Hierarchical Lexicon and the Argument/Adjunct Distinction

Adam Przepiórkowski
Institute of Computer Science, Polish Academy of Sciences
Institute of Philosophy, University of Warsaw

Proceedings of the LFG'17 Conference

University of Konstanz

Miriam Butt, Tracy Holloway King (Editors)

2017

CSLI Publications

pages 348–367

http://csli-publications.stanford.edu/LFG/2017

Keywords: lexicon, valency, FrameNet, argument-adjunct distinction

Abstract

The first aim of this paper is to propose a fully hierarchical organisation of valency information in LFG, inspired by FrameNet and by recent LFG work on using templates to encode valency. The second aim is to demonstrate that this proposal meshes particularly well with a recent proposal *not* to distinguish arguments from adjuncts.

1 Introduction

In many constraint-based linguistic theories, as well as in some lexicographic projects, lexical information is organised hierarchically. In such a hierarchy, internal nodes represent generalisations pertaining to various portions of the lexicon. These generalisations are inherited by 'lower' nodes. The 'lowest' nodes – the 'leaves' in the hierarchy – typically correspond to specific lexical items, which inherit generalisations from all the nodes on the way up to the root of the hierarchy, and only add truly idiosyncratic information such as the orthographic form. This approach to the lexicon is an important aspect of Head-driven Phrase Structure Grammar (cf., e.g., Flickinger 1987 and Davis 2001), but similar proposals have also been made within Lexicalized Tree-Adjoining Grammar (Vijay-Shanker and Schabes 1992) and within Categorial Grammar (van der Linden 1992), *inter alia*. Hierarchical organisation is also an important feature of WordNet (Miller et al. 1990, Fellbaum 1998) and FrameNet (Fillmore et al. 2003, Fillmore and Baker 2015, Ruppenhofer et al. 2016). While in all these approaches hierarchies represent mainly syntactic and semantic generalisations, Network Morphology (Corbett and Fraser 1993), based on the lexical representation language DATR (Evans and Gazdar 1996), is concerned with morphological and morphosyntactic generalisations.

The possibility of adopting such a comprehensive taxonomic approach to the lexicon has never been seriously entertained within Lexical Functional Grammar. The first aim of this paper is to propose an organisation of the LFG lexicon that is close to that of FrameNet. The technical side of this proposal is relatively straightforward, assumes the Glue approach to LFG semantics (Dalrymple 1999, 2001), makes heavy use of templates (Dalrymple et al. 2004, Asudeh et al. 2008, 2013), and does not require any formal extensions to the underlying LFG machinery, but does require some care to avoid the spurious multiple introduction of meaning constructors. An accompanying paper, Przepiórkowski 2017a, which shares with the current paper most of the material of the initial three sections, demonstrates the feasibility of this FrameNet-inspired approach within the standard LFG.

The second aim of this paper is to argue that such a hierarchical organisation of lexical information fills a gap in the recent proposal *not* to distinguish arguments from adjuncts in LFG, made in Przepiórkowski 2016. That proposal follows the conservative assumption that typical arguments are introduced in the lexicon and

[†]I am grateful to both reviewers for their comments. The research reported here is partially supported by the Polish Ministry of Science and Higher Education within the CLARIN ERIC programme 2016–2018 (http://clarin.eu/).

typical adjuncts are introduced by syntactic rules (with the proviso that some arguments may be introduced constructionally in the syntax and the so-called 'obligatory adjuncts' may be introduced in the lexicon). Importantly, together with other approaches to adjunction, that approach assumes that typical adjuncts may in principle modify any head (of an appropriate grammatical category), subject to some vague 'semantic compatibility' requirement. Here, we show that the hierarchical lexicon approach makes it possible to encode the distribution of adjuncts more precisely. As a by-product, the last vestige of the argument/adjunct distinction is removed from the analysis of Przepiórkowski 2016: according to the analysis proposed here, both types of dependents are normally introduced lexically (which does not preclude the possibility of some being introduced constructionally, pace Asudeh et al. 2013).

