jäger ging in den Wald und fing einen Hasen.



der Jäger	ging	in den Wald	(und)		
vf	*cf*	*mf*			
[der Jäger]			fing	einen Hasen	
vf			*cf*	*mf*	

[5]The topological field model goes back to early work in descriptive grammar, and was introduced in formal syntactic theory by Höhle (1983b). The model gives a topological characterisation of German clausal syntax: Arguments and adjuncts can occur in three phrasal fields: Vorfeld (*vf*), Mittelfeld (*mf*) or Nachfeld (*nf*). They are delimited by the complementizer field *cf* and the verbal complex *vc*, where *cf* can only host complementizers or the finite verb, while *vc* admits verbal elements.

Here, the additional tabular representation represents the linearisation of the combinatorial factor (*der Jäger*) (in bold face) to the second conjunct's tier (indicated by brackets and underlining of the linearised phrase). Linearisation of the combinatorial factor preserves its topological label (here *vf*). The coordination structure is wellformed iff the Topological Construal Condition (22) is satisfied (see Kathol 1999).

(21) **Topological Construal Condition**: A coordinated construction is well-formed if the linear factor's topological assignment yields a valid topological sequence on each conjunct tier.

However, if this model is applied to an SGF construction (here an interrogative V1 variant), linearisation of the combinatorial factor (i.e., the phrase to be distributed, or 'linearised') to the second conjunct's tier yields an *invalid* topological sequence (22).[6]

(22) Ging der Jäger in den Wald und fing einen Hasen?

Ging **der Jäger** in den Wald	(und)	
cf *mf* *mf*		
[der Jäger]	fing	einen Hasen
* *mf*	*cf*	*mf*

To account for the special type of *asymmetric* (SGF) coordination structures, Kathol introduces a Subject Functor Linearisation condition (clause A), which is later extended by clause B.

(23) **Subject Functor Linearization** (Kathol 1999:332,334)
A. The subject of a verb-initial conjoined predicate counts as a linear factor only if it occurs in the *Vorfeld*.
B. In the absence of any other linear factor, a constituent occurring in the *Vorfeld* counts as a linear factor (regardless of its status as combinatorial factor).

Clause A restricts linearisation of a *subject* combinatorial factor *in verb-initial coordination structures* to subjects that occur in the vorfeld position. Since in verb-initial structures the subject is either in a vorfeld or a middle field position, clause A excludes linearisation of the combinatorial subject (i.e., its distribution to the second conjunct tier) *exactly* in those — exceptional — cases that characterise the SGF coordination construction: i.e., those cases in which the subject is contained in the middle field of a verb-fronted coordination structure, but is interpreted as the subject of both conjuncts. Due to clause A, none of the combinatorial SGF subjects in (24) is linearised to the second tier. While this yields the correct results for (24a) and (24b) (*cf* <

[6]The underlying tree structure for (22) is identical to the one in (20).

mf is a valid topological sequence), it also admits the ungrammatical serialisation (24c) (cf. (9b) above).

(24) a.

Ging	**der Jäger**	in den Wald	(und)		
cf	*mf*	*mf*			
			fing	einen Hasen	
			cf	*mf*	

b.

In den Wald	ging	**der Jäger**	(und)	
vf	*cf*	*mf*		
			fing	einen Hasen
			cf	*mf*

c.

*In den Wald	ging	**der Jäger**	(und)	
vf	*cf*	*mf*		
			einen Hasen	fing
			vf	*cf*

Here then, clause B comes into effect, positing that "In the absence of any other linear factor, [any] constituent occurring in the *Vorfeld* counts as a linear factor (regardless of its status as combinatorial factor)". That is, in case no constituent has been linearised, a constituent in the Vorfeld can be counted as a linear factor in the second conjunct, whether or not it can be considered as a combinatorial factor, i.e. a phrase to be distributed. This further amendment does, in the end, account for the facts, as illustrated in (25), but at a high price: linearisation of phrases to the second tier can be motivated for *combinatorial factors* — phrases that are interpreted as arguments or adjuncts of both conjunct heads — but is lacking any justification for phrases that are not interpreted as shared with the second conjunct. As a consequence, clause B weakens the otherwise crucial notion of a *combinatorial factor*.

(25) b.

In den Wald	ging	**der Jäger**	(und)	
vf	*cf*	*mf*		
[In den Wald]			fing	einen Hasen
vf			*cf*	*mf*

c.

*In den Wald	ging	**der Jäger**	(und)	
vf	*cf*	*mf*		
[In den Wald]			einen Hasen	fing
**vf*			*vf*	*cf*

In sum, the *Subject Functor Linearisation* conditions — designed to account for the special properties of SGF coordination — are introduced without motivation or supporting evidence. They are tailored to meet the exceptional configuration of SGF coordinations.

12.5 An LFG Analysis of Asymmetric Coordination

In what follows we develop a multi-factorial LFG analysis of asymmetric coordination. It builds on well-established grammatical principles of the LFG theory, in particular principles of correspondence between c– and f–structure, and the notion of *grammaticalised discourse functions* (GDF). Our analysis of asymmetric coordination introduces a new concept — *asymmetric* GDF *projection* — that will be motivated by relating the discourse-functional properties of asymmetric coordination to the well-known discourse subordination effects of modal subordination.

12.5.1 Symmetric analysis with asymmetric GDF projection

Grammatical functions can be classified along various dimensions, e.g., argument vs. non-argument functions, discourse vs. non-discourse functions (cf. Bresnan 2001:97f.). Bresnan further introduces the notion of a *grammaticalised discourse function*, covering TOPIC, FOCUS and SUBJ: "These functions are the most salient in discourse and often have c–structure properties that iconically express this prominence, such as preceding or c-commanding other constituents in the clause."

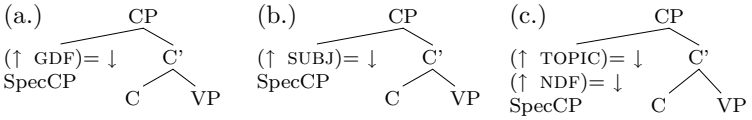
(26) *Grammaticalised Discourse Functions* (GDF)

TOPIC FOCUS SUBJ OBJ OBJ_θ OBL_θ COMP ADJ

Discourse Functions *Argument Functions*

In a verb second language like German, we can characterise the GDF functions as the class of functions that occupy the specifier position of CP. From the abstract functional annotation principle in (27a) we can derive alternative GDF instantiations in (27b) and (27c).[7]

(27)



This language-specific characterisation of GDF functions is in accordance with Bresnan's general characterisation: functions that occupy the specifier position of CP qualify as most salient in discourse (Choi 2001), and are c–structurally prominent, in terms of both precedence and c–command.

Our analysis of asymmetric coordination is summarised in the following (extended) definition of the CP coordination rule: (28) defines

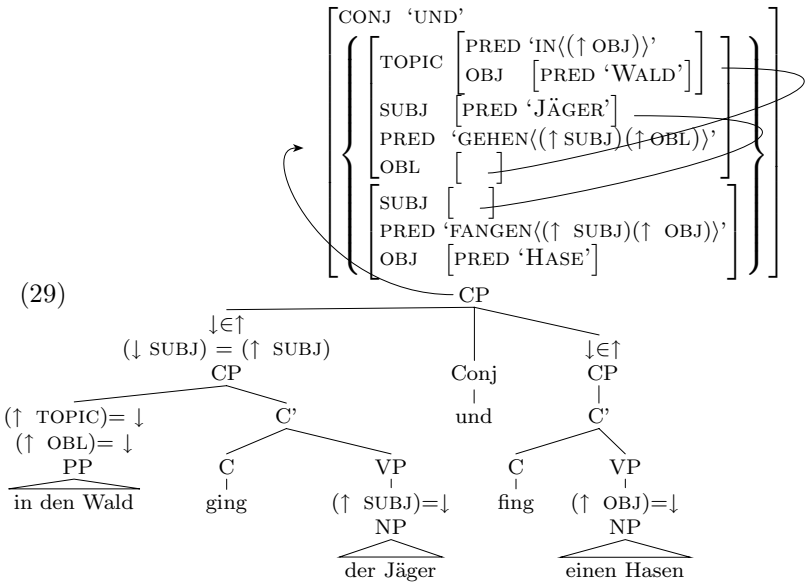
[7] Projection of a discourse function typically involves additional projection of a non-discourse function NDF (27c).

symmetric CP coordination in c–structure, with projection of the conjunct f–structures by classical ↓∈↑ annotations. As an extension to this classical *symmetric* coordination analysis we allow, at the level of f–structure, for *optional, asymmetric projection of a* GDF *function* of the left conjunct to the level of the coordination. As we shall see, this extension accounts for the major syntactic properties of SGF coordination.

(28) CP ⟶ CP Conj CP
↓∈↑ ↑ =↓ ↓∈↑
((↓ GDF) = (↑ GDF))

An example is given in (29). Here, GDF is chosen to instantiate to SUBJ. The annotation (↓ SUBJ) = (↑ SUBJ) defines the first conjunct's SUBJ (*Jäger*) as the SUBJ of the coordination as a whole, i.e. the set-valued f–structure. Due to the distributional character of grammatical functions, the SUBJ defined for the set is distributed to *all* elements of the set. While it is already defined for the left conjunct, it is now introduced in the right conjunct, filling the notorious subject gap.

(29)



12.5.2 Syntactic properties revisited

We will now investigate the predictions of this analysis, reconsidering the syntactic and semantic properties discussed in Section 12.3.2.

Number and Type of Gaps. We had seen that asymmetric SGF coordination is restricted to a *single* gap, and to *subject* gaps only. The

relevant examples — (7) and (8) from page 266 — are reproduced in (30) and (31). How does our analysis by asymmetric GDF-projection account for these restrictions?

(30) * Einen Wagen$_j$ kaufte Hans$_i$ und meldete e$_i$ e$_j$ an.

(31) * Gestern kaufte Hans den Wagen$_i$ und meldete Max e$_i$ an.

We need to consider two cases: Instantiation of GDF to (i) SUBJ or (ii) a discourse function DF.

(i) Instantiation of GDF to SUBJ: In (30) asymmetric projection of SUBJ enables distribution of the first conjunct's SUBJ (*Hans*) to the second conjunct, satisfying the Completeness constraint of *anmelden* regarding its SUBJ. However, the obligatory OBJ function is not locally defined, and *cannot* be satisfied by alternative means: asymmetric GDF projection in (28) can only be instantiated once, and has been chosen to project SUBJ. The sentence is ungrammatical due to the missing object.

The ungrammaticality of non-subject gaps as in (31) is explained as follows: Since the subjects of the two conjuncts are distinct, asymmetric projection of SUBJ (instantiating GDF to SUBJ) leads to an inconsistency in f–structure, due to conflicting values for SUBJ in the second conjunct. Moreover, given SUBJ projection, the object gap cannot, at the same time, be asymmetrically projected from the first to the second conjunct.

(ii) Instantiation of GDF to TOPIC/FOCUS: We further need to prove that examples (30) and (31) are ruled out in the alternative case: instantiation of GDF to a discourse function, e.g. TOPIC.[8]

In (30) the TOPIC (*Wagen*) is identical to the first conjunct's OBJ (cf. (27c)). By asymmetric projection of TOPIC, the TOPIC is distributed to the second conjunct. The SUBJ function of the second conjunct, by contrast, remains unfilled, leading to a violation of Completeness.

In a similar way, (31) with a non-subject gap is ruled out if GDF is set to TOPIC. The structural TOPIC position is occupied by a non-OBJect function, here an adjunct. Its projection to the second conjunct leaves the crucial object gap unfilled.

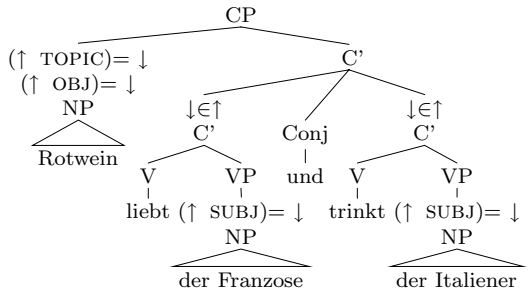
Principle of Economy of Expression. Asymmetric GDF projection as defined in (28) predicts the basic functional properties of SGF coordination. However, besides the cases discussed above, it predicts an asymmetric analysis for data such as (32), which are, however, cases of classical, symmetric ATB extraction.

(32) Rotwein liebt der Franzose und trinkt auch der Italiener.
Red wine loves the Frenchman and drinks also the Italian.

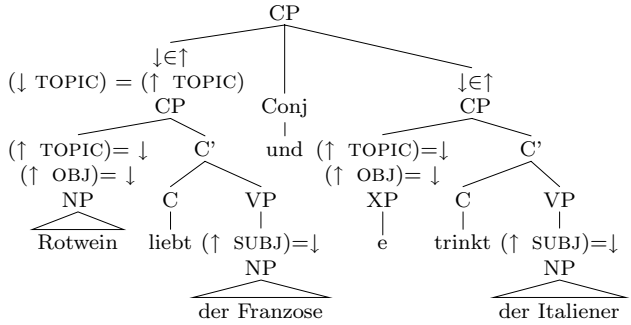
[8] We restrict discussion to the TOPIC function, the case of FOCUS being equivalent.

The classical analysis of ATB extraction as in (32) is given in (33a). The topicalised OBJ is realised outside the C' coordination. The (coreferent) OBJ and TOPIC functions are distributed to both conjuncts, as shown in (33c). However, the same f-structure is now obtained by an alternative analysis, in terms of asymmetric GDF projection, as in (33b). In c–structure, the (shared) TOPIC is realised *within* the first CP conjunct. With GDF instantiated to TOPIC, the TOPIC is asymmetrically projected to the second conjunct. In addition, an empty SpecCP position is required within the second conjunct, to equate TOPIC and OBJ functions. The analysis projects the very same f–structure that we obtain for the regular ATB extraction analysis, namely (33c).

(33a)



(33b)



(33c)

$$\left[\left\{\left[\begin{array}{ll}\text{TOPIC} & \left[\text{PRED 'ROTWEIN'}\right] \\ \text{OBJ} & \square \\ \text{PRED} & \text{'LIEBEN}\langle(\uparrow \text{SUBJ})(\uparrow \text{OBJ})\rangle\text{'} \\ \text{SUBJ} & \left[\text{PRED 'FRANZOSE'}\right]\end{array}\right], \left[\begin{array}{ll}\text{TOPIC} & \square \\ \text{OBJ} & \square \\ \text{PRED} & \text{'TRINKEN}\langle(\uparrow \text{SUBJ})(\uparrow \text{OBJ})\rangle\text{'} \\ \text{SUBJ} & \left[\text{PRED 'ITALIENER'}\right]\end{array}\right]\right\}\right]$$

This unwarranted spurious ambiguity is, however, ruled out on the basis of the *Principle of Economy of Expression* (34), (Bresnan 2001:91). This principle requires the choice of the smallest c–structure that allows for the satisfaction of f–structure constraints and the expression of the intended meaning (see also Dalrymple 2001:85).

(34) **Economy of expression:** All syntactic phrase structure nodes are optional and are not used unless required by independent principles (completeness, coherence, semantic expressivity).

The alternative analyses (33a,b) yield identical f-structure representations, on the basis of different c–structure representations. However, the structural complexity — measured in terms of the number of syntactic nodes employed, excluding lexical and preterminal nodes — is higher for the asymmetric coordination analysis (10 nonterminal nodes) as opposed to the regular ATB extraction analysis (9 nonterminal nodes).

Following the Principle of Economy of Expression, then, the more "verbose" structural backbone, the asymmetric analysis in (33b), is not admitted as an alternative grammatical analysis.

Quantifier Scope. Example (35) — introduced as (10) on page 267 — shows the peculiar property of SGF coordination of allowing wide scope of a quantified subject from the middle field position of the first conjunct. That is, the SGF coordination (35b) is semantically equivalent to the VP coordination construction (35a) (modulo the topicalised adverbial).

(35) a. Die wenigsten Leute [kaufen ein Auto] und [fahren mit dem Bus].
b. [Daher kaufen die wenigsten Leute ein Auto] und [fahren mit dem Bus].

The key to this puzzling behaviour is already implied by our asymmetric GDF projection analysis, where the inherent asymmetry of the construction is captured in the c– to f–structure correspondence: by asymmetric projection of the SUBJ to the second conjunct we derive the very same f–structure representations for the symmetric and asymmetric coordination examples (again, modulo the adjunct in (35b)).

Since in LFG theory semantic interpretation, including quantificational scope, is computed on the basis of the f–structure representation, we predict the equivalent f–structures for symmetric and asymmetric coordinations in (35) to yield identical scopal interpretations.

In *Glue Semantics* (Dalrymple 1999), meaning is constructed compositionally, in parallel to a linear logic derivation that assembles parts of the f–structure that contribute to the sentence meaning. For co-

ordination with shared arguments, the semantics is built on identical f–structure representations, schematically displayed in (36).

(36) f
$$\left[\begin{array}{l}\textsc{conj}\ \ \text{`und'}\\ \left\{\begin{array}{ll} f_1 & \left[\begin{array}{ll}\textsc{pred} & \ldots\\ \textsc{subj} & h\left[\ \ \right]\end{array}\right]\\ f_2 & \left[\begin{array}{ll}\textsc{pred} & \ldots\\ \textsc{subj} & h\left[\ \ \right]\end{array}\right]\end{array}\right\}\end{array}\right]$$
$\lambda P.\lambda Q.\lambda X.[P(X) \wedge Q(X)]$:
$[h_\sigma \multimap f_{1_\sigma}] \multimap [[h_\sigma \multimap f_{2_\sigma}] \multimap [h_\sigma \multimap f_\sigma]]$ (Dalrymple 2001:379)

An analysis attributed to Crouch and Asudeh is sketched in (37) (Dalrymple 2001:p.376ff): the semantic contributions of the conjoined predicates (corresponding to $h_\sigma \multimap f_{1_\sigma}$ and $h_\sigma \multimap f_{2_\sigma}$ in the glue part) are consumed, leading to an open, conjoined predicate in the corresponding meaning part $\lambda X.[P(X) \wedge Q(X)]$. Quantifying in of the shared subject, referred to by h_σ, then leads to a wide scope reading in the case of a quantified subject. The important steps of the derivation for (35) are illustrated in (37).

(37)
$$\left[\begin{array}{l}\textsc{conj}\ \ \text{`und'}\\ \left\{\begin{array}{ll} f_1 & \left[\begin{array}{ll}\textsc{subj} & h\left[\textsc{pred}\ \text{`Leute'}\right]\\ \textsc{pred} & \text{`kaufen}\langle(\uparrow \textsc{subj})(\uparrow \textsc{obj})\rangle\text{'}\\ \textsc{obj} & \left[\textsc{pred}\ \text{`Wagen'}\right]\end{array}\right]\\ f_2 & \left[\begin{array}{ll}\textsc{subj} & h\left[\ \ \right]\\ \textsc{pred} & \text{`fahren}\langle(\uparrow\ \textsc{subj})\rangle\text{'}\\ \textsc{adj} & \left\{\left[\begin{array}{ll}\textsc{pred} & \text{`mit}\langle(\uparrow \textsc{obj})\rangle\text{'}\\ \textsc{obj} & \left[\textsc{pred}\ \text{`Bus'}\right]\end{array}\right]\right\}\end{array}\right]\end{array}\right\}\end{array}\right]$$
$\lambda X.[\lambda x.kaufen(x, wagen)(X) \wedge \lambda x.fahren(x, bus)(X)] : h_\sigma \multimap f_\sigma$
$wenige(x, leute(x), kaufen(x, wagen) \wedge fahren(x, bus)) : f_\sigma$

The puzzle of word order asymmetry. We are left with the special word order restrictions observed in Section 12.3.2. We need to explain why the specifier position of the right conjunct CP cannot be overtly realised. That is, why is (38b) ungrammatical, as opposed to the general availability of topicalised non-subjects as in (38c)?

These order restrictions are particularly challenging for a symmetric c–structure analysis, where the second conjunct offers a SpecCP position, and thus predicts (38b) to be grammatical.

(38) a. In den Wald ging der Jäger und fing einen Hasen.
b. * In den Wald ging der Jäger und einen Hasen fing.
c. Einen Hasen fing der Jäger.

In discussion of Kathol's approach we argued that his attempt to derive these word order restrictions from syntactic constraints leads to rather ad-hoc conditions, lacking independent grammatical motivation.

In Frank (2002) we investigated the OT-model of word order in Choi (2001) to explain these special word order restrictions. Choi (2001) derives word order properties observed in typologically distinct languages from a set of interacting constraints between different levels of grammatical description, in particular structural, functional-syntactic and discourse properties represented in c–, f–, and i–structure. However, when applied to the word order properties of SGF or VL/VF coordinations, the model fails to rule out the order in (38b) as suboptimal.

12.5.3 Solving the puzzle: A discourse-functional analysis

In the following we will investigate the special discourse-functional properties of asymmetric coordination constructions. We will relate these properties to the well-known discourse subordination effects of modal subordination. We establish general licensing conditions for this kind of discourse-functional subordination that will explain the mysterious word order restrictions on both SGF and VL/VF coordination.

Discourse-functional properties of asymmetric coordination. The following set of examples gives pairwise contrasts between "regular" coordination or discourse sequences, as opposed to what we will call "discourse(-functional) subordination contexts" (cf. Frank 1994).

In (39) we observe a striking contrast of interpretation between the symmetric (VL/VL) and the asymmetric (VL/VF) coordination: the asymmetric variant (b) only allows for a nonsensical interpretation where I go for walks if it is summer and winter at the same time, while in the symmetrical (a) example I go for walks in summer *or* winter.[9]

(39) Symmetric vs. VL/VF Coordination

a. [[Wenn es Sommer ist] und [wenn es Winter ist]], gehe ich
when it summer is and when it winter is, go I
spazieren.
for walks.
'When it is summer and when it is winter, I go for walks.'

b. ≠ [[Wenn es Sommer ist] und [es ist Winter]], gehe ich spazieren.
'When it is summer and it is winter, I go for walks.'

(40) shows a related contrast involving SGF coordination. In the symmetrical case, the question focusses on possibly different times: the

[9] The example was brought up in discussion by Ellen Brandner (see Frank 1994).

time when Peter calls the dog and the time when he takes him for a walk. The SGF construction, though, can only be understood as a question about the time of a single, complex event or situation, when Peter calls the dog to take him for a walk.

(40) Symmetric vs. SGF Coordination

a. [Wann ruft Peter den Hund] und [wann geht Peter mit ihm
When calls Peter the dog and when goes Peter with it
spazieren]?
for walks
'When does Peter call the dog and when does Peter take him for a walk?'

b. [Wann ruft Peter den Hund] und [geht mit ihm spazieren]?
'When does Peter call the dog and take him for a walk?'

The modal subordination examples in (41) show a related pattern. The first sentence is — under standard analyses of the discourse semantics of conditionals — an island for binding of anaphoric pronouns. However, in (41b) the same syntactic configuration allows for the extension of the conditional's scope, as indicated by the binding of *es* to *ein Pferd*.

(41) Modal Subordination

a. Wenn Fritz ein Pferd hätte, würde er es lieben. # Er reitet *es* jeden Tag.

b. Wenn Fritz ein Pferd hätte, würde er es lieben. Er würde *es* jeden Tag reiten.
'If Fritz had a horse, he would love it. He (#rides | would ride) *it* every day.'

While analyses of modal subordination differ in various respects (cf. Frank 1997), an abstract characterisation of the crucial aspects involved can be stated as follows: Modal subordination can occur in contexts of complex situations (or eventualities), by extension of the scope of a modal operator to otherwise inaccessible material. Domain extension is only licensed if the discourse-subordinated elements do not display *independent* domain marking. This condition is violated in (41a), where indicative mood signals reference to the actual world; as opposed to (41b), where subjunctive mood accords with the context of hypothetical worlds set up by the subordinating modal operator.

We can generalise these conditions to an abstract characterisation of generalised *discourse subordination*, involving (i) the subordinating *domain extension* of an *operator*, (ii) in a complex situation, (iii) lacking *independent* domain marking of the discourse-subordinated elements.

Licensing conditions for asymmetric GDF projection. We consider asymmetric coordination as a syntactic instance of this general notion of *discourse subordination*. Unlike extension of a modal operator's scope, we encounter extension of a functional domain, which is marked by a complementiser or a genuine discourse function, both typical elements of the clause's functional projection. This extension of the default discourse-functional domain is brought about and modeled by our notion of (asymmetric) projection of a grammaticalised discourse function GDF, and is subject to various constraints. In particular, extension of a discourse-functional domain is incompatible with *independent domain marking* of the subordinated elements, by complementisers or discourse functions TOPIC or FOCUS. This is summarised in (42).

(42) Asymmetric coordination: discourse-functional domain extension
- Complementisers (C) and discourse functions TOPIC, FOCUS are syntactic markers of discourse functional domains.
- Extension of a discourse functional domain is modeled by (asymmetric) GDF projection.
- It occurs in coordinated conjuncts, conceived or presented as a complex situation.
- Independent domain marking of functionally subordinated conjuncts is prohibited.

The conditions in (42) are met by the asymmetric (39b) and (40b): the functional domain established by the first conjunct (by a complementiser or FOCUS phrase) is extended to the second conjunct, which is lacking domain marking by a complementiser or discourse function.

Word order properties explained. The assumptions stated in (42) account for the word order properties of asymmetric coordination. In (43) we associate the different serialisations of both types of asymmetric coordinations with their respective discourse-functional domain markers: it is brought out that an introducing domain marked by a TOPIC or complementiser (Compl) may be extended, by asymmetric GDF projection, provided the subordinated conjunct is not independently domain-marked by another complementiser or a genuine discourse function. A SUBJ function in the second conjunct is a neutral element for functional domain marking.

(43) In den Wald ging der Jäger ...
a. und fing einen Hasen. TOPIC-OBL & SUBJ
b. * und einen Hasen fing. * TOPIC-OBL & TOPIC-OBJ

(44) Wenn Du in ein Kaufhaus kommst ...
  a. und hast kein Geld, ... Compl & subj
  b. und Du hast kein Geld, ... Compl & subj
  c. * und kein Geld hast Du, ... * Compl & topic-obj

Final support for relating asymmetric coordination to a general notion of discourse subordination comes from the general forward direction of domain extension to the right (cf. the ungrammatical backwards serialisations in (45)). Restriction of forward-directed scope extension is also observed for modal subordination (cf. Frank 1997).

(45) a. * Ging in den Wald und gestern fing der Jäger einen Hasen.
'Went to the forest and yesterday caught the hunter a rabbit.'
  b. * Kommst in ein Kaufhaus und wenn Du kein Geld hast, kannst Du nichts kaufen.
'Enter a shop and if you have no money, you cannot buy anything.'

12.6 Conclusion

Our analysis of asymmetric coordination is built on a minimal extension of the classical LFG analysis of constituent coordination. In contrast to syntactic frameworks that are based on a single, constituent-based structure, the flexible correspondence architecture of LFG theory allows a solution for the asymmetry paradox. The asymmetry is captured in the c– to f–structure mapping, where we license the projection of a grammaticalised discourse function to the coordination level. This analysis predicts the basic functional syntactic and semantic properties of asymmetric coordinations, and is motivated by taking into account the discourse properties of asymmetric coordination. We argued that asymmetric coordination is a special instance of a more general notion of discourse subordination, by relating it to modal subordination. From this notion of discourse subordination we derived special licensing conditions for functional-syntactic discourse subordination that account for the peculiar word order restrictions of asymmetric coordination.

A similar analysis by "spreading equations" has been proposed in Sadler (2006) for coordination in Welsh, where both subj and tense can asymmetrically project to non-initial conjuncts. Sadler assembles further data, from typologically distinct languages, where asymmetric coordination has been or can be analysed by projecting subject or tense information across coordinated conjuncts. This is in accordance with a general asymmetric feature spreading analysis, where the characteristic functional features of a functional sentence projection, gdf for CP or tense for IP, can extend to a discourse-subordinated conjunct.

Acknowledgments

My first contact with Ron Kaplan dates back to 1991, in my first year as a researcher, when I was looking at problems of binding and coordination. In search of advice I wrote a message to NLTT, and received stimulating input from Ron and his group about a new device that was 'freshly baked' at PARC. The new device, f-precedence, was exactly what I needed, at the time, to handle the basics of coordination and binding. The puzzle of asymmetric coordination, however, I couldn't solve. I took this up much later again, in 2001, after I had left XRCE Grenoble.

During the ten years that lay between my two encounters with asymmetric coordination, and ever since, I have had many opportunities to discuss and exchange ideas with Ron. I greatly admire his sharp thinking and deep insight in formal and linguistic problems, and deeply appreciate his open and friendly mind, where intellectual combat and humour go hand in hand.

References

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Oxford, United Kingdom: Blackwell.

Büring, Daniel and Katharina Hartmann. 1998. Asymmetrische Koordinationen. *Linguistische Berichte* 174:172–201.

Choi, Hye-Won. 2001. Phrase structure, information structure, and resolution of mismatch. In P. Sells, ed., *Formal and Empirical Issues in Optimality Theoretic Syntax*, pages 17–62. Stanford, CA: CSLI Publications.

Crysmann, Berthold. 2006. Syntax - categories: Coordination. In K. Brown, ed., *Encyclopedia of Language and Linguistics*, vol. 3, pages 183–196. Oxford, United Kingdom: Elsevier, 2nd edn.

Dalrymple, Mary, ed. 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. Cambridge, MA: The MIT Press.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*, vol. 34 of *Syntax and Semantics*. San Diego, CA: Academic Press.

Frank, Anette. 1994. V2 by underspecification or by lexical rule. Arbeitspapiere des SFB 340 Nr. 43, University of Stuttgart.

Frank, Anette. 1997. *Context Dependence in Modal Constructions*. Ph.D. thesis, University of Stuttgart.

Frank, Anette. 2001. Treebank conversion. Converting the NEGRA treebank to an LTAG grammar. In *Proceedings of the Workshop on Multi-layer Corpus-based Analysis, EUROLAN 2001 Summer Institute on Creation and Exploitation of Annotated Language Resources*, pages 29–43. Iasi, Romania.

Frank, Anette. 2002. A (Discourse) Functional Analysis of Asymmetric Coordination. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2002 (LFG'02)*, pages 174–196. Athens, Greece: CSLI Online Publications.

Haider, Hubert. 1988. Matching projections. In A. Cardinaletti, G. Cinque, and G. Giusti, eds., *Constituent Structure: Papers from the 1987 GLOW Conference*, pages 101–121. Dordrecht, The Netherlands: Foris. Annali di Ca' Foscari XXVII.

Heycock, Caroline and Anthony Kroch. 1993. Verb movement and coordination in a dynamic theory of licensing. *The Linguistic Review* 11:257–283.

Höhle, Tilman. 1983a. Subjektlücken in Koordinationen. Unpublished manuscript, University of Cologne.

Höhle, Tilman. 1983b. Topologische Felder. Unpublished manuscript, University of Cologne.

Höhle, Tilman. 1990. Assumptions about asymmetric coordination in German. In J. Mascaró and M. Nespor, eds., *Grammar in Progress*, pages 221–235. Dordrecht, The Netherlands: Foris.

Kathol, Andreas. 1995. *Linearization-based German Syntax*. Ph.D. thesis, Ohio State University.

Kathol, Andreas. 1999. Linearization vs. phrase structure in German coordination constructions. *Cognitive Linguistics* 10(4):303–342.

Roberts, Craige. 1989. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy* 12:683–721.

Sadler, Louisa. 2006. Function spreading in coordinate structures. *Lingua* To appear.

Steedman, Mark. 1990. Gapping as constituent coordination. *Linguistics and Philosophy* 13:207–264.

Wunderlich, Dieter. 1988. Some problems of coordination in German. In U. Reyle and C. Rohrer, eds., *Natural Language Parsing and Linguistic Theories*, pages 289–316. Dordrecht, The Netherlands: Reidel.

13

The Insufficiency of Paper-and-Pencil Linguistics: the Case of Finnish Prosody

Lauri Karttunen

13.1 Introduction

It is a basic scientific practice to examine a limited amount of data in the light of some theoretical framework and to develop an analysis that accounts for the primary facts and extends to unseen data. If the predictions are correct, the analysis stands and lends further support for the framework in which it is conceived.

This paper focuses on a case where the analysis turns out to be wrong. It highlights the need for the formalization and computational implementation of linguistic theories. Paper-and-pencil methods are insufficient to test a theory with real data. The problem is particulary acute for optimality theory (Prince and Smolensky 1993, Kager 1999, McCarthy 2002). The ot framework assumes two levels of representation, a set of *inputs* and, for each input, a possibly infinite set of *outputs*. The mapping between the two levels is subject to a set of ranked constraints. It is typically the case that no output candidate satisfies all the constraints. A winning candidate is the one that incurs the fewest violations of the most highly ranked constraint still in play that cannot be satisfied by any of the other surviving output candidates.

We consider two closely related ot analyses of Finnish prosody by Elenbaas (1999) and Kiparsky (2003). In both cases the input consists of a sequence of phonemes and the outputs are sequences of metrical

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

feet that consist of syllables with stress marks, as shown in (1) for the input *opiskelijakin*.

(1) (ó.pis).(kè.li).(jà.kin) 'even the student' (Sg. Nom.)

Here the acute accent indicates primary stress and the two grave accents mark secondary stress. Periods mark syllable boundaries and feet are enclosed in parentheses.

In general, Finnish prosody is trochaic with the main stress on the first syllable and a secondary stress on every other following syllable. Finnish also has a ternary stress pattern that surfaces in words where the stress would fall on a light syllable that is followed by a heavy syllable. A light syllable ends with a short vowel (*ta*); a heavy syllable ends with a coda consonant (*jat*, *an*) or a long vowel (*kuu*, *aa*) or a diphthong (*voi*, *ei*). Example (2a) shows the correct ternary prosody for the input *rakastajattarenako* 'mistress' (Sg. Ela., QP).

(2) a. (rá.kas).ta.(jàt.ta).(rè.na).ko
b. *(rá.kas).(tà.jat).(tà.re).(nà.ko)

(2b) shows the binary stress pattern that is incorrect because of the (tà.jat) foot where the stress falls on a light syllable followed by a heavy syllable. The initial (rá.kas) syllable in (2a) is actually a violation of the same stress constraint but it is allowed by a more highly ranked constraint, specific to Finnish, that requires the main stress on the initial syllable.[1]

It has been claimed that the ternary prosodic pattern arises naturally, in the context of Optimality Theory (OT), from the interaction of independently motivated optimality constraints such as *Lapse and StressToWeight. The idea has its origins in Hanson and Kiparsky (1996). It has been explored in depth in the Ph.D. thesis of Nine Elenbaas (1999) and summarized in the articles by Elenbaas and Kager (1999) and Kiparsky (2003).

This paper formalizes the Elenbaas and Kiparsky analyses in finite-state terms using lenient composition (Karttunen 1998) to prune the candidate set. It shows that the two OT analyses yield incorrect results in cases such as (3).

(3) *(ká.las).te.(lè.mi).nen 'fishing' (Sg. Nom.)

The specific conclusion is that the explanation for the ternary meter offered by Elenbaas and Kiparsky fails systematically for certain input patterns, but a more general point is that OT phonology badly needs

[1] Kiparsky and Elenbaas treat the third syllable of a dactyl as extrametrical, that is, (rá.kas).ta. instead of (rá.kas.ta). This decision of not recognizing a ternary foot as a primitive is of no consequence as far as the topic of this paper is concerned.

computational support. It is difficult to get globally correct results from a handful of examples with the traditional tableau method.

13.2 OT Constraints for Finnish Prosody

Under Kiparsky's analysis (p. 111), the prosody of Finnish is characterized by the system in (4). The constraints are listed in the order of their priority.

(4) a. *Clash: No stresses on adjacent syllables.
b. Left-handedness: The stressed syllable is initial in the foot.
c. Main Stress: The primary stress in Finnish is on the first syllable.
d. FootBin: Feet are minimally bimoraic and maximally disyllabic.
e. *Lapse: Every unstressed syllable must be adjacent to a stressed syllable or to the word edge.
f. Non-Final: The final syllable is not stressed.
g. Stress-To-Weight: Stressed syllables are heavy.
h. License-σ: Syllables are parsed into feet.
i. All-Ft-Left: The left edge of every foot coincides with the left edge of some prosodic word.

Elenbaas (1999) and Elenbaas and Kager (1999) give essentially the same analysis except that they replace Kiparsky's Stress-To-Weight constraint with the more specific one in (5).

(5) *(ÌLH): If the second syllable of a foot is heavy, the stressed syllable should not be light.