2 Inheritance in FrameNet

FrameNet organises lexical knowledge with reference to cognitive structures called *frames*. Various lexical items may evoke the same frame. For instance, the Apply_heat frame is evoked by verbs such as BAKE, FRY, GRILL, STEW, etc. Frames also define *frame elements*, i.e. – simplifying a little – semantic roles which are normally expressed by dependents of lexical items evoking the frame. In the case of Apply_heat, typical frame elements are the Cook and the Food, but also the Container that holds the Food to which heat is applied, the Medium through which heat is applied to Food, etc. In the following examples, verbs evoking the Apply_heat frame are in bold:[1]

(1) **Boil** [the potatoes]$_{\text{Food}}$ [in a medium-sized pan]$_{\text{Container}}$.

(2) [Drew]$_{\text{Cook}}$ **sauteed** [the garlic]$_{\text{Food}}$ [in butter]$_{\text{Medium}}$.

Frames are linked via a number of relations, including the hierarchical multiple-inheritance relation. For example, Apply_heat inherits semantic roles from both Activity and Intentionally_affect frames, the latter inherits from Intentionally_act, which in turn inherits from Event (see Figure 1). It is not clear whether this is a design feature of FrameNet or just a reflection of its work-in-progress status, but it happens in current versions of FrameNet (including the latest at the time of writing this paper, 1.7) that the same role is introduced multiple times in the hierarchy. For example, within the fragment of the inheritance hierarchy in Figure 1, the Agent role is introduced independently at Activity, at Objective_influence (where it is called Influencing_entity; see below) and at Intentionally_act.

Another feature of FrameNet is that inherited roles, as they acquire more specialised meanings, may change names.[2] For example, the agentive role introduced at Objective_influence is called Influencing_entity there, but gets renamed to Agent

[1]These made up examples are taken from the description of the Apply_heat frame at the FrameNet web interface, at https://framenet2.icsi.berkeley.edu/.

[2]This correspondence between frame elements of different frames is currently not shown in the web interface to FrameNet, but it is explicitly defined in the distributed version of the lexicon, in the file `frRelation.xml`.

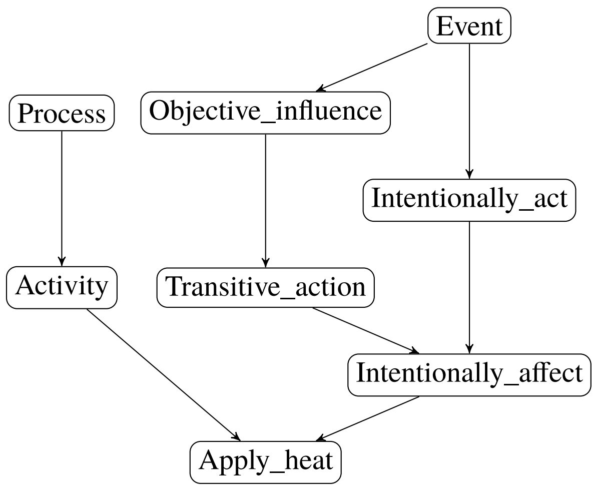


Figure 1: A fragment of the FrameNet 1.7 inheritance hierarchy – all frames from which Apply_heat inherits

when it is inherited by Transitive_action. The Agent roles of both Transitive_action and Intentionally_act correspond to the (single) Agent role of Intentionally_affect, but the role of Apply_heat corresponding to the Agent roles of both Activity and Intentionally_affect is renamed to Cook. Similarly, the Food role of Apply_heat corresponds to the Patient role of Intentionally_affect and Transitive_action above it (where it is renamed from the Dependent_entity role of Objective_influence). Below, we will simplify by adopting single names for roles related via inheritance hierarchy. For example, instead of Cook and Food, the respective roles of Apply_heat will be called Agent and Patient, as on the superordinate frames. But, as always, it should be borne in mind that a role on a subordinate frame will usually carry more entailments than the homonymous role on a superordinate frame.