Kiparsky, Elenbaas and Kager construct the ranking of these constraints by considering all possible output candidates for a fair number of multisyllabic words. They show that only the ordering in (4) yields the right outcome. For example, the fact that (2a) is preferred over (2b) indicates that License-σ is dominated by Stress-To-Weight (or *(ÌLH)). The contrast between (6a) and (6b) indicates that Stress-To-Weight in turn is dominated by *Lapse.

(6) a. (rá.vin).(tò.lat) 'restaurant' (Pl. Nom.)
b. *(rá.vin).to.lat

13.3 Finite-State Approximation of OT

As we will see shortly, classical OT constraints such as those in (4) and (5) are regular (= rational) in power. They can be implemented by finite-state networks. Nevertheless, it has been known for a long time

(Frank and Satta 1998, Karttunen 1998, Eisner 2000) that OT as a whole is not a finite-state system. Although the official OT rhetoric suggests otherwise, OT is fundamentally more complex than finite-state models of phonology such as classical Chomsky-Halle phonology (Kaplan and Kay 1994) and Koskenniemi's two-level model (Koskenniemi 1983). The reason is that OT takes into account not just the ranking of the constraints but the number of constraint violations. For example, (7a) and (7b) win over (7c) because (7c) contains two violations of *Lapse whereas (7a) and (7b) have no violations.[2]

(7) a. (ér.go).(nò.mi).a 'ergonomics' (Nom. Sg.)
b. (ér.go).no.(mì.a)
c. (ér.go).no.mi.a

Furthermore, for GRADIENT constraints such as All-Ft-Left, it is not just the number of instances of non-compliance that counts but the SEVERITY of the offense. Candidates (7a) and (7b) both contain one foot that is not at the left edge of the word. But they are not equally optimal. In (7a) the foot not conforming to All-Ft-Left, (nò.mi), is two syllables away from the left edge whereas in (7b) the noncompliant (mì.a) is three syllables away from the beginning. Consequently, (7b) with three violations of All-Ft-Left loses to (7a) that only has two violations of that constraint.

If the number of constraint violations is bounded, the classical OT theory of Prince and Smolensky (1993) can be approximated by a finite-state cascade where the input is first composed with a transducer, GEN, that maps the input to a set of output candidates (possibly infinite) and the resulting input/output transducer is then "leniently" composed with constraint automata starting with the most highly ranked constraint. We will use this technique, first described in Karttunen (1998), to implement the two OT descriptions of Finnish prosody. The key operation, LENIENT COMPOSITION, is a combination of ordinary composition and PRIORITY UNION (Kaplan and Newman 1997).

The basic idea of lenient composition can be explained as follows. Assume that `R` is a relation, a mapping that assigns to each input form some number of outputs, and that `C` is a constraint that prohibits some of the output forms. The lenient composition of `R` and `C`, denoted as `R .O. C`, is the relation that eliminates all the output candidates of a given input that do not conform to `C`, provided that the input has at least one output that meets the constraint. If none of the output candidates of a given input meet the constraint, lenient composition

[2]It is important to keep in mind that the actual scores, 0 vs. 2, are not relevant. What matters is that (7a) and (7b) have **fewer** violations than (7c).

spares all of them. Consequently, every input will have at least one output, no matter how many violations it incurs.[3]

In order to be able to give preference to output forms that incur the fewest violations of a constraint C, we first mark the violations and then select the best candidates using lenient composition. We set a limit n, an upper bound for the number of violations that the system will consider, and employ a set of auxiliary constraints, V_{n-1}, V_{n-2}, ..., V_0, where V_i accepts the output candidates that violate the constraint at most i times. The most stringent enforcer, V_0, allows no violations. Given a relation R, a mapping from the inputs to the current set of output candidates, we mark all the violations of C and then prune the resulting R' with lenient composition: R' .O. V_{n-1} .O. V_{n-2} ... .O. V_0. If an input form has output candidates that are accepted by V_i, where $n > i \geq 0$, all the ones that are rejected by V_i are eliminated; otherwise the set of output candidates is not reduced. The details of this strategy are explained in section 13.4.2.

For the sake of efficiency, we may compose all the inputs with the GEN relation and leniently compose the result with all the constraints into a single finite-state transducer that maps each input form directly into its optimal surface realizations, and vice versa.

13.4 Finite-State OT Prosody

In this section, we will show in as much detail as space allows how the two OT descriptions of Finnish prosody in Section 13.2 can be implemented in a finite-state system. The regular expression formalism in this section and the **xfst** application used for computation are described in the book *Finite State Morphology* (Beesley and Karttunen 2003). This technology is the result of a long line of research started by Ronald M. Kaplan and Martin Kay in the early 1980s.

13.4.1 The GEN Function

The task of the GEN function is to provide each input with all conceivable output candidates. In keeping with the hallmark OT thesis of "freedom of analysis", every candidate, however bizarre, should be available for evaluation by the constraints.

A GEN function for prosody must accomplish three tasks: (1) parse the input into syllables, (2) assign optional stress, and (3) combine syllables optionally into metrical feet. Each of these tasks can be performed by a finite-state transducer. The GEN function for Finnish prosody can thus be defined as the composition of the three compo-

[3]Frank and Satta (1998:8–9) call this operation "conditional intersection."

nents:[4] `Syllabify .o. Stress .o. Scan`, where `.o.` represents ordinary composition, as opposed to `.O.` for lenient composition. With the help of this regular expression, we can define the GEN function for Finnish prosody as in (8)

(8) `define GEN(X) [X .o. Syllabify .o. Stress .o. Scan]`

where `X` can be a single input form or a symbol representing a set of input forms or an entire language. The result of evaluating `GEN(X)` is a transducer that maps each input form in `X` into all of its possible output forms.

The initial task, syllabification, is non-trivial in Finnish because the nucleus of a syllable may consist of a short vowel, a long vowel, or a diphthong. Adjacent vowels that cannot constitute a diphthong must be separated by a syllable boundary. For example, the first vowel pair in the input *kielien* 'tongue' (Pl. Gen.) constitutes a diphthong but the second pair does not because of its position in the word. The correct syllabification is kie.li.en.[5]

Because stress assignment and foot assembly are optional, GEN produces a large number of alternative prosodic structures for even short inputs. For example, for the input *kala* 'fish' (Sg. Nom.), `GEN({kala})` produces the 33 output forms shown in (9).

(9) kà.là, kà.lá, kà.la, kà.(lá), kà.(là), ká.là, ká.lá, ká.la, ká.(lá), ká.(là), ka.là, ka.lá, ka.la, ka.(lá), ka.(là), (ká).là, (ká).lá, (ká).la, (ká).(lá), (ká).(là), **(ká.la)**, (ká.lá), (ká.là), (kà).là, (kà).lá, (kà).la, (kà).(lá), (kà).(là), (kà.la), (kà.lá), (kà.là), (ka.lá), (ka.là)

As the analyses by Elenbaas and Kiparsky predict, the correct output is (ká.la).

13.4.2 The Constraints

There are two types of violable OT constraints. For CATEGORICAL constraints, the penalty is the same no matter where the violation occurs. For GRADIENT constraints, the site of violation matters. For example, All-Feet-Left assigns to non-initial feet a penalty that increases with the distance from the beginning of the word.[6]

[4] For details, see the **xfst** script in
`http://www.stanford.edu/~laurik/fsmbook/examples/FinnishOTProsody.html`.

[5] Instead of providing the syllabification directly as part of GEN, it would of course be possible to generate a set of possible syllabification candidates from which the winners would emerge through an interaction with OT constraints such as HaveOnset, FillNucleus, NoCoda, etc.

[6] The current status of gradient constraints is controversial. McCarthy (2003) argues that gradient constraints are unnecessary and harmful. According to him, alignment constraints such as (4i) should be categorical. See also Eisner (2000).

Our general strategy is as follows. We first define an evaluation template for the two constraint types and then define the constraints themselves with the help of the templates. We use asterisks as violation marks and use lenient composition to select the output candidates with the fewest violation marks. Categorical constraints mark each violation with an asterisk. Gradient constraints mark violations with sequences of asterisks starting from one and increasing with the distance from the word edge.

The initial set of output candidates is obtained by composing the input with GEN. As the constraints are evaluated in the order of their ranking, the number of output forms is successively reduced. At the end of the evaluation, each input form typically should have just one correct output form.

An evaluation template for categorical constraints, shown in (10), needs four arguments: the current output mapping, a regular expression pattern describing what counts as a violation, a left context, and a right context.[7]

(10)
```
define Cat(Candidates, Violation, Left, Right) [
  Candidates .o. Violation -> ... "*" || Left _ Right
  .O. Viol3 .O. Viol2 .O. Viol1 .O. Viol0
  .o. Pardon ];
```

The first part of the definition composes the candidate set with a rule transducer that inserts an asterisk whenever it sees a violation that occurs in the specified context.[8] The second part of the definition is a sequence of lenient compositions. The first one eliminates all candidates with more than three violations, provided that some candidates have only three or fewer violations. Finally, we try to eliminate all candidates with even one violation. This will succeed only if there are some output strings with no asterisks. The auxiliary terms `Viol3`, `Viol2`, `Viol1`, `Viol0` limit the number of asterisks. For example, `Viol1`, is defined as `~[$"*"]^2`. It prohibits having two or more violation marks. The third part, `Pardon`, is defined as `"*" -> 0`. It removes any remaining violation marks from the output strings. Because we are counting violations only up to three, we cannot distinguish strings that have four violations from strings with more than four violations. It turns out that three is an empirically sufficient limit for our categorical prosody constraints.

The evaluation template for gradient constraints counts up to 14 violations and each violation incurs more and more asterisks as we

[7] Some constraints can be specified without referring to a particular left or right context. The expression `?*` stands for any unspecified context.

[8] The formalism is explained in Chapter 2 of Beesley and Karttunen (2003).

count instances of the left context. The definition is given in (11).

(11)
```
define GradLeft(Candidates, Violation, Left, Right) [
Candidates
.o. Violation -> "*" ... ||.#. Left _ Right
.o. Violation -> "*"^2 ... ||.#. Left^2 _ Right
.o. Violation -> "*"^3 ... ||.#. Left^3 _ Right
.o. Violation -> "*"^4 ... ||.#. Left^4 _ Right
.o. Violation -> "*"^5 ... ||.#. Left^5 _ Right
.o. Violation -> "*"^6 ... ||.#. Left^6 _ Right
.o. Violation -> "*"^7 ... ||.#. Left^7 _ Right
.o. Violation -> "*"^8 ... ||.#. Left^8 _ Right
.o. Violation -> "*"^9 ... ||.#. Left^9 _ Right
.o. Violation -> "*"^10 ... ||.#. Left^10 _ Right
.o. Violation -> "*"^11 ... ||.#. Left^11 _ Right
.o. Violation -> "*"^12... || .#. Left^12 _ Right
.o. Violation -> "*"^13 ... ||.#. Left^13 _ Right
.o. Violation -> "*"^14 ... ||.#. Left^14 _ Right
.O. Viol14 .O. Viol13 .O. Viol12 .O.Viol11 .O. Viol10
.O. Viol9 .O. Viol8 .O. Viol7 .O. Viol6 .O. Viol5
.O. Viol4 .O. Viol3 .O. Viol2 .O. Viol1 .O.  Viol0
.o. Pardon ];
```

Using the two templates in (10) and (11), we can now give very simple definitions for Kiparsky's nine constraints in (4). We only need a few auxiliary concepts listed in (12). We omit the simple definitions here.

(12)
a. `Light`: Light Syllable (A syllable with a short vowel and without a coda)
b. `MSS`: Syllable with Main Stress
c. `SV`: Stressed Vowel
d. `SS`: Stressed Syllable
e. `US`: Unstressed Syllable
f. `S`: Syllable
g. `E`: Edge: Syllable Boundary or Word Edge.
h. `B`: Boundary (Edge or Foot Boundary)

The constraints are defined in (13).

(13)
a. *Clash: No stress on adjacent syllables.
`define Clash(X) Cat(X, SS, SS B, ?*);`
b. Left-handedness: The stressed syllable is initial in the foot.
`define AlignLeft(X) Cat(X, SS, ".", ?*);`

c. Main Stress: The primary stress in Finnish is on the first syllable.
```
define MainStress(X)
  Cat(X, ~[B MSS ~$MSS], .#., .#.);
```
d. Foot-Bin: Feet are minimally bimoraic and maximally bisyllabic.
```
define FootBin(X)
  Cat(X, "(" Light ")" | "(" S ["." S]^>1, ?*, ?*);
```
e. Lapse: Every unstressed syllable must be adjacent to a stressed syllable or to the word edge.
```
define Lapse(X) Cat(X, US, [B US B], [B US B]);
```
f. Non-Final: The final syllable is not stressed.
```
define NonFinal(X) Cat(X, SS, ?*, ~$S .#.);
```
g. Stress-To-Weight: Stressed syllables are heavy.
```
define StressToWeight(X) Cat(X, SS & Light, ?*, B);
```
h. License-σ: Syllables are parsed into feet.
```
define Parse(X) Cat(X, S, E, E);
```
i. All-Ft-Left: The left edge of every foot coincides with the left edge of some prosodic word.
```
define AllFeetFirst(X)
  GradLeft(X, "(", ~$"." "." ~$".", ?*);
```

To take just one example, let us consider the `StressToWeight` function. The violation part of the definition, `SS & Light`, picks out syllables such as *tí* and *tì* that are light and contain a stressed vowel. The left context is irrelevant, represented as `?*`. The right context matters. It must be some kind of boundary; otherwise perfectly well-formed outputs such as (má.te).ma.(tìik.ka) would get two violation marks: (má*.te).ma.(tì*ik.ka). That is because *tì* by itself is a stressed light syllable but *tìik* is not. The violation mark on the initial syllable *má* is correct but has no consequence because the higher-ranked `MainStress` constraint has removed all competing output candidates for *matematiikka* 'mathematics' (Sg. Nom.) that started with a secondary stress, *mà*, or without any stress, *ma*.

13.4.3 Combining GEN with the Constraints

Having defined both the gen function and Kiparsky's nine prosody constraints, we can now put it all together creating a single function, `FinnishProsody`, that should map any Finnish input into its correct prosodic form. The definition is given in (14).

(14)
```
define FinnishProsody(Input) [ AllFeetFirst( Parse(
StressToWeight( NonFinal( Lapse( FootBin( MainStress(
AlignLeft( Clash( GEN( Input ))))))))) ];
```

A regular expression of the form `FinnishProsody(X)` is computed "inside-out." First the GEN function defined in (8) maps each of the input forms in `X` into all of its possible output forms. Then the constraints defined in Section 13.4.2 are applied in the order of their ranking to eliminate violators, making sure that at least one output form remains for all the inputs that have at least one output form that does not run afoul of some unviolable constraint.

Applying `FinnishProsody` to the input *opettamassa* 'teaching' (Sg. Ine.) results in the input/output relation shown in (15) that correctly represents the ternary prosodic pattern of the word. The incorrect trochaic output competitor, (ó.pet).(tà.mas).sa, has been eliminated.

(15)
```
o   p e t    t a    m a s   s a
( ó . p e t ) . t a . ( m à s . s a )
```

Getting the right result for one input is of course no guarantee that all possible inputs get the desired output. To provide a quick test for the correctness of the analysis we defined `FinnWords` as the set of 25 input words collected from Kiparsky and Elenbaas illustrating various patterns of light and heavy syllables. It is by no means a complete inventory, but it is sufficient to reveal that both analyses are flawed. The compilation of the regular expression `FinnishProsody(FinnWords)` with Kiparsky's constraints produces the output forms shown in (16).

(16) (ér.go).(nò.mi).a, (íl.moit).(tàu.tu).mi.(sès.ta), (íl.moit).(tàu.tu).(mì.nen), (ón.nit).(tè.le).(mà.ni).kin, (ó.pis).(kè.li).ja, (ó.pet).ta.(màs.sa), (vói.mis).te.(lùt.te).le.(màs.ta), (strúk.tu).ra.(lìs.mi), (rá.vin).(tò.lat), (rá.kas).ta.(jàt.ta).(rè.na).ko, (ré.pe).(ǎ̀.mä), (pé.ri).jä, (pú.he).li.(mèl.la).ni, (pú.he).li.(mìs.ta).ni, (mǻ́.ki), (má.te).ma.(tìik.ka), (mér.ko).(nò.min), (kái.nos).(tè.li).jat, (ká.las).te.(lèm.me), **(ká.las).te.(lè.mi).nen**, (ká.las).(tè.let), (kú.nin).gas, **(jǎ̋r.jes).tel.(mǎ̀l.li).syy.(dèl.lä).ni**, (jǎ̋r.jes).(tèl.mät).tö.(mỳy.des).(tǎ̀n.sä), (jǎ̋r.jes).(tèl.mäl).(lìs.tä).mä.(tǒ̀n.tä)

For a native speaker of Finnish, it is immediately obvious that the two bold-faced outputs in (16) are incorrect. The correct outputs are (ká.las).(tè.le).(mì.nen) and (jǎ̋r.jes).(tèl.mäl).li.(sỳy.del).(lǎ̀.ni). Why are the rightful winners losing to undeserving competitors?

13.5 Error Analysis

The GEN function produces 70,653 output candidates for *kalasteleminen* 'fishing' (Nom. Sg.). The six most highly ranked constraints, `Clash`, `AlignLeft`, `MainStress`, `FootBin`, `Lapse` and `NonFinal`, eliminate nearly all of them, leaving just two candidates to be evaluated by the next constraint, `StressToWeight`: (ká.las).te.(lè.mi).nen and (ká.las).(tè.le).(mì.nen). As shown in (17), the desired winner, (17b), has one `StressToWeight` violation more than its competitor (17a).

(17) a. (ká*.las).te.(lè*.mi).nen
b. (ká*.las).(tè*.le).(mì*.nen)

Consequently, the incorrect (17a) is left as the sole survivor.

The same problem arises in the case of *järjestelmällisyydelläni* 'with my systematicity' (Sg. Ade.). After the six most highly ranked constraints have been applied, out of the initial set of 21,767,579 output candidates, 36 candidates are left. As shown in (18), the desired winner, (18b), contains one `StressToWeight` violation whereas 8 others, including the actual winner (18a), satisfy the `StressToWeight` constraint.

(18) a. (jä̆r.jes).tel.(mä̀l.li).syy.(dèl.lä).ni
b. (já̆r.jes).(tèl.mäl).li.(sỳy.del).(lä̀*.ni)

In these two cases, the desired result could be obtained by switching the ranking of `Parse` and `StressToWeight`. However, the new ranking would have undesirable consequences elsewhere. In particular, it would produce the wrong result in the case of *rakastajattarenako* 'mistress' (Sg. Ess., QP) and *voimisteluttelemasta* 'having someone do gymnastics' (Sg. Ela.). Instead of the correct result shown in (16), they would surface with an unwanted trochaic pattern as in (19).

(19) a. *(rá.kas).(tà.jat).(tà.re).(nà.ko)
b. *(vói.mis).(tè.lut).(tè.le).(màs.ta)

Replacing Kiparsky's STRESS-TO-WEIGHT by the more specific *(ÌLH) constraint proposed in Elenbaas (1999) and in Elenbaas and Kager (1999) yields the correct result in the case of *järjestelmällisyydelläni* because (lä̀.ni) in the last foot of (18b) is not a violation of *(ÌLH). However, this switch does not help in the case of *kalasteleminen*. Instead of (17), we now get (20). The correct output, (20b), is still eliminated because it has one violation more than its competitor.

(20) a. (ká*.las).te.(lè.mi).nen
b. (ká*.las).(tè.le).(mì*.nen)

The specter of an unexpected competitor suddenly emerging to eliminate the desired winner is the bane of OT analyses.

The appendix of Elenbaas (1999) contains an extensive list of Finnish syllable patterns and a sample output for each pattern, e.g. XXLHLL (má.te).ma.(tìik.ka).ni. Here L and H stand for light and heavy syllable, respectively, and X can be either L or H. Although the list appears complete, there are gaps. The missing patterns include at least four where the Elenbaas analysis in fact gives an incorrect result: XXLLLH *(ká.las).te.(lè.mi).nen, XXHHLH *(há.pa).roi.(tùt.ta).vaa, XXLHHLH *(pú.hu).(tè.tuim).(mìs.ta).kin and XXHLLH *(kú.ti).tet.(tù.ja).kin. Kiparsky's analysis also fails with the XXLLLH, XXHHLH and XXLHHLH patterns but succeeds with (kú.ti).(tèt.tu).(jà.kin).

In the case of the XXHHLH pattern, the input *haparoituttavaa* 'of the kind that causes one to fumble' (Sg. Par.) has four remaining output candidates at the point where Kiparsky's Stress-To-Weight and Elenbaas' *(ÌLH) constraints come into play. As shown in (21), the desired winner, (21d), loses to (21a) and (21b), which have no violations.

(21) a. (há.pa).roi.(tùt).ta.vaa
b. (há.pa).roi.(tùt.ta).vaa
c. (há.pa).(roi).tut.(tà*.vaa)
d. (há.pa).(ròi.tut).(tà*.vaa)

At the next step, the contest between the remaining two incorrect outputs, (21a) and (21b), is decided in favor of (21b) because it has fewer violations of the `Parse` constraint (Kiparsky's License-σ, Elenbaas' Parse-Syl) than (21a). As in the case of (17) and (18), giving `Parse` a higher rank would give us the right result for the XXHHLH pattern. But, as we have already seen in (19), it would lead to errors elsewhere.

In the case of the XXLHHLH input *puhutetuimmistakin* 'even those who have been made to talk the most' (Pl. Ela.), the desired winner, (22b), loses at the end because it has more violations of the All-Ft-Left constraint than its only remaining competitor, (22a).

(22) a. (pú.hu).**(tè.tuim).****(mìs.ta).kin
b. (pú.hu).te.***(tùim.mis).*****(tà.kin)

In the case of the XXHLLH input *kutitettujakin* 'even the ones that have been tickled' (Pl. Par.), the expected winner, (23c), is eliminated under the Elenbaas analysis because it violates the *(ÌLH) constraint, whereas one of its two competitors, (23a), has no *(ÌLH) violation.

(23) a. (kú.ti).tet.(tù.ja).kin
b. (kú.ti).(tèt).tu.(jà*.kin)
c. (kú.ti).(tèt.tu).(jà*.kin)

Kiparsky's analysis does better in this case because the (tù.ja) foot in (23a) is a violation of his more general Stress-To-Weight constraint. Thus all three candidates in (23) tie on Stress-To-Weight, and License-σ gets to pick (23c) as the rightful winner.

As far as we can see, there is no ranking of the nine constraints in (4) that would produce the right outcome in all the cases we have discussed. Replacing Stress-To-Weight by *(ÌH) does not help.

13.6 Conclusion

The assumption that is common to Kiparsky, Elenbaas and Kager is that the alternation between binary and ternary patterns in Finnish arises in a natural way from the interaction of independently motivated universal constraints. The ranking of the constraints can presumably be discovered by examining a limited set of examples. It may ultimately turn out to be the right assumption for some set of constraints. But it is not true for the constraints that have been proposed so far.

Optimality Prosody is a difficult enterprise. There are computational tools such as **OTSoft**[9] and **Praat**[10] that can select the most optimal output candidate provided that the user has explicitly specified the competing output forms and has manually marked up the violations.[11] These tools presuppose that (1) all the relevant input types are covered and (2) all the possible output candidates are included. Neither condition is met in the Elenbaas and Kiparsky studies. Without a gen function to enumerate all the possible outputs, it is easy to miss the actual winner even if one is a native speaker of the language and an expert in the field.[12]

A finite-state approximation of ot guards against some errors made by a human gen and eval. Instead of working with individual words one-by-one, the phonologist can collect a set of possible inputs of all the different types and apply the constraint system to the whole corpus at once to see the global effect of any change. But even with the **xfst** techniques described in this paper, debugging ot constraints is a very hard problem.

Acknowledgments

Thanks to Arto Tapani Anttila, Kenneth R. Beesley, Mary Dalrymple, Tracy King, Paul Kiparsky, Annie Zaenen and an anonymous reviewer for their helpful comments on earlier versions of this paper.

[9] http://www.linguistics.ucla.edu/people/hayes/otsoft/

[10] http://www.fon.hum.uva.nl/praat/

[11] OTSoft can mark some types of simple violations automatically but not others.

[12] Quandoque bonus dormitat Homerus.

References

Beesley, Kenneth R. and Lauri Karttunen. 2003. *Finite State Morphology*. Stanford, CA: CSLI Publications.

Eisner, Jason. 2000. Directional constraint evaluation in Optimality Theory. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pages 257–263. Saarbrücken, Germany.

Elenbaas, Nine. 1999. *A Unified Account of Binary and Ternary Stress*. Utrecht, Netherlands: Graduate School of Linguistics.

Elenbaas, Nine and René Kager. 1999. Ternary rhythm and the lapse constraint. *Phonology* 16:273–329.

Frank, Robert and Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24(2):307–316.

Hanson, Kristin and Paul Kiparsky. 1996. A theory of metrical choice. *Language* 72:287–436.

Kager, René. 1999. *Optimality Theory*. Cambridge, England: Cambridge University Press.

Kaplan, Ronald M. and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20(3):331–378.

Kaplan, Ronald M. and Paula S. Newman. 1997. Lexical resource reconciliation in the Xerox Linguistic Environment. In *ACL/EACL'98 Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, pages 54–61. Madrid, Spain.

Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. In *Proceedings of Finite-State Methods in Natural Language Processing (FSMNLP'98)*. Ankara, Turkey: Bilkent University. cmp-lg/9804002.

Kiparsky, Paul. 2003. Finnish noun inflection. In D. Nelson and S. Manninen, eds., *Generative Approaches to Finnic and Saami Linguistics: Case, Features and Constraints*, pages 109–161. Stanford, California: CSLI Publications.

Koskenniemi, Kimmo. 1983. Two-level morphology. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

McCarthy, John J. 2002. *The Foundations of Optimality Theory*. Cambridge, England: Cambridge University Press.

McCarthy, John J. 2003. OT constraints are categorical. *Phonology* 20(1):75–138.

Prince, Alan and Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar*. Rutgers, New Jersey: Cognitive Science Center. ROA Version, 8/2002.

14

Gender Resolution in Rumanian

Louisa Sadler

14.1 Introduction

This paper offers a contribution to the treatment of agreement phenomena in LFG by providing an analysis of Rumanian nominal agreement, focussing on gender.

I take up two issues concerned with gender marking and agreement in Rumanian. The first of these is the apparent mismatch between the number of nominal controller genders (three), and the number of target genders (two): nouns appear to make more gender distinctions than the elements which agree with them. This phenomenon, which is not unique to Rumanian, has engendered a number of analyses and on the face of it is a challenge to approaches to agreement by token identity or co-specification. I show how this can be accommodated straightforwardly in LFG. Second, Rumanian is a language in which syntactic resolution of gender under coordination is limited to inanimate NPs: conjoined inanimates resolve to the feminine plural unless all of them are masculine but mixed sex animates resolve to the masculine (Farkas 1990, Lumsden 1992, Corbett 1991, Farkas and Zec 1995, Wechsler and Zlatić 2003, Wechsler 2002). It is therefore interesting in terms of understanding how syntactic and semantic resolution interact, an issue which arises in various forms in a substantial number of languages. I formulate an approach which combines the set-based approach to syntactic gender resolution of Dalrymple and Kaplan (2000) with a specification of semantic resolution. This paper started out in a very practical fashion, in that I needed to get to grips with the implementation of closed sets as values in the xle, a grammar engineering platform for LFG (Crouch

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

et al. 2006), and needed a domain. The analysis proposed here is implemented as an xle grammar fragment, and as usual, the experience of writing a grammar fragment showed that the problem had intricacies which were not at first apparent, and has ultimately helped clarify my thinking about the problem and in particular about the interaction of syntactic and semantic resolution. At several points in this paper I footnote minor divergences between the theoretical descriptions and the notation the xle supports.

Section 14.2 reviews the data concerning the number of genders in Rumanian, showing the mismatch between the number of controller genders and target genders. This section also illustrates the agreement properties of coordinations of inanimate nouns. Section 14.3 looks at gender in animate nouns, presenting data concerning agreement patterns in the coordination of animate nouns, and also cases in which natural and grammatical gender diverge. I then turn to analyses of the data, starting with a brief review of the most comprehensive approach in the literature, that of Farkas (Farkas 1990, Farkas and Zec 1995), in section 14.4. Section 14.5 provides an LFG analysis of target and controller genders in Rumanian, proposing that targets underspecify the features of their controllers. Section 14.6 extends this approach to deal with gender resolution for coordinations of inanimate NPs, building on the proposals of Dalrymple and Kaplan (2000). I then discuss coordination of animate nouns in section 14.7 and formulate the agreement generalization for these coordinate structures. Section 14.8 shows how the agreement for all NP coordinations can be captured succinctly, and section 14.9 concludes with some additional data.

14.2 Three Nominal Genders

Rumanian nouns fall straightforwardly into three distinct gender classes when we consider their behaviour in construction with agreement targets such as adjectives, as illustrated in (1–6) below. In Rumanian, participles and predicate adjectives show predicate-argument agreement with the subject, and determiners and adjectives within NP agree with the head noun (head-modifier agreement), as shown in (1–4) for masculine and feminine nouns (examples from Farkas and Zec 1995, glosses slightly altered for consistency):

(1) un copac frumos
a.m tree.msg beautiful.msg
'a beautiful tree'

(2) doi copaci frumoşi
two.m trees.mpl beautiful.mpl
'two beautiful trees'

(3) o rochie frumoasă
a.F dress.FSG beautiful.FSG
'a beautiful dress'

(4) două rochii frumoase
two.F dresses.FPL beautiful.FPL
'two beautiful dresses'

There is a third class of nouns shown in (5–6) and glossed as neuter, which show a mixed behaviour:

(5) un scaun frumos
a.M chair.NSG beautiful.MSG
'a beautiful chair'

(6) două scaune frumoase
two.F chairs.NPL beautiful.FPL
'two beautiful chairs'

Assignment to a gender class is partly driven by formal factors in Rumanian — nouns ending in [e] are MASC or FEM, those ending in any other vowel are FEM, and nouns ending in a consonant are MASC or NEUT (Farkas and Zec 1995), but there is also a semantic dimension to syntactic gender assignment: nouns referring to males are MASC in gender while those referring to females are FEM. Nouns referring to inanimate objects may be in any of three classes.

Note that neuter is not an inquorate gender, that is, a gender with a very small number of members (Corbett 1991:170), but rather is a class fully on a par with the MASC and the FEM genders. This third class of nouns controls agreement forms identical to the MASC in the singular, and forms identical to the FEM in the plural. The agreement patterns determined by Rumanian nouns are summarised in Table 1.

TABLE 1 Nominal Agreement Patterns

N	Target	N	Target
FSG	FSG	FPL	FPL
MSG	MSG	MPL	MPL
NSG	MSG	NPL	FPL

How should we interpret this third class of nouns? One theoretical possibility is that this (large) class of lexemes simply belongs to two different syntactic genders — they really are MASC in the singular and FEM in the plural (as found with Somali gender polarity), with the existence of this "third" class being essentially a fact internal to the morphology. Such a proposal is found in recent work by Bateman and Polinsky (2005) who propose that Rumanian has just two noun classes in the singular and two in the plural, with membership determined on both formal and semantic grounds. A similar position is adopted in Wechsler and Zlatić (2003:157): "the so-called neuter is really a class of inquorate nouns that are masculine in the singular but feminine in the plural".

On the other hand, in his wide-ranging study of gender as a morphosyntactic category, Corbett (1991) reasserts the traditional view and argues that the existence of three distinct agreement classes is itself enough to merit recognition of three genders in Rumanian, with a distinction emerging between controller and target genders. There is, furthermore, indication of a three way syntagmatic distinction in the syntax. In particular, there is clear evidence from coordination, where the behaviour of neuter singular nouns is evidently distinct from that of masculine singular nouns, that neuter should be differentiated as a third syntactic gender. Note the agreement patterns exemplified in the following data (Farkas and Zec 1995:96) for coordinations of singular nouns.

(7) a. Podeaua şi plafonul sînt albe.
floor.DEF.FSG and ceiling.DEF.MSG are white.FPL
'The floor and the ceiling are white.'

b. Scaunul şi dulapul sînt albe.
chair.DEF.NSG and cupboard.DEF.NSG are white.FPL
'The chair and the cupboard are white.'

c. Peretele şi scaunul sînt albe.
wall.DEF.MSG and chair.DEF.NSG are white.FPL
'The wall and the chair are white.'

d. Podeaua şi scaunul sînt albe.
floor.DEF.FSG and chair.DEF.NSG are white.FPL
'The floor and the chair are white.'

e. Podeaua şi uşa sînt albe.
floor.DEF.FSG and door.DEF.FSG are white.FPL
'The floor and the ceiling are white.'

f. Nucul şi prunul sînt uscaţi.
walnut.DEF.MSG and plum tree.DEF.MSG are dry.MPL
'The walnut tree and the plum tree are dry.'

This data highlights the difficulty for the view that neuter nouns are simply members of a class MSG/FPL. On this view, a coordination of two MSG nouns should be indistinguishable from a coordination of two NSG nouns, which is clearly not the case (indeed, Bateman and Polinsky 2005 explicitly leave the resolution behaviour under coordination as a problem in their account.). Table 2 summarizes.

The data considered in this section shows that Rumanian is a language which distinguishes three agreement classes (Corbett 1991:147) among nouns but has only two target genders — masculine and feminine. This mismatch phenomenon is found in other languages also — Corbett (1991) briefly discusses Telugu (Dravidian) as having three

TABLE 2 Nominal Agreement (Inanimates) under Coordination

NP1	NP2	AP	NP1	NP2	AP
NSG	NSG	FPL	FSG	MSG	FPL
FSG	FSG	FPL	FSG	NSG	FPL
MSG	MSG	MPL	MSG	NSG	FPL

controller genders and two target genders, and Lak (Caucasian) with four controller genders, three target genders in the singular and two target genders in the plural, as well as a number of other languages.

14.3 Coordination of Animate Nouns

It is well known that in some languages gender resolution in animate coordinate structures is semantically based, rather than taking account of the grammatical gender of the conjuncts. This is evident in particular when natural and grammatical gender diverge, as shown in the following example from French, a language with syntactic resolution for inanimates.

(8) La sentinelle et la personne à la barbe ont été
the sentry.FSG and the person.FSG to the beard have been
pris /*prises en otage.
taken.MPL /taken.FPL hostage
'The sentry and the person with the beard were taken hostage.'
(Wechsler 2002:10)

In general, then, we must allow for syntactic resolution to exist alongside other resolution processes in one and the same language. The following examples illustrate the resolution patterns for coordinations of animate nouns in Rumanian.[1]

(9) Maria şi tata au fost văzuti.
Maria.FSG and father.MSG were seen.MPL
'Maria and father were seen.' (Moosally 1998:112)

(10) Maria şi mama au fost văzute.
Maria.FSG and mother.FSG were seen.FPL
'Maria and mother were seen.' (Farkas and Zec 1995:94)

(11) Ion şi tata au fost văzuti.
Ion.MSG and father.MSG were seen.MPL
'Ion and father were seen.' (ibid. 95)

[1]Glosses have been added as appropriate, where they were absent from the original.

(12) un vizitator şi o turistă mult interesaţi
a visitor.MSG and a tourist.FSG very interested.MPL
'a very interested (male) visitor and a very interested (female) tourist' (Maurice 2001:237)

As these examples show, unlike coordination of inanimates, coordinations of animate nouns determine masculine agreement if any of the conjuncts are male-denoting. Confirmation that the determining factor is semantic rather than grammatical gender assignment comes both from nominals which are not (semantic) gender specific, but which are feminine in form (*persoană*, 'person'), and those which denote a male individual but are feminine in form (*popă*, 'priest').

Such nouns control agreement of adjectives, determiners, participles and predicative adjectives in terms of their grammatical gender, but participate in semantically based agreement in coordination. The pronominal anaphor referring back to nouns such as *persoană* also reflects the natural gender of the denotata.

(13) Persoană cu barbă a fost văzută. El trebuie
person.DEF.FSG with beard was seen.FSG he must
arestat imediat.
arrested.MSG immediately
'The person with a beard was seen. He must be arrested immediately.' (Farkas and Zec 1995:94)

(14) Maria şi santinelă au fost căsătoriti de catre
Maria and sentry.DEF.FSG were married.MPL by
protul local.
priest.DEF local
'Maria and the sentry were married by the local priest.' (Wechsler and Zlatić 2003:188)

(15) Maria şi persoană cu rochie au fost văzute.
Maria and person.DEF.FSG with dress have been seen.FPL
'Maria and the person with a dress have been seen.' (Farkas and Zec 1995:94)

In summary, for animates the resolution behaviour under coordination refers to natural rather than grammatical gender, and animate nouns may show a mismatch between grammatical and natural gender: Table 3 compares with Table 2 in the previous section.

TABLE 3 Animate Nominal Agreement under Coordination

NP1	NP2	Target
FEMALE	MALE	MPL
FEMALE	FEMALE	FPL
MALE	MALE	MPL

14.4 Previous Accounts

The facts outlined in the previous sections are described in the general descriptive and typological literature on agreement (Corbett 1991 is a typical example) and have attracted some theoretical attention, including Farkas (1990), Lumsden (1992), Farkas and Zec (1995), Wechsler (2002), and Wechsler and Zlatić (2003). The most comprehensive discussion is that of Farkas (Farkas 1990, Farkas and Zec 1995). In this section I briefly review this approach, which is based on underspecifying the gender value of the nouns.

Farkas (1990) takes agreement to be a directional process of feature copying from the agreement trigger to the agreement target, which initially has unspecified features. Feminine nouns are lexically specified as [+Fem], masculine as [−Fem] and neuter nouns are lexically unspecified for gender. A feature co-occurrence restriction (a feature-filling rule, applying thus only to neuter nouns) gives a gender value for neuter plural nouns:

(16) $\begin{bmatrix} < [+\text{N}][-\text{V}] > \\ +\text{PLURAL} \end{bmatrix} \rightarrow [\ +\text{FEM}\ ]$

Neuter singulars are masculine by the Elsewhere Principle, on the assumption that [−Fem] is the default value in the system. This is encoded in the following Feature Specification Default:

(17) $[\quad] \rightarrow [\ -\text{FEM}\ ]$

On this view, then, neuter singular Ns are masculine (though not lexically specified as such) and neuter plural Ns are feminine (though again, not by lexical specification). Adjectives and determiners in agreement with neuter nouns will therefore be masculine or feminine depending on the number of the noun and will acquire features in the syntax copied from the controller noun. This approach effectively holds that there are just two syntactic genders in Rumanian: the difference between masculine nouns and neuter nouns in the singular coming down to whether the [−Fem] feature is introduced lexically or by a feature specification default.

Farkas and Zec (1995), which is largely concerned with patterns of agreement under coordination for both animate and inanimate nouns, adopts a slightly revised version of this proposal. As before, neuter nouns are lexically unspecified. The following rules are postulated (ordered by the Elsewhere Condition), which provide values for the gender feature:[2]

(18) $[\emptyset\text{F}\ ] \rightarrow [-\text{F}]$

(19) $\begin{bmatrix} \emptyset\text{F} \\ \text{Number} \quad [+\text{PL}] \end{bmatrix} \rightarrow [+\text{F}]$

Thus consider an example like (5) repeated here for convenience as (20). The neuter noun is lexically unspecified for gender, but the default in (18) specifies a − value for the feature F: the noun will thus behave syntactically as a MASC noun.

(20) un scaun frumos
a.M chair.NSG beautiful.MSG
'a beautiful chair'

Consideration of Rumanian agreement patterns leads Farkas and Zec (1995) to abandon the "morphosyntactic resolution rules" approach to coordinate noun phrases. Coordinate structures are taken to be headless: in the absence of a head, the content of morphosyntactic agreement features are determined by the following generalization for animates (Farkas and Zec 1995:95):

(21) **Gender Assignment to groups (animate)**
a. If the discourse referent includes a male individual, its gender is [−F].
b. Otherwise, the referent receives no gender specification.

In case b, the rule in (19) will determine the syntactic gender assignment as feminine.

Because non-coordinate NPs are lexically headed they inherit the agreement features of the head: thus animate 'mismatch' nouns control morphosyntactic agreement (targets agree with the syntactic gender features, so that *persoană cu barbă* occurs with a FSG participle or adjective). On the other hand, pronouns are always governed by discourse factors, so that for animates, male referents determine [−F] pronouns, and female referents [+F] pronouns: thus in the case of mismatch nouns,

[2]The approach to agreement between controller and target differs from Farkas (1990) in that it is agnostic on the choice between a directional copying approach and a feature matching approach.

pronouns reflect the natural gender rather than the grammatical gender (see (13)).

For inanimate coordinate phrases, Farkas and Zec (1995:97) propose the following generalization:

(22) **Gender Assignment to groups (inanimate)**

a. If all the components of a composite discourse referent are [−F], the discourse referent inherits this specification.

b. Otherwise, the referent receives no gender specification.

Again, the intention is that if the composite discourse referent fails case a, then the rules in (18) and (19) will be relevant and provide a syntactic gender assignment.

The relevant cases concern the following contrasting behaviour between MASC and NEUT nouns under coordination. We provide the lexical specifications according to Farkas and Zec (1995) in parentheses in Table 4.

TABLE 4 Lexical Specifications

NP1	NP2	Target Morphology
MSG (−F)	MSG (−F)	MPL
MSG (−F)	NSG (∅F)	FPL
NSG (∅F)	NSG (∅F)	FPL

The intention is clearly that case b apply whenever there is a neuter conjunct, allowing (19) to determine the syntactic gender assignment to the group as FEM. But for this to happen it is crucial that the default in (18) *fail* to apply at the level of the conjuncts themselves. Otherwise the effect would be to resolve the underspecified ∅F on all the NSG nouns to −F, resulting in the assignment shown in Table 5.

TABLE 5 Specifications After (18)

NP1	NP2	Target Morphology
MSG (−F)	MSG (−F)	MPL
MSG −F	NSG −F	MPL
NSG −F	NSG −F	MPL

The problem is that it is not clear how the rules in (18) and (19) are to be prevented from applying as described to the conjuncts, which lack a lexical specification for the gender feature: note that a modifier

such as a numeral, quantifier or attributive adjective shows only a binary distinction between ±F and thus the NP will be determinate for gender, whether the agreement mechanism is feature copying or feature matching.

Working within a constraint-based formalism, Wechsler and Zlatić (2003)[3] develop a related approach, within the wider context of a theory concerning the interaction of syntactic and semantic resolution. For Wechsler and Zlatić (2003), coordinate NPs necessarily lack an inherent gender because they are headless. They postulate the following universal generalizations for such cases:

(23) Gender agreement with an animate NP that lacks inherent gender is always interpreted semantically. (Wechsler and Zlatić 2003:150)

(24) Rule for deriving gender of inanimate aggregate discourse referents:
D.R. [{ [GEND γ_1], ... [GEND γ_n] }] $\Leftrightarrow$
D.R. [GEND γ_1] $\cap$ .... $\cap\gamma_n \cap G_s$]
where γ_1 ... γ_n are null or unary sets and G_s is the set of s-gender features in the grammar (Wechsler and Zlatić 2003:152)

The rule in (24) for coordinations of inanimates states that the value of GEND for the coordinate NP is the intersection of the semantically-interpretable genders (typically masculine and feminine) of the conjunct daughters (gender features on this proposal are the empty set and singleton sets, e.g. {F}). In the case of Rumanian, as noted above, they assume that neuter nouns are simply members of a mixed class MSG/FPL, and thus it seems that all nouns will have an s-gender feature on this proposal. For inanimate coordinations falling under (24), they take FEM as the resolution class. One problem with this approach is that it predicts that a coordination of MSG with NSG will resolve in precisely the same manner as a coordination of two MSG nouns, because NSG and MSG are indistinguishable.

These accounts, then, are based on an approach which posits only two syntactic genders for Rumanian nouns. The account we develop in the following sections, on the other hand, recognises three nominal genders but only two target genders on adjectives and participles.

14.5 Targets as Underspecified

Rather than take neuter nouns as lexically underspecified for gender, or as members of a mixed class, we will propose instead that the *targets* of agreement underspecify the agreement features of their controllers.

[3]As an alternative reference, Wechsler (2002) covers precisely the same ground.

Nouns are specified as belonging to one of the three nominal genders (we will modify the expression of this approach to use sets as values for the GEND feature shortly). For example:

(25)

copac	(↑ PRED) = 'TREE'	*rochie*	(↑ PRED) = 'DRESS'
	(↑ GEND) = MASC		(↑ GEND) = FEM
	(↑ NUM) = SG		(↑ NUM) = SG
scaun	(↑ PRED) = 'CHAIR'		
	(↑ GEND) = NEUT		
	(↑ NUM) = SG		

Adjectives and determiners place constraints along the lines shown in (26).

(26)

frumoasă	*frumoşi*
((ADJ ∈ ↑) GEND) = FEM	((ADJ ∈ ↑) GEND) = MASC
((ADJ ∈ ↑) NUM) = SG	((ADJ ∈ ↑) NUM) = PL
(↑ PRED) = 'BEAUTIFUL'	(↑ PRED) = 'BEAUTIFUL'

frumos	*frumoase*
((ADJ ∈ ↑) GEND) ¬ = FEM	((ADJ ∈ ↑) GEND) ¬ = MASC
((ADJ ∈ ↑) NUM) = SG	((ADJ ∈ ↑) NUM) = PL
(↑ PRED) = 'BEAUTIFUL'	(↑ PRED) = 'BEAUTIFUL'

Entries along the lines of the first two are appropriate for all FSG and MPL modifiers — these share the gender value of their head.[4] The second two entries, for MSG and FPL forms of nominal modifiers, are underspecified: the masculine singular form cannot combine with a feminine noun but combines freely with a masculine or neuter singular, and similarly the feminine plural will combine with the feminine or neuter plural.

Participles and predicate adjectives would be similarly specified. We show below the entry for a MSG predicative adjective:[5]

(27) Un trandafir alb e scump.
a.MSG rose white.MSG is expensive.MSG
'A white rose is expensive.' (Farkas 1990:539)

[4]Some determiners are probably inflectional in Rumanian (see the data in (7)). Whether they are inflectional or co-heads, we assume they directly constrain the agreement features of the f-structure they share with the noun.

[5]For simplicity, we treat predicative and attributive adjectives by means of separate lexical entries in this paper. This fails to reflect the fact that they agree in precisely the same way with their controller — what differs is simply the *path* to the controller. The use of paths and local names in lexical entries permits the agreement properties of attributive and predicative adjectives to be given a unitary characterisation: see Otoguro (2006) for an approach to case and agreement along these lines.

(28) *scump* $\neg$ (↑ SUBJ GEND) = FEM
(↑ SUBJ NUM) = SG
(↑ PRED) = 'EXPENSIVE'

As noted above, coordination of inanimate NPs always results in NPs which control FPL agreement unless all conjuncts are MSG (see Table 2 above, and recall that reference to NP1 and NP2 in this table does not encode facts about the linear order of the conjuncts): FPL morphology on a target can correspond to a neuter or a feminine plural controller. The resolution facts may be summarised as follows:[6]

(29) Rumanian Resolution:
- If all conjuncts have the same gender, the coordinate structure has that gender.
- Otherwise the feminine form is used.

In the following section we replace the atomic gender values with set-valued features to extend our analysis to take account of agreement with coordinate (inanimate) controllers. We then turn to coordinations involving animate conjuncts.

14.6 Agreement and Coordination

Dalrymple and Kaplan (2000) propose an approach to the syntactic resolution of agreement features in coordinate structures which treats GEND as a set-valued rather than an atomic feature. On this approach, syntactic resolution reduces to the simple operation of set union. The value of the GEND feature of the coordinate structure as a whole is defined as the smallest set containing the values of the individual conjuncts, as in (30).

(30) NP $\longrightarrow$ NP CONJ NP
$\downarrow \in \uparrow$ $\downarrow \in \uparrow$
$(\downarrow \text{ GEND}) \subseteq (\uparrow \text{ GEND})$ $(\downarrow \text{ GEND}) \subseteq (\uparrow \text{ GEND})$

(31) $x \cup y$ is the smallest set z such that $x \subseteq z \ \wedge \ y \subseteq z$

The approach makes use of a notion of a *set designator* which indicates that the value of a feature is a set and also *exhaustively enumerates* the elements of the set. For example, the equation (↑ CASE) = {NOM, ACC} (in which {NOM, ACC} is a set designator) defines the value of CASE (for the f-structure in question) to be the set {NOM, ACC}, and the con-

[6] (29) interprets a coordination of NSG as resolving to NPL — the agreeing FPL form follows from the underspecified requirements placed by FPL targets (see (26) above). An alternative, which we do not pursue here, is to assume that a coordination of NSG conjuncts itself resolves to FPL under the elsewhere clause.

straint (↑ SUBJ GEND) $=_c$ {M} requires the value to be the (singleton) set {M} (Dalrymple and Kaplan 2000).

Following this approach to the GEND feature, we can represent the Rumanian nominal genders as follows:

(32) Rumanian:

MASC	{M}
FEM	{M, N}
NEUT	{N}

(33)

copac	(↑ GEND) = {M}	*rochie*	(↑ GEND) = {M, N}
	(↑ NUM) = SG		(↑ NUM) = SG
	(↑ PRED) = 'TREE'		(↑ PRED) = 'DRESS'
scaun	(↑ GEND) = {N}		
	(↑ NUM) = SG		
	(↑ PRED) = 'CHAIR'		

The lexical entries for predicative adjectives (as in 34–37) are along the lines shown in (38–41).[7,8]

(34) Un trandafir alb e scump.
a.MSG rose.MSG white.MSG is expensive.MSG
'A white rose is expensive.' (Farkas 1990:539)

(35) O garoafă albă e scumpă.
a.FSG carnation.FSG white.FSG is expensive.FSG
'A white carnation is expensive.' (ibid:539)

(36) Un scaun confortabil e folósitor.
a.MSG chair.NSG comfortable.MSG is useful.MSG
'A comfortable chair is useful.' (ibid: 540)

(37) Nişte scaune confortabile e folositoare.
some.FPL chair.NPL comfortable.FPL are useful.FPL
'Some comfortable chairs are useful.' (ibid: 540)

[7]The XLE does not appear to permit $=_c$ over closed sets as values, as shown in (38) and (43). This is encoded instead as a conjunction of constrained membership statements in the XLE:

(i) {N} $\in_c$ (↑ GEND)
{M} $\in_c$ (↑ GEND)

[8]XLE does not implement negation of closed sets as shown in (40) and (44). The negation shown on the entry for *frumos* (MSG) can be re-expressed as a negation over a conjunction of membership statements:

(i) ¬ [{N} ∈ (↑ GEND) ∧ {M} ∈ (↑ GEND)]

The negation shown for *frumoase* FPL can be re-expressed as a positive requirement that {N} is in the set.

(38) *scumpă* (SUBJ GEND *must be* FEM)
($\uparrow$ SUBJ GEND) $=_c$ {M, N}
($\uparrow$ SUBJ NUM) = SG
($\uparrow$ PRED) = 'EXPENSIVE'

(39) *scumpi* (SUBJ GEND *must be* MASC)
($\uparrow$ SUBJ GEND) $=_c$ {M}
($\uparrow$ SUBJ NUM) = PL
($\uparrow$ PRED) = 'EXPENSIVE'

(40) *scump* (SUBJ GEND *can't be* FEM)
($\uparrow$ SUBJ GEND) $\neg$ = {M, N}
($\uparrow$ SUBJ NUM) = SG
($\uparrow$ PRED) = 'EXPENSIVE'

(41) *scumpe* (SUBJ GEND *can't be* MASC)
($\uparrow$ SUBJ GEND) $\neg$ = {M}
($\uparrow$ SUBJ NUM) = PL
($\uparrow$ PRED) = 'EXPENSIVE'

For example in (37) *scaune* is lexically specified ($f1$ GEND) = {N} and the FPL adjective *folositoare* (like *scumpe* in (41)) specifies ($f1$ GEND) $\neg$ = {M}, that is, requires the GEND value not to be the closed set containing the single element {M}, hence allowing the GEND value to be either {N} or {M, N}. Given that there is a limited set of possibilities here, we can alternatively express this negative constraint as the equivalent:

(42) {N} $\in_c$ ($\uparrow$ SUBJ GEND)

Attributive adjectives place constraints along the lines shown in (43) and (44), and other NP internal modifiers such as numerals, demonstratives and quantifiers will be similar.

(43)

frumoasă (FSG)	*frumoşi* (MPL)
(($\uparrow$ ADJ $\in$) GEND) $=_c$ {M, N}	(($\uparrow$ ADJ $\in$) GEND) $=_c$ {M}
(($\uparrow$ ADJ $\in$) NUM) = SG	(($\uparrow$ ADJ $\in$) NUM) = PL

(44)

frumos (MSG)	*frumoase* (FPL)
(($\uparrow$ ADJ $\in$) GEND) $\neg$ = {M, N}	(($\uparrow$ ADJ $\in$) GEND) $\neg$ = {M}
(($\uparrow$ ADJ $\in$) NUM) = SG	(($\uparrow$ ADJ $\in$) NUM) = PL

We now turn to the coordination examples in (7), restricting attention for the moment to the behaviour of inanimate conjuncts. According to the analysis of syntactic resolution in Dalrymple and Kaplan (2000), the GEND feature of the coordinate NP as a whole is the smallest set which has the GEND values of the conjunct daughters as subsets (see (30) and (31)). Table 6 summarises the results of this analysis.

TABLE 6 Nominal Coordination with Set Values

NP1		NP2		NPCoord	Target Morph
{M N}	(FSG)	{M}	(MSG)	{M N}	FPL
{M}	(MSG)	{N}	(NSG)	{M N}	FPL
{M N}	(FSG)	{N}	(NSG)	{M N}	FPL
{N}	(NSG)	{N}	(NSG)	{N}	FPL
{M N}	(FSG)	{M N}	(FSG)	{M N}	FPL
{M}	(MSG)	{M}	(MSG)	{M}	MPL

On this first pass, the phrase structure rule for Rumanian is constrained to apply only to inanimate NPs because, as we have seen, animate NPs undergo semantic resolution under coordination. We assume that nouns are lexically specified as ANIM + or −. The following rule is restricted so that only coordinate structures in which all conjuncts are inanimate undergo resolution by set union.[9]

(45) NP ⟶ NP CONJ NP
$\downarrow \in \uparrow$ $\downarrow \in \uparrow$
$(\downarrow \text{ GEND}) \subseteq (\uparrow \text{ GEND})$ $(\downarrow \text{ GEND}) \subseteq (\uparrow \text{ GEND})$
$(\downarrow \text{ ANIM}) = -$ $(\downarrow \text{ ANIM}) = -$

To conclude this section, we observe that a simple account of the different numbers of controllers and targets can be given by the simple method of using negative conditions. Moreover, the otherwise slightly puzzling (inanimate) agreement pattern of two neuters under coordination is straightforwardly accommodated under an approach using closed sets for agreement features and set union for syntactic resolution.

14.7 Semantic Resolution

In very many languages, the sort of syntactic resolution under coordination of gender features modelled in the proposal of Dalrymple and Kaplan (2000) by set descriptors and set union is one aspect of the phenomenon and exists alongside other processes and in particular semantically-based resolution in the case of conjoined animates (see Corbett 1991, Wechsler and Zlatić 2003 for some discussion). Coordinations of animate NPs in Rumanian do not resolve syntactically, but according to the following generalization:

[9] An alternative is to declare the feature ANIM as distributive, which would additionally rule out mixed animacy coordination. We will return to this issue shortly.

(46) a. If one conjunct denotes a male animate then M is used.
b. If all conjuncts are M, then M is used.
c. Otherwise, F is used. (Corbett 1991:289)

We now consider how the approach to syntactic resolution in the previous section, directly modelled on Dalrymple and Kaplan (2000), can be combined with a formulation of semantic resolution.[10] Our approach starts from the observation that if any of the conjuncts refers to a MALE individual, then the f-structure corresponding to the coordinate structure as a whole is marked as having masculine gender. To encode the notion of reference to a male individual (or set of individuals) I posit an additional f-structure feature SEMGEND with values MALE and FEMALE. A similar feature is used to encode semantic gender in Network Morphology lexical networks, although such analyses do not deal with semantic resolution in the syntax for coordinate structures (Corbett and Fraser 2000). I assume that lexical entries which denote male individuals are lexically specified as (↑ SEMGEND) = MALE (including mismatch nouns which are syntactically FEM), and those which denote female individuals are likewise marked as (↑ SEMGEND) = FEMALE. Nouns which lack an inherent semantic gender are ambiguous out of context and thus in principle may undergo either coordination schema.[11]

For clarity, I proceed by considering first what sorts of annotations would be necessary to encode the generalization in (46). In (47), the functional uncertainty on the second conjunct daughter will be interpreted existentially: it succeeds if there is a member of the set with SEMGEND = MALE.[12]

[10]Wechsler and Zlatić (2003) discusses languages which exhibit both syntactic and semantic resolution under coordination. Although their approach is not formalized in detail, it takes LFG as its framework of reference. The essence of their proposal is that the GEND feature of an animate coordinate structure will have a semantic value while the GEND feature of a inanimate coordinate structure will have a set-valued feature. Semantically assigned values are taken to be semantic forms such as 'female', 'non-female', with the assumption that "the negatively defined semantic feature 'non-female' is not distributive (since negation itself is not distributive): a 'non-female' group is a group that fails to meet the description of a 'female' group (namely a group of females). Thus any group containing at least one male is a 'non-female' group" (Wechsler and Zlatić 2003:151). There are clearly a number of issues concerning how such an account might be formalized, but discussion of these matters would take us too far afield.

[11]The rules in (47) and (48) as formulated predict that if a mismatch noun which is MASC in syntactic gender but refers to a FEMALE individual is coordinated with another noun which refers to a FEMALE individual, the set as a whole will control FEM agreement. I do not currently have any grammatical/natural gender mismatch data to confirm or contradict this.

[12]The constraint (↑ ∈ SEMGEND) = MALE is arbitrarily placed on the second

(47) NP ⟶ NP CONJ NP
↓ ∈ ↑ ↓ ∈ ↑
(↓ ANIM) = + (↓ ANIM) = +
(↑ ∈ SEMGEND) = MALE
(↑ GEND) = {M}

This rule will only succeed if one member (at least) is MALE. Otherwise, all the daughters have SEMGEND = FEMALE and the syntactic gender is set to feminine ({M N}) for the set as a whole.

(48) NP ⟶ NP CONJ NP
↓ ∈ ↑ ↓ ∈ ↑
(↓ ANIM) = + (↓ ANIM) = +
(↓ SEMGEND) = FEMALE (↓ SEMGEND) = FEMALE
(↑ GEND) = {M, N}

These rules can be combined into one. (48) requires all the members of the coordinate set to have SEMGEND = FEMALE. Since negation in a functional uncertainty is given a wide scope interpretation, that is, is interpreted as a universal (not an existential), we can express this condition as:

(49) (↑ ∈ SEMGEND) ¬= (MALE)
(there is no member of the set for which SEMGEND = MALE is true)

(50) NP ⟶ NP CONJ NP
↓ ∈ ↑ ↓ ∈ ↑
(↓ ANIM) = + (↓ ANIM) = +
[(↑ ∈ SEMGEND) ¬= MALE
(↑ GEND) = {M, N}
| (↑ GEND) = {M}]

14.8 Combining Syntactic and Semantic Resolution

We have proposed two rules for coordinate structures, one for animate conjuncts and one for inanimate conjuncts, capturing the generalizations concerning agreement and especially agreement under coordination discussed in Farkas and Zec (1995). These rules, repeated as (51) and (52) below, follow the assumption in Farkas and Zec (1995) that coordinate NPs combine either animate or inanimate conjuncts but do not mix the two: the two separate rules in (52) and (51) ensure that the animacy features of the conjuncts matched.

conjunct and could as well be associated with the CONJ daughter. Clearly the rules can also be extended to cover additional conjuncts by adding a Kleene-plus to the first conjunct.

(51) NP ⟶ NP CONJ NP
$\downarrow \in \uparrow$ $\downarrow \in \uparrow$
$(\downarrow \text{ ANIM}) = +$ $(\downarrow \text{ ANIM}) = +$
$[\ (\uparrow \in \text{SEMGEND})\ \neg{=}\ \text{MALE}$
$(\uparrow \text{ GEND}) = \{\text{M, N}\}$
$|\ \ (\uparrow \text{ GEND}) = \{\text{M}\}\ ]$

(52) NP ⟶ NP CONJ NP
$\downarrow \in \uparrow$ $\downarrow \in \uparrow$
$(\downarrow \text{ ANIM}) = -$ $(\downarrow \text{ ANIM}) = -$
$(\downarrow \text{ GEND}) \subseteq (\uparrow \text{ GEND})$ $(\downarrow \text{ GEND}) \subseteq (\uparrow \text{ GEND})$

However though this is the assumption made in Farkas and Zec (1995), Moosally (1998) observes that this does not seem to be correct and gives the following examples of mixed animate/inanimate coordinations:

(53) Mašina şi sofer-ul au fost văzuti ieri.
car.FSG and driver.MSG have.PL been seen.MPL yesterday
'The car and the driver (male referent) were seen yesterday.' (Moosally 1998:113)

(54) Mašina şi sofer-ita au fost văzute ieri.
car.FSG and driver.FSG have.PL been seen.FPL yesterday
'The car and the driver (female referent) were seen yesterday.' (Moosally 1998:113)

The following example also shows that the "animate" strategy is not limited to human animates but extends to (at least some) animals:

(55) Pisica şi ciinele sing inometati.
cat.FSG and dog.MSG are hungry.MPL
'The cat and the dog are hungry.' (Moosally 1998:114)

Clearly more research is needed to determine precisely what the facts are here, but on the assumption that Moosally's data are correct, it is possible to give a rather succinct statement of the resolution facts for Rumanian, using templates to encode generalizations. We define first the following templates:[13]

(56) a. CONJUNCT = $\downarrow \in \uparrow$
b. RES-GEND = $(\uparrow \in \text{SEMGEND})\ \neg = \text{MALE}$
$(\downarrow \text{ GEND}) \subseteq (\uparrow \text{ GEND})$
c. SEM-GEND = $(\uparrow \in \text{SEMGEND}) =_c \text{MALE}$
$(\uparrow \text{ GEND}) = \{\text{M}\}$

[13] We ignore PERS and NUM here.

(57) NP-CONJUNCT @ CONJUNCT
@ RES-GEND | SEM-GEND

(56b) and (56c) together state the disjunction: either some conjunct has the feature SEMGEND = MALE defined, in which case the GEND of the coordinate structure is {M} (semantic resolution), or no conjunct has the feature SEMGEND = MALE defined, in which case the GEND of the coordinate structure is given by syntactic resolution (set union).

(58) NP ⟶ NP CONJ NP
@NP-CONJUNCT @NP-CONJUNCT

Clearly, it is also possible to encode relatively succinctly (by using templates) the situation holding in a language in which it is impossible to mix animate and inanimate conjuncts (i.e. a language in which (51) and (52) are the operative rules).

14.9 Conclusion and Further Data

Finally, it is worth noting that the discussion in the previous literature (Farkas 1990, Farkas and Zec 1995, Lumsden 1992, Wechsler 2002, Wechsler and Zlatić 2003) is concerned with coordinations of noncoreferring singular NPs, but additional data suggests the existence of further agreement patterns. For example, Maurice (2001) notes that with inanimates in a coordination of SG and PL it is the PL which determines agreeing forms, as shown by the following contrast:[14]

(59) Satelitul şi avionele au fost doborîte.
satellite.DET.MSG and airplane.DET.NPL have been shot.down.FPL
'The satellite and the airplanes have been shot down.' (Maurice 2001:238)

(60) Satelitii şi avionul au fost doborîţi.
satellite.DEF.MPL and airplane.DEF.NSG have been shot.down.MPL
'The satellites and the airplane have been shot down.' (ibid:238)

If two plurals are combined the predicate agrees with the closest conjunct:

(61) Sateliţii şi avioanele au fost doborîte.
satellite.DEF.MPL and airplane.DEF.NPL have been shot.down.FPL
'The satellites and the airplanes have been shot down.' (ibid:238)

(62) Avioanele şi sateliţii au fost doborîţi.
airplane.DEF.NPL and satellite.DEF.MPL have been shot.down.MPL
'The airplanes and the satellites have been shot down.' (ibid:238)

[14] In (59) we gloss the agreement form as FPL in line with our practice elsewhere, but Maurice glosses it as NPL. This has no bearing on her point.

Aurora Petan (p.c.) gives the following, with two nouns denoting the same entity, with closest conjunct agreement and a singular verb:

(63) Speranta şi viitorul meu este acest copil.
hope.FSG and future.NSG my.MSG is this child.
'My hope and future is this child.'

(64) Viitorul şi speranta mea este acest copil.
future.NSG and hope.FSG my.FSG is this child.
'My future and hope is this child.'

I leave these patterns to one side, but clearly a more comprehensive account of Rumanian agreement under coordination would have to take account of these patterns and their distribution.

This paper has shown that LFG permits a relatively simple and straightforward account of the intricacies of gender agreement in Rumanian, a language which both displays a separation between the number of target and controller gender and in which animate and inanimate noun phrases undergo different gender resolution patterns under coordination. The account posits three controller genders and two target genders and uses underspecification on targets to capture the agreement facts. The treatment of resolution under coordination builds on the set-based approach to resolution of Dalrymple and Kaplan (2000) and reduces these resolution patterns to a simple disjunction: if any conjunct is animate male, then the coordinate structure is marked as MASC, and otherwise, gender is resolved by set union.

Acknowledgments

I am grateful to the editors for the opportunity to contribute to this volume in honour of Ron Kaplan, and in celebration and appreciation of his many and varied contributions to the field. Above all I am deeply grateful to Ron for his unflagging enthusiasm for working with linguists in the formalization of linguistic descriptions and generalizations and for his commitment to communicating formal notions. Over the years I have benefitted greatly from such interactions. Thanks to Nicoleta Bateman, Kakia Chatsiou, Mary Dalrymple, Despina Kazana, Tracy Holloway King, John Maxwell, Aurora Petan, Andrew Spencer and two anonymous reviewers for data, comments and feedback which have improved this paper — all errors are mine.

References

Bateman, Nicoleta and Maria Polinsky. 2005. Rumanian as a two-gender language. Unpublished LSA handout.

Corbett, Greville G. 1991. *Gender*. Cambridge, United Kingdom: Cambridge University Press.

Corbett, Greville G. and Norman M. Fraser. 2000. Gender assignment: a typology and a model. In G. Senft, ed., *Systems of Nominal Classification*, pages 293–325. Cambridge, United Kingdom: Cambridge University Press.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, John T. Maxwell III, and Paula Newman. 2006. XLE Documentation. Palo Alto Research Center.

Dalrymple, Mary and Ronald M. Kaplan. 2000. Feature indeterminacy and feature resolution in description-based syntax. *Language* 76(4):759–798.

Farkas, Donka F. 1990. Two cases of underspecification in morphology. *Linguistic Inquiry* 21(4):539–550.

Farkas, Donka F. and Draga Zec. 1995. Agreement and pronominal reference. In G. Cinque and G. Giusti, eds., *Advances in Roumanian Linguistics*, pages 83–101. Philadelphia, PA: John Benjamins.

Lumsden, John. 1992. Underspecification in grammatical and natural gender. *Linguistic Inquiry* 23(3):469–486.

Maurice, Florence. 2001. Deconstructing gender — The case of Rumanian. In M. Hellinger and H. Bussmann, eds., *Gender across Languages: The linguistic representation of men and women*, vol. 1, pages 229–252. Amsterdam, The Netherlands: John Benjamins.

Moosally, Michelle J. 1998. *Noun Phrase Coordination: Ndebele Agreement Patterns and Cross-Linguistic Variation*. Ph.D. thesis, University of Texas at Austin.

Otoguro, Ryo. 2006. *Morphosyntax of Case: A theoretical investigation of the concept*. Ph.D. thesis, University of Essex.

Wechsler, Stephen. 2002. 'Elsewhere' in gender resolution. In K. Hanson and S. Inkelas, eds., *The Nature of the Word — Essays in Honor of Paul Kiparsky*, pages 1–39. Boston, MA: The MIT Press. To appear.

Wechsler, Stephen and Larisa Zlatić. 2003. *The Many Faces of Agreement*. Stanford, CA: CSLI Publications.

15

Animacy and Syntactic Structure: Fronted NPs in English

Neal Snider and Annie Zaenen

15.1 Animacy in Natural Language

It has long been known that whether the referent of a nominal is animate or not can be important in determining its syntactic or morphological realization. To describe this effect, researchers have proposed a number of hierarchies. The original hierarchy due to Silverstein (1976) conflates definiteness distinctions, animacy distinctions and person distinctions into one ordering called the 'animacy hierarchy'. We follow Aissen (2003) (based on Croft 1988) in distinguishing separate hierarchies because they refer to different aspects of entity representation within language: the *definiteness dimension* is linked to the status of the entity as already known or not yet known at a particular point in the discourse, the *person hierarchy* depends on the participants within the discourse, and the *animacy status* is an inherent characteristic of the entities referred to. We moreover assume that the traditional definiteness hierarchy, which looks at the morphological marking of the nominal, is in fact a proxy for an ordering according to information status (see below). Each of these three aspects, however, contributes to making entities more or less *salient* or *accessible* at a particular point in the discourse.

As long as one's attention is limited to the distinction between grammatical and ungrammatical sentences, the importance of the animacy hierarchy is mainly relevant for languages with a richer morphology than English. In such languages animacy distinctions can influence

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

grammaticality of, e.g., case-marking and voice selection. To give just two examples, in Navaho, a *bi*-form is used when the patient is animate and the agent is inanimate, whereas the *yi*-form is used when the agent is animate and the patient is inanimate, as illustrated in (1) (from Comrie 1989:193).

(1) a. At'ééd nímasi bi-díílíd
girl potato burnt
'The potato burnt the girl.'

b. At'ééd nímasi yi-díílíd
girl potato burnt
'The girl burnt the potato.'

In Spanish, animate direct objects are introduced by *a*, whereas inanimates are bare NPs, as illustrated in (2):

(2) a. Vi *el* libro.
'I saw the book.'

b. Vi *al* niño.
'I saw the child.'

As discussed, inter alia, in Aissen (2003), this animacy distinction interacts with definiteness. It is conceptually desirable to distinguish between animacy and definiteness or information status but in practice it is frequently the case that a linguistic phenomenon is conditioned by multiple conceptually independent factors (see e.g. Comrie 1989 for discussion). This needs to be reflected in the way the data is analyzed and modeled. We discuss this later in more detail.

Recent linguistic studies have highlighted the importance of animacy distinctions in languages such as English. For instance, the choice between the Saxon genitive and the *of*-genitive (Leech et al. 1994, Rosenbach 2002, 2003, O'Connor et al. 2004), between the double NP and the prepositional dative (Bresnan et al. 2005), between active and passive (Bock et al. 1992, McDonald et al. 1993) and between pronominal and full noun reference (Dahl and Fraurud 1996, based on Swedish data) have all been shown to be conditioned by the animacy or inanimacy of the referents of the arguments whose realization can vary. In these cases, the difference between animate and inanimate does not lead to a difference between a grammatical or an ungrammatical sentence, as in the cases exemplified above, but to a difference in acceptability.

As some of the references given above indicate, psycholinguists too have investigated the importance of animacy in language. For them it is one of the many factors that play a role in sentence production. The mainstream hypotheses rely on a notion of salience or accessibility that

is an amalgam of different functional factors. Prat-Sala (1998), for instance, lists the following from the literature: predictability, semantic priming, animacy, concreteness, prototypicality, as well as some that are linked to the shape of the words: word length, metrical structure, phonological priming and word frequency. In her own study she adds discourse factors. The lists are not exhaustive but it is clear that it is difficult to study all the listed factors at once. Ideally, though, this should be done as these factors combine to make one entity more salient than another. In experimental studies one typically tries to keep all the factors except one constant. In corpus studies one has to try to tease out the contribution of the various factors statistically. In the study reported on here, we examined the influence and the interactions of weight (represented by the number of words in a constituent), information status and animacy. In this paper we focus on what the results tell us about the role of animacy. Information status is discussed in more detail in Snider (2006).

There are currently two main hypotheses about how animacy influences syntactic realization: the first assumes that the grammatical function realization of a semantic argument follows the 'Syntactic Accessibility Hierarchy' or Grammatical Function hierarchy postulating the following ordering: SUBJ > OBJ > OBJ2 > OBL (henceforth GFH; see, for instance, Bock and Warren 1985). The second hypothesizes that the surface linearization of arguments is directly conditioned by accessibility (henceforth WOH, see for instance Kempen and Harbush 2004).[1]

For English it is difficult to distinguish between these two hypotheses and several sets of data are compatible with both, e.g. the studies done on the choice between passive and active in Prat-Sala (1998), Bock et al. (1992), and McDonald et al. (1993).