An important aspect of FrameNet is that frame elements correspond to both arguments and adjuncts. For example, among the roles associated with Apply_heat are roles realised by typical adjuncts, such as Manner, Time and Place. A FrameNet reflex of the argument/adjunct dichotomy is its categorisation of roles into core (corresponding to arguments) and non-core (corresponding to adjuncts), but the criteria used for deciding whether a role is core or not suffer from the usual problems of providing only partial tests or being pairwise incompatible. In particular, Fillmore and Baker 2015: 801, admitting that "there are clear cases and unclear cases in trying to draw this distinction", propose the following partial criteria for coreness:

- if a role is obligatorily expressed, it is core,
- roles realised as subjects and direct objects are core (but there are exceptions),
- roles "expressed by phrases with lexically specific morphological marking" are core.

As discussed in Przepiórkowski 2016: 562–563, dependents may be obligatory in various ways and for different reasons, some of them pragmatic in nature. For example, the apparently obligatory adjuncts (Grimshaw and Vikner 1993) occurring after *The house was built* (see below) are arguably (Goldberg and Ackerman 2001) obligatory only in a rather weak sense of being often – but not always – needed to satisfy Grice's maxim of quantity:

(3) #This house was built.

(4) This house was built {yesterday / in ten days / in a bad part of town / only with great difficulty / by a French architect}.

Probably the same very weak notion of obligatoriness is invoked in FrameNet in the case of the core roles Time and Place of the Event frame (neither of the other two criteria for coreness applies here). This frame is directly evoked by verbs such as HAPPEN or OCCUR, which might seem to require the presence of Time or Place: #*This event occurred.* However, just as in the case of the "obligatory adjuncts" combining with *The house was built*, it is easy to find fully acceptable occurrences of HAPPEN and OCCUR without any realisation of Time or Place, e.g.:

(5) If it's not on Facebook it didn't happen.[3]

(6) Scientists manipulate brains of mice to make them think fake event really occurred.[4]

Hence, obligatoriness is not a fully operational criterion for distinguishing core from non-core, even when treated as just a unidirectional test (*if* it is obligatory, *then* it is core).

Also the second criterion above, referring to subjects and direct objects, is not reliable even when construed as unidirectional. Fillmore and Baker 2015: 801 (fn. 4) exempt from this criterion direct objects in dative constructions such as *you* in *I'll bake you a cake*, as well as direct objects in resultative constructions such as *my plate* in *I ate my plate clean*. Since no meta-criteria are given for the applicability of this criterion, it is – again – not operational even as a partial test.

Finally, the third unidirectional criterion, referring to the possibility to express a role with "lexically specific morphological marking", is also difficult to apply in practice, as specific morphological marking on dependents is rarely a feature of single lexemes, but usually a matter of smaller or larger classes of lexemes. Ruppenhofer et al. 2016: 24 illustrate this criterion with the verb DEPEND and its lexically specified dependent introduced by the preposition ON, but even in this case this kind of dependent is a feature of a coherent class of verbs, which includes RELY, COUNT, LEAN, etc. What seems to matter here is the *size* of the class of verbs or frames with which a given morphosyntactic type of dependent occurs bearing a specific meaning. This is made clear in the following quote (Ruppenhofer et al. 2016: 24): "The preposition ON does not occur as a marker of the same meaning [i.e. the same as in the case of DEPEND – AP] with predicates in many other frames. In its basic spatial

[3] http://www.wordyard.com/2016/01/19/if-its-not-on-facebook-it-didnt-happen/

[4] http://www.independent.co.uk/news/science/is-it-inception-total-recall-no-science-fact-false-implanted-in-mice-brains-8732466.html; this is a headline, hence the lack of articles.

sense of 'in contact with and supported by', ON occurs in many different frames; as a marker of Place or Location frame elements it is totally unremarkable and does not suggest core status for these [frame element]s." But without any specification of the minimal size of the class of verbs or frames which justifies the assignment of the non-core status to a role with a specific morphosyntactic realisation, this criterion is again not operational.