15.2 Animacy and Fronting Constructions

One way of distinguishing between the two hypotheses is to look at other languages where word order precedence does not correlate with the Grammatical Function hierarchy. Another way is to look at cases in English where elements come earlier in a sentence without being higher on the Grammatical Function hierarchy. In this paper we pursue the latter and present a comparison of the occurrence of animates and inanimates in Left Dislocation and Topicalization with the occurrence

[1] Hierarchies are generally assumed to be totally ordered, but in fact most studies only look at two adjacent elements, e.g. subjects and objects, or objects and second objects.

of animates and inanimates as in situ arguments.

The two constructions studied are illustrated in (3):

(3) a. Topicalization: 'Brains you're born with. A great body you have to work at.' [Brooke Shields, in health club commercial]
b. Left Dislocation: 'That guy, I met him last week in the grocery store.'

Note that we use *Topicalization* here to refer to a syntactic construction without implying a link to pragmatic topichood. In fact, this syntactic configuration most likely has several different pragmatic uses. We will refer to the sentence initial constituents in these constructions as the *fronted elements* and to the subcategorization relation between the fronted element and the verb as the in-situ grammatical function. For Left Dislocation, the in-situ realization of the fronted element is a pronoun, while for Topicalization there is no overt realization.[2] We did not study other fronting constructions that might occur in the corpus. In examining the role of animacy in these two constructions, we need to extend the hypotheses given above to these cases. As stated, they do not take long distance dependencies into account. Especially for Topicalization, we need to examine the importance of the in-situ grammatical role that the fronted element is linked to. In written text, it is impossible to see whether the subject of the highest clause is 'topicalized' or in-situ. Only the Topicalization of a subject in an embedded clause leads to a marked word order and such cases are very rare (in our corpus, Topicalization from non-subject position is 330 times more frequent than Topicalization from subject position). Subjects tend to be animate, and topicalized NPs overwhelmingly have a gap in a non-subject position. Taking this into consideration, the extension of the predictions of WOH is rather straightforward: given that the hypothesis takes surface word order as the factor that animacy influences, it predicts that, when other factors are controlled for, fronted elements will tend to be animates, regardless of their link to in-situ grammatical functions. Under the assumptions of GFH, it is not so clear what to expect. One could assume topicalized NPs to be animate or non-animate to the same degree as a referent realized in the in-situ position would be. For Left Dislocation, one could reason that the anaphoric binding to a pronoun in-situ might also lead to the same animacy preferences as an in-situ constituent, although this reasoning seems to be weaker in this case than in the case of Topicalization because the identification

[2]We realize that it is not so clear what should go under the term Left Dislocation, but we have limited ourselves to cases where the fronted element can be linked to an in-situ pronoun.

between the fronted constituent and the pronoun can be considered to be weaker (in LFG terms, anaphoric binding versus functional control). Alternatively, one could place the fronted functions on the hierarchy of grammatical functions. From a purely formal point of view, one can propose that they are lower or that they are higher than the subcategorized functions.[3]

To analyze the data we use logistic regression because it gives the researcher great power to determine the factors that influence construction choice, even in rare constructions such as Left Dislocation and Topicalization. In particular, logistic regression provides a means to overcome problems that are inherent in the use of corpus data: correlated variables and multiple speakers. Before going into more details about methods and results we briefly describe the corpus data we used.

15.2.1 The Data

The corpus

We use a part of the Switchboard corpus, a corpus of spoken English that was compiled from telephone dialogues involving speakers from different parts of the United States (Godfrey et al. 1992). This corpus has been annotated over the years to make it more useful for syntactic studies. The Treebank Project (Marcus et al. 1993) of the Linguistic Data Consortium released a version of Switchboard annotated for part of speech and hierarchical syntactic structure. We relied on this annotation to extract the Left Dislocations and Topicalizations from the corpus.

The Switchboard corpus has also been annotated for various semantic and pragmatic features. The Edinburgh-Stanford LINK project on Paraphrase annotated the nominals in subsections of the corpus for animacy and for information status (see Nissim et al. 2004 and Zaenen et al. 2004 for a detailed description of these efforts). The animacy annotation was done for the whole syntactically annotated subpart of the corpus by Stanford, while Edinburgh did the information status annotation for a smaller subpart.

We will briefly describe how the nominals in the corpus were annotated for the various factors and how this information was used in the current study. For a more extensive discussion see Snider and Zaenen (2006).

[3]The psycholinguistic interpretations of these two choices are not equally natural, as we will discuss in the conclusion.

Animacy categories

One major problem with devising an animacy hierarchy is that the linguistically relevant notion of animacy does not directly correspond to biologically-based distinctions. Another is that it is not clear how many distinctions are linguistically relevant nor whether the same distinctions play a role in all languages. Binary animacy distinctions such as human/non-human and animate/inanimate have been proposed as well as more fine-grained ones.

The LINK project opted for a nine-valued scale based on Garretson et al. (2004), distinguishing *humans* (HUM), *organizations* (ORG), *animals* (ANIMAL), *intelligent machines* (MAC), *vehicles* (VEH), other *concrete entities* (CONC), *places* (PLACE), *times* (TIME), and other *non-concrete entities* (NONCONC).[4]

The nominals in the syntactically annotated part of the Switchboard were coded for these categories by three annotators. Evaluation (Zaenen et al. 2004) showed that the interannotator agreement in general was very high. But it also showed that some of the distinctions were not reliably annotated. Moreover, the amount of data that we have available in the corpus for Topicalization and Left Dislocation does not allow us to make a nine-way distinction. There would have been too many variables for the number of facts. Therefore, we collapsed the nine-valued scale into a binary distinction between *animates* and *inanimates*, the animates comprising HUM, ORG, ANIMAL, MAC, and VEH, and the inanimates of CONC, PLACE, TIME, and NONCONC.

The details of the information status annotations are not the focus of this paper. They are described in Nissim et al. (2004) and discussed in Snider (2006). Suffice it to say that a three-way distinction between *old*, *mediated*, and *new*, based on Prince (1981), was used.

15.2.2 Characteristics of the data and analysis techniques

The Switchboard corpus records the speech of a great variety of speakers, but we want to come as close as possible to a general picture of the speech community as a whole. Therefore we need to control for speaker variation. It is also well known that animacy effects are correlated with other factors such as information status (definiteness), as pointed out in the introduction. To overcome these problems we used logistic regression techniques.

[4]The somewhat esoteric categories 'vehicle' and 'intelligent machine' were introduced because it has been claimed that humans treat moving objects like cars and computers as human. There were not enough examples of these categories in the corpus to test this.

Correlated factors

Correlations pervade naturalistic linguistic data. This has often led to reductive theories that attempt to explain the data in terms of one factor. To give just one example, Hawkins (1994) proposes a theory of linear order in sentences. He postulates that shorter expressions occur earlier in sentences to facilitate processing and assumes that because discourse givenness is correlated with shorter, less complex constituents, apparent effects of givenness reduce to the preference to postpone syntactically complex (longer) phrases later. Arnold et al. (2000) and Bresnan et al. (2005) show how regression techniques can be used as a test of such theories. Logistic regression allows one to control for many factors, even correlated ones, simultaneously, so their independent effects can be measured. Bresnan et al. (2005) found that givenness, animacy, and length factors all have independent effects in predicting the dative alternation.

As we said above, the factors interacting with animacy that we are considering in this study are grammatical function, information status and weight. Logistic regression allowed us to determine the independent effects of these factors, and as we will see in more detail in the next section, all the factors do indeed have an independent effect.

Multiple Speakers

Another possible problem with corpora is that the data is pooled from many different speakers. Corpora such as the Switchboard corpus used here explicitly include data from many different speech communities. Newmeyer (2003) claims "There is no way that one can draw conclusions about the grammar of an individual from usage facts about communities, particularly communities from which the individual receives no speech input." Bresnan et al. (2005) point out that this is an empirical question, one that can be answered using modern statistical techniques. They use *bootstrap sampling* to show that data from different speakers does not affect their logistic regression model, which supports the idea that their conclusions about the dative alternation represent generalizations about many English speech communities, whose differences are not significant relative to other substantive factors. Another way to control for different speakers is the one employed here, the mixed model. In a mixed model logistic regression, the speaker is modeled as a random factor, in that each speaker is allowed to have a different base rate of producing the construction in question, assuming only that the inter-speaker variation in rate is normally distributed. Once the presence of different speakers is controlled in this way, the model allows us to draw conclusions about the factors that independently influence

their construction choice.

15.2.3 Interpreting logistic regression models

Regression models are well suited to corpus analysis because they allow one to determine the effects of factors, while controlling for others. They do this by fitting a mathematical model which contains coefficients for all the factors in the data relevant to the researcher's hypotheses. The models used in corpus analyses such as this one are a special type of regression, called logistic regression, which is suited for modeling categorical dependent variables (such as construction choice). These models predict the *odds ratio* of occurrence. Odds ratios should be familiar from their use in horse racing: a bookmaker might put the odds of a horse winning as 1 : 10 or 0.1, that is, there is a $\frac{1}{11} = 9.1\%$ chance the horse will win. In a linguistic example, the odds of a particular construction occurring, as opposed to another, might be 50%, with an odds ratio of 1. When a logistic regression models the effects of various factors on the odds ratio, the coefficients associated with the factors are interpreted in terms of how much they increase (or decrease) the odds ratio of the construction's occurrence. If the factor is itself categorical, say animate vs. inanimate, then the regression coefficient is interpreted as how much more likely one value makes the construction to occur over another value of the factor. For example, if the value for the animacy coefficient is 2, then animates make the construction twice as likely to occur as inanimates. If the factor is continuous, like a length in words, then the coefficient represents the increase in odds of the construction for each increment of the factor. For example, if the length coefficient has an odds ratio of 0.5, then the construction is 50% more likely for each one word increase in length. Finally, when using logistic regressions to draw inductive inferences about the general behavior of a construction, and not merely the specific structure of the corpus, one needs to be careful not to build models that "over-fit" the data. This can happen when the model has more degrees of freedom than the data allow. This caveat is relevant to this work in that the topicalization data set is so small, it only allows three degrees of freedom (three independent variables) per model. In the topicalization data below, we report results for more than three independent variables, so all the results were verified by constructing sub-models that were limited to three degrees of freedom. Most importantly, the animacy results are the same in a model that only contains animacy, information status, and grammatical function as predictors.

TABLE 1

Factors	Coefficient	F-value	p-value
(Intercept)	-2.032766	57.61879	$< .0001$
animacy		9.36251	0.0022
inanimate vs animate	1.477782		
status		4.48857	0.0113
old vs mediated	-1.908018		
new vs mediated	-0.199473		
gf		89.80565	$< .0001$
subj vs non-subj	-5.846039		
weight	0.219027	6.23979	0.0125

TABLE 2

Topicalization	Topics	Subjects	Objects
animates	10	8929	2237
inanimates	87	2665	7135

15.3 Results

15.3.1 Topicalization

As predicting factors, we used a binary animate/inanimate distinction, three values for information status, as well as weight and in-situ grammatical function. For Topicalization, the coefficients for the linear logistic regression are as given in table 1.

As the table shows, all the factors are independently significant predictors of Topicalization. In odds-ratio terms: *mediated* nominals are 6.7 times more likely to be topicalized than *old* nominals, and *mediated* nominals are 1.2 times more likely to be topicalized than *new* ones. The first result is not surprising in the light of a theory of Topicalization such as that of Prince (1998). What is more surprising is the second: that new information tends to be in fronted position more often than old. This is discussed further in Snider (2006).

The big surprise, however, is that inanimates are 4.3 times more likely than animates to be in a topic-position. This result contradicts WOH rather directly, but it is also not in agreement with GFH: at first one might be tempted to attribute this result to the fact that most topicalized constituents are linked to non-subject in-situ elements. As animacy correlates strongly with grammatical function, with animates being attracted toward higher levels on the grammatical function hierarchy, one might be tempted to conclude that most topicalized constituents tend to be inanimate by virtue of their link to non-subject

table 3

Factors	Coefficient	F-value	p-value
(Intercept)	-1.212015	360.0704	$< .0001$
animacy		3.8718	0.0492
inanimate vs animate	-0.420939		
status		27.3778	$< .0001$
old vs mediated	-1.961611		
new vs mediated	-0.270017		
gf		157.8188	$< .0001$
subj vs non-subj	-3.209837		
weight	0.401827	144.0611	$< .0001$

grammatical functions and interpret the results as being in favor of one way of amending GFH described in 15.2. But the independent effect of animacy shows that this cannot be the whole explanation for the prevalence of inanimates in topic position: the tests above show that animacy has an independent effect at $p = 0.0022$ and that the effect is that inanimates are favored in topic-position. If we look at the raw numbers given in table 2, we see that these also show that the distribution of animates and inanimates is very different in topic position from what it is in subject position, as we overwhelmingly find inanimates in topic position. The raw data also show that the distribution is different from that of animates and inanimates in object position: the proportion in topicalized position is 1 to 9 whereas for objects it is about 2 to 8.

With respect to weight, the results show that heavier constituents are more likely to be topicalized than light ones, again a result that goes against the grain of purely linear order-based accounts.

15.3.2 Left Dislocation

Information status is significant at $p < .0001$. Thus, it is clear that information status is a significant predictor of Left Dislocation, with *mediated*-coded entities most likely to left-dislocate. A *mediated* NP is 1.3 times more likely to be in a left-dislocated position than a *new* NP, and *mediated* NPs are 7.1 times as likely to be left-dislocated than *old* NPs. These results will be discussed further in Snider (2006). Here we just note that the behavior of *mediated* elements is as expected, but the ratio between *new* and *old* is somewhat surprising given most theories about Left Dislocation. Animacy and grammatical function are also significant factors. And the tests above show that, for this construction too, each of these factors has an independent effect, because each signif-

TABLE 4

Left Dislocation	LD	Subjects	
animates	227	8929	
inanimates	173	2665	$p < .001$

TABLE 5

Factors	Coefficient	F-value	p-value
(Intercept)	-2.40407	1866.1625	$< .0001$
animacy		2.7000	0.1004
inanimate vs animate	0.13064		
gf		516.2203	$< .0001$
subj	-3.03187		
weight	0.41943	1041.6754	$< .0001$

icantly increases the likelihood of the model when added individually. Grammatical function is significant at the $p < .0001$ level, and animacy is significant at the $p < 0.05$ level. In odds-ratio terms, animates are 1.5 times more likely than inanimates to be in a Left Dislocation construction. Here the results are weakly consistent with GFH. When we look at the raw numbers for left-dislocated elements and subjects in table 4, we see that indeed they are more similar, although the proportion of inanimates is higher in Left Dislocation.

The role of weight is similar to that in Topicalization.

15.3.3 Analysis of Larger Animacy Set

In order to further test the animacy effects, we analyzed a larger data set. This was possible because, as mentioned above, the animacy annotation has greater coverage than the information status annotation. In this data set, we were able to use all 399 Left Dislocations and 106 Topicalizations, but the models had fewer factors (only animacy, GF, and weight). Tables 5 and 6 show that using more data and fewer factors, the animacy effect in Left Dislocation disappears ($p > 0.1$) and strengthens for Topicalizations. These differences might just be a measure of the effect of information status that is now lost given that we now have a much poorer model. But they suggest that the anti-animacy effect in Topicalization is not a fluke due to the small dataset.

TABLE 6

Factors	Coefficient	F-value	p-value
(Intercept)	-3.237999	161.16239	$< .0001$
animacy		48.39225	$< .0001$
inanimate vs animate	1.972179		
gf		144.72518	$< .0001$
subj	-6.949687		
weight	0.318986	82.20584	$< .0001$

15.4 Discussion

There have been no previous studies that examined the effects of animacy on Left Dislocation and Topicalization. The above results suggest that there is no effect of animacy on left-dislocated NPs and show a tendency for topicalized NPs to be inanimate. The logistic regression shows that this inanimacy effect for Topicalizations is not merely due to the fact that they are extracted from a non-subject position, where inanimates are preferred, because this factor was included in the regression model.

One should not read too much into the results of one rather small study. We need to do a further analysis of the data to see whether other factors might play a role. For instance, the relative animacy of the subject and the fronted element might be important. Our impression is that in the Switchboard corpus, sentences with fronted elements tend to have animate subjects, but we have not counted them. Another, perhaps more promising hypothesis to pursue is the following: Topicalization involves a long distance dependency and one can hypothesize that as such it carries an extra processing load. It might be that to compensate for this, the argument structure of the sentences where it is used tends to be canonical, i.e. of the animate subject, inanimate object type. If there is such a tendency, it could lead to the results we found. Whatever the exact explanation, if the result stands, the tendency for inanimates to topicalize, documented here, is problematic for theories of production that predict the saliency of referents to directly influence linearization of NPs in the clause (Kempen and Harbush 2004). Such theories would predict that a construction that caused a referent to occur first in the clause would choose the most salient referent. Animate NPs are inherently more salient than inanimates, so these theories would predict that animates should topicalize more. The data in this study show that this is not the case. There is other evidence that a simple 'animate-first' theory is inadequate. For

instance, data from the ordering of temporal adjunct PPs shows that the animacy of the subject does not affect the realization of adjunct PPs before the subject or at the end of the clause (Cueni et al. 2005).

Our results do not support a simple version of the Grammatical Function Hierarchy hypothesis either: according to one version of that hypothesis there would be no effect on Topicalization or Left Dislocation of animacy, but we see that for Topicalization there is a negative effect. As we discussed, this effect cannot be due to the overwhelming influence of information status: even when information status is taken into account, the effect remains. If the explanation is a version of the processing load hypothesis we sketch above, there could be a version of the GFH that is compatible with it.

To make things more complicated, our results contradict WOH, but there are data, from languages other than English, in favor of WOH. Kempen and Harbush (2004) show that in the middle field in German, GFH does not hold but that the hypothesis that animacy is correlated with simple surface word order accounts for the data. One could reconcile the data of our study and those of Kempen and Harbush (2004) by proposing that the linear order effects occur only in the sentence internal domain, the IP domain, where subcategorization plays a direct role, and not in the periphery, the CP domain. This is descriptively adequate but it is puzzling for some psycholinguistic models.

From a psycholinguistic perspective, it is plausible to assume that salient entities are accessed and expressed first. However, here we see that elements sometimes occur first despite the fact that they should be less salient, according to the usual criteria of salience (animacy in this case). At first this may suggest that the influence of salience on ordering might be more construction bound than has been assumed: when a so-called unmarked order is used, the most salient element comes first but marked constructions can be used in which other elements are first. This way of interpreting the data, however, might call into question the very notion of salience that is used: an inanimate in topic position does not strike one as less salient, it rather strikes one as an non-typical salient element. The construction seems to treat as salient an element that is not salient by normal criteria.[5] We do not have a model of sentence production that would allow for such construction-dependent reversals in signaling salience, but it is clear that more attention needs to be paid to variation in syntactic constructions than is done in the production models currently proposed.

[5]Note also that order alone is not what achieves this effect: in Left Dislocation the effect of animacy is not the same as in Topicalization, but both are fronting constructions.

Acknowledgments

The second author thanks Ron for years of interesting discussion and collaboration on various linguistic issues. Thanks also to Tom Wasow and Anna Cueni for initial discussions and to Tom, two anonymous reviewers, and the participants in the Spring 2006 Laboratory Syntax Class for comments. All remaining errors are as usual our own.

References

Aissen, Judith. 2003. Differential object marking: Iconicity vs. economy. *Natural Language and Linguistic Theory* 21:435–483.

Arnold, Jennifer, Thomas Wasow, Anthony Losongco, and Ryan Ginstrom. 2000. Heaviness vs. newness: The effects of structural complexity and discourse status on constituent order. *Language* 76:28–55.

Bock, J. Kathryn, Helga Loebell, and Randal Morey. 1992. From conceptual roles to structural relations: Bridging the syntactic cleft. *Psychological Review* 99:150–171.

Bock, J. Kathryn and Richard K. Warren. 1985. Conceptual accessibility and syntactic structure in sentence formation. *Cognition* 21:47–67.

Bresnan, Joan, Anna Cueni, Tatiana Nikitina, and Harald Baayen. 2005. Predicting the dative alternation. In *Proceedings of the Royal Netherlands Academy of Science, Workshop on Foundations of Interpretation*. Amsterdam, The Netherlands.

Comrie, Bernard. 1989. *Language Universals and Linguistic Typology: Syntax and Morphology*. Chicago, IL: University of Chicago Press.

Croft, William. 1988. Agreement vs. case marking and direct objects. In M. Barlow and C. A. Ferguson, eds., *Agreement in Natural Language: Approaches, Theories, Descriptions*, pages 159–179. Stanford, CA: CSLI Publications.

Cueni, Anna, Neal Snider, and Annie Zaenen. 2005. Boundaries to the influence of animates. Paper presented at the LSA Annual Winter Meeting.

Dahl, Osten and Kari Fraurud. 1996. Animacy in grammar and discourse. In T. Fretheim and J. K. Gundel, eds., *Reference and Referent Accessibility*, pages 47–64. Amsterdam, The Netherlands: John Benjamins.

Garretson, Gregory, Mary C. O'Connor, Barbora Skarabela, and Marjorie Hogan. 2004. *Coding practices used in the project Optimal Typology of Determiner Phrases*. Boston, MA: Boston University.

Godfrey, John, Edward Holliman, and Jane McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'92)*, pages 517–520. San Francisco, CA.

Hawkins, John A. 1994. *A Performance Theory of Order and Constituency*. Cambridge, United Kingdom: Cambridge University Press.

Kempen, Gerard and Karin Harbush. 2004. A corpus study into word order variation in German subordinate clauses: Animacy affects linearization independently of grammatical function assignment. In T. Pechmann and C. Habel, eds., *Multidisciplinary approaches to language production*, pages 173–181. Berlin, Germany: Mouton De Gruyter.

Leech, Geoffrey, Brian Francis, and Xfueng Xu. 1994. The use of computer corpora in the textual demonstrability of gradience in linguistic theories. In C. Fuchs and B. Victorri, eds., *Continuity in linguistic semantics*, pages 57–76. Amsterdam, The Netherlands: John Benjamins.

Marcus, Mitch, Beatrice Santorini, and Mary A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics,* 19:313–330.

McDonald, Janet L., J. Kathryn Bock, and Michael H. Kelly. 1993. Word and word order: Semantic, phonological, and metrical determinants of serial position. *Cognitive Psychology* 25:188–230.

Newmeyer, Frederick. 2003. Grammar is grammar and usage is usage. *Language* 79:682–707.

Nissim, Malvina, Shipra Dingare, Jean Carletta, and Mark Steedman. 2004. An annotation scheme for information status in dialogue. In *Proceedings of the 4th Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal.

O'Connor, Mary C., Arto Anttila, Vivienne Fong, and Joan Maling. 2004. Differential possessor expression in English: Re-evaluating animacy and topicality effects. Paper presented at the Annual Meeting of the Linguistic Society of America, Boston, MA.

Prat-Sala, Mercè. 1998. *The Production of Different Word Orders: A Psycholinguistic and Developmental Approach*. Edinburgh, United Kingdom: Doctoral dissertation, University of Edinburgh.

Prince, Ellen. 1981. Toward a taxonomy of given-new information. In P. Cole, ed., *Radical Pragmatics*, pages 223–256. New York, NY: Academic Press.

Prince, Ellen. 1998. On the limits of syntax, with reference to left-dislocation and topicalization. In P. Culicover and L. McNally, eds., *Syntax and semantics. Vol. 29. The limits of syntax*, pages 281–302. New York, NY: Academic Press.

Rosenbach, Anette. 2002. *Genitive Variation in English. Conceptual Factors in Synchronic and Diachronic Studies*. Berlin, Germany/New York, NY: Walter de Gruyter.

Rosenbach, Anette. 2003. Aspects of iconicity and economy in the choice between the s-genitive and the of-genitive in English. In G. Rohdenburg and B. Mondorf, eds., *Determinants of Grammatical Variation in English*. Berlin, Germany/New York, NY: Walter de Gruyter.

Silverstein, Michael. 1976. Hierarchy of features and ergativity. In R. M. W. Dixon, ed., *Grammatical Categories in Australian Languages*, pages 112–171. Canberra, Australia: Australian Institute of Aboriginal Studies.

Snider, Neal. 2006. Left dislocation and topicalization in English (working title). Unpublished Manuscript.

Snider, Neal and Annie Zaenen. 2006. Animacy and salience. Unpublished Manuscript.

Zaenen, Annie, Jean Carletta, Gregory Garretson, Joan Bresnan, Andrew Koontz-Garboden, Tatiana Nikitina, Mary C. O'Connor, and Tom Wasow. 2004. Animacy encoding in English: Why and how. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), Workshop on Discourse Annotation*, pages 118–125. Barcelona, Spain.

16

Accounting for Discourse Relations: Constituency and Dependency

Bonnie Webber

At the start of my career, I had the good fortune of working with Ron Kaplan on Bill Woods' Lunar system (Woods et al. 1972). One day, in talking with Ron, I marvelled to him over the range of syntactic constructions I was able to implement in Lunar's ATN grammar formalism. Ron replied was that you could implement *anything* in an ATN: the point was, rather, to identify the minimal machinery required for a task. This sensible advice I subsequently sought to follow, and in this paper for Ron's festschrift, I try to apply it to understanding and comparing accounts of discourse relations.

16.1 Introduction

Theories of discourse that attempt to explain how the meaning of a text is more than the sum of the meaning of its component sentences have often been presented in ways that discourage easy comparison. One attempt to remedy this was made by Moore and Pollack (1992), who suggested a distinction between an *intentional* organization of discourse and an *informational* organization. Intentional organization could be described in terms of the speaker's plans with respect to his/her utterances and how the parts of the plan relate to each other (Grosz and Sidner 1990, Lochbaum 1998), while informational organization could be described in formal semantic and pragmatic terms, with discourse relations such as *consequence*, *cause*, *contrast*, *narration*, etc.

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

While this solved part of the problem, it did not contribute to understanding how theories concerned with the informational organization of discourse differ from one another. In this paper, I want to suggest that recasting them in terms of the common linguistic concepts of *constituency* and *dependency* might help us to better understand their similarities and differences.

16.2 Constituency and Dependency

16.2.1 Constituency

In syntax, *constituency* has been defined in the following terms:

1. Sentences have parts, which may themselves have parts.
2. The parts of sentences belong to a limited range of types.
3. The parts have specific roles or functions within the larger parts they belong to. (Huddleston and Pullum 2002:20)

Of course, different theories of syntax posit different parts, and different parts may be appropriate for different languages. Nevertheless, it is this idea of parts within parts that is basic to *constituency*, as is the idea that parts have specific roles or functions. Both ideas are integral to the compositional interpretation of constituent structure (i.e., *compositional semantics*).

Another aspect of constituency is that parts are *continuous spans*. A constituent can become *discontinuous* if (1) part of it moves somewhere else — e.g., to the front of the sentence in questions (the discontinuous PP in "Which pocket is your wallet in?"), to the rear of the sentence when the informational content of a relative clause is more than that of the main predicate (the discontinuous NP in "A man came in who was wearing a green hat."); or (2) the constituent is interrupted by a parenthetical phrase (the discontinuous VP in "I can tell you, since you ask, more about my parents."). But even in languages with free word order, the parts of one constituent will not be arbitrarily scrambled with those of others.

16.2.2 Dependency

There are several notions of *dependency* in linguistics. One is a *syntactic* notion related to morphology, where the morphological realization of a lexico-syntactic element depends on another element elsewhere in the sentence or clause, including realisation as an empty element or *gap*. Such a syntactic dependency may be unbounded or bounded.

In English, bounded syntactic dependencies include gender and number agreement, as in:

(1) a. These$_i$ boys$_i$ shave$_{(i)}$ themselves$_i$.

b. This$_i$ boy$_i$ shaves$_{(i)}$ himself$_i$.

Here, i is used to co-index the dependent element (simple subscript) and the element it depends on (parenthesized subscript). The dependency shows itself in the morphological features of the determiner, noun, verb and reflexive pronoun.

Unbounded dependency constructions in English (Huddleston and Pullum 2002:1079) contain a *gap* in a position that syntax requires to be filled, as in

(2) a. This is [the book]$_{(i)}$ which$_i$ I think Fred said he wrote $\_\_{}_i$.
b. [The other chapters]$_{(i)}$ I think Fred said he wrote $\_\_{}_i$ himself.

Here $\_\_{}_i$ indicates the gap in the object position in both (2a) and (2b), with i being the index of the element it depends on. *Unbounded* refers to the arbitrary depth to which the clause containing the gap can be embedded. Syntactically, *dependency* refers to the fact that the relative pronoun in (2a) and the topicalized NP in (2b) require an associated gap. The semantic consequences are that the gap in both examples draws its interpretation from the constituent that it is co-indexed with and that the topicalised NP in (2b) would be taken as being in contrast with something else in the discourse context.

A second notion of dependency between words underlies *dependency grammar*. Hudson defines this sense of dependency in terms of *support*: A dependent word is *supported* by the word it depends on, which allows it to occur in a sentence and also constrains such features as its possible location in a sentence and its possible morphological form.[1] This generalizes the previous notion of dependency, although it is limited to word-to-word relations, rather than relations between lexico-syntactic constituents, including gaps.

While both these notions of dependency are syntactic, they can also have semantic consequences in terms of how a sentence is interpreted. On the other hand, *semantic dependency* affects only interpretation — and in particular, truth-conditional semantics. Linguistic elements that exert such influence include quantifiers (3a), negation (3b) and adjuncts (3c), and what they can influence includes both reference and truth values.

(3) a. I need *a student* to work on *every project*.
b. I do *not* love you *because you have red hair*.
c. *A woman* has been elected president *for the second time*. (Huddleston and Pullum 2002:719)

[1] http://www.phon.ucl.ac.uk/home/dick/enc/syntax.htm#dependency, *The Encyclopedia of Dependency Grammar*, accessed April 2006.

Their range of influence is called their *scope*, which will be either *narrow* or *wide*. For example, if *every project* is taken to have narrow scope in (3a), the term *a student* will be interpreted as referring independently of the set of projects being iterated over (meaning that one student is needed for the whole set of projects). With wide scope, *a student* will be interpreted as dependent on the set, meaning that the student that is needed depends on the project. In (3b), it is a truth value that is affected: if *not* has narrow scope, it is the proposition headed by *love* that is negated (i.e., because you have red hair is the reason I *don't* love you), while if it has wide scope, it is the proposition headed by *because* that is negated (i.e., because you have red hair is *not the reason* that I do love you). Finally, in (3c), if the adjunct *for the second time* is taken to have narrow scope, the referent of *a woman* is independent of the election (i.e., the same woman has been elected twice), while with wide scope the referent is dependent on it (i.e., a possibly different woman has been elected each time). It should be clear that scope does not correlate directly with relative linear order: within a clause, a scope-bearing element can appear to the left or right of the elements it has scope over.

The final notion of dependency is *anaphoric dependency*. Here, all or part of an element's interpretation depends on what is available in the discourse context. For example, the interpretation of *he* in (4a) depends on, and is coreferential with, the man introduced in the previous sentence.[2] In (4b), the interpretation of *another man* is dependent on the same thing, but only in part: it is a man other than that one. In (4c), the interpretation of *one* depends on set descriptions available in the context: here, *man*, while in (4d), the interpretation of the ellipsed VP depends on available predicates. In (4e), the interpretation of the demonstrative pronoun *that* depends on an *abstract object* (i.e., a fact, proposition, eventuality, claim, etc. (Asher 1993, Webber 1991)) available from the previous discourse — here the action of ordering a single malt. Finally, in (4f), the interpretation of the adverbial *instead* depends on available predications that admit alternatives — here, *refusing a drink*, which admits the alternative *accepting one* — i.e., instead of accepting a drink, she started talking (Webber et al. 2003).

(4) a. A man walked in. *He* sat down.
b. A man walked in. *Another man* called him over.
c. A tall man walked in, then a short *one*.

[2]This notion of *anaphoric dependency* does not cover such intra-clausal co-reference as "John shaves his father" and "John shaves himself", which fit better under the notion of *(bounded) syntactic dependency*.

d. A man walked in and ordered a single malt. Then a woman *did*.
e. A man walked in and ordered a single malt. *That* showed he had good taste.
f. The woman refused a drink. *Instead* she started talking.

Only *syntactic dependency* is part of syntax. *Semantic dependency* is part of the semantic composition process that operates alongside or on the result of syntactic analysis, while *anaphoric dependency* is separate from both syntactic analysis and semantic composition (but cf. footnote 2).

Since syntactic dependencies can impact the power of a grammar, one on-going challenge in linguistics has been to understand whether a given phenomenon is a matter of syntactic, semantic or anaphoric dependency. For example, it was once thought that the relation induced by the adverbial *respectively* was a matter of syntactic dependency between elements, such as in (5a), where the two parts of the conjoined subject are paired in order with the two parts of the conjoined verb phrase (VP) — i.e., John washing and Mary painting, and as in (5b), where the three parts of the conjoined subject are so paired with the three parts of the conjoined VP.

(5) a. I think that John and Mary will *respectively* wash his car and paint her boat today.
b. I think John, Mary and Kim will *respectively* wash his car, paint her boat, and clean their room today.

If these pairings come from the grammar, it would mean two crossing dependencies (one crossing point) for (5a) and three crossing dependencies (three crossing points) for (5b). More generally, a sentence with N conjoined subjects, N conjoined VPs and *respectively*, would have N crossing dependencies and N*(N-1)/2 crossing points. This would require a grammar to have more than context-free power.

Later, however, it was noted that negation removed the need for such pairing:

(6) a. I think John and Mary will wash his car and paint her boat, but probably not *respectively*.
b. I think John, Mary and Kim will wash his car, paint her boat, and clean their room, but probably not *respectively*.

This would not happen if the dependency were syntactic or even semantic. Thus, the individual dependencies associated with *respectively* must be the result of inference based on anaphoric dependency.[3]

[3]The fact that *respectively* can be paraphrased with the demonstrative *that* — i.e., "in *that* order" — provides additional evidence for this conclusion.

In the following sections, I will use these notions of *constituency* and *dependency* to characterize the source of discourse relations in several theories of discourse. This will, I hope, illuminate both their similarities and differences. However, I do not have the space here for what would have to be an extended discussion of SDRT (Asher and Lascarides 2003), given the deep involvement of *inference* in how it derives discourse relations.

16.3 Dependency as a Source of Discourse Relations

The theory of discourse articulated by Halliday and Hasan (1976) is one in which discourse relations can be seen to arise solely from *anaphoric dependency*. Specifically, Halliday and Hasan (1976) consider *cohesion* to be what relates parts of a discourse, where *cohesion* is defined as holding when one part cannot be effectively interpreted except by recourse to the interpretation of another part. Halliday and Hasan posit five types of cohesion, each associated with a particular set of lexical or syntactic elements:

1. elements expressing referential identity or dependence, such as pronouns and other forms of anaphora;
2. substitution, as with *one(s)*, *so* and *do so*;
3. ellipsis, including nominal ellipsis (e.g., "the best hat" → "the best"), verb phrase ellipsis, and clausal ellipsis;
4. lexical cohesion, as in the reiteration of the same word, or a synonym or near synonym, or a superordinate term or generic word;
5. conjunction, as expressed through coordinating and subordinating clausal conjunctions, adverbials like *later on*, and prepositional phrases like *in that case*.

The fifth is the source of discourse relations in Halliday and Hasan (1976) and is, in fact, the sole source. That is, discourse relations arise from identifying what other part of the discourse the interpretation of a conjunctive element depends on. It should be clear that *cohesion* as a source of discourse relations is essentially *anaphoric dependency*, with three interesting features. First, there is no theoretical constraint on its locality, since any part of a text can theoretically depend on any other part. Secondly, there are no constraints on how many parts of the text a given part may depend on: in the analysis given in Halliday and Hasan (1976, Ch. 8.3), multiple cohesive links between the parts of a sentence (including the sentence as a whole) and the previous discourse are the norm, rather than the exception. Thirdly, there are no constraints on what parts of a discourse can be linked together, so cohesive links are as likely to cross one another as to be embedded.

Note that Halliday and Hasan explicitly reject any notion of structure or *constituency* in discourse, saying for example:

> Whatever relation there is among the parts of a text — the sentences, the paragraphs, or turns in a dialogue — it is not the same as structure in the usual sense, the relation which links the parts of a sentence or a clause. (Halliday and Hasan 1976:6)

> We doubt whether one can demonstrate generalized structural relations into which sentences enter as the realisation of functions in some higher unit as can be done for all units below the sentence. (Halliday and Hasan 1976:10)

> Between sentences, there are no structural relations. (Halliday and Hasan 1976:27)

Thus, only the notion of *anaphoric dependency* is needed in describing the source of discourse relations in Halliday and Hasan (1976): neither *constituency* nor *scope* plays any role.

16.4 Constituency as a Source of Discourse Relations

In contrast with Halliday and Hasan (1976), both Mann and Thompson's *Rhetorical Structure Theory* (RST) and Polanyi's *Linguistic Discourse Model* (LDM) take *constituency* as the sole basis for discourse relations. Both provide an exhaustive top-down context-free constituency analysis of a text, associating with many (LDM) or all (RST) constituents, a formal pragmatic account in terms of discourse relations. Differences between the two lie in what they take to be a constituent and how they associate discourse relations with constituents.

16.4.1 Rhetorical Structure Theory

In RST (Mann and Thompson 1988), the constituency structure of a discourse consists of instantiated *schemas* which specify discourse relations (called here *rhetorical relations*) between adjacent spans, which may be clauses or the projection of instantiated schemas. The set of *schemas* essentially defines a context-free (CF) grammar on a single non-terminal, which we can call D. RST has five kinds of *schemas*, which differ with respect to how they re-write D in terms of (i) rhetorical relations that hold between right-hand side (RHS) sisters; (ii) whether or not the RHS has a head (called in RST, a *nucleus*); and (iii) whether there are two, three, or arbitrarily many sisters (the latter like Kleene plus).

The first is a headed binary-branching schema in which a specific rhetorical relation such as CIRCUMSTANCE or EVIDENCE holds between the head daughter and its sister (called a *satellite*). As the order of

the daughters does not matter, such a schema actually stands for two standard CF rules. This type of schema is illustrated in Figures 1(a) and 2(a), where the former follows the conventions used in Mann and Thompson (1988), and the latter follows more standard tree-drawing conventions. RST also allows for an N-ary version of this schema, which retains a single head daughter, but with multiple sisters, all of which bear the same rhetorical relation with the head.

The second is a headed binary-branching schema in which the rhetorical relation CONTRAST holds between two daughters of equal headedness. This schema is illustrated in Figures 1(b) and 2(b), where again, the former follows the graphic conventions used in Mann and Thompson (1988), and the latter follows more standard tree-drawing conventions.

The third is an N-ary branching schema called JOINT, in which no rhetorical relation is taken to hold between sisters that nevertheless belong together in some way, and all sisters are equal in headedness.

The fourth is a ternary-branching schema in which the head has a MOTIVATION relation to one sister and a ENABLEMENT relation to the other. As in the first type of schema, the order of the daughters does not matter, so that this schema actually stands for six standard CF rules, each corresponding to a different order of the three sisters. This schema is illustrated in Figures 1(d) and 2(d).

The final type of schema is a multi-headed N-ary branching rule in which each sister except the first is related to its left-adjacent sister by a SEQUENCE relation. This is illustrated in Figures 1(e) and 2(e). Since one's choice of schema commits one to a particular discourse relation (or none) holding between sisters, there is no way to say that two adjacent spans in a discourse are related, without saying what the relation is. This is one way that RST differs from the *Linguistic Data Model*, to be discussed next.

An RST analysis of a discourse is a tree whose root non-terminal covers the entire string span and where adjacent non-terminals in any cut across the tree cover adjacent string spans. A possible RST analysis of the short discourse in Example (7) is shown in Figure 3, with Figure 3(a) following RST conventions and Figure 3(b) following more standard tree-drawing conventions.

(7) a. You should come visit.
b. Edinburgh is lovely in early fall
c. and there are no rabbits around.

Much of Mann and Thompson (1988) is concerned with delineating the conditions under which it is appropriate for someone analysing a discourse to assert that a particular relation holds between adjacent

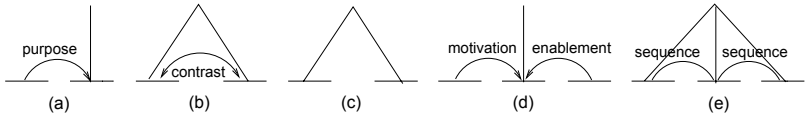


FIGURE 1 RST schema types in RST notation

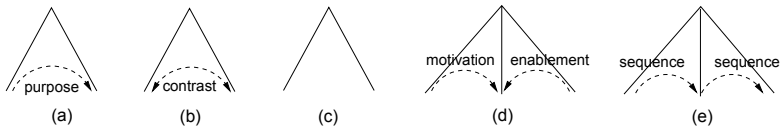


FIGURE 2 RST schema types in standard tree notation

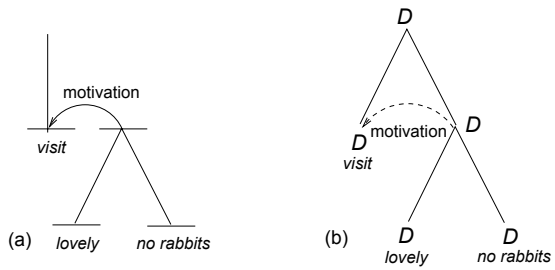


FIGURE 3 RST analysis of Example 7

spans, which then determines what schema applies. Such conditions can be placed on (1) the informational content of the spans themselves, (2) the speaker's perceived intent with respect to each span and its role with respect to its sister(s), and (3) the intended effect of the span. RST requires an analyst to produce a single RST analysis of a discourse, so judgments must be made in cases of perceived ambiguity and in cases where more than one rhetorical relation can simultaneously be taken to hold between sisters.

What RST does not do is place conditions on anything elsewhere in the discourse. Thus it views discourse relations solely in terms of context-free *constituency*. They can hold only between sisters on the RHS of some schema, although those sisters may correspond to either terminal nodes or the non-terminal nodes of instantiated RST schema. (This latter point will become relevant in comparing RST and the *GraphBank* approach presented in Section 16.5.2.)

16.4.2 Linguistic Discourse Model (LDM)

The Linguistic Discourse Model (Polanyi 1988, Polanyi and van den Berg 1996, Polanyi et al. 2004) is a theory of discourse and discourse parsing that resembles RST in constructing an explicit tree-structured representation of discourse constituents, but differs in separating discourse structure from discourse interpretation. It does this by having three (and only three) context-free re-write rules, each associated with a rule for semantic composition:

1. an N-ary branching rule for *discourse coordination*, in which all the RHS sisters bear the same relationship to their common parent (used for elements of lists and narratives). Here there is no specific relation between sisters. The interpretation of the parent node is the information common to all its daughters.
2. a binary branching rule for *discourse subordination*, in which one sister (considered *subordinate*) elaborates an entity or situation described in the other (considered *dominant*). Here, ELABORATION is the discourse relation between the sisters, and the interpretation of the parent node is the interpretation of the dominant daughter.
3. an N-ary branching rule in which the RHS sisters are related by a logical or rhetorical relation, or by a genre-based or interactional convention. Here, the interpretation of the parent node derives from the interpretation of each daughter and from the relationship between them. Evidence for that relationship can come from lexical and/or syntactic information. It appears that this rule

could apply without the particular relationship being fully specified, awaiting further specification later on. Such a move would not be possible in RST.

Recent work on the LDM has been concerned with the issue of automatically parsing discourse efficiently, with respect to the model. But the issue of concern here — whether the source of discourse relations in the LDM is *constituency* or *dependency* — comes down firmly for the former.

16.5 Mixed Approaches to Discourse Relations

In contrast with the theories presented earlier, both D-LTAG (Webber et al. 2003) and the theory underlying GraphBank (Wolf and Gibson 2005) exploit both *constituency* and *anaphoric dependency* in their accounts of discourse relations, but in very different ways.

16.5.1 A Lexicalized TAG for Discourse (D-LTAG)

D-LTAG is a lexicalized approach to discourse relations (Webber et al. 2001, Forbes et al. 2003, Webber et al. 2003, Webber 2004, Forbes-Riley et al. 2006). Lexicalization means that D-LTAG provides an account of how lexical elements (including some phrases) anchor discourse relations and how other parts of the text provide arguments for those relations.

D-LTAG arose from a belief that the mechanisms for conveying discourse relations were unlikely to be entirely different from those for conveying relations within the clause. Because the latter can be anchored on lexical items, D-LTAG was developed as a *lexicalized grammar* for discourse — in particular, a lexicalized Tree Adjoining Grammar (Schabes 1990). A lexicalized TAG (LTAG) differs from a basic TAG in taking each lexical entry to be associated with the set of tree structures that specify its local syntactic configurations. These structures can be combined via either *substitution* or TAG's *adjoining* operation, in order to produce a complete sentential analysis. In D-LTAG, elementary trees are anchored (by and large) by discourse connectives (representing predicates), whose substitution sites (arguments) can be filled by clauses or other trees.

Elementary trees anchored by a *structural connective* (i.e., a coordinating or subordinating conjunction, a subordinator such as *in order to*, *so that*, etc.) or what we call an *empty connective* are used to build constituent structure. The compositional interpretation of this structure is in terms of discourse relations between arguments (Forbes-Riley et al. 2006). Discourse adverbials, on the other hand, exploit *anaphoric*

dependency to convey a discourse relation between the *abstract object* (AO) interpretation of its matrix clause and the AO interpretation of a previous clause, sequence of clauses, or nominalization in the discourse. That discourse adverbials such as *instead*, *afterwards*, *as a result*, etc. differ from structural connectives in terms of the distribution of their arguments is demonstrated on theoretical grounds in Webber et al. (2003) and on empirical grounds in Creswell et al. (2002). An explanation for the anaphoric character of discourse adverbials is given in Forbes (2003) and Forbes-Riley et al. (2006).

Both *constituency* and *dependency* can be seen in the D-LTAG analysis of Example 8:

(8) John loves Barolo.
So he ordered three cases of the '97.
But he had to cancel the order
because he *then* discovered he was broke.

The analysis is shown in Figure 4. It involves a set of elementary trees for the connectives (*so, but, because, then*) and a set of leaves ($T1$-$T4$) corresponding to the four clauses in Example (8), minus the connectives. Through the operations of *substitution* (solid lines) and *adjoining* (dashed lines) recorded in the *derivation tree* (here shown to the right of the arrow), a *derived tree* is produced (here shown at the head of the arrow). More detail on both the representation of connectives and D-LTAG derivations is given in Webber et al. (2003). A preliminary parser for D-LTAG is described in Forbes et al. (2003).

Compositional interpretation of the derivation tree produces the discourse relations associated with *because*, *so* and *but*, while anaphor resolution produces the other argument to the discourse relation associated with *then* (i.e., the ordering event), just as it would if *then* were paraphrased as *soon after that*, with the pronoun *that* resolved anaphorically. Details on D-LTAG's syntactic-semantic interface are given in Forbes-Riley et al. (2006), along with a detailed discussion of *discourse adverbials* and the various sources of their anaphoric links with the previous discourse. Empirical data on the predicate-argument structure of discourse connectives are now available in Release 1.0 of the annotated Penn Discourse TreeBank[4] (Dinesh et al. 2005, Miltsakaki et al. 2004a,b, Prasad et al. 2004, Webber 2005). An early effort to use data in the PDTB to develop a procedure for resolving the anaphoric argument of the discourse adverbial *instead* is described in Miltsakaki et al. (2003), and the effect of *Information Structure* on the preferred

[4] http://www.seas.upenn.edu/~pdtb

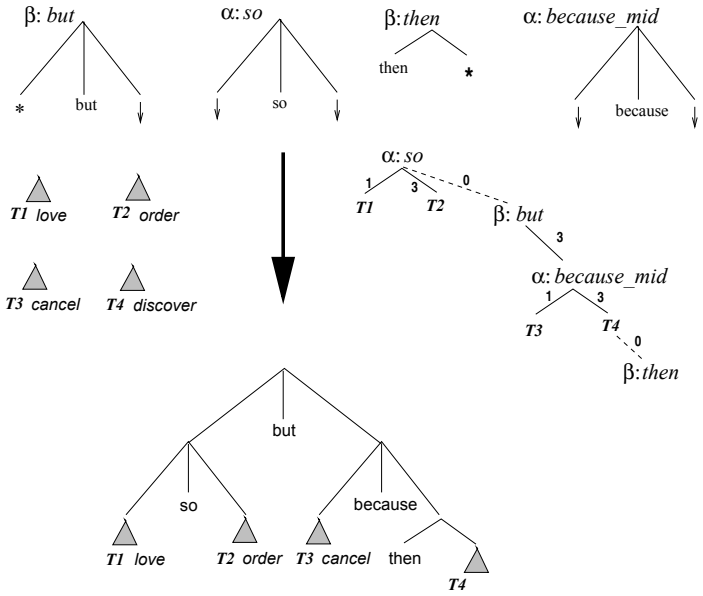


FIGURE 4 Derivation of Example 8

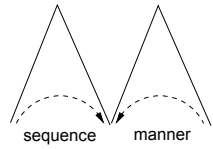


FIGURE 5 Simple multi-parent structure

argument of the discourse adverbial *otherwise* is described in Kruijff-Korbayová and Webber (2001).

One final note: although D-LTAG produces only trees, it is acknowledged in Webber et al. (2003) — as noted earlier by Bateman (1999) and Gardent (1997) — that a discourse unit must be allowed to participate in one constituent structure with left-adjacent material and another with right-adjacent material, as in Figure 5. While RST handles this as a special schema (cf. Figures 1(d) and 2(d)), D-LTAG would have to incorporate such structures in a more general way.

16.5.2 Wolf and Gibson (2005)

Wolf and Gibson (2005) present a view of discourse related to RST (Section 16.4.1), but different in one important way: Rather than analyzing

a text as a recursive structure of discourse spans with discourse relations holding between sisters, Wolf and Gibson (2005) claim that discourse structure should be seen as a relatively shallow graph of discourse segments linked to one or more previous adjacent or non-adjacent segments (or segment *groupings*, see below) via discourse relations (here called *coherence relations*).

More specifically, Wolf and Gibson (2005) assume two types of basic discourse segments: clauses and attributions. The latter are related to what Huddleston and Pullum (2002) call *reporting frames* (e.g., "John asserted that ..."). Clause fragments are also treated as discourse segments if they result from the interruption of a clause by a discourse segment such as the reporting frame in (9).

(9) The economy,
according to some analysts,
is expected to improve by early next year.
(Wolf and Gibson 2005: 255, Ex. 17).

In this case, the non-adjacent clausal fragments are linked into a discourse segment through a *coherence relation* called SAME. (SAME is only used within a sentence. It is never used inter-sententially to form a segment from two non-adjacent segments.)

While Wolf and Gibson make use of eleven broad classes of binary relations in their analysis, including SAME, CONDITION, ATTRIBUTION, CAUSE-EFFECT, CONTRAST, ELABORATION, and GENERALIZATION, they do not require a relation to hold between the entire content of the segments so linked together. For example, in (10)

(10) a. Difficulties have arisen in enacting the accord for the independence of Namibia
b. for which SWAPO has fought for many years.
(Wolf and Gibson 2005: Ex. 18)

an ELABORATION relation is taken to hold between the non-restrictive relative clause that constitutes (10b) and the matrix clause (10a), even though (10b) only elaborates an NP within (10a) — i.e., "the independence of Namibia". This is significant because it means that the NP does not have to be analysed as a separate discourse segment, as the remainder of the matrix clause would then have to be.

The only hierarchical structuring in Wolf and Gibson's approach comes from *grouping* a sequence of adjacent discourse segments to serve as one argument to a coherence relation whose other argument is a previous discourse segment or *grouping*. The basis for a grouping is

common attribution or common topic.[5] Within a grouping, a coherence relation can hold between segments and the same is true between a within-grouping segment and one outside the grouping. Because a grouping of segments on a common topic might itself contain groupings on common sub-topics, it appears that groupings could determine a partial hierarchical structure for parts of a text, and that grouping is a matter of *constituency*. But this is the only hierarchical structure in Wolf and Gibson's approach: unlike RST (Section 16.4.1) and LDM (Section 16.4.2), the existence of a coherence relation between two segments does not produce a new segment that can serve as argument to another coherence relation.

Procedurally, a text is analyzed in a sequence of left-to-right passes. First, the text is segmented in a left-to-right pass, then groupings are generated, and finally, the possibility of a coherence relation is assessed between each segment or grouping and each discourse segment or grouping to its left. This produces a rather flat discourse structure, with frequent crossing arcs and nodes with multiple parents that Wolf and Gibson argue should be represented as a *chain graph* — that is, a graph with both directed and undirected edges, whose nodes can be partitioned into subsets within which all edges are undirected and between which, edges are directed but with no directed cycles.[6] This is illustrated in the discourse structure ascribed to Example (11), shown in Figure 6.

(11) 1. "The administration should now state
2. that
3. if the February election is voided by the Sandinistas
4. they should call for military aid,"
5. said former Assistant Secretary of State Elliot Abrams.
6. "In these circumstances, I think they'd win."
(Wolf and Gibson 2005: Ex. 26)

Figure 6 contains two *groupings* — one from segments 3 and 4, the other from segments 1 and 2 and the first grouping, while among the coherence relations are ones that hold within a grouping (COND) and between a segment and a grouping (ATTRIBUTION). The approach has been used to analyze a corpus of 135 news articles called the *Discourse GraphBank*, available from the LDC catalogue as LDC2005T08.

Wolf and Gibson's claim that discourse structure is best modelled as

[5]Documentation for the Discourse GraphBank (Wolf et al. 2003) states that a grouping should only be assumed if otherwise truth conditions are changed.

[6]A *Directed Acyclic Graph* (DAG) is a special case of a chain graph, in which each subset contains only a single node.

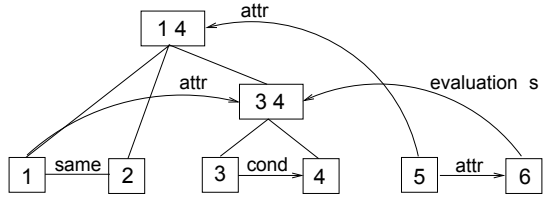


FIGURE 6 Coherence graph for Example (11)

a chain graph, rather than a tree, shows that they associate discourse relations with *constituency* structure alone. However, two things suggest that *anaphoric dependency* plays a significant role. The first is the extent to which the discourse relation ELABORATION underpins their discourse structure, coupled with arguments by Knott et al. (2001) that OBJECT-ATTRIBUTE ELABORATION is actually a matter of *anaphora* — that is, *anaphoric dependency*. The second parallels the argument in Section 16.2.2 that pairings in the *respectively* construction arise from *anaphoric dependency* rather than *constituency*.

ELABORATION is the most common discourse relation in Wolf and Gibson's *Discourse GraphBank*, comprising 43.6% of the total. Many instances of ELABORATION are claimed to hold between non-adjacent discourse segments, such as between (12.4) and (12.2).

(12) 1. Susan wanted to buy some tomatoes
2. and she also tried to find some basil
3. because her recipe asked for these ingredients.
4. *The basil* would probably be quite expensive this time of year.
(Wolf and Gibson 2005: Ex. 21)

Example (12) exemplifies OBJECT-ATTRIBUTE ELABORATION, which Mann and Thompson (1988) take as holding when one segment presents an object, and the related segment subsequently presents an attribute of that object. But as in Knott et al. (2001), the claim for that relation holding relies on the fact that *the basil* in (12.4) is anaphorically related to *some basil* in (12.2).[7]

[7]The *Discourse GraphBank* also has cases of elaboration holding between adjacent segments. But many of these involve syntactic constructions such as a relative clause (as in (10) above), an appositive (as in the relation between 1*a* and 1*b* in (i) below), or an adjunct (as in the relation between grouping (2–3) and grouping (1a–1b) in the same example).

(i) 1. $_{1a}$[Mr. Baker's assistant for inter-American affairs,] $_{1b}$[Bernard Aronson,]
2. while maintaining
3. that the Sandinistas had also broken the cease-fire,
4. acknowledged:
5. "It's never very clear who starts what." (Wolf and Gibson 2005: Ex. 23)

Knott et al. (2001) came to their view of OBJECT-ATTRIBUTE ELABORATION as *anaphoric dependency* in trying to automatically generate passages of text similar to that found in museum guidebooks. They first analysed such passages in terms of RST, adhering to RST's assumption that the spans linked by a relation must either be *adjacent* or if not adjacent, that any intervening spans must also be linked to the initial span by the same relation (cf. Section 16.4.1). But they found that in passages such as (13), they had to analyse non-adjacent segments as standing in an ELABORATION relation, here (13.4) elaborating (13.2).

(13) (1) In the women's quarters the business of running the household took place. (2) Much of the furniture was made up of chests arranged vertically in matching pairs (. . .). (3) Female guests were entertained in these rooms, which often had beautifully crafted wooden toilet boxes with fold-away mirrors and sewing boxes, and folding screens, painted with birds and flowers.
(4) Chests were used for the storage of clothes . . . .

This, however, violated RST's adjacency assumption. In analysing all the cases where they saw ELABORATION relations holding between non-adjacent segments, they noticed that in each case, the elaborating segment re-introduced and described an entity that had been mentioned earlier in the discourse. [8] Subsequent segments then continued that description until another previously mentioned entity was re-introduced and elaborated.

To explain this, Knott et al. (2001) assumed that discourse was locally structured as an *entity chain* — i.e.,

> a sequence of Rhetorical Structure (RS) trees, each constructed just as in RST, but minus the ELABORATION relation. These trees can either be simple trees consisting of just one text span, or more complex trees with several layers of hierarchy. In each case, we can define the *top nucleus* of the tree to be the leaf-level text span which is reached by following the chain of nuclei from its root. A legal *entity chain* whose *focus* is entity **E** is one where the top nucleus of each tree is a fact about **E**.

and globally structured as a sequence of entity chains, as in Figure 7, where the focussed entity in each chain is mentioned in a proposition somewhere within the previous N chains. For the current discussion, the most important thing to notice is that the links between entity

[8] The entity could have been introduced by one or more noun phrases, or by one or more clauses. In the latter case, reference was via a demonstrative pronoun (*this* or *that*) or definite or demonstrative NP.

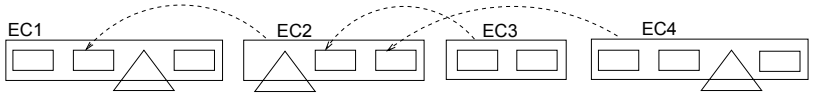


FIGURE 7 A legal sequence of entity-chains (Knott et al. 2001: Fig. 4)

chains are based on *anaphoric relations*, and that such links can and will cross (Figure 7).

As noted, the *Discourse GraphBank* contains many instances of ELABORATION, many holding between non-adjacent segments (including ones at a considerable distance apart), as in the following example from file 62 (segment numbering as in the source file):

(14) 0 First lady Nancy Reagan was saluted Monday night for her sense of style and her contribution to the American fashion industry.
1 Mrs. Reagan was presented with a Lifetime Achievement Award by the Council of Fashion Designers of America at its eighth annual awards ceremony,
. . .
46 Also among those recognized at the ceremony was designer Geoffrey Beene,
47 who was saluted for making "fashion as art."
. . .
48 Performer Liza Minelli,
49 one of many women attending the ceremony who dressed in Mrs. Reagan's favorite color,
. . .
54 Actress Audrey Hepburn,
55 wearing a red gown designed by Givenchy,
56 presented a Lifetime Achievement Award to photographer Richard Avedon,
. . .

Here annotators have recorded ELABORATION relations between segment 46 and segment 1, between segment 48 and segment 1, between segment 54 and segment 1, and between several later segments and segment 1. But each of these appears to be based on an anaphoric bridging relation between the people mentioned in segments 46, 48, 54 and elsewhere, and the attendees at the awards ceremony mentioned in segment 1. (Recall that Wolf and Gibson do not require a relation to hold between the entire contents of linked segments, as in Example (10).) Other anaphoric relations are discernable in other instances

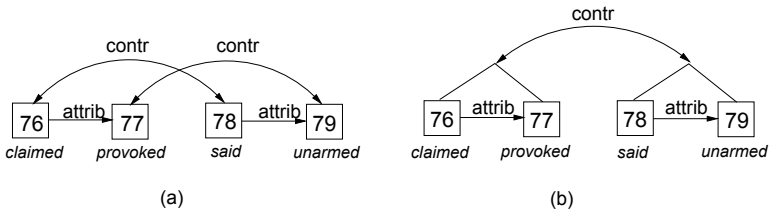


FIGURE 8 Coherence graph for Example (15)

of ELABORATION. Thus what is being captured here by Wolf and Gibson is a discourse relation based on *anaphoric dependency* rather than *constituency*.

A second such case can be found in constructions that resemble *respectively* at the discourse level, as in Example (15) taken from file 105 of the *Discourse GraphBank* (segment numbering as in the source file), which has been annotated with the discourse relations shown in Figure 8(a).

(15) 76 Washington claimed
77 the downings were provoked by aggressive and threatening actions by armed Libyan aircraft,
78 while Libya said
79 its jets were unarmed and on a reconaissance flight over international waters.

If from the perspective of constituency, this were analysed as the simpler structure in Figure 8(b), then as with the *respectively* construction, the individual pairings would be a matter of anaphoric dependency, derived only when appropriate. While such examples might not occur frequently in discourse, the *respectively* construction shows there are alternative explanations for crossing dependencies that do not involve arbitrary inter-leaving of *constituency structure*. And once instances of discourse relations are understood to arise from *anaphoric dependency*, it calls into question Wolf and Gibson's leap upward in complexity from trees to *chain graphs* as a model for discourse structure.

16.6 Conclusion

I started this paper by mentioning Ron's advice that one should seek to identify the minimal machinery required for a task. Here, the task was to understand and compare the source of discourse relations in five different theories: Halliday and Hasan's theory of discourse cohesion, Mann and Thompson's Rhetorical Structure Theory, Polanyi's Linguis-

tic Data Model, Wolf and Gibson's theory of discourse graphs, and the lexically-based theory that I have been involved in, D-LTAG. What I have tried to show is that the paired notions of *constituency* and *dependency* provide a useful way of understanding some of the significant similarities and differences between these theories. If I have succeeded, I may also have convinced the reader that they are equally important in understanding discourse relations.

Acknowledgments

I would like to thank all those who took the time to read and comment on earlier versions of this manuscript — Mark Steedman, Rashmi Prasad, Nikhil Dinesh, Eleni Miltsakaki, Annie Zaenen and two anonymous reviewers. The paper has benefitted from their comments, and any remaining errors are my own.

References

Asher, Nicholas. 1993. *Reference to Abstract Objects in Discourse*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Asher, Nicholas and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge, United Kingdom: Cambridge University Press.

Bateman, John. 1999. The dynamics of 'surfacing': An initial exploration. In *Proceedings of the International Workshop on Levels of Representation in Discourse (LORID'99)*, pages 127–133. Edinburgh, United Kingdom.

Creswell, Cassandre, Katherine Forbes, Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2002. The discourse anaphoric properties of connectives. In A. Branco, T. McEnery, and R. Mitkov, eds., *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC2002)*. Lisbon, Portugal: Edições Colibri.

Dinesh, Nikhil, Alan Lee, Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2005. Attribution and the non-alignment of syntactic and discourse arguments of connectives. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Workshop on Frontiers in Corpus Annotation*. Ann Arbor, MI.

Forbes, Katherine. 2003. *Discourse Semantics of S-Modifying Adverbials*. Ph.D. thesis, Department of Linguistics, University of Pennsylvania.

Forbes, Katherine, Eleni Miltsakaki, Rashmi Prasad, Anoop Sarkar, Aravind Joshi, and Bonnie Webber. 2003. D-LTAG System: Discourse parsing with a lexicalized Tree-Adjoining Grammar. *Journal of Logic, Language and Information* 12(3):261–279.

Forbes-Riley, Katherine, Bonnie Webber, and Aravind Joshi. 2006. Computing discourse semantics: The predicate-argument semantics of discourse connectives in D-LTAG. *Journal of Semantics* 23(1):55–106.

Gardent, Claire. 1997. Discourse tree adjoining grammars. CLAUS Report Nr. 89, University of the Saarland, Saarbrücken, Germany.

Grosz, Barbara and Candace Sidner. 1990. Plans for discourse. In P. Cohen, J. Morgan, and M. Pollack, eds., *Intentions in Communication*, pages 417–444. Cambridge, MA: The MIT Press.

Halliday, Michael and Ruqaiya Hasan. 1976. *Cohesion in English*. London, United Kingdom: Longman.

Huddleston, Rodney and Geoffrey Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge, United Kingdom: Cambridge University Press.

Knott, Alistair, Jon Oberlander, Mick O'Donnell, and Chris Mellish. 2001. Beyond elaboration: The interaction of relations and focus in coherent text. In T. Sanders, J. Schilperoord, and W. Spooren, eds., *Text Representation: Linguistic and psycholinguistic aspects*, pages 181–196. Amsterdam, The Netherlands: John Benjamins.

Kruijff-Korbayová, Ivana and Bonnie Webber. 2001. Information structure and the semantics of "otherwise". In *Proceedings of the 13th European Summer School on Logic, Language and Information (ESSLLI'01), Workshop on Information Structure, Discourse Structure and Discourse Semantics*, pages 61–78. Helsinki, Finland.

Lochbaum, Karen. 1998. A collaborative planning model of intentional structure. *Computational Linguistics* 24(4):525–572.

Mann, William and Sandra Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text* 8(3):243–281.

Miltsakaki, Eleni, Cassandre Creswell, Kate Forbes, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2003. Anaphoric arguments of discourse connectives: Semantic properties of antecedents versus non-antecedents. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), Workshop on Computational Treatment of Anaphora*. Budapest, Hungary.

Miltsakaki, Eleni, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004a. Annotating discourse connectives and their arguments. In *Proceedings of the Human Language Technology Conference and the 4th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04), Workshop on Frontiers in Corpus Annotation*. Boston, MA.

Miltsakaki, Eleni, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004b. The Penn Discourse TreeBank. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal.

Moore, Johanna and Martha Pollack. 1992. A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics* 18(4):537–544.

Polanyi, Livia. 1988. A formal model of the structure of discourse. *Journal of Pragmatics* 12:601–638.

Polanyi, Livia and Martin H. van den Berg. 1996. Discourse structure and discourse interpretation. In P. Dekker and M. Stokhof, eds., *Proceedings of the 10th Amsterdam Colloquium*, pages 113–131. Amsterdam, The Netherlands.

Polanyi, Livia, Chris Culy, Martin H. van den Berg, Gian Lorenzo Thione, and David Ahn. 2004. A rule based approach to discourse parsing. In *Proceedings of the Human Language Technology Conference and the 4th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04), 5th SIGdial Workshop on Discourse and Dialogue*, pages 108–117. Cambridge, MA.

Prasad, Rashmi, Eleni Miltsakaki, Aravind Joshi, and Bonnie Webber. 2004. Annotation and data mining of the Penn Discourse TreeBank. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), Workshop on Discourse Annotation*, pages 88–95. Barcelona, Spain.

Schabes, Yves. 1990. *Mathematical and Computational Aspects of Lexicalized Grammars*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Webber, Bonnie. 1991. Structure and ostension in the interpretation of discourse deixis. *Language and Cognitive Processes* 6(2):107–135.

Webber, Bonnie, Alistair Knott, and Aravind Joshi. 2001. Multiple discourse connectives in a lexicalized grammar for discourse. In H. Bunt, R. Muskens, and E. Thijsse, eds., *Computing Meaning*, vol. 2, pages 229–249. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Webber, Bonnie, Matthew Stone, Aravind Joshi, and Alistair Knott. 2003. Anaphora and discourse structure. *Computational Linguistics* 29(4):545–587.

Webber, Bonnie. 2004. D-LTAG: Extending Lexicalized TAG to discourse. *Cognitive Science* 28:751–779.

Webber, Bonnie. 2005. A short introduction to the Penn Discourse TreeBank. In *Copenhagen Working Papers in Language and Speech Processing*. Copenhagen, Denmark.

Wolf, Florian, Edward Gibson, Amy Fisher, and Meredith Knight. 2003. A procedure for collecting a database of texts annotated with coherence relations. Documentation accompanying the *Discourse GraphBank*, LDC2005T08.

Wolf, Florian and Edward Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational Linguistics* 31(2):249–287.

Woods, William, Ron Kaplan, and Bonnie Nash-Webber. 1972. The lunar sciences natural language information system: Final report. Tech. Rep. 2378, Bolt Beranek and Newman, Cambridge, MA.

Part IV

Semantics and Inference

17

Direct Compositionality and the Architecture of LFG

Ash Asudeh

17.1 Introduction

The principle of compositionality is arguably the foundational principle of formal semantics. It is quoted in (1) from Janssen (1997:419), but can be found in innumerable sources in much the same formulation.

(1) *The Principle of Compositionality*
The meaning of a complex expression is a function of the meanings of its parts.

This principle is often called 'Frege's Principle', although it is doubtful whether Frege himself formulated it (Janssen 1997, Hodges 1998, 2001).[1] Hodges (2001:7) suggests that compositionality in the modern sense is more readily attributable to Tarski (1983 [1935]), the foundational work in truth-conditional semantics. Heim and Kratzer (1998:1–3) point out that Tarskian truth-conditional schemas can only be informative in light of compositionality.

Despite its generally acknowledged importance to modern semantic theory, compositionality is in danger of becoming a shibboleth, because it is typically formulated sufficiently broadly that just about any semantic theory would satisfy it in some sense or other.[2] In this light, recent

[1] Nevertheless, the principle is clearly in the spirit of Frege's later works, hence the common attribution (Janssen 1997:421).

[2] Zadrozny (1994) notes that 'the standard definition of compositionality is formally vacuous', given his theorem that any semantics can be made compositional.

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

work by Jacobson (1999, 2002, 2004, 2005) on 'the hypothesis of Direct Compositionality' is important, because it features an in-depth defense of one well-articulated substantive conception of compositionality that simultaneously suggests why certain other modern approaches do not satisfy the conception. The non-directly compositional approach that forms Jacobson's main target is semantics based on Logical Form (LF) in Principles and Parameters Theory (P&P; Chomsky 1981, 1995). May (1977) is an early and influential precursor of the LF approach, but its principal modern articulation is Heim and Kratzer (1998). Lexical Functional Grammar (LFG; Kaplan and Bresnan 1982) superficially appears to similarly fall into the non-directly compositional class of formalisms, due to its postulation of a grammatical level of *semantic structure* (s-structure) in its parallel projection architecture (Kaplan 1987, Halvorsen and Kaplan 1988, Kaplan 1989, Dalrymple 1993).

In this paper, I argue that this superficial impression is incorrect, expanding on some remarks in Asudeh (2005:433–439). Both that initial treatment of the problem and this expanded treatment are based on Ron Kaplan's foundational work on LFG's grammatical architecture. Rather than entering into a direct comparison of LF in P&P with s-structure in LFG, I will show that strings in LFGs can be assigned a directly compositional interpretation. Therefore, despite whatever similarities between LF and semantic structure suggest themselves without delving deeper, LFG grammars are *not* outside the class of directly compositional grammars in Jacobson's sense. Nevertheless, I will argue, based on Kaplan's insights, that a grammatical architecture like LFG's — an architecture that posits many intermediate structures between form and meaning, but which crucially treats the structures as eliminable — is preferable to an architecture that allows *only* a very direct mapping between form and meaning.

The paper is structured as follows. In section 17.2, I present Kaplan's notion of a parallel projection architecture and a synthesis of subsequent LFG-theoretic architectural proposals in the literature (based on Asudeh 2004:32–35, with some modifications). In section 17.3, I present Jacobson's work on direct compositionality. Then, in section 17.4, I present the apparent problem that Jacobson's work poses for LFG and show how the problem can be resolved. Lastly, in section 17.5, I take the opposing tack and consider the hypothesis of direct compositionality in light of LFG's grammatical architecture.

17.2 The Parallel Projection Architecture

The original architecture of LFG (Kaplan and Bresnan 1982) consisted of two syntactic levels: constituent structure (c-structure) and functional structure (f-structure). C-structures are represented as trees, which are described in the usual manner (with a set of nodes, a labeling on the set, and functions for dominance and precedence). The level of c-structure represents syntactic information about precedence, dominance, and constituency. F-structures are represented as feature structures (attribute-value matrices), described by a set of recursive functional equations on a set of symbols. The level of f-structure is another aspect of syntactic representation — it is not a semantic representation. However, f-structure represents more abstract aspects of syntax, such as grammatical functions, predication, subcategorization, and local and non-local dependencies. C-structure and f-structure are projected from lexical items, which specify their c-structure category and f-structure feature contributions. Variables in lexical items are instantiated by the c-structure parse. The two syntactic representations are present simultaneously, in parallel. They are related by the ϕ projection function, also known as a correspondence function. The ϕ function maps c-structure nodes (i.e., tree nodes) to f-structure nodes (i.e., feature structures). The original grammatical architecture of LFG is shown schematically in (2).

(2) *The original LFG architecture*:

$$\text{constituent structure} \xrightarrow{\phi} \text{functional structure}$$

An LFG representation of an expression on this view is a triple consisting of a c-structure, an f-structure and a ϕ projection function that maps the c-structure to the f-structure: $\langle c,\ f,\ \phi \rangle$.