It is clear, then, that the core/non-core dichotomy in FrameNet is as ill-understood as the infamous argument/adjunct distinction. For this reason we will mostly ignore this dichotomy below, but we will still model it by making core roles obligatory and non-core roles optional. It should be borne in mind, though, that this is a vast oversimplification, since – as discussed above – there is no strong assumption in FrameNet that all core roles are syntactically obligatory. Moreover, semantic obligatoriness cuts across the class of non-core roles and is the basis of the distinction within this class between peripheral roles (semantically obligatory but – unlike core roles – not central to the meaning of the frame) and extra-thematic roles (semantically optional). Again, the issue of the proper modelling of these various aspects of obligatoriness requires substantial research.

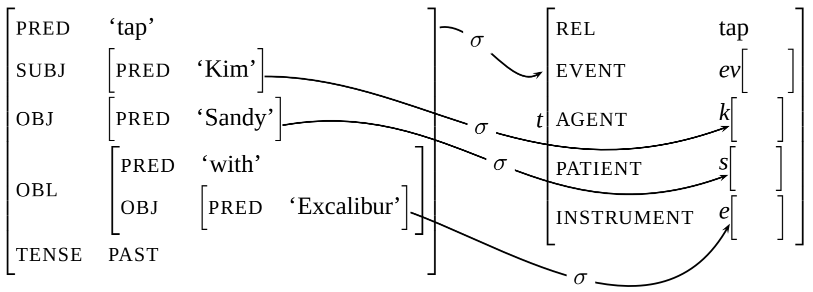
What is interesting is that inheritance may change the coreness status of a role. For example, as discussed above, at the Event frame the roles Time and Place are marked as core, but the same roles are treated as non-core on almost all of the 27 frames directly subordinate to Event.[5] Obviously, the Event frame pertains to situations which normally occur at some time and at some place, so in this sense Time and Place are semantically obligatory, but the same can be said about all frames inheriting from Event, on which, however, Time and Place are not marked as core. The reverse situation happens in case of the Existence frame, where Time and Place are non-core, but become core on its directly subordinate frame, Circumbscribed_existence. Such changes of coreness seem to bring non-monotonicity to the otherwise monotonic inheritence relation in FrameNet. Below, we will see how such apparently non-monotonic behaviour can be modelled via the monotonic means of Lexical Functional Grammar.

3 Valency in LFG

As is common in LFG, we assume the existence of a level of representation which encodes the semantic argument structure, i.e., which contains information about semantic (or thematic) roles such as Agent or Goal. Traditionally, semantic forms – values of PRED – served this purpose in Lexical Functional Grammar (Kaplan and Bresnan 1982). Alternatively, we could employ the distinct level of argument structure of Butt et al. 1997. Instead, we build here on more recent work and assume the formalisation of argument structure within the semantic structure (Asudeh and Giorgolo 2012, Asudeh et al. 2014, Findlay 2016). For example, Asudeh and Giorgolo 2012: 78, propose the following f-structure and s-structure for the sentence *Kim tapped Sandy with Excalibur*:

[5]The only exception is the Emergency frame, which treats Time as core.

(7)



Much of the work in this thread assumes that some of the semantic attributes – those which correspond to arguments in the scope of the (Lexical) Mapping Theory (Bresnan and Kanerva 1989) as formalised in Findlay 2016 – are called ARG$_1$, ARG$_2$, etc., instead of the more mnemonic names such as AGENT and PATIENT, but – as we are not concerned with LMT in this paper – we will continue to use these more intuitive names.

What kind of lexical entries give rise to such f- and s-structures? Let us consider a simpler case, that of the transitive verb *devoured*:

(8) lexical entry for *devoured*, 1st version:

devoured V $(\uparrow \text{PRED}) = \text{`DEVOUR'}$
$(\uparrow_\sigma \text{ REL}) = \text{DEVOUR}$
$\lambda e.\ devour(e) : (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$
$(\uparrow \text{SUBJ})_\sigma = (\uparrow_\sigma \text{AGENT})$
$\lambda P \lambda x \lambda e.\ P(e) \wedge agent(e,x) :$
$[(\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \text{AGENT}) \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$
$(\uparrow \text{OBJ})_\sigma = (\uparrow_\sigma \text{PATIENT})$
$\lambda P \lambda x \lambda e.\ P(e) \wedge patient(e,x) :$
$[(\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \text{PATIENT}) \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$
$(\uparrow \text{TENSE}) = \text{PAST}$
$\lambda P.\ \exists e\ P(e) \wedge past(e) : [(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma$