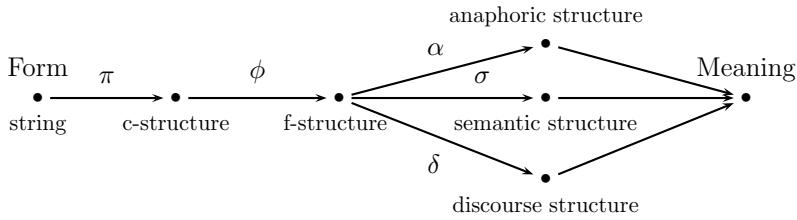
C-structures and f-structures are constructed by simultaneous constraint satisfaction. LFG is a declarative, non-transformational theory. The fact that c-structure and f-structure are represented using distinct data structures (trees and feature structures) distinguishes LFG from both transformational theories such as P&P, which represents all syntactic information in a tree, and non-transformational theories such as Head-Driven Phrase Structure Grammar (Pollard and Sag 1987, 1994), which represents all grammatical information, whether syntactic or not, in a directed acyclic graph. LFG uses mixed data structures related by structural correspondences, rather than a single monolithic data structure.

The LFG architecture was subsequently further generalized to a parallel projection architecture (Kaplan 1987, Halvorsen and Kaplan 1988,

Kaplan 1989). According to this architecture, there are various levels of linguistic representation (not just syntactic ones) called *projections* that are present in *parallel* and are related by structural correspondences (i.e., projection functions) which map elements of one projection onto elements of another. C-structure and f-structure are still the best-understood projections, but they are now two among several levels of representation and the projection function ϕ is now one of many. For example, f-structures are mapped onto s(emantic)-structures by the σ-function (Halvorsen 1983, Dalrymple 1993, Dalrymple et al. 1999b, Dalrymple 2001).

Kaplan (1987, 1989) gives (3) as a hypothetical example of the projection architecture, representing the decomposition of a single mapping, Γ, from form to meaning.

(3) *Kaplan's hypothetical parallel projection architecture*:



Two of the projections proposed in (3) — anaphoric structure and discourse structure — never received much further attention in the LFG literature, at least not in the way that Kaplan originally suggested. Anaphors have been handled at semantic structure (Dalrymple 1993, 2001), and discourse structure has been pursued instead as information structure (i-structure; Butt and King 2000), which encodes notions like discourse topic and focus and old and new information.

Importantly, the correspondence functions between levels can be composed (see below for details), since the domain of each successive function is the range of the previous one. This is summarized in the following passage from Kaplan (1987:363):

> Although the structures related by multiple correspondences might be descriptively or linguistically motivated levels of representation, justified by sound theoretical argumentation, they are formally and mathematically, and also computationally, eliminable ... Obviously there is a structural correspondence that goes from the word string to the f-structure, namely the composition of π with ϕ. ... So as a kind of formal, mathematical trick, you can say 'Those intermediate levels of representation are not real, they are just linguistic fictions, useful for stating the necessary constraints'.

There are two key points in this passage. First, intermediate levels are eliminable through composition of correspondence functions. Second, although such elimination is possible, it may nevertheless be desirable to have separate levels. I will pick up on both of these points in sections 17.4 and 17.5 below.

Kaplan observes that we can compose π and ϕ to go directly from strings to f-structures. We can further compose $\pi \circ \phi$ with σ, moving directly from the string to semantic structure. The nature of these mapping functions is important to consider. The postulation of a projection function is tantamount to the claim that there is a function from a structure of type A to a structure of type B. The range of the function may, however, be the empty set. Or there may be more than one such function. Each projection function therefore represents a family of functions. For example, consider the mapping π from strings to c-structures. For each string there is a π function mapping the string to c-structure. An unparseable string — one that has no structural analysis — will not be mapped to anything by π. A parseable but unambiguous string will be in the domain of exactly one π function. An ambiguous string will be in the domain of more than one π function. Similarly, a string may have only one c-structure, but there may be multiple instances of the ϕ mapping if the c-structure is f-structurally ambiguous. The same comments apply to the σ function from f-structure to s-structure and all the other projection functions.

The various levels of grammatical representation in the projection architecture are simultaneously present, but each level is governed by its own rules and representations. This separation of levels allows one to make simple theoretical statements about just the aspects of grammar that the level in question models. It is also possible to split up correspondences in novel ways. Since the projection functions are functions in the mathematical sense, we can always regain the original function through composition of the new functions. This is exemplified by the Butt et al. (1997) proposal for argument structure, discussed below, which separates the original ϕ function into α and λ functions. Another important feature of this architecture is that there can be systematic mismatches between grammatical levels. For example, null pronoun subjects in pro-drop languages are not present at c-structure, because they are unmotivated by the aspects of syntax that are represented at that level. Rather, null pronouns are present at f-structure, where they can participate in agreement, binding, and other syntactic processes modeled at that level.

Although the exact specification of the projection architecture is not the main point of this paper, it is useful from a general LFG-theoretic

perspective to stop and take stock of certain subsequent augmentations that have been proposed in the LFG literature. Information structure, the alternative to discourse structure mentioned above, is just one of several subsequent proposals for new projections. Three other proposals are argument structure (a-structure; Butt et al. 1997), morphological structure (m-structure; Butt et al. 1996, 1999, Frank and Zaenen 2002) and phonological structure (p-structure; Butt and King 1998), the latter of which should perhaps be called prosodic structure, since it is concerned with phrasal phonology and prosody. Butt et al. (1997) propose that argument structure should be interpolated between c-structure and f-structure, with the ϕ projection function broken up into the α function from c-structure to a-structure and the λ function from a-structure to f-structure. The original ϕ function would then be the composition of these two new functions: $\phi = \alpha \circ \lambda$ (this will be slightly revised below, in light of m-structure). Information structure and phonological structure have both been proposed as projections from c-structure. There has been some debate over the proper location for morphological structure in the architecture. Butt et al. (1996, 1999) treat it as a projection from c-structure. Frank and Zaenen (2002) argue that although this is adequate for the phenomena for which Butt et al. (1996, 1999) use morphological structure (auxiliaries), there are reasons to prefer morphological structure as a projection from f-structure. I assume that morphological information should feed both argument structure and functional structure; I therefore place m-structure between c-structure and a-structure. This also means that Butt et al. (1997)'s α projection function now maps from m-structure to a-structure, rather than from c-structure to a-structure (their original α is the composition of μ and my α). The original ϕ function of Kaplan and Bresnan (1982) is thus the composition of μ, α and λ (that is, $\mu \circ \alpha \circ \lambda$).

Figure 1 shows an architecture resulting from the addition of these proposals to Kaplan's hypothetical architecture in (3) (note that anaphoric structure and discourse structure have been removed). The architecture in figure 1 is considerably more complex than the original LFG architecture in (2), or even the initial parallel architecture in (3). However, it rests on Kaplan's simple, but powerful, fundamental idea: there is a series of functions, the domain of each subsequent one being the range of the previous one, that map from linguistic form to linguistic meaning.

Let me spell out the mapping to semantics in figure 1 in a little more detail. First, I will define a function that captures the mapping from c-structure to s-structure that is represented by the smaller functions. This function has three components, representing the three paths

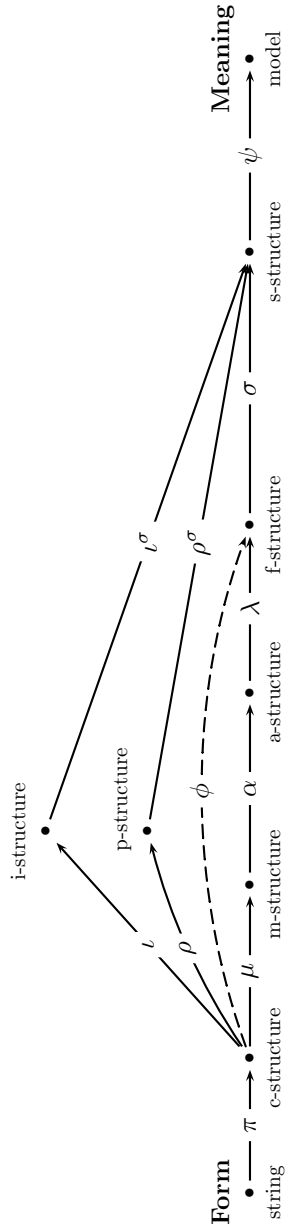


FIGURE 1 The parallel projection architecture (incorporating certain recent proposals)

of information flow from c-structure to s-structure: via morphological structure, argument structure and functional structure, via information structure, and via phonological structure. The function is thus a mapping from c-structure nodes to a triple of s-structure information. Let us call the function Σ ('Big Sigma') in homage to the original σ function. Big Sigma can be characterized as follows:

(4) $\Sigma = \lambda y.\, \langle (\phi \circ \sigma)(y),\ (\iota \circ \iota^{\sigma})(y),\ (\rho \circ \rho^{\sigma})(y) \rangle$
where y is a c-structure node

Note that the ϕ function in the body of Big Sigma is the new ϕ (i.e., $\mu \circ \alpha \circ \lambda$).

As in other theories of grammar, most work on LFG semantics has focused on the mapping from syntax to semantics, leaving aside the semantic contributions of information structure and phonological/prosodic structure. Thus, semantics in LFG has focused on the first member of the Big Sigma triple, which we can access via projection on the triple:

(5) $\mathsf{first}(\Sigma) = \lambda y.(\phi \circ \sigma)(y)$

In Glue Semantics for LFG (Dalrymple 1999, 2001), s-structure nodes and lexically-defined logical operations on s-structure nodes form the input to a linear logic (Girard 1987) proof of an expression's semantics. Linear logic provides the 'glue language' that specifies how meanings are put together; that is, linear logic is the logic of semantic composition. The linear logic proof is directly related to a model-theoretic semantics via the Curry-Howard Isomorphism between formulas and types (Curry and Feys 1958, Howard 1980). This will be illustrated with respect to a specific example in section 17.4.

The crucial point, though, is that semantic structure forms the input to semantic composition. It thus seems that semantic structure is an indispensable pre-semantic level of representation, on a par with LF in Principles and Parameters Theory. In the next section, I review Jacobson's criticism of LF semantics based on the hypothesis of direct compositionality. Then, in section 17.4, I show that LFG semantic structure is not analogous to Logical Form in the relevant sense through a demonstration that LFG semantics can satisfy 'Strong Direct Compositionality' (Jacobson 2002).

17.3 Direct Compositionality

The hypothesis of Direct Compositionality (DC) has been discussed in some detail in recent work by Jacobson (1999, 2002, 2004, 2005). Jacobson (1999) characterizes the hypothesis as follows:

> [S]urface structures directly receive a model-theoretic interpretation without being mapped into another level (i.e., LF). (Jacobson 1999:117)

In later work, Jacobson (2002, 2004, 2005) characterizes DC slightly differently:

> [T]here is a set of syntactic rules which prove the well-formedness of the set of sentences (or other expressions) in the language . . . Coupled with each syntactic rule is a semantic rule specifying how the meaning of the larger expression is derived from the meaning of the smaller expressions. (Jacobson 2002:603)

The latter form of DC is part of Jacobson's characterization of *Strong Direct Compositionality*, which is one of three successively weaker notions of DC, the other two being *Weak(er) Direct Compositionality* and *Deep Compositionality* (Jacobson 2002). However, it is clear from Jacobson's latest work (2004, 2005), that the notion of compositionality laid out in the second quote is meant as a characterization of the general form of DC, as evidenced by the following introductory passage from Jacobson (2004):

> The hypothesis of Direct Compositionality. . . is that the syntax and semantics work "in tandem". The syntax is a system of rules . . . which prove the well-formedness of linguistic expressions while the semantics works simultaneously to provide a model-theoretic interpretation for each expression as it is proved well-formed in the syntax.

There are thus two characterizations of the DC hypothesis (the one in the first quote and the one in the second two quotes); these can be summed up as follows:

(6) Hypothesis of Direct Compositionality
   a. Surface structure is directly model-theoretically interpreted without mapping to an intervening level.
   b. Model-theoretic interpretation is a function of syntactic well-formedness.

Jacobson tends to treat these two characterizations of DC equivalently, but they are logically distinct. LFG semantics upholds (6b), but it seems similar to LF semantics in contravening (6a).

The second characterization of DC, construed broadly, seems to be a corollary of the principle of compositionality. As Janssen notes, a 'more precise version' of the principle (Janssen 1997:462) than the version quoted in (1) involves reference to syntax, as in the following formulation from Partee et al. (1993:316):

(7) *The Principle of Compositionality (version 2)*
The meaning of a complex expression is a function of the meanings of its parts and of the syntactic rules by which they are combined.

It is clear, though, that one can have a syntax–semantics architecture that respects (7), and therefore clause (6b) of DC, without respecting clause (6a). In fact, the very system that Jacobson (1999:117) cites as a crucial early exemplar of the direct compositionality approach — Montague's semantics in *The proper treatment of quantification in ordinary English* (PTQ; Montague 1973) — is an instance of the separation of (6a) and (6b). In PTQ, strings of English are first translated into expressions of intensional logic (IL), and it is these IL expressions that are subsequently model-theoretically interpreted. PTQ therefore postulates a level, intensional logic, between surface forms and their interpretations, thus contravening (6a). However, it clearly respects (6b) in its foundational presentation of rule-by-rule translation. Thus, (6b) does not logically depend on (6a).

Similarly, one can imagine systems that respect (6a) by not positing any intervening level between surface structure and semantics, but which contravene (6b) through appeal to non-compositionality. Indeed, such proposals have been made; relevant examples and discussion can be found in Partee (1984) and Janssen (1997:437–441). It is not clear how interesting the proposals are, in light of theorems about the syntactic and semantic side of compositionality (Janssen 1986, Zadrozny 1994), which together show that 'without constraints on syntax and semantics, there are no counterexamples to compositionality' (Janssen 1997:456–457). In other words, these proposals are arguably more relevant to the question of proper constraints on syntax and semantics than to the question of compositionality. However, the proposals do show that (6a) does not depend on (6b) (although, according to such proposals, surface syntax is not the sole determinant of interpretation). Therefore, the two parts of DC are independent.

Jacobson's conflation of (6a) and (6b) is understandable, though, when seen in light of the broader context of her work on direct compositionality. This work is part of a research program, set out in detail in Jacobson (1999), that seeks to argue for a directly compositional (in both senses of (6)), variable-free semantics in opposition to semantics in the tradition of Logical Form, for which Heim and Kratzer (1998) is a key touchstone, both for Jacobson and in the semantics literature more generally. As it happens, Heim and Kratzer (1998) deny both parts of (6) in the sense that Jacobson has in mind, although they do not deny

(6b) under a different construal (i.e., their semantics *is* compositional). Their denial of (6a) is obvious, since their semantics interprets Logical Forms, which are not surface structures. Their denial of (6b) is much more subtle, though.

Heim and Kratzer (1998:49) propose the following principle for semantic interpretation:

(8) *Principle of Interpretability*
All nodes in a phrase structure tree must be in the domain of the interpretation function ⟦ ⟧.

Given that the phrase structure tree in question must be well-formed according to some syntax, it initially seems that Heim and Kratzer's system does support (6b). However, it is clear that what Jacobson means by 'syntax' is whatever proves well-formedness of the surface strings of the language. But this is not the kind of syntax that yields the phrase structure trees of interest in (8): those are LF trees and LF is not surface structure (the yields of LF trees are not the surface strings of the language, since movement operations can occur at LF).

Here is another way to think about the difference between the position of Jacobson and that of Heim and Kratzer. Jacobson assumes that semantic interpretability and syntactic well-formedness are mutually entailing: if a string is syntactically well-formed, it receives an interpretation, and if a string receives an interpretation, it is syntactically well-formed. Heim and Kratzer explicitly deny this:

> In sum, we are adopting a view of the grammar as a whole in which syntax and semantics are independent modules. Each imposes its own constraints on the grammatical structures of the language, and we expect there to be structures that are interpretable though syntactically illegitimate, as well as structures that are syntactically correct but uninterpretable. (Heim and Kratzer 1998:49)

Thus, Jacobson assumes an extremely tight relationship between syntax and semantics, and Heim and Kratzer assume a looser relationship, although for them interpretation still depends compositionally on the level of Logical Form.

The question now is how the architecture of Lexical Functional Grammar fits into this picture, assuming that Glue Semantics is providing the semantic theory. I will call this combination 'LFG-Glue'. With respect to the question of the relationship between model-theoretic interpretation and syntactic well-formedness, LFG-Glue provides a potentially interesting intermediate position between the Jacobson and Heim & Kratzer positions. The linear logic proofs in Glue Semantics depend on the syntax to instantiate semantic structure nodes in

the premises for the proofs. If the syntactic input to the proofs is ill-formed, the proofs will consequently fail due to improper instantiation of premises. Thus, successful semantic interpretation depends on syntactic well-formedness. Under certain circumstances, syntactically ill-formed structures may have informative *partial* interpretations (Asudeh 2004:321–334). However, these structures are not *fully* interpretable, contra the picture sketched in the Heim and Kratzer quote above, which specifically countenances 'structures that are interpretable though syntactically illegitimate'. In sum, (6b) is upheld in Glue Semantics: complete interpretation is a function of syntactic well-formedness, where the syntax in question is the syntax of the surface strings in LFG-Glue.

Jacobson's position further entails that if a structure is syntactically well-formed, it is interpretable. In general, this is also true in Glue Semantics, because lexical items specify their semantic types and there is, as in most theories, a strong correlation between semantic type and syntactic category. The upshot is that as long as the lexical items are assigned motivated meanings, syntactically well-formed structures will be interpretable. However, the syntax of composition in Glue Semantics is divorced from the syntax of string formation, unlike in Categorial Grammar, the framework that Jacobson assumes (see Jacobson 1999). Therefore, if one assigns a type to a lexical item in LFG-Glue that is not reflected by the item's syntactic information, there could be a syntactically well-formed structure that is not fully saturated (i.e., not of type t). Such cases do not happen in practice, but are possible in principle.

For example, suppose we had a syntax with the following annotated phrase structure rule for c-structure:

(9) $$\begin{array}{lcccc} \text{S} & \longrightarrow & \text{N} & & \text{V} \\ & & (\uparrow \textsc{subj}) = \downarrow & & \uparrow = \downarrow \end{array}$$

Now suppose we have a verb in our lexicon, e.g. *floobles*, which has the semantically transitive type $\langle e, \langle e, t \rangle \rangle$, but which does not syntactically select for an object. The verb would have the usual syntactic category of V. Assuming appropriate category N lexical entries, our syntax would then derive sentences like *John floobles*, which is *syntactically* well-formed, but which is uninterpretable, according to the type specification of *floobles*.

LFG-Glue therefore upholds (6b) in Jacobson's strict sense, because successful interpretation does entail well-formedness of the string-yielding syntax. As far as interpretability entailing well-formedness, LFG-Glue and Categorial Grammar are thus in agreement, contra LF semantics. However, LFG-Glue is like the latter, contra Catego-

rial Grammar, in denying that syntactic well-formedness *in principle* entails interpretability (although it does in practice). This difference between LFG-Glue and Categorial Grammar is directly traceable to the logic of composition. Categorial Grammar posits that the syntax of string analysis (parsing/generation) and the syntax of composition are essentially the same. The non-commutativity of syntax is thus passed on to semantics. However, it is questionable whether the logic of semantic composition should itself be non-commutative, because the fundamental operation in compositional semantics, functional application, is commutative (Asudeh 2004:76–77). Glue Semantics, through its use of the commutative linear logic for semantic composition, separates the non-commutativity of string analysis (provided by an LFG syntax in LFG-Glue) from the commutativity of semantic composition. This is, at heart, what gives LFG-Glue a theoretical position with respect to the modularity of syntax and semantics that is intermediate between syntactic well-formedness and semantic interpretability being mutually entailing (Jacobson's position) or fully independent (Heim and Kratzer's position).

The LFG architecture thus respects (6b), the second part of Direct Compositionality, even on a strict interpretation. However, it seems that the LFG architecture does not support (6a), because there is a level of semantic structure immediately before interpretation. In the next section, I show that semantic structure is dispensable and that LFG therefore upholds (6a), despite initial appearances.

17.4 Directly Compositional LFG

Jacobson (2002:603ff.) uses quantifier scope ambiguity to illuminate the hypothesis of direct compositionality. In this section, I will first introduce an analysis of scope ambiguity in LFG-Glue; this will help put the subsequent discussion of direct compositionality in LFG-Glue on an even footing with Jacobson's presentation, which considers other frameworks.

The example that Jacobson (2002:603) uses is:

(10) Some man read every book.

Appropriate simplified lexical entries for this sentence in LFG-Glue are shown in (11).

(11) *some* D $(\uparrow \textsc{pred}) = \text{`some'}$

$$\begin{array}{l} \mathit{some}' : \\ [((\textsc{spec} \uparrow)_\sigma \textsc{var}) \multimap ((\textsc{spec} \uparrow)_\sigma \textsc{restr})] \multimap \\ \quad [((\textsc{spec} \uparrow)_\sigma \multimap X) \multimap X] \end{array}$$

man	N	$(\uparrow \textsc{pred}) = \text{`man'}$
		$man' : (\uparrow_\sigma \textsc{var}) \multimap (\uparrow_\sigma \textsc{restr})$
read	V	$(\uparrow \textsc{pred}) = \text{`read}\langle(\uparrow \textsc{subj}), (\uparrow \textsc{obj})\rangle\text{'}$
		$(\uparrow \textsc{tense}) = \textsc{past}$
		$read' : (\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$
every	D	$(\uparrow \textsc{pred}) = \text{`every'}$
		$every' :$ $[((\textsc{spec} \uparrow)_\sigma \textsc{var}) \multimap ((\textsc{spec} \uparrow)_\sigma \textsc{restr})] \multimap$ $[((\textsc{spec} \uparrow)_\sigma \multimap Y) \multimap Y]$
book	N	$(\uparrow \textsc{pred}) = \text{`book'}$
		$book' : (\uparrow_\sigma \textsc{var}) \multimap (\uparrow_\sigma \textsc{restr})$

The last part of each of these lexical entries is a Glue Semantics *meaning constructor*. Each meaning constructor has the following form:

(12) $\mathcal{M} : G$

$\mathcal{M}$ is a term from the meaning language — the language that is model-theoretically interpreted; this is a fragment of a logic that supports the lambda calculus. G is a term from the glue language, a fragment of linear logic. The meaning constructors for an expression form the premise set for a linear logic proof of the expression's semantics. Terms in $\mathcal{M}$ and G are systematically related in the proof by the Curry-Howard Isomorphism (CHI) between formulas and types (Curry and Feys 1958, Howard 1980). The only aspects of the CHI that will be relevant in what follows are the correspondences between functional application in the meaning language and implication elimination ($\multimap_\mathcal{E}$; *modus ponens*) in the linear logic glue language and between abstraction and implication introduction ($\multimap_\mathcal{I}$).[3]

These lexical entries form the terminals in a constituent structure parse of the string in (10). The constituent structure rules are annotated phrase structure rules (Kaplan and Bresnan 1982). The annotations are typically functional structure equations that are defined in terms of two variables over c-structure nodes — $*$ for the current node and $\widehat{*}$ for the current node's mother[4] — and the ϕ projection function from c-structures to f-structures. The annotation $\phi(*)$ therefore means the f-structure correspondent of the current node and the annotation $\phi(\widehat{*})$

[3]For further details on the CHI and Glue Semantics, see Dalrymple et al. (1999a), Crouch and van Genabith (2000), Dalrymple (2001), and Asudeh (2004, 2005).

[4]These notions are ultimately defined in terms of N, the set of c-structure nodes, and the mother function on nodes, M: N $\rightarrow$ N; see Kaplan (1987) for a concise overview.

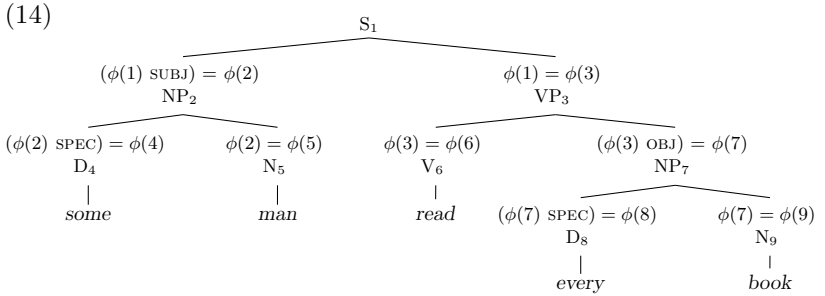
means the f-structure correspondent of the current node's mother. $\phi(*)$ and $\phi(\widehat{*})$ are conventionally written as $\downarrow$ and $\uparrow$ respectively.

For expository purposes, I assume the following very simple set of c-structure rules for the analysis of (10):

(13)
$$\begin{array}{lcccc} \text{S} & \longrightarrow & \text{NP} & & \text{VP} \\ & & (\uparrow \textsc{subj}) = \downarrow & & \uparrow = \downarrow \\ \text{NP} & \longrightarrow & \text{D} & & \text{N} \\ & & (\uparrow \textsc{spec}) = \downarrow & & \uparrow = \downarrow \\ \text{VP} & \longrightarrow & \text{V} & & \text{NP} \\ & & \uparrow = \downarrow & & (\uparrow \textsc{obj}) = \downarrow \end{array}$$

The lexical entries in (11) and the c-structure rules in (13) give the following c-structure for (10):

(14)



The nodes in the c-structure have been assigned unique numbers as names (see Kaplan 1987); these node names are used to instantiate the f-structure variables in node annotations. The terminals are the lexical entries from (11), including all the equational information, but have been labeled in an abbreviated form.

Solving the f-structure equations in the lexical entries and c-structure, we get the following functional structure:

(15)
$$r\begin{bmatrix} \textsc{pred} & \text{`read}\langle(r\ \textsc{subj}),(r\ \textsc{obj})\rangle\text{'} \\ \textsc{subj} & m\begin{bmatrix} \textsc{pred} & \text{`man'} \\ \textsc{spec} & \begin{bmatrix} \textsc{pred} & \text{`some'} \end{bmatrix} \end{bmatrix} \\ \textsc{obj} & b\begin{bmatrix} \textsc{pred} & \text{`book'} \\ \textsc{spec} & \begin{bmatrix} \textsc{pred} & \text{`every'} \end{bmatrix} \end{bmatrix} \\ \textsc{tense} & \textsc{past} \end{bmatrix}$$

I have followed the convention of labeling f-structures mnemonically based on their PRED value. In this case, this means that $r = \phi(1) = \phi(3) = \phi(6)$, $m = \phi(2) = \phi(5)$, and $b = \phi(7) = \phi(9)$.

The σ projection function from f-structure to s-structure maps from nodes in (15) to the s-structure in (16). Note that I have labeled $(m_\sigma \text{VAR})$ as *v1*, etc.; these abbreviations will be useful below. I have also followed the convention of writing $\sigma(x)$ as x_σ.

(16) $$r_\sigma\begin{bmatrix} \end{bmatrix} \quad m_\sigma\begin{bmatrix} \text{VAR} & v1\begin{bmatrix} \end{bmatrix} \\ \text{RESTR} & r1\begin{bmatrix} \end{bmatrix} \end{bmatrix} \quad b_\sigma\begin{bmatrix} \text{VAR} & v2\begin{bmatrix} \end{bmatrix} \\ \text{RESTR} & r2\begin{bmatrix} \end{bmatrix} \end{bmatrix}$$

Notice that semantic structure is both very sparse and unconnected. It is unconnected because no notion of a semantic structure head path has been defined on a par with the f-structure head paths defined by $\uparrow = \downarrow$ equations. Such a semantic notion of head could easily be constructed through the specification $\uparrow_\sigma = \downarrow_\sigma$, but a theoretical need for this has yet to be identified.

The nodes of s-structure fill in variables in lexically contributed meaning constructors, yielding the set of premises in (17) for the linear logic proof of the semantics, based on the contributions in (11).

(17) 1. $some' : (v1 \multimap r1) \multimap ((m \multimap X) \multimap X)$
2. $man' : v1 \multimap r1$
3. $read' : b \multimap m \multimap r$
4. $every' : (v2 \multimap r2) \multimap ((b \multimap Y) \multimap Y)$
5. $book' : v2 \multimap r2$

Based on these premises, we can construct two valid linear logic proofs. Both proofs share the same initial sub-proof, shown in (20). The proofs then diverge, depending on which quantifier is scoped first. The proof in figure 2 provides the surface scope reading and the proof in figure 3 provides the inverse scope reading. For presentational purposes, I have left implicit in figures 2 and 3 the sub-proofs that show the composition of the quantificational determiners with their nominal restrictions; these are presented separately in (18) and (19).

(18) $$\frac{some' : (v1 \multimap r1) \multimap ((m \multimap X) \multimap X) \qquad man' : v1 \multimap r1}{some'(man') : ((m \multimap X) \multimap X)} \multimap_\varepsilon$$

(19) $$\frac{every' : (v2 \multimap r2) \multimap ((b \multimap Y) \multimap Y) \qquad book' : v2 \multimap r2}{every'(book') : ((b \multimap Y) \multimap Y)} \multimap_\varepsilon$$

(20) $$\dfrac{\dfrac{read' : b \multimap m \multimap r \quad [y : b]^1}{read'(y) : m \multimap r} \multimap_\varepsilon \qquad [x : m]^2}{read'(y)(x) : r} \multimap_\varepsilon$$

In Glue Semantics, the two alternative scopings are thus completely based on alternative linear logic derivations on the same set of premises.

$$\dfrac{\begin{array}{c}\vdots\\ some'(man') :\\ ((m \multimap X) \multimap X)\end{array} \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ every'(book') :\\ ((b \multimap Y) \multimap Y)\end{array} \qquad \dfrac{\begin{array}{c}\vdots\\ read'(y)(x) : r\end{array}}{\lambda y.read'(y)(x) : b \multimap r}\ \multimap_{\mathcal{I},1}}{every'(book')(\lambda y.read'(y)(x)) : r}\ \multimap_{\mathcal{E}},\ [r/Y]}{\lambda x.every'(book')(\lambda y.read'(y)(x)) : m \multimap r}\ \multimap_{\mathcal{I},2}}{some'(man')(\lambda x.every'(book')(\lambda y.read'(y)(x))) : r}\ \multimap_{\mathcal{E}},\ [r/X]$$

FIGURE 2 Surface scope proof

$$\dfrac{\begin{array}{c}\vdots\\ every'(book') :\\ ((b \multimap Y) \multimap Y)\end{array} \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ some'(man') :\\ ((m \multimap X) \multimap X)\end{array} \qquad \dfrac{\begin{array}{c}\vdots\\ read'(y)(x) : r\end{array}}{\lambda x.read'(y)(x) : m \multimap r}\ \multimap_{\mathcal{I},2}}{some'(man')(\lambda x.read'(y)(x)) : r}\ \multimap_{\mathcal{E}},\ [r/X]}{\lambda y.some'(man')(\lambda x.read'(y)(x)) : b \multimap r}\ \multimap_{\mathcal{I},1}}{every'(book')(\lambda y.some'(man')(\lambda x.read'(y)(x))) : r}\ \multimap_{\mathcal{E}},\ [r/Y]$$

FIGURE 3 Inverse scope proof

No syntactic Quantifier Raising ambiguity is assumed — there is a single c-structure and f-structure for (10) — and there is no type shifting. The Glue Semantics approach to scope ambiguity is therefore distinct from both Logical Form and Categorial Grammar approaches.

With this exposition of scope in Glue Semantics in hand, we can now return to the question of direct compositionality. The use of semantic structure as an input to the linear logic derivation of the semantics of (10) is an apparent rejection of part (6a) of the hypothesis, which postulates that there is no intermediate level between surface structure and model-theoretic interpretation. However, as discussed in section 17.2, in LFG the intervening levels between form and meaning are dispensable, via composition of the correspondence functions. The composition in question is the following:

(21) $\Gamma = \pi \circ \phi \circ \sigma \circ \psi$

Recalling the discussion of ambiguity in section 17.2, this function is short for a family of functions. Thus, each function in (21) admits several instances, and there may therefore be multiple Γ functions that map the string to different meanings.

In this case there is an ambiguity in meaning. The two instances of Γ functions are shown here:

(22) $\Gamma^1(\textit{some man read every book})$
$= \textit{some}'(\textit{man}')(\lambda x.\textit{every}'(\textit{book}')(\lambda y.\textit{read}'(y)(x)))$

(23) $\Gamma^2(\textit{some man read every book})$
$= \textit{every}'(\textit{book}')(\lambda y.\textit{some}'(\textit{man}')(\lambda x.\textit{read}'(y)(x)))$

The ambiguity arises only at the last point, in the ψ mapping (characterized by linear logic proofs) from semantic structure to meaning. The ambiguity derives from multiple proofs from a single set of premises.

The fact that alternative scopings arise from the same premise set means that Glue Semantics espouses a notion of purely semantic ambiguity. In other words, Glue Semantics rejects the conception of compositionality in which there is a functional relation between syntactic structures and meanings (as in, for example, the classic Montague Semantics of Montague 1970, 1973). In functional compositionality, for each distinct meaning there is at most one distinct syntactic structure: there are no one-to-many mappings from syntax to meanings. The result of functional compositionality is that every semantic ambiguity forces a syntactic ambiguity; in other words, there is no pure notion of semantic ambiguity. Glue Semantics instead espouses a relational view of compositionality: there can be a one-to-many mapping from syntax to semantics. This preserves semantic ambiguity without forcing syn-

tactic ambiguity. However, each distinct interpretation corresponds to a distinct proof. There is therefore a purely functional mapping from the syntax of semantic composition (proofs) to model-theoretic interpretations: each proof maps to a single interpretation.

Although LFG-Glue brings relational compositionality to the fore, note that this view of compositionality is also a feature of other modern approaches to the syntax–semantics interface. In LF semantics, a string can be assigned multiple logical forms. Thus, although there is a functional mapping from logical forms to model-theoretic interpretation (each LF has a single interpretation), the mapping from a *string* to its interpretation(s) is relational. Similarly, consider Categorial Grammar. There is a functional mapping from each categorial proof of syntax to model-theoretic interpretation. However, unary operations like type-shifting mean that the set of lexical items that parse the string can correspond to multiple syntactic analyses. Indeed, quantifier scope ambiguity is precisely a case that leads to multiple logical forms in LF semantics and to type-shifting in Categorial Grammar. If we are considering the mapping from a string to its interpretations, the real question is thus not whether compositionality is relational — the existence of ambiguity dictates that at some point there has to be a one-to-many mapping from a string to its interpretations. The real question is: What is the point identified by the grammatical architecture at which the mapping from syntax to semantics becomes purely functional? In Glue Semantics, this point happens very late in the pipeline from form to meaning, at the proof level. In LFG-Glue terms, it happens in the ψ mapping from s-structure to model-theoretic interpretation. In LF semantics it also happens late, at the point of mapping from logical forms to model-theoretic interpretation. In Categorial Grammar, it happens early, in the syntactic analysis.

Let me unpack in a little more detail how the composition of projection functions works, since the pieces are already in place. The initial π projection function from the string in (10) to the c-structure in (14) is characterized by the annotated phrase structure rules in (13). The ϕ projection function maps the c-structure in (14) to the f-structure in (15). The σ projection function maps the f-structure in (15) to the s-structure in (16). Lastly, the ψ function maps from the s-structure to model-theoretic meaning. The ψ function is characterized by a fragment of linear logic. The Curry-Howard Isomorphism relates operations in the linear logic to operations in the related meaning language. Interpretation of the meaning language yields the model-theoretic meaning. In sum, although there are many different levels between the string and its meaning in LFG-Glue, including a level between the syntax

and semantics (s-structure), these levels are all 'formally and mathematically, and also computationally, eliminable' (Kaplan 1987:363). Thus, although LFG postulates a level of semantic structure between syntax and model-theoretic meaning, the theory nonetheless upholds the first part of the hypothesis of direct compositionality, because the level in question is eliminable. Nevertheless, as Kaplan (1987:363) also notes, '[T]he structures related by multiple correspondences might be descriptively or linguistically motivated levels of representation, justified by sound theoretical argumentation.' For example, in Asudeh (2005), I argue that a crucial distinction between pronouns and relational nouns can be explained by a theoretically motivated distinction at s-structure.

Lastly, let us consider in what precise sense LFG-Glue meets direct compositionality. Jacobson (2002:603) characterizes a grammar that satisfies Strong Direct Compositionality as one that uses context-free phrase structure rules or the equivalent for its syntax. More generally, trees or other structured objects in the relevant grammars should constitute proofs of string well-formedness, but should not be directly referred to by the grammar. The phrase structure component of LFG satisfies this conception, since LFG has a context-free base (Kaplan 1987). In fact, Roach (1985) shows that this context-free base can in certain circumstances be further reduced to a finite-state base (Kaplan 1987:364). Thus, LFG-Glue not only satisfies direct compositionality, it satisfies the strongest version, Strong Direct Compositionality.

17.5 The Proper Use of Intermediate Structures

The grammatical architecture of LFG-Glue in principle upholds direct compositionality, in both senses of (6); however, the architecture also permits the direct mapping from surface structure to models to be taken apart. This allows relevant linguistic generalizations to be made straightforwardly about points in the mapping (intermediate structures in the projection architecture) that are hidden in the direct mapping.

This sort of architecture is similar in principle to that of Montague's PTQ, Jacobson's exemplar of direct compositionality. PTQ's intermediate representation is intensional logic, but this is merely an eliminable intermediate step to interpretation (Gamut 1991), as demonstrated by Montague (1970), which provides a model-theoretic interpretation for a fragment of English without using IL. Janssen (1997) provides a useful discussion of this:

> Since meanings are generally formalized as model-theoretic entities, such as truth values, sets of sets, etc., functions have to be spec-

> ified which operate on such meanings ... Such descriptions are not easy to understand, nor convenient to work with. Therefore almost always a logical language is used to represent meanings and operations on meanings ... So in practice associating meanings with natural language amounts to translating sentences into logical formulas. (Janssen 1997:434)

Janssen (1997:434) provides a telling example of a complicated and hard to understand operation from Montague (1970) that translates into the simple intensional logic formula $\hat{}\lambda t \lambda u[t = u]$. In other words, Montague's PTQ is a perfect example of what Kaplan is pointing out in the quote on page 366: although intermediate levels can be eliminated, this may be lead to a linguistic theory that is harder to understand and that, as a result, is less revealing than an equivalent theory that uses the intermediate levels appropriately.

Although the architecture of LFG-Glue can uphold direct compositionality, other work in Glue Semantics has postulated that proof-theoretic properties of linear logic proofs can explain linguistic phenomena, such as grammatical violations of the Coordinate Structure Constraint (Asudeh and Crouch 2002a) and scope parallelism in ellipsis (Asudeh and Crouch 2002b). This amounts to a denial of direct compositionality, because the linear logic proofs are themselves considered an aspect of interpretation in this work, rather than merely an eliminable step to model-theoretic interpretation.

With respect to this use of linear logic proofs, the following continuation of the passage by Janssen is pertinent:

> Working in accordance with compositionality of meaning puts a heavy restriction on the translations into logic, because the goal of the translations is to assign meanings. The logical representations are just a tool to reach this goal. The representations are not meanings themselves, and should not be confused with them. This means for instance, that two logically equivalent representations are equally good as representation [sic.] of the associated meaning. (Janssen 1997:434)

Linear logic proofs have strong identity criteria that allow them to be properly individuated, thus avoiding the trap of false distinctions that Janssen identifies here. For the fragment of linear logic used in Glue Semantics, there are two convergent ways of stating the identity criteria. One is based on the Curry-Howard Isomorphism and the other on proof normalization/cut elimination (Prawitz 1965); see Asudeh and Crouch (2002b) for some discussion with respect to linguistic generalizations and Crouch and van Genabith (2000) for detailed theoretical discussion.

17.6 Conclusion

In this paper, I have built on Ron Kaplan's work on LFG's parallel projection architecture and presented a synthesis of subsequent proposals for the architecture. I considered the question of whether the architecture satisfies the hypothesis of direct compositionality, discussed in recent work by Jacobson, in the context of LFG with Glue Semantics as its semantic theory. I identified two components of the hypothesis and argued that LFG-Glue satisfies both components. Lastly, I argued that the grammatical architecture of LFG-Glue sheds new light on the hypothesis of direct compositionality: intermediate levels of representation can be appropriate and useful if well-understood, a point long anticipated by Ron Kaplan.

Acknowledgments

Many thanks to Mary Dalrymple, Ida Toivonen, and an anonymous reviewer for their invaluable comments. Thanks also to Mary for her very patient editorial and formatting work. I am especially indebted to Chris Potts for his thoughtful feedback on the paper in general, and section 17.4 in particular. Any remaining errors are my own.

References

Asudeh, Ash and Richard Crouch. 2002a. Coordination and parallelism in Glue Semantics: Integrating discourse cohesion and the element constraint. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2002 (LFG'02)*, pages 19–39. Athens, Greece: CSLI Online Publications.

Asudeh, Ash and Richard Crouch. 2002b. Derivational parallelism and ellipsis parallelism. In L. Mikkelsen and C. Potts, eds., *Proceedings of the 21st West Coast Conference on Formal Linguistics (WCCFL XXI)*, pages 1–14. Somerville, MA: Cascadilla Press.

Asudeh, Ash. 2004. *Resumption as Resource Management*. Ph.D. thesis, Stanford University.

Asudeh, Ash. 2005. Relational nouns, pronouns, and resumption. *Linguistics and Philosophy* 28:375–446.

Butt, Miriam, María-Eugenia Niño, and Frédérique Segond. 1996. Multilingual processing of auxiliaries in LFG. In *Natural Language Processing and Speech Technology: Results of the 3rd KONVENS Conference, Bielefeld*, pages 111–122. Berlin, Germany: Mouton de Gruyter.

Butt, Miriam, Mary Dalrymple, and Anette Frank. 1997. An architecture for linking theory in LFG. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 1997 (LFG'97)*. San Diego, CA: CSLI Online Publications.

Butt, Miriam and Tracy H. King. 1998. Interfacing phonology with LFG. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 1998 (LFG'98)*. Brisbane, Australia: CSLI Online Publications.

Butt, Miriam, Tracy H. King, María-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. Stanford, CA: CSLI Publications.

Butt, Miriam and Tracy H. King. 2000. Null elements in discourse structure. In K. V. Subbarao, ed., *Papers from the NULLS Seminar*. Delhi, India: Motilal Banarsidass. To appear.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Dordrecht, The Netherlands: Foris.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: The MIT Press.

Crouch, Richard and Josef van Genabith. 2000. Linear logic for linguists. Ms., PARC and Dublin City University. `http://www2.parc.com/istl/members/crouch/`; checked 24/04/2006.

Curry, Haskell B. and Robert Feys. 1958. *Combinatory Logic*, vol. 1. Amsterdam, The Netherlands: North-Holland.

Dalrymple, Mary. 1993. *The Syntax of Anaphoric Binding*. Stanford, CA: CSLI Publications.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell, III, and Annie Zaenen, eds. 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Dalrymple, Mary, ed. 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. Cambridge, MA: The MIT Press.

Dalrymple, Mary, Vineet Gupta, John Lamping, and Vijay Saraswat. 1999a. Relating resource-based semantics to categorial semantics. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*, pages 261–280. Cambridge, MA: The MIT Press.

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat. 1999b. Overview and introduction. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*, pages 1–38. Cambridge, MA: The MIT Press.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. San Diego, CA: Academic Press.

de Groote, Philippe, ed. 1995. *The Curry-Howard Isomorphism*, vol. 8 of *Cahiers du Centre de Logique*. Louvain-la-Neuve, Belgium: Academia.

Frank, Anette and Annie Zaenen. 2002. Tense in LFG: Syntax and morphology. In H. Kamp and U. Reyle, eds., *How do we say WHEN it happens: Contributions to the theory of temporal reference in natural language*, pages 17–51. Tübingen, Germany: Niemeyer.

Gamut, L.T.F. 1991. *Intensional Logic and Logical Grammar*, vol. 2 of *Logic, Language, and Meaning*. Chicago, IL: University of Chicago Press.

Girard, Jean-Yves. 1987. Linear logic. *Theoretical Computer Science* 50:1–102.

Halvorsen, Per-Kristian. 1983. Semantics for Lexical-Functional Grammar. *Linguistic Inquiry* 14:567–615.

Halvorsen, Per-Kristian and Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'88)*, pages 1116–1122. Tokyo, Japan. Reprinted in Dalrymple et al. (1995:279–292).

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Oxford, United Kingdom: Blackwell.

Hodges, Wilfrid. 1998. Compositionality is not the problem. *Logic and Logical Philosophy* 6:7–33.

Hodges, Wilfrid. 2001. Formal features of compositionality. *Journal of Logic, Language, and Information* 10:7–28.

Howard, William A. 1980. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, eds., *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. London, United Kingdom: Academic Press. Circulated in unpublished form from 1969. Reprinted in de Groote (1995:15–26).

Jacobson, Pauline. 1999. Towards a variable-free semantics. *Linguistics and Philosophy* 22:117–184.

Jacobson, Pauline. 2002. The (dis)organization of the grammar: 25 years. *Linguistics and Philosophy* 25:601–626.

Jacobson, Pauline. 2004. Direct Compositionality: Is there any reason why not? MS., Brown University. Presented at the University of Michigan Workshop on Linguistics and Philosophy, Ann Arbor, MI.

Jacobson, Pauline. 2005. Direct Compositionality and variable-free semantics: The case of 'Principle B' effects. In C. Barker and P. Jacobson, eds., *Direct Compositionality*. Oxford, United Kingdom: Oxford University Press. To appear.

Janssen, Theo M. V. 1986. *Foundations and Applications of Montague Grammar. Part I: Philosophy, Framework, Computer Science*. No. 19 in CWI Tracts. Amsterdam, The Netherlands: Centre for Mathematics and Computer Science.

Janssen, Theo M. V. 1997. Compositionality. In J. van Benthem and A. ter Meulen, eds., *Handbook of Logic and Language*, pages 417–473. Cambridge, MA: The MIT Press. Co-published with Elsevier Science B.V., Amsterdam, The Netherlands.

Kaplan, Ronald M. and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, ed., *The Mental Representation of Grammatical Relations*, pages 173–281. Cambridge, MA: The MIT Press.

Kaplan, Ronald M. 1987. Three seductions of computational psycholinguistics. In P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, and P. Bennett, eds., *Linguistic Theory and Computer Applications*, pages 149–181. London, United Kingdom: Academic Press. Reprinted in Dalrymple et al. (1995:339–367).

Kaplan, Ronald M. 1989. The formal architecture of Lexical-Functional Grammar. In C.-R. Huang and K.-J. Chen, eds., *Proceedings of the ROC Computational Linguistics Workshops II (ROCLING II)*, pages 3–18. Taipei, ROC. Reprinted in Dalrymple et al. (1995:7–27).

May, Robert. 1977. *The Grammar of Quantification*. Ph.D. thesis, MIT.

Montague, Richard. 1970. English as a formal language. In B. Visentini et al., eds., *Linguaggi nella Societá e nella Tecnica*, pages 189–224. Milan, Italy: Edizioni di Communità. Reprinted in Montague (1974:188–221).

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes, eds., *Approaches to Language*, pages 221–242. Dordrecht, The Netherlands: Reidel. Reprinted in Montague (1974:247–270).

Montague, Richard. 1974. *Formal Philosophy: Selected Papers of Richard Montague*. New Haven, CT: Yale University Press. Edited and with an introduction by R. H. Thomason.

Partee, Barbara H. 1984. Compositionality. In F. Landman and F. Veltman, eds., *Varieties of Formal Semantics*, pages 281–311. Dordrecht, The Netherlands: Foris. Reprinted in Partee (2004:153–181).

Partee, Barbara H., Alice ter Meulen, and Robert E. Wall. 1993. *Mathematical Methods in Linguistics*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Partee, Barbara H. 2004. *Compositionality in Formal Semantics: Selected Papers of Barbara H. Partee*. Oxford, United Kingdom: Blackwell.

Pollard, Carl and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics*. Stanford, CA: CSLI Publications.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago, IL and Stanford, CA: The University of Chicago Press and CSLI Publications.

Prawitz, Dag. 1965. *Natural Deduction: A Proof-Theoretical Study*. Stockholm, Sweden: Almquist and Wiksell.

Roach, Kelly. 1985. The mathematics of LFG. Ms., Xerox Palo Alto Research Center.

Tarski, Alfred. 1983 [1935]. The concept of truth in formalized languages. In *Logic, Semantics, Metamathematics*, pages 152–278. Indianapolis, IN: Hackett, 2nd edn. Translated by J. H. Woodger. Edited by J. Corcoran.

Zadrozny, Wlodek. 1994. From compositional semantics to systematic semantics. *Linguistics and Philosophy* 17:329–342.

18

Packed Rewriting for Mapping Text to Semantics and KR

Dick Crouch

18.1 Introduction

This paper describes the use of a packed rewriting engine for constructing semantic and knowledge representations from f-structures. The f-structures are obtained by using the XLE system (Maxwell and Kaplan 1996) and a broad coverage LFG grammar of English (Butt et al. 1999, Riezler et al. 2002) to parse open text. The knowledge representations produced by this system are intended for tasks such as knowledge-based question answering or detection of textual entailments and contradictions.

The paper has two goals. The first is to provide a snapshot of some of the work being done on broad coverage semantic and knowledge representation (KR) in Ron Kaplan's research group at PARC. But principally it is to provide a tutorial description of the free-choice packing mechanisms developed by Kaplan and Maxwell (1995) for managing ambiguity in non-context free parsing, and to show how they can be applied beyond syntactic analysis.

The paper is organized as follows. Section 18.2 describes the kinds of knowledge representations being produced from texts, discusses some of the obstacles to producing these representations, and informally illustrates the use of packing to represent alternative interpretations. Section 18.3 discusses the packed rewriting system, and goes into greater detail about the theoretical and computational issues involved in managing ambiguity through packing. Section 18.4 discusses how semantics

Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan.
Miriam Butt, Mary Dalrymple, and Tracy Holloway King (eds.).

and KR differ from f-structures, and describes examples of the kind of rewrite rules used to map between the representations. Section 18.5 concludes.

18.2 Knowledge Representation for Text

Mapping text to KR proceeds via a number of stages: (1) parse the sentence, mapping it onto an f-structure, (2) map the f-structure onto a semantic representation, (3) map the semantic representation onto an abstract knowledge representation, and optionally (4) map the abstract knowledge representation onto a concrete knowledge representation formalism as used by some knowledge representation or reasoning system, e.g. Cyc (Lenat 1995) or Knowledge Machine (Clark and Porter 1998). We will focus here on describing abstract knowledge representation (AKR) which is the final level of representation explicitly constrained by linguistic considerations before mapping into other formalisms which are more constrained by the demands of automated reasoning. Differences between AKR and semantic representations will be discussed in Section 18.4.

18.2.1 Abstract KR

It is easiest to introduce the form of AKR we will be using by means of examples:

(1) Bush claimed that Iraq possessed WMDs.

(2) context(t),
context(claim_cx1),
context_relation(informationConveyed(claim_ev1), t, claim_cx1)
sub_concept(claim_ev1, InformationClaimEvent)
sub_concept(Bush2, USPresident43)
sub_concept(possess_ev3, PossessingObject)
sub_concept(Iraq4, CountryOfIraq)
sub_concept(WMD5, WeaponOfMassDestruction)
role(performedBy, claim_ev1, Bush2)
role(informationConveyed, claim_ev1, claim_cx1)
role(possessor, possess_ev3, Iraq4)
role(thingPossessed, possess_ev3, WMD5)
role(cardinality, WMD5, plural)
temporalRel(precedes, claim_ev1, Now)
temporalRel(precedes, possess_ev3, Now)
instantiable(claim_ev1, t)
instantiable(Bush2, t)
instantiable(possess_ev3, claim_cx1)

instantiable(Iraq4, claim_cx1)
instantiable(WMD5, claim_cx1)

The AKR shown in (2) introduces two contexts: a top level context "t", representing the commitments of the speaker of sentence (1), and an embedded context "claim_cx1" representing the state of affairs according to Bush's claim. The two contexts are related via the "informationConveyed" role of the claim.

The representation contains terms like "claim_ev1" or "Bush2" which refer to the kinds of object that the sentence is talking about.[1] The "sub_concept" facts explicitly link these terms to their concepts in some chosen ontology. Thus "claim_ev1" is stated to be some sub-kind of the type InformationClaimEvent, and "WMD5" to be some sub-kind of the type WeaponOfMassDestruction. It is important to bear in mind that terms like "claim_ev1" and "WMD5" do not refer to individuals, but to concepts (or types, or kinds, which we will use interchangeably). Saying that there is some sub-concept of the kind WeaponOfMassDestruction, where this sub-concept is further restricted to be a kind of WMD possessed by Iraq, does not thereby commit you to saying that there are any *instances* of this sub-concept. However, some concepts, like CountryOfIraq and USPresident43, are such that (a) it is generally known that there is an instance of the concept, and that (b) it is a unique instance.

The role relations further restrict the sub-concepts. Thus, "possess_ev3" is not just any kind of PossessingObject state, it is one where the CountryOfIraq does the possessing, and some sub-kind of WMD is what is possessed. The temporal relations also restrict the sub-concepts: e.g. the kind of possession that holds before Now.

The instantiable assertions commit the representation to the existence of the kinds of object described. In the top-level context "t", there is a commitment to an instance of USPresident43 and of an InformationClaimingEvent made by him. However, there is no top-level commitment to any instances of WeaponOfMassDestruction possessed by the CountryOfIraq. These commitments are only made in the embedded "claim_cx1" context. It is left open whether these embedded commitments correspond, or not, to the beliefs of the speaker.

Two distinct levels of structure can thus be discerned in AKR (2): a conceptual structure and a contextual structure. The conceptual structure, through use of "sub_concept" and "role" assertions, indicates the subject matter. The contextual structure indicates differing commit-

[1]The names of these terms are strictly arbitrary, but in this paper they will be made mnemonic to ease readability.

ments to the existence of the subject matter via instantiability assertions linking concepts to contexts, and via context relations linking contexts to contexts.

This two-fold structure, and more particularly the absence of reference to individual objects, is somewhat unusual amongst standard, mostly first-order knowledge representations (Lenat 1995, Clark and Porter 1998). As Condoravdi et al. (2001) argue, reference to concepts rather than individuals makes it easier to deal with a variety of downward monotone contexts where the non-existence of individuals is entailed, such as:

(3) The technician prevented an accident.

(4)
```
context(t)
context(prevent_cx1)
context_relation(prevents(prevent_ev1), t, prevent_cx1)
sub_concept(prevent_ev1, Event)
sub_concept(technician2, Technician)
sub_concept(accident3, AccidentEvent)
role(doneBy, prevent_ev1, technician2)
temporalRel(precedes, prevent_ev1, Now)
instantiable(prevent_ev1, t)
instantiable(technician2, t)
instantiable(accident3, prevent_cx1)
uninstantiable(accident3, t)
```

The AKR in (4) says that there is an instance in "t" of some kind of Event done by some kind of Technician, and that, whatever this kind of event was, it manifests a "prevent" relation between "t" and another context "prevent_cx1". There is also some sub-concept of AccidentEvent, "accident3", such that in "t" this sub-concept is uninstantiable (i.e., it does not occur), whereas in the "prevent_cx1" context that the technician's action prevents, the sub-concept is instantiable. For more details on how this kind of analysis differs from first-order or intensional higher-order ones, see Condoravdi et al. (2001) and Bobrow et al. (2005).

18.2.2 Ontologies for AKR

An ontology provides a collection of concepts and relations between concepts, often arranged in some kind of hierarchy (either single inheritance or multiple inheritance). It may also provide more specific axioms, e.g. for any object O, if it is moved to location L, then O is located at L immediately after the movement event. Not all ontologies provide such axioms, however. The ontology may also provide lexical

information, stating which words map onto which concepts. Again, not all ontologies provide such lexical information, and many that do are neither particularly complete nor detailed with respect to that mapping.

One of the chief obstacles in mapping text to AKR is to find a suitable ontology, where suitability means three things.

1. The ontology should be a good staging post for mapping to alternatives in case there is to be a final mapping from abstract KR to a concrete KR; different KR formalisms tend to use different ontologies.
2. The ontology should be rich enough to support the inferential distinctions that are actually made in natural language.
3. The ontology should either come with or be paired with lexical mappings, showing how words and phrases align with the ontology.

These criteria are hard to satisfy jointly. A large scale ontology like Cyc (Lenat 1995) makes many inferential distinctions, but its lexical pairing is patchy. To the extent that WordNet (Fellbaum 1998) and VerbNet (Kipper et al. 2000) encode ontologies, they have broad lexical coverage. But they are not set up to encode information important to inference in the way done by Cyc and other ontologies explicitly designed for reasoning. As for converting between different ontologies, WordNet is often used as a *de facto* standard where ontologies state how their concepts align with WordNet synonym sets; but this is in danger of being a form of conversion via the lowest common denominator.

Considerable work has been put into unifying diverse lexical and ontological resources to provide suitable word to concept and role pairings (Crouch and King 2005, Gurevich et al. 2006). Two alternative strategies have been pursued.

- Use the Cyc ontology as a target and estimate mappings for unknown words onto Cyc when there are sufficient similar examples available, and failing this, back off to a shallower (pseudo) ontology such as WordNet synsets and VerbNet roles.
- Uniformly use just WordNet synsets and VerbNet roles as a shallow ontology but with fairly broad lexical coverage.

Unifying lexicons and ontologies is beyond the scope of the current paper, however; it is mentioned just to point out that it is a problematic area where more work is required. In this paper we will assume a lexical pairing with a Cyc-like ontology for the purposes of illustration.

18.3 Packing and Packed Rewriting

This section describes the mechanism that is used to run the rules that map f-structures first onto semantic representations and then onto KR representations. Discussion of the differences between these levels of representation and the kinds of rules required to bring about the mapping are left to the next section. Here the aim is to describe the principles behind the packed rewriting system. To do this, it is first necessary to describe packed representations. Packed representations were originally devised by Ron Kaplan and John Maxwell to efficiently compute and encode syntactic ambiguities in f-structures. However, the technique is quite general, and the discussion here will be pitched in terms of packed AKRs.

18.3.1 Packed Representations

Although constraints can sometimes be applied at the KR level to resolve syntactic ambiguities, others will pass through to the KR level, and yet more may be introduced by such things as word sense ambiguity.

Alternative interpretations are represented in a packed form. An example will give an idea of what these packed representations are like:

(5) John saw a man with a telescope.

(6)
```
choice:  (A1 xor A2) iff 1
    1:   context(t)
    1:   sub_concept(see_ev1, Seeing)
    1:   sub_concept(john2, Person)
    1:   sub_concept(man3, MaleAdult)
    1:   sub_concept(telescope4, TelescopeOpticalDevice)
    1:   role(doneBy, see_ev1, john2)
    1:   role(objectPerceived, see_ev1, man3)
    1:   role(name_of, john2, 'John')
   A1:   role(instrumentUsed, see_ev1, telescope4)
   A2:   role(thingPossessed, man3, telescope4)
    1:   instantiable(see_ev1, t)
    1:   instantiable(john2, t)
    1:   instantiable(man3, t)
    1:   instantiable(telescope4, t)
```

The standard prepositional attachment ambiguity is reflected in AKR (6) by two alternative role restrictions: the telescope is either the instrumentUsed in the seeing event, or the thingPossessed by the man. The two alternatives are labeled by the distinct choices "A1" and "A2".

As the first line in the representation states, "A1" and "A2" are mutually exclusive (xor = exclusive or) ways of partitioning the true choice labeled "1". Most parts of the representation are common to both possible interpretations, and are thus labeled with the choice "1". It is only the two role assignments for "telescope4" that are put under distinct choice labels.

A slightly more complex case of prepositional attachment ambiguity gives rise to the following AKR (shown somewhat abbreviated):

(7) John saw a man in a park with a telescope.

(8)
```
choice:   (A1 xor A2) iff 1
choice:   (B1 xor B2 xor B3) iff A1
choice:   (C1 xor C2) iff A2

    1:    context(t)
    1:    sub_concept(see_ev1, Seeing)
    1:    sub_concept(john2, Person)
    1:    sub_concept(man3, MaleAdult)
    1:    sub_concept(park4, ParkOpenArea)
    1:    sub_concept(telescope5, TelescopeOpticalDevice)
    1:    role(doneBy, see_ev1, john2)
    1:    role(objectPerceived, see_ev1, man3)
    1:    role(name_of, john2, 'John')
   A1:    role(location, man3, park4)
   A2:    role(location, see_ev1, park4)
B1 or C1: role(instrumentUsed, see_ev1, telescope5)
B2 or C2: role(thingPossessed, park4, telescope5)
   B3:    role(thingPossessed, man3, telescope5)
```

Here there are some interactions between the attachments: if the location of the man is the park ("A1"), then *with a telescope* can modify either the seeing ("B1"), the park ("B2"), or the man ("B3"). But if the location of the seeing event is the park ("A2"), then *with a telescope* can only modify either the seeing ("C1") or the park ("C2"). This is reflected in the choice structure, which says that "A1" and "A2" are a disjoint partition of "1", and that "A1" is in turn partitioned into "B1", "B2", and "B3", while "A2" is partitioned into "C1" and "C2".

Note that the five possible readings for (7) are represented in not much more space than the two readings for (5). It is possible to count the number of readings by looking only at the choice space: "A1" has three alternatives sitting under it, "A2" has two, and "A1" and "A2" are disjoint, so there are $3 + 2 = 5$ alternatives altogether.

18.3.2 Free Choice Packing

Packing relies on the observation that a large number of alternative natural language analyses are typically generated by a small number of (relatively) independent ambiguities. For example, n independent 2-way ambiguities in a sentence will lead to 2^n possible interpretations. Thus, in a (rather contrived) sentence like

(9) The deer led the sheep to the cabbage.

where *deer* can be singular/plural, *sheep* can be singular/plural, and *cabbage* can be count/mass, three independent two-way ambiguities give rise to $2^3 = 8$ readings. This could be represented (schematically) as:

(10)
```
choice:  (A1 xor A2) iff 1
choice:  (B1 xor B2) iff 1
choice:  (C1 xor C2) iff 1

    A1:  role(cardinality, deer1, singular)
    A2:  role(cardinality, deer1, plural)
    B1:  role(cardinality, sheep2, singular)
    B2:  role(cardinality, sheep2, plural)
    C1:  role(divisibility, cabbage3, count)
    C2:  role(divisibility, cabbage3, mass)
   ...   ...
```

Here we can make a completely free choice between "A1/A2", "B1/B2" and "C1/C2".

The problem is that ambiguities are not always completely independent of one another. For example, in the prepositional attachment ambiguity (7) one might be tempted to say that there is one 2-way choice — is the man or the seeing in the park — and a second independent 3-way choice — is the seeing, the man or the park with the telescope. But this will not do. If the seeing is in the park, then it cannot be the man who is with the telescope: the phrase structure tree for this attachment would have crossing branches.

One possibility for (7) would be a representation that includes a 'nogood' on the choice space:

(11)
```
choice:   (A1 xor A2) iff 1
choice:   (B1 xor B2 xor B3) iff 1
nogood:   (A2 and B3)

   1:   context(t)
   1:   sub_concept(see_ev1, Seeing)
   1:   sub_concept(john2, Person)
   1:   sub_concept(man3, MaleAdult)
   1:   sub_concept(park4, ParkOpenArea)
   1:   sub_concept(telescope5, TelescopeOpticalDevice)
   1:   role(doneBy, see_ev1, john2)
   1:   role(objectPerceived, see_ev1, man3)
   1:   role(name_of, john2, 'John')
  A1:   role(location, man3, park4)
  A2:   role(location, see_ev1, park4)
  B1:   role(instrumentUsed, see_ev1, telescope5)
  B2:   role(thingPossessed, park4, telescope5)
  B3:   role(thingPossessed, man3, telescope5)
```

The choice alternatives indicate that you have a completely free choice of either "A1" or "A2" and of "B1", "B2", or "B3". But the nogood is a bit of 'contractual small print' saying that the choice is not quite as free as it appears to be, and that the combination of "A2" and "B3" is disallowed.

A key observation made by Kaplan and Maxwell (1995) is that nogoods can always be eliminated from the choice space. Thus the choice structure in (8), repeated below as (13), is equivalent to the choice space with a nogood from (11), repeated below as (12).

(12)
```
choice:   (A1 xor A2) iff 1
choice:   (B1 xor B2 xor B3) iff 1
nogood:   (A2 and B3)
```

(13)
```
choice:   (A1 xor A2) iff 1
choice:   (B1 xor B2 xor B3) iff A1
choice:   (C1 xor C2) iff A2
```

What has happened in going from (12) to (13) is that the original 3-way "B"-choice has been unpacked a little to give one 3-way choice under "A1" ("B1 ... B3") and another 3-way choice under "A2" ("C1 ... C3"). The nogood is then cashed out to eliminate the "C3" choice under "A2", leaving just "C1 ... C2".

What is the benefit of eliminating the nogood in this way? On the surface, it would appear that it just succeeds in increasing the size of the representation: more labels in the choice space, and more complex

Boolean combinations of labels attached to the facts in the representation. But the free-choice spaces obtained by eliminating nogoods have three computationally important properties:

1. The choice structure constitutes an and-or graph, where the satisfiability of Boolean combinations of choice labels is very simple to check. A conjunction of labels is unsatisfiable iff the lowest node in the and-or tree dominating both of them is an or-node. What this means is that you are trying to conjoin two mutually exclusive arms of a disjunction, which cannot be done. This contrasts favorably with the complexity of the general propositional satisfiability problem, which is NP-complete, and which is what would result from leaving nogoods in the choice structure.
2. Any single reading can be read out in a time linear in the length of the representation. Starting at the first choice, pick an alternative ("A1" or "A2"). Then descend through all the other choice alternatives picking further alternatives compatible with the choices already made. For instance, in (8), if you have chosen "A1" you can choose any one of "B1" to "B3" and ignore all the alternatives. Then go through each fact in the representation in turn, determining whether its Boolean combination of labels is made true by the choices made. The absence of nogoods means that you can safely descend through the choice structure, making choices compatible with what you have already decided, without running the risk of later discovering that the combination of choices was ruled out. This is what makes it a free-choice structure.
3. Free choice structures allow you to efficiently compute the number of analyses without having to enumerate each one. This is key for efficient stochastic disambiguation of packed structure (Riezler and Vasserman 2004). Moreover, the counting can be done without any reference to the kind of representation being packed, whether f-structure, semantic representation or AKR. This means that methods for stochastic disambiguation of packed structures can be applied to different kinds of representation (though presently, training material has only been assembled for c- and f-structure disambiguation).

But there is a price to be paid for these properties. Re-arranging the choice space to eliminate nogoods increases the size of the choice space, and the complexity of the choices decorating individual clauses. Creating new Boolean combinations of choices can have the same effect. In the worst case, where all choices interact with all other choices, the choice space can expand to a size proportional to the (exponential)

total number of possible analyses. That is, in the worst case, packing degrades into a disjunctive normal form enumeration of each analysis. The Kaplan-Maxwell wager is that these bad cases rarely arise for linguistically typical input. Choices from distant parts of sentences or texts usually do not interact (coordinations and long distance dependencies being exceptions), and free choice spaces normally remain a manageable size.

Free Choice Packing and Underspecification

Free choice packing differs from underspecification as a technique for ambiguity management. Underspecification delivers a set of fragmentary analyses plus a set of constraints recording interactions between fragments and limiting the ways they can be put together. The constraints perform the role of nogoods in non-free choice packing. Free choice packing delivers a chart-like structure that records all completely assembled analyses, and does not require further constraint satisfaction checks to read out or count individual analyses. Underspecification is a form of procrastination: work is not done on evaluating constraints until it is necessary, in the expectation that for many of the constraints it will never become necessary. Packing computes everything, whether it needs to or not, on the basis that it is often easier to do everything at once than it is to carefully distinguish what needs to be computed now from what can be left until later. While underspecification in semantics has attracted a lot of attention, packing has been relatively and unjustly neglected. To keep packed representations a manageable size, it is important to pull disjunctions of meanings out into the choice space, and not to represent them explicitly as modalized disjunctions within the semantic representation (c.f. Ramsay 1999). The latter approach tends to multiply out the size of representations when disjunctions are embedded and/or distributed in other disjunctions.

18.3.3 The Packed Rewriting System

XLE parsing produces packed f-structures. The goal of f-structure to AKR mapping is to produce packed AKRs from packed f-structures without having to unpack; that is, without enumerating each f-structure analysis separately, converting it to AKR, and then packing the AKRs back together again. A packed rewriting system is used to achieve this.

The rewrite system is a reimplemented version of a transfer component originally devised by Martin Kay for use in machine translation (Frank 1999). As with Oepen (2004) and Wahlster (2000), rewriting is a resource-sensitive process. The input is a set of clauses, and rewrite rules apply in an ordered, stepwise manner to progressively replace in-

put clauses by output clauses. The system's novel feature is that it is able to operate directly on packed input.

Rule Formalism

A somewhat contrived[2] example of a rewrite rule is:

(14)
```
        PRED(%X, eat),
        SUBJ(%X, %S),
        OBJ(%X, %O),
        –OBL(%X, %%)
==>
        sub_concept(%X, EatingEvent),
        role(doneBy, %X, %S),
        role(objectConsumed, %X, %O).
```

This rule looks at a set of clauses describing an f-structure to see if there is some node %X (the % is used to indicate a variable), with a subject %S and object %O, but no oblique. If the left hand side of the rule is matched, the matching PRED, SUBJ and OBJ clauses are removed from the description, and are replaced by the sub_concept and role clauses on the right hand side of the rule.

More generally, the format for rewrite rules is shown in figure 1. The left hand sides of rules contain boolean combinations of patterns over clauses. Clauses are atomic predicates heading a set of argument terms, where the terms may be non-atomic: e.g. "SUBJ(var(0), var(1))", where SUBJ is the predicate and var(0) and var(1) are the non-atomic arguments. In patterns over clauses, some of the argument terms can be, or can contain, variables. For example, the pattern "SUBJ(%X, var(%Y))" will match "SUBJ(var(0), var(1))", setting %X to var(0) and %Y to 1. Second-order quantification over atomic predicates is also available, where "qp(%P, [%X, %Y])" matches "SUBJ(var(0), var(1))", setting the predicate variable %P to SUBJ and the list of argument variables [%X, %Y] to var(0) and var(1).

By prefixing a clause pattern on the left hand side of a rule with a "+", you can indicate that the rule should check for the presence of a matching clause in the input without deleting the clause. Likewise, a prefix of "–" checks that a pattern is not matched by the input. Boolean combinations of clause patterns are possible, as are calls to external procedures. External procedures are forbidden from directly manipulating the full set of input clauses; instead they allow you to perform table lookup or tests on terms, such as subsumption checking

[2]Individual lexical mappings would not in practice be handled by rules like (14). See (15) for a more realistic treatment.

Rule	::=	LHS ==> RHS.	*Obligatory rewrite*
		LHS ?=> RHS.	*Optional rewrite*
		LHS *=> RHS.	*Recursive rewrite*
		|– Clause.	*Permanent, unresourced fact*
LHS	::=	Clause	*Match & delete atomic clause*
		+Clause	*Match & preserve atomic clause*
		LHS, LHS	*Boolean conjunction*
		(LHS | LHS)	*Boolean disjunction*
		–LHS	*Boolean negation*
		{ProcedureCall}	*Procedural attachment*
RHS	::=	Clauses	*Set of replacement clauses*
		0	*Empty set of replacement clauses*
		stop	*Abandon the analysis*
Clause	::=	Atom(Term,. . . ,Term)	*Clause with atomic predicate*
		Atom	*Atomic clause*
		qp(Variable, [Term_i,. . . , Term_n])	*Clause with unknown predicate and n arguments*
Term	::=	Variable	
		Clause	

FIGURE 1 Format of Rewrite Rules

in the Cyc generalization hierarchy, or looking up the synset of a word in WordNet.

The right hand side of a rule can be a comma separated set of clause patterns, including the empty set represented as "0". The right hand side can also be the directive "stop", which means that the analysis path should be deleted.

Rules can be obligatory, optional or recursive rewrites, and can also introduce permanent non-consumable facts. If the left hand side of an obligatory rule is matched, then the consumed clauses (i.e. those not marked with a "+" or a "–") have to be removed from the set of input clauses, and replaced by the clauses on the right hand side of the rule. For an optional rule, conceptually speaking, there is a fork in the set of output clauses. On one fork the rule applies, and the consumed clauses on the left hand side are replaced by those on the right hand side. On the other fork the rule does not apply, and the set of clauses remains unchanged. But instead of forking the sets of clauses, the choice space is split to record the alternatives where the rule is and is not applied (see below). A recursive rule can re-apply to its own output, provided that each recursion also consumes some of the input that was present before the first recursive application; this ensures termination of the recursion.