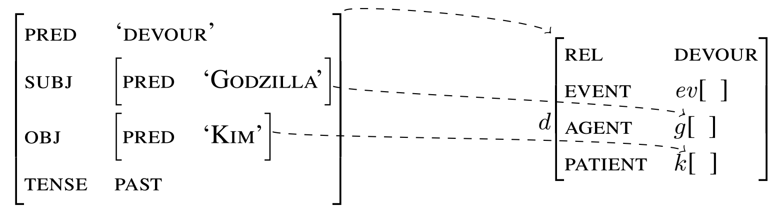
There are four natural parts of this lexical entry: 1) the idiosyncratic part, defining PRED,[6] as well as the corresponding s-structure attribute REL,[7] and introducing the basic meaning constructor containing the *devour* relation in its meaning representation; 2) the part saying that the subject grammatical function realises the agent semantic relation: 3) the analogous part defining the correspondence between the object and the patient; 4) the part adding tense information to the f-structure and to the meaning representation, as well as defining the existential closure over the event

[6]As noted already in Dalrymple et al. 1993: 13–14, and Kuhn 2001: § 1.3.3, Glue makes PRED largely superfluous. Here I follow Asudeh and Giorgolo 2012 and retain PRED, but with values reflecting the predicate *sans* its valency – hence, the value of PRED in (8) is defined as 'DEVOUR' rather than 'DEVOUR⟨SUBJ, OBJ⟩'.

[7]The existence of the semantic attribute REL is assumed – but not formally introduced – in a number of recent papers, including Asudeh and Giorgolo 2012, Asudeh et al. 2008, 2013, 2014 and Findlay 2016; the following lexical entries make the introduction of this attribute explicit.

variable. In the case of the sentence *Godzilla devoured Kim*, this lexical entry gives rise to the f- and s-structures in (9) below, as well as to the instantiated meaning constructors in (10).

(9)



(10) 1. $\lambda e.\ devour(e) : ev \multimap d$

2. $\lambda P \lambda x \lambda e.\ P(e) \wedge agent(e,x) : [ev \multimap d] \multimap g \multimap ev \multimap d$

3. $\lambda P \lambda x \lambda e.\ P(e) \wedge patient(e,x) : [ev \multimap d] \multimap k \multimap ev \multimap d$

4. $\lambda P.\ \exists e\ P(e) \wedge past(e) : [ev \multimap d] \multimap d$

These instantiated meaning constructors, together with the following instantiated meaning constructors introduced by the lexical entries of *Godzilla* and *Kim*, may be used to derive the expected meaning representation for the whole sentence: $\exists e\ devour(e) \wedge agent(e, godzilla) \wedge patient(e, kim) \wedge past(e)$.

(11) 5. $godzilla : g$

6. $kim : k$

Here is one possible proof:

(12) 7. $\lambda x \lambda e.\ devour(e) \wedge agent(e,x) : g \multimap ev \multimap d$ (from 2 and 1)

8. $\lambda e.\ devour(e) \wedge agent(e, godzilla) : ev \multimap d$ (from 7 and 5)

9. $\lambda x \lambda e.\ devour(e) \wedge agent(e, godzilla) \wedge patient(e,x) : k \multimap ev \multimap d$ (from 3 and 8)

10. $\lambda e.\ devour(e) \wedge agent(e, godzilla) \wedge patient(e, kim) : ev \multimap d$ (from 9 and 6)

11. $\exists e\ devour(e) \wedge agent(e, godzilla) \wedge patient(e, kim) \wedge past(e) : d$ (from 4 and 10)

Obviously, apart from the first – idiosyncratic – part of the lexical entry (8), the other three parts will also occur in many other lexical entries, so it makes sense to encode them as templates (Dalrymple et al. 2004):[8]

(13) $\textsc{agent} := (\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{agent})$
$\lambda P \lambda x \lambda e.\ P(e) \wedge agent(e,x) :$
$[(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \textsc{agent}) \multimap (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

(14) $\textsc{patient} := (\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{patient})$
$\lambda P \lambda x \lambda e.\ P(e) \wedge patient(e,x) :$
$[(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \textsc{patient}) \multimap (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

[8] In the actual templates, the parts defining the correspondence between a semantic argument and a grammatical function will be more complex, to allow for diathesis; see Asudeh et al. 2014 and Findlay 2016.

(15) $\textsc{past} := (\uparrow \textsc{tense}) = \textsc{past}$

$$\lambda P.\, \exists e\; P(e) \wedge past(e) : [(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma$$

With these templates in hand, the lexical entry for *devoured* simplifies to:

(16) lexical entry for *devoured*, 2nd version:

devoured V $(\uparrow \textsc{pred}) = \text{‘}\textsc{devour}\text{’}$
$(\uparrow_\sigma \textsc{rel}) = \textsc{devour}$
$\lambda e.\, devour(e) : (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$
$@\textsc{agent} \quad @\textsc{patient} \quad @\textsc{past}$

LFG templates may call other templates, and in this sense they form a kind of hierarchy. In the context of valency, this possibility was explored by Asudeh et al. (2008, 2013) in their account of English and Swedish *way*-constructions, as in: *Bill elbowed his way through the crowd.* Similarly, Asudeh et al. (2014) make use of such embedded template calls for example when defining a prototypical transitive argument structure:

(17) $\textsc{agent-patient} := @\textsc{agent} \;\; @\textsc{patient}$

The proposal presented below may be seen as taking the approach summarised above to its logical conclusion.

4 Lexically introduced adjuncts in LFG

4.1 Frame elements via templates

As shown in Przepiórkowski 2017a, the main gist of the current proposal is compatible with standard LFG and the traditional feature architecture of f-structures, but here we assume the approach of Patejuk and Przepiórkowski 2016 (inspired by Alsina 1996 and work within HPSG), on which all syntactically realised dependents of a predicate are members of the DEPS set,[9] with just a couple of grammatical functions – typically SUBJECT and OBJECT – additionally singled out via separate attributes. This is illustrated in Figure 2, which contains the desired f- and s-structures

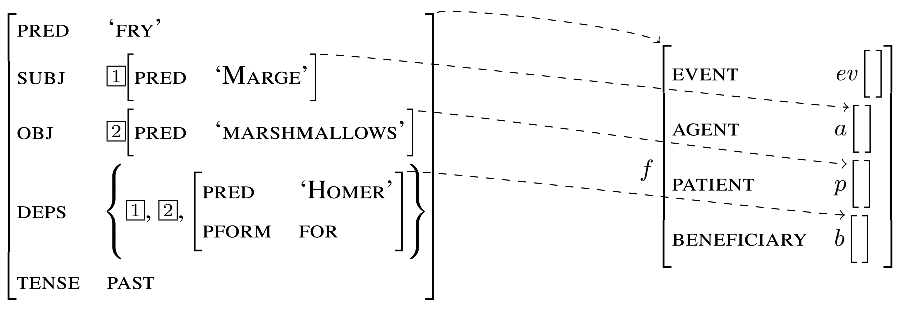


Figure 2: Functional and semantic structures for *Marge fried marshmallows for Homer*

[9]In Patejuk and Przepiórkowski 2016 and Przepiórkowski 2016, DEPS is list-valued and encodes the functional hierarchy, but here we assume for simplicity that it is set-valued.

for the sentence *Marge fried marshmallows for Homer*. In this setup, parts of the lexical entry for *fried* may look as in (18):

(18) lexical entry for *fried*, 1st version:

fried I (↑ PRED) = 'FRY'
$\lambda e.fry(e) : (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$
$(\uparrow_\sigma \text{ AGENT}) = (\uparrow \text{ SUBJ})_\sigma \quad (\uparrow \text{ SUBJ}) \in (\uparrow \text{ DEPS})$
$\lambda P \lambda x \lambda e.\ P(e) \wedge agent(e,x) :$
$[(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \text{ AGENT}) \multimap (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$
$(\uparrow_\sigma \text{ PATIENT}) = (\uparrow \text{ OBJ})_\sigma \quad (\uparrow \text{ OBJ}) \in (\uparrow \text{ DEPS})$
$\lambda P \lambda x \lambda e.\ P(e) \wedge patient(e,x) :$
$[(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \text{ PATIENT}) \multimap (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$
$((\uparrow_\sigma \text{ BENEFICIARY}) = (\uparrow \text{ DEPS } \in)_\sigma$
$\lambda P \lambda x \lambda e.\ P(e) \wedge beneficiary(e,x) :$
$[(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \text{ BENEFICIARY}) \multimap (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma)$
...
$\lambda P.\exists e\, P(e) \wedge past(e) \ : \ [(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma$
(↑ TENSE) = PAST

How do we know that various pointers to elements of DEPS, e.g., (↑ SUBJ) ∈ (↑ DEPS) and $(\uparrow_\sigma$ BENEFICIARY$) = (\uparrow$ DEPS $\in)_\sigma$ pick out different elements of this set? This follows directly from the resource sensitivity of Glue: if, say, the above two references to elements of DEPS picked out the same element, then also $(\uparrow_\sigma$ AGENT$)$ and $(\uparrow_\sigma$ BENEFICIARY$)$ would be the same semantic object, so the two corresponding meaning constructors above, referring to $(\uparrow_\sigma$ AGENT$)$ and $(\uparrow_\sigma$ BENEFICIARY$)$ in the linear formulae, would ultimately require the consumption of two linear resources corresponding to this semantic object. But only one such resource is introduced by the phrase which gave rise to the DEPS element which was selected twice. Hence, if two references to elements of DEPS accidentally pick out the same element, the linear logic proof will fail.

Obviously, just as above, such extended valency specifications may again be made concise via the use of template calls such as @AGENT, @PATIENT and @BENEFICIARY:

(19) lexical entry for *fried*, 2nd version:

fried V (↑ PRED) = 'FRY'
$\lambda e.fry(e) : (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$
@AGENT @PATIENT (@BENEFICIARY) ...
@PAST

The definitions of the templates AGENT and PATIENT in (13)–(14) must be trivially modified, by adding the functional schemata (↑ SUBJ) ∈ (↑ DEPS) and (↑ OBJ) ∈ (↑ DEPS), respectively.[10] The basic definition of BENEFICIARY is similar:

[10]Again, this should be a part of a more general analysis of diathesis.

(20) $\textsc{beneficiary} := (\uparrow_\sigma \textsc{beneficiary}) = (\uparrow \textsc{deps} \in)_\sigma$
$\lambda P \lambda x \lambda e.\ P(e) \wedge \mathit{beneficiary}(e,x) :$
$[(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \textsc{beneficiary}) \multimap$
$(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

In order to ensure that only prepositional phrases headed by *for* play the role of a Beneficiary, an additional constraint should make sure that the value of PFORM is FOR; this can be done with the use of a local name (Dalrymple 2001: 146–148):[11]

(21) $\textsc{beneficiary} := \%\textsc{b} = (\uparrow \textsc{deps} \in)$
$(\%\textsc{b}\ \textsc{pform}) =_c \textsc{for}$
$(\uparrow_\sigma \textsc{beneficiary}) = \%\textsc{b}_\sigma$
$\lambda P \lambda x \lambda e.\ P(e) \wedge \mathit{beneficiary}(e,x) :$
$[(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma \textsc{beneficiary}) \multimap$
$(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

On this analysis, *for* is assumed to be an asemantic (PRED-less) preposition, with the lexical entry in (22), and combining with the following NP as a co-head via a rule such as (23).

(22) *for* P $(\uparrow \textsc{pform}) = \textsc{for}$

(23) PP $\rightarrow$ P NP
$\downarrow=\uparrow$ $\quad