Rules are ordered: rule 1 applies to the input, rule 2 applies to the output of rule 1, and so on. Rule ordering can be exploited in useful ways, e.g. to encode sequences of defaults, but the feeding and bleeding behavior needs to be handled with some care. The rule ordering also means that the scope of any recursion is strictly limited to a single rule.

Clauses preceded by a |– are included as permanent, non-consumable facts. These are part of neither the input nor the output, but can be called on to provide tests or data. For example, a more sensible way of achieving the effects of (14) would be:

```
(15) |– concept-map(eat, V-SUBJ-OBJ, Eating,
                    doneBy, objectConsumed).
     |– concept-map(drink, V-SUBJ-OBJ, Drinking,
                    doneBy, objectConsumed).
     ...
            PRED(%X, %P),
            SUBJ(%X, %S),
            OBJ(%X, %O),
            –OBL(%X, %%),
            concept-map(%P, V-SUBJ-OBJ, %C, %SR, %OR)
      ==>
            sub_concept(%X, %C),
            role(%SR, %X, %S),
            role(%OR, %X, %O).
```

In this way, a large number of lexical mappings can be asserted permanently, and a single rule takes care of the concept mapping for transitive verbs.

The formalism also allows macros to be used to parameterize commonly occurring patterns in rules, and templates to parameterize commonly occurring sequences of rules. This is an alternative to using a type hierarchy (Oepen 2004) for producing compact rule sets.

Rewriting and the Choice Space

There is nothing especially innovative about the range of rewrite rules available. What is significant is that the rules can be applied to packed input. Suppose that we have the following rule and packed input:

(16) Rule: p(%X), q(%X) ==> r(%X).
Packed input: `A1:`p(a) `B2:`q(a) `C3:`s(a)

The rule matches the input only in the conjunction of choices `A1` and `B2`, where both patterns on the left hand side can simultaneously be matched. This leads to the production of a new output fact, r(a), only under this conjoined choice (`A1 & B2`), and deletion of the matching input facts only under the same conjoined choice. The matched input facts remain present under the residual choices that do not overlap with `A1 & B2`. The input fact s(a) is unaffected by the rule, and remains in its initial state. Thus the results of the rule application are:

(17) Packed output:

`A1 & ¬B2:`	p(a)	`B2 & ¬A1:`	q(a)
`A1 & B2:`	r(a)	`C3:`	s(a)

In creating these new conjoined choices, the rewriting system will alter the choice space to keep it in a free choice form. Typically this involves introducing new choice labels, so that `A1 & B2` will correspond to a new label, e.g. `D`, that is dominated by both `A1` and `B2` in the and-or choice structure graph. It is important to keep the choice space in a free choice form, since the applicability of the rule depends on the Boolean `A1 & B2` being satisfiable, and testing satisfiability needs to be kept efficient.

Application of an optional rule conceptually splits an analysis into two disjoint parts. Rather than forking two separate rewrite subprocesses (Oepen 2004), new choices are introduced to pack the two analysis paths together into the same process. For example:

(18) Rule: p(%X), q(%X) ?=> r(%X).
Packed input: `A1:` p(a) `B2:` q(a) `C3:` s(a)

The rule matches under choice `A1 & B2`. This choice must now be split into two disjoint parts: `O1` where the rule is applied, and `O2` where the rule is not applied. This new split in the choice space can be represented via the following choice definition:

(19) choice: `(O1 xor O2)` iff `A1 & B2`

which says that if you are under `A1 & B2` then you have a completely free choice of exactly one of `O1` or `O2`, and that if you are under either `O1` or `O2`, then you must also be under the choice `A1 & B2`. The output of the rule is:

(20) Packed output:

`¬(A1 & B2) or O2:`	p(a)	`¬(A1 & B2) or O2:`	q(a)
`O1:`	r(a)	`C3:`	s(a)

Rewrite rules can also introduce new choices through the resolution of resource conflicts. Suppose that we have the following:

(21) |– concept-map(bank, N, BankFinancialInstitution).
|– concept-map(bank, N, BankEarthBarrier).

PRED(%X, %P),
+NOUN(%X,+),
concept-map(%P, N, %C)
==>
sub_concept(%X,%C).

Input: `1:` PRED(sk3, bank) `1:`NOUN(sk3,+).

The two concept mapping facts mean that the rewrite rule can apply in two distinct ways to consume the input clause. But they cannot both simultaneously consume the same input clause. The rewriting system

detects such competition over input resources, and resolves the conflict by introducing a new split in the choice space. One rule application will apply on one side of the split, and the other rule application on the other. Hence the output is:

(22) `S1:` sub_concept(sk3,BankFinancialInstitution)
`S2:` sub_concept(sk3,BankEarthBarrier)
`1:` NOUN(sk3,+)
(New) choice: `(S1 xor S2)` iff `1`

The rule 'ambiguates' the word "bank", but ensures that the two senses sit under mutually exclusive choices. Thus "bank" cannot mean both financial institution and mound of earth, but means either one or the other.

18.4 F-Structure, Semantics and AKR

So far, we have seen what the target AKRs for sentences are like, considered packing as a means for efficiently representing ambiguity, and looked at a rewriting system that can operate on packed structures. It is now time to pull this together, and see how packed rewriting can transform f-structures into abstract knowledge representations.

This transformation is done in two steps. First, f-structures are mapped onto semantic representations. Then, semantic representations are mapped onto AKRs. The reason for this two step mapping is in part historical. In the original implementation of the system, f-structures were first mapped to semantic representations via Glue Semantics (Dalrymple 2003, Crouch 2005), at which point packed rewriting took over to map the semantic representations onto AKR. However, packing within Glue has not yet been properly implemented, so that packed f-structures had to be unpacked, interpreted and then repacked. The introduction of recursive rules into the rewrite system gave it sufficient expressive power to do semantic interpretation without the inefficiency of unpacking and repacking. In addition, despite the possibility of using the same rewriting mechanism to map f-structures to semantics to AKR, there are theoretical reasons for distinguishing these levels of representation.

This section therefore first motivates separate levels of semantics and AKR, and then describes some of the mapping rules used.

18.4.1 F-Structure and Semantic Representation

In representing the grammatical functional structure of sentences, f-structure encodes a basic form of predicate-argument structure. As shown in van Genabith and Crouch (1999), f-structures can be placed

in correspondence with underspecified semantic representations such as Quasi Logical Form (Alshawi and Crouch 1992) and Underspecified Discourse Representation Structures (Reyle 1993). Two observations should be drawn from this. First, f-structures need to be notationally transformed in various ways to become semantic representations with the familiar machinery of connectives, quantifiers and variables for supporting logical inference. Second, f-structures only correspond to *underspecified* versions of such representations: there is semantic information that is not encoded in f-structure, even in a notationally alternative form.

With regard to the first observation, although differences in notational form may seem superficial, surfaces can run remarkably deep: notation matters.[3] It is not just the absence of devices like logical variables or connectives that matters. It is also that f-structure can deemphasize some distinctions that, while grammatically minor, are of major logical and inferential significance.

One example of this is the distinction between (a) first-order, entity denoting arguments of predicates (approximately, nominal subjects, objects, etc.), and (b) proposition denoting arguments (approximately, clausal COMPs and XCOMPs, etc.). Propositional arguments require logically special treatment, increase the complexity of any underlying inference system, and need to be marked as different. F-structures mark this distinction only in passing, if at all. This typing information can be indirectly obtained by looking at the grammatical function of the argument (COMP or XCOMP vs. SUBJ or OBJ). But in some adjunct structures (e.g. for sentential modifiers like "allegedly" or "possibly"), even this indirect typing information can be missing.

Other examples of these differences in notational emphasis between f-structure and semantics include: (i) Uniform representation of coordination, despite the semantically diverse effects of noun phrase, noun, verb phrase, sentential, and adjunct coordination; (ii) The "inside-out" representation of modifiers in which semantic representations wrap modifiers around the modifyee to alter the environment in which the modifyee is interpreted, whereas f-structure places the modifiers in an adjunct set within the modifyee; and (iii) Different representations of modifier scope / surface order.

With regard to the second observation that f-structures correspond to underspecified semantic representations, it follows that a single f-structure can give rise to a number of distinct interpretations. This means that f-structures lack certain semantic information. Sometimes

[3] A point on which Ron Kaplan has always been adamant.

the missing information is predictable. For example, with quantifier scope ambiguity, an enumeration procedure can generate the alternative interpretations, so that the f-structures do not lack important semantic information. But sometimes f-structures lack information that is not systematically recoverable. For example, some verbs permit semantically distinct internal and external modification:

(23) John put up the tent for half an hour.

where it is either the putting-up activity that lasts half and hour, or the amount of time the tent is left standing. Other verbs do not give rise to this distinction, e.g. *John watched the tent for half an hour.* Syntactically, the two verbs behave similarly, and f-structures for the two sentences will contain no hint of their divergent semantic behavior.

To conclude, the motivation for a semantic representation is both to reformat information in a way that better supports semantic processing, and to spell out interpretive variations that are not made explicit in the syntax.

18.4.2 Semantics and KR

What is the motivation for a level of knowledge representation over and above a semantic representation? Surely natural language, and hence natural language semantics, is the most natural knowledge representation there is? Given an ideal analysis of language and its semantics on the one hand, and an ideal analysis of knowledge representation and implementation of reasoning on the other, you might reasonably expect the two to coincide. A number of researchers have assumed no fundamental distinction between linguistic semantics and (the current state of the art in) KR. In some cases, this is a practical consequence of working on language understanding for limited domains. Here, linguistic analysis can be specially tailored to meet the domain, e.g. eliminating many forms of ambiguity at the outset, and/or avoiding anything other than special case analyses of inferentially troublesome higher-order or intensional constructions (Allen et al. 1996). In other cases this is a consequence of the assumption that natural language semantics can be made to coincide with what is currently tractable in automated reasoning: either through care and ingenuity in formulating a sophisticated first-order analysis (Hobbs 1985, Blackburn et al. 2001) or through the assumption that cases falling outside of an essentially quantifier-free first-order logic are sufficiently rare as to be insignificant (Moldovan and Rus 2001).

Even if first-order logic were sufficient for NL semantics, there is still a clash of compositionalities between semantics and KR to be overcome.

Semantic representations must respect the syntactic composition of the texts from which they are derived, to achieve a general and systematic syntax-semantics mapping. Consequently, the semantic representations assigned to sentences tend to be more complex, and different, than the representations a knowledge engineer would assign on a case-by-case basis when targeting a particular knowledge base. Additional processing is required to overcome such "impedance mismatches", which are more evident when one is trying to produce, say, database queries in SQL from semantic representations (Rayner 1993). Semantics to AKR transformations include:

1. Map words / word senses onto terms in the target ontology, though this can also be done in semantics.

2. Make meaning postulates / lexical entailments explicit in the KR. For example, in:

(24) Bush knew that Iraq possessed WMDs.

the speaker is committed to the assertion that Iraq possessed WMDs. The AKR should make this commitment explicit by propagating instantiability claims from the embedded context to the upper context (Nairn et al. 2006):

(25) context(t)
context(know_cx1)
context_relation(informationConveyed(know_ev1), t, know_cx1)
sub_concept(know_ev1, KnowledgeState)
sub_concept(Bush2, USPresident43)
sub_concept(possess_ev3, PossessingObject)
sub_concept(Iraq4, CountryOfIraq)
sub_concept(WMD5, WeaponOfMassDestruction)
role(performedBy, know_ev1, Bush2)
role(informationConveyed, know_ev1, know_cx1)
role(possessor, possess_ev3, Iraq4)
role(thingPossessed, possess_ev3, WMD5)
role(cardinality, WMD5, plural)
temporalRel(precedes, know_ev1, Now)
temporalRel(precedes, possess_ev3, Now)
instantiable(know_ev1, t)
instantiable(Bush2, t)
instantiable(possess_ev3, know_cx1)
instantiable(possess_ev3, t)
instantiable(Iraq4, know_cx1)
instantiable(Iraq4, t)
instantiable(WMD5, know_cx1)
instantiable(WMD5, t)

3. Canonicalize compositionally distinct but equivalent semantic representations onto the same KR. The sentences in (26) receive different compositional semantic analyses, but are both mapped to the same KR (27):

(26) The technician cooled the room.
The technician lowered the temperature of the room.

(27) decreasesCausally(lower_ev22, room24, temperatureOfObject)
role(doneBy, lower_ev22, technician23)
sub_concept(technician23, TechnicianWorker)
sub_concept(lower_ev22, ScalarStateChange)
sub_concept(room24, RoomInAConstruction)
instantiable(lower_ev22, t)

4. Eliminate ontologically ill-formed analyses. For example, in "John saw a man with Mary", selectional restrictions in the ontology may rule out the parse where the prepositional phrase modifies the verb "see".

5. Reformulate intensional and higher-order aspects of the semantic representation in a form more amenable for KR, as illustrated in (4)

and (25).

18.4.3 Examples of Semantic Mapping Rules

We will illustrate the use of recursive rules to flatten out the contextual structure implicit in f-structure. F-structures are recursive, with one node being embedded inside another, yet it is straightforward to represent this as a flat set of clauses. For (24) these might be along the (abbreviated) lines of:

(28) PRED(var(0), know)
PRED(var(1), Bush)
PRED(var(2), Iraq)
PRED(var(3), possess)
PRED(var(4), WMD)
SUBJ(var(0), var(1))
COMP(var(0), var(3))
SUBJ(var(3), var(2))
OBJ(var(3), var(4))
TNS-ASP(var(0), var(5))
TENSE(var(5), past)
TNS-ASP(var(3), var(6))
TENSE(var(6), past)

The f-structure node var(3) is embedded under var(0), and var(2) is in turn embedded under var(3). But not all of the f-structure embeddings lead to context embeddings in the semantics: in fact, it is only the nodes var(0) and var(3) that introduce semantic contexts.

The f-structure to semantics rewrite rules therefore need to make a recursive traversal of the f-structure, linking each f-structure node to the nearest dominating node that introduces a semantic context. This is achieved in three stages. First, nodes introducing a context are identified and labeled (where "c(...)" is wrapped around a node to indicate its context), e.g.

(29) +COMP(%N1, %N2)
==>
new_context(%N2, c(%N2))
in_context(%N2, c(%N2)).

Second, all immediate links between f-structure nodes are labeled, e.g.

(30) +SUBJ(%N1, %N2) ==> link(%N1, %N2).
+OBJ(%N1, %N2) ==> link(%N1, %N2).

+COMP(%N1, %N2) ==> link(%N1, %N2).
+TNS-ASP(%N1, %N2) ==> link(%N1, %N2).

Finally, a recursive rule traverses the links propagating the in_context labels:

(31)
+in_context(%N1, %C),
link(%N1, %N2),
–new_context(%N2,%%)
==>
in_context(%N2, %C).

Each recursive step will consume one of the "link(. . . ,. . .)" facts, ensuring that the recursion terminates. The recursion will simultaneously start at all the nodes initially labeled as being in_context by rule (29), and the negative test on new_context ensures that nodes are only connected back to their immediately dominating context. It is important to remember that this rule only applies recursively to its own output. This is unlike more general recursion in a set of unordered rules, where rules can recursively apply to the output of other rules.

18.4.4 Examples of AKR Mapping Rules

Coverage and robustness are major issues when deriving KR from free text. Full, detailed KR coverage of texts is not likely to be achieved, but gaps in coverage should not lead to a failure to produce an analysis. Resourced rewriting allows one to specify an ordered sequence of backoffs and defaults to call on rules and external data of varying degrees of reliability. First, one applies carefully hand-coded rules to consume as much of the input as possible. Then a first level of backoff is applied to mop up some of the remaining input. For example, the system described here calls on language-to-KR mapping rules imported from Cyc. A second level of backoff consults other external data, such as WordNet and VerbNet, in order to guess plausible concepts for unknown words bearing some similarity to known words. Finally, for any original input still remaining, concept names are systematically invented. The most extreme form of backoff is to leave elements of the input unchanged. Invented concepts are of lesser use to a backend reasoning system with no ontological information about them, although they still enable inferences based on strict identity of concepts. Similar levels of backoff are used to deal with other phenomena, such as clausal complement verbs that set up contexts that are veridical (e.g. "know"), averidical (e.g "believe") or anti-veridical (e.g. "refute") (Nairn et al. 2006). Unknown contexts default to being averidical.

Packing and conflict resolution handle ambiguous mappings, e.g.

(21). The system can also eliminate ambiguities. A rule for mapping the verb "hire" and its grammatical arguments onto concepts and roles is:

(32)
```
PRED(%E, hire), SUBJ(%E,%X), OBJ(%E,%Y),
+sub_concept(%X,%CX), +sub_concept(%Y,%CY),
{genls(%CX, Organization), genls(%CY,Person)}
==>
sub_concept(%E, EmployingEvent),
role(performedBy,%E,%X),
role(personEmployed,%E,%Y).
```

Consider "the bank hired Smith", assuming that rules have already enforced a choice split and assigned the nominal arguments to concepts, so that the "bank" corresponds to either BankFinancialInstitution or BankEarthBarrier. Given that a procedural call to the concept generalization (genls) hierarchy confirms that BankFinancialInstitution is a subtype of Organization, but BankEarthBarrier is not, the rule above will only apply under the choice where "bank" maps to BankFinancialInstitution. Applying the rule will consume the PRED, SUBJ and OBJ inputs, but leave them in place under the alternate choice where bank maps to BankEarthBarrier. After other concept and role mapping rules come disambiguation rules like:

(33)
```
SUBJ(%%,%%) ==> stop.
OBJ(%%,%%) ==> stop.
```

If there are any choices under which the SUBJ and OBJ facts still remain, by the time the rules above apply, these choices are abandoned. Thus, if no other rules have used up the SUBJ input under the EarthBarrier choice, then these rules eliminate the sortally ill-formed reading.

Softer, stochastic disambiguation can in principle be applied to the packed output KR, using the same mechanisms as for MaxEnt disambiguation of f-structures (Riezler et al. 2002). This has yet to be done, and depends on (a) defining a set of disambiguation features, and (b) obtaining lightly annotated material to train weights on these features.

18.5 Further Work

Evaluation material is not available for assessing the *quality* of the full text to KR system. Existing evaluation material like PropBank does not go as deeply as required, though it will be used to assess argument selection in the parser. In terms of *quantity*, full coverage is obtained.

The best that can be done in the meantime is to report, somewhat

anecdotally, on the utility of the rewriting system. It is distributed as a fully integrated part of the XLE, under the XLE's research license. It has been used for a variety of tasks inside and outside PARC, including sentence condensation (Riezler et al. 2003, Crouch et al. 2004), converting the TIGER dependency annotations to f-structures (Forst 2003), constructing FrameNet annotations for German (Frank and Semecky 2004), tutoring systems (Burton, this volume), and also machine translation. Timing figures are only moderately informative, but with the current set of mapping rules, packed semantic input from 30-40 word sentences can typically be converted in well under a second on a 2GHz processor.

Major effort is still required to build up the AKR mapping rules to minimize the number of times mapping backs off to default rules. There is also scope for further improvement in the rewriting system, including parallel rule application, improved rule indexing, and further optimization of the order in which LHS patterns are matched.

Acknowledgments

This work builds on the efforts of numerous people: John Maxwell and Ron Kaplan for the library interface to XLE ambiguity management routines; Tracy King in particular, and Annie Zaenen, Anette Frank, Martin Forst, Stefan Riezler and Anubha Kothari for using and breaking the rewrite system while it was under initial development but coming back for more; Danny Bobrow, Cleo Condoravdi, and Valeria Paiva for assistance in defining AKR and writing AKR mapping rules; Reinhard Stolle, John Everett and Liz Coppock for other work on KR; and Martin Kay for a prototype version of the rewrite system. This paper is a much revised and extended version of one that appeared in the 6th International Workshop on Computational Semantics. The work was supported in part by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program.

References

Allen, James F., Bradford W. Miller, Eric K. Ringger, and Teresa Sikorski. 1996. A robust system for natural spoken dialogue. In A. Joshi and M. Palmer, eds., *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL'96)*, pages 62–70. Santa Cruz, CA: Morgan Kaufmann.

Alshawi, Hiyan and Richard Crouch. 1992. Monotonic semantic interpretation. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL'92)*, pages 33–39. Newark, DE.

Blackburn, Patrick, Johan Bos, Michael Kohlhase, and Hans de Nivelle. 2001. Inference in computational semantics. In H. Bunt, R. Muskens, and E. Thijsse, eds., *Computing Meaning*, vol. 2, pages 11–28. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Bobrow, Daniel G., Cleo Condoravdi, Richard Crouch, Ronald M. Kaplan, Lauri Karttunen, Tracy H. King, Valeria de Paiva, and Annie Zaenen. 2005. A basic logic for textual inference. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-05), Workshop on Inference for Textual Question Answering*. Pittsburgh, PA.

Butt, Miriam, Tracy H. King, María-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. Stanford, CA: CSLI Publications.

Clark, Peter and Bruce Porter. 1998. *KM — The Knowledge Machine 2.0: The Users Manual*. AI Lab, University of Texas at Austin.

Condoravdi, Cleo, Richard Crouch, Martin van den Berg, Reinhard Stolle, Valeria de Paiva, John O. Everett, and Daniel G. Bobrow. 2001. Preventing existence. In C. A. Welty and B. Smith, eds., *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS-2001)*, pages 162–173. Ogunquit, ME: ACM Press.

Crouch, Richard, Tracy H. King, John T. Maxwell, III, Annie Zaenen, and Stefan Riezler. 2004. Exploiting f-structure input for sentence condensation. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2004 (LFG'04)*, pages 167–187. Christchurch, New Zealand: CSLI Online Publications.

Crouch, Richard. 2005. Packed rewriting for mapping semantics to KR. In *Proceedings of the 6th International Workshop on Computational Semantics (IWCS-6)*, pages 103–114. Tilburg, The Netherlands.

Crouch, Richard and Tracy H. King. 2005. Unifying lexical resources. In *Proceedings of Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*, pages 32–37. Saarbrücken, Germany.

Dalrymple, Mary. 2003. *Lexical Functional Grammar*. San Diego, CA: Academic Press.

Fellbaum, Christiane, ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: The MIT Press.

Forst, Martin. 2003. Treebank conversion — creating an f-structure bank from the TIGER Corpus. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 2003 (LFG'03)*, pages 205–216. Albany, NY: CSLI Online Publications.

Frank, Anette. 1999. From parallel grammar development towards machine translation. In *Proceedings of Machine Translation Summit VII. MT in the Great Translation Era*, pages 134–142. Singapore: Kent Ridge Digital Labs.

Frank, Anette and Jiri Semecky. 2004. Corpus-based induction of an LFG syntax-semantics interface for frame semantic processing. In S. Hansen-Schirra, S. Oepen, and H. Uszkoreit, eds., *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), 5th International Workshop on Linguistically Interpreted Corpora (LINC-04)*, pages 39–46. Geneva, Switzerland.

Gurevich, Olga, Richard Crouch, Tracy H. King, and Valeria de Paiva. 2006. Deverbal nouns in knowledge representation. In *Proceedings of the 19th Florida Artificial Intelligence Research Society Conference (FLAIRS'06)*. Melbourne Beach, FL.

Hobbs, Jerry R. 1985. Ontological promiscuity. In *Proceedings of the 23th Annual Meeting of the Association for Computational Linguistics (ACL'85)*, pages 61–69. Chicago, IL.

Kaplan, Ronald M. and John T. Maxwell, III. 1995. A method for disjunctive constraint satisfaction. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell, III, and A. Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, pages 381–401. Stanford, CA: CSLI Publications.

Kipper, Karin, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00) and the 12th Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-00)*, pages 691–696. Austin, TX.

Lenat, Douglas B. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the Association for Computing Machinery (ACM)* 38(11):33–38.

Maxwell, John T., III and Ronald M. Kaplan. 1996. An efficient parser for lexical functional grammar. In M. Butt and T. H. King, eds., *Proceedings of the International Lexical-Functional Grammar Conference 1996 (LFG'96)*. Grenoble, France: CSLI Online Publications.

Moldovan, Dan I. and Vasile Rus. 2001. Logic form transformation of Wordnet and its applicability to question answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*, pages 394–401. Toulouse, France.

Nairn, Rowan, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of thr 5th Conference on Inference in Computational Semantics (ICoS-5)*. Buxton, United Kingdom.

Oepen, Stephan. 2004. Som å hoppe etter wirkola. Unpublished Manuscript.

Ramsay, Allan. 1999. Dynamic & underspecified semantics without dynamic and underspecified logic. In H. Bunt and R. Muskens, eds., *Computing Meaning*, vol. 1, pages 57–72. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Rayner, Manny. 1993. *Abductive Equivalential Translation and its application to Natural Language Database Interfacing*. Ph.D. thesis, Royal Institute of Technology, Stockholm.

Reyle, Uwe. 1993. Dealing with ambiguities by underspecification: Construction, representation and deduction. *Journal of Semantics* 10(2):123–179.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 271–278. Philadelphia, PA.

Riezler, Stefan, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for Lexical-Functional Grammar. In *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, pages 118–125. Edmonton, Canada.

Riezler, Stefan and Alexander Vasserman. 2004. Incremental feature selection and l_1 regularization for relaxed maximum-entropy modeling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*, pages 174–181. Barcelona, Spain.

van Genabith, Josef and Richard Crouch. 1999. Dynamic and underspecified semantics for LFG. In M. Dalrymple, ed., *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach.*, pages 209–260. Cambridge, MA: The MIT Press.

Wahlster, Wolfgang, ed. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Berlin, Germany: Springer Verlag.

Index

Agreement, 301–320
  adjective, German, 16
  gender, Rumanian, 301–320
  noun class, Gĩkũyũ, 205, 215, 217
  verb, Urdu, 250n
  with reflexive pronoun, German, 160
Animacy, 243, 323–335
  and gender resolution, 301–320
  hierarchy, 323, 328–335
Applicative, 206, 237, 239n
Argument structure, 236, 237, 245, 367, 368, 370, *see also* Linking theory; Parallel projection architecture

C-structure, *see* Constituent structure
CAT predicate, 175, 226–228
Categorial Grammar, 374, 375, 380, 381
Causative
  Chichewa, 242–243
  French, 242–243
  Hindi, 243
  Sanskrit, 243n
  Urdu, 245–256
Causatives, 242–254, *see also* Complex predicates
Coherence, *see also* Lexical coherence; Phrasal coherence
  at f-structure, 128, 227, 228n, 244, 254, 255
Coherence relations, 352, 353
COMP, 151–152, 178, 406
Completeness, 227, 228n, 244, 254, 255
Complex predicates, 235–256
Compositionality, 340, 343, 348–350, 363–384, 407, 409
  direct, 364, 370–373
  principle of, 363
Constituent structure, 54, 112, 218n, 226, 227, 228n, 230, 237, 240, 262–263, 340, 365–370, 374, 376–378, 380–382
Coordination, 107, 115, 125, 146–150, 216, 259–283, 301–320, 344, 348, 349, 383, 399
  asymmetric, 147–148, 260–283
  constituent, 259–260
Curry-Howard isomorphism, 370, 376, 381, 383

D-LTAG, 349–351
Data sets
  AP treebank, 113, 114, 124
  ATB treebank, 126
  ATIS, 113, 122
  British National Corpus, 119
  Cast3LB treebank, 125, 126

COMLEX, 118, 119
CTB treebank, 125, 126
DCU 105 F-Structure Bank, 117
Europarl, 44
Kyoto University Corpus, 126
OALD, 118, 119
P7T treebank, 126
PARC 700, 117, 121–124
Penn treebank, 95, 98, 100, 113–116, 118–125, 127, 128
PropBank, 117, 123, 124
SprogTeknologisk Ordbase, 169–185, 188, 190, 193, 194
SUSANNE treebank, 114, 119
Switchboard, 327–329, 334
TIGER treebank, 124–126, 137, 138, 145, 146, 149–151, 154, 156–158, 161, 163, 413
VerbNet, 393, 411
WordNet, 393, 402, 411
Dependency
anaphoric, 342–345, 349–357
syntactic, 340–344
Direct syntactic encoding, 255
Discourse
organization
informational, 339
intentional, 339
relations, 339–358
constituency, 345–349
Distribution, 122
distributive features, 263, 315n
into sets, 148, 259, 263, 266, 268, 275–277
Dual-projection hypothesis, 223–226

Economy of expression, 261, 263, 276–278
Ellipsis, 152–153, 202n, 344, 383
Endocentricity, 201, 222, 224, 229, 231
Extended head theory, 228–231

F-score, 97, 105, 128n
definition, 97n, 161
F-structure, *see* Functional structure
Finite state morphology, 89, 96, 173, 235, 249–252, 256, 287–299
Urdu, 249–252
XFST, 291
Frege's principle, 363
Functional structure, 53–55, 81, 112–113, 218n, 236, 237, 262–263, 365–370, 374, 376–378, 380–382
f-description, 55

Generation, 19–32, 42, 53–70, 126–128, *see also* Translation; XLE
algorithms, 19–32
chart generation, 19–23, 67–68
from packed representation, 21, 23–32
head-driven, 20
shake-and-bake, 20
ambiguity-preserving, 31, 69–70
definition, 19
from acyclic structures, 60–68
from fully specified structures, 59–60
decidability theorem, 60
from induced grammars, 126–128
from underspecified structures, 42, 56–59
decidability theorem, 57
from unknown predicates, 42
in statistical machine translation, 36–37
off-line generability restriction, 57–60, 70
probabilistic, 126–128
evaluation, 128
with restriction operator, 68–69
decidability theorem, 68
Gerunds, *see* Mixed categories

Glue semantics, 255, 278–279, 363–384, 405
Grammar development, 137–164, *see also* Grammar induction; ParGram; XLE
  ambiguity management, 87, 155–160, 169, 184–185, *see also* Tagging
  efficiency techniques, 154–160
  evaluation, 160–164
  grammar specialization, 81, 155–158
  lexical resources, 118–120, 167–195
  long-distance dependencies, 158–160
  robustness techniques, 86, 88–89, 137–164, *see also* XLE, fragment grammar, normalization of input text
Grammar induction, 67, 111–129
  domain variation, 121–122
  evaluation, 116–117, 119–120, 122–124
  f-structure annotation algorithm, 113–117
  long-distance dependencies, 115–116, 119–121
Grammatical functions, 178–180, *see also* COMP, XCOMP
  grammaticized discourse functions, 261, 274, 276
  hierarchy, 325–327, 331–333, 335

Haspelmath's Generalization, 220–224

Infinito sostantivato, *see* Mixed categories
Inside-out function application, 155, 160, 226
Intensional logic, 372, 382, 383

Knowledge representation, 390–413
  and semantics, 407–410
  conceptual structure, 391
  contextual structure, 391
  Cyc, 390, 393, 402, 411
  Knowledge Machine, 390
  ontologies, 392–393

Left dislocation, 325–335
Lenient composition, 288, 290–293, *see also* Finite state morphology
  definition, 290
Lexical coherence, 213, 214, 225
Lexical integrity, 201, 222, 231, 251n
Lexical Mapping Theory, *see* Linking theory
LFG-DOP, 113
Linear logic, 370
Linguistic Discourse Model, 348–349
Linking theory, 226, 237, 239, 244
Logical Form, 364, 370, 372, 373, 380
Logistic regression, 327–330
  definition, 330
Long distance dependencies, 35, 112, 326–335, 341, 399, *see also* Grammar development; Grammar induction; Left dislocation; Topicalization

Machine translation, *see* Translation
Mapping Theory, *see* Linking theory
Mixed categories, 201–231
  infinito sostantivato, 202–203, 222, 223
  dual-projection hypothesis, 223–226
  Gĩkũyũ agentive nominalizations, 203–229
  gerunds, 107, 193, 202–203
  single-projection hypothesis, 221–223

Modal subordination, 280–283

Off-line generability restriction, 57–60, 70
Optimality theory, 237, 287–299
  and finite state morphology, 289
  categorical constraints, 292, 293
  GEN function, 291–292
  gradient constraints, 290, 292, 293

Packed representations, 20, 97, 389–413, *see also* Generation, algorithms, from packed representation
  definition, 396–399
Parallel projection architecture, 237–238, 245, 254, 365–384, *see also* Constituent structure, functional structure, argument structure, semantic (σ) structure
Parentheticals, 150–152
ParGram, 167, 235, 245, 249, *see also* Grammar development; XLE
  Danish grammar, 167–195
  English grammar, 39, 44, 80n, 89, 95–107, 137, 167
  German grammar, 39, 44, 137–164
  Norwegian grammar, 168, 169
  Urdu grammar, 235, 249–254
Parsing, *see also* XLE
  chart parsing, 20
  definition, 19
  integrated architecture, 120–121
  pipeline architecture, 120–121
  robustness techniques, 47
Part-of-speech tagging, *see* Tagging
Passive, 29, 118, 122, 170, 175, 180, 181, 183, 188–190, 193, 237, 238, 239n, 244, 246, 247n, 252–254, 324, 325
Permissive, 239–242
Phrasal coherence, 213, 222, 229
Priority union, 290

Realizational Morphology, 235, 251n
Resolution rules, 301–320
  semantic resolution, 305–306, 315–319
  syntactic resolution, 307–315, 317–319
Restriction operator, 68–69, 238–256
Rhetorical Structure Theory, 345–348

Semantic (σ) structure, 56, 251n, 364–382, *see also* Glue semantics; Parallel projection architecture; XLE, semantic interpretation
Single-projection hypothesis, 221–223
Subject Condition, 253–255

Tagging, 91–107
  ambiguity reduction, 103–107
Templates, *see* XLE, lexical structure
Tokenization, *see under* XLE
Topicalization, 170, 325–335
Transfer rules, *see under* XLE
Translation, 3–17, 31–32, 35–48, *see also* Generation, ambiguity-preserving
  statistical machine translation, 35–48, 128–129
    Pharaoh system, 38, 39, 42, 44
    phrase-based, 35, 37–39, 44–48
  theory of, 3–17

with induced grammars, 128–129
word alignment, 37
Treebanks, *see* Data sets
Tutoring systems, 75–89
SOPHIE, 75, 80

XCOMP, 89, 159, 160, 178, 406
XFST, *see* Finite state morphology
XLE, 75–89, 96–97, 169, 172–174, 244, 253, 301, 302, 313n, 389–413, *see also* Packed representations
fragment grammar, 42, 44, 45, 47, 97, 100, 152, 161, 162
generation, 58, 70, 191–192
grammar libraries, 172n, 192
guesser, 96
lexical structure, 159, 175–192
morphological analysis, 41, 138, 142–144, 173–174, 188–192, *see also* Finite state morphology
vs. full form lexicon, 173
normalization of input text, 140–141
optimality marks, 42
semantic interpretation, 83–84
skimming, 100, 160–163
syntactic analysis, 81–83
tokenization, 96, 138–141, 144, 145
transfer rules, 42–48, 83–86, 89, 394–413
contiguity constraint, 40–41
extraction, 39–41
syntax of, 84

CSLI Lecture Notes
Number 179

This volume collects papers that are at the cutting edge of research in computational as well as theoretical linguistics. As all of the papers represent research areas in which Ronald M. Kaplan has made foundational contributions, the papers in the volume represent a tribute to the vital role he has played in the development of computational linguistic research and linguistic theory, particularly within Lexical-Functional Grammar (LFG).

Part one, "Generation and Translation," contains contributions on the design of the most optimal architecture for machine translation, parsing and generation, as well as proposals for a machine translation system which successfully combines statistical methods with deep natural language processing.

Part two, "Grammar Engineering and Applications," focuses on practical natural language processing such as using the LFG grammar development platform XLE for implementing tutoring systems, building large lexicons and grammars, exploring interactions of tagging and parsing, and building large grammars and lexical resources from treebanks.

Part three, "Formal Issues," examines difficult linguistic data and their treatment in formal linguistic theory. These papers range from contributions on Optimality Theoretical vs. finite-state treatments of Finnish prosody to mixed-category constructions to theories of discourse and coordination. Foundational issues addressed include interactions between morphology and syntax in complex predicates, coordination and agreement, and the resolution of coordination asymmetries via f-structure analysis.

Part four, "Semantics and Inference," examines the fundamental issue of compositionality in syntactic and semantic theory, and presents cutting edge research on theoretical and practical issues in mapping from linguistic structures to knowledge representations.

Miriam Butt is Professor of Theoretical and Computational Linguistics at the Department of Linguistics, Universität Konstanz. ***Mary Dalrymple*** is University Lecturer in General Linguistics at the University of Oxford and Fellow of Linacre College. ***Tracy Holloway King*** is a senior member of the research staff at the Palo Alto Research Center and adjunct associate professor in Symbolic Systems at Stanford University.

CSLI
PUBLICATIONS
Center for the Study of
Language and Information
Stanford University

ISBN-13 978-1-57586-532-4
ISBN-10 1-57586-532-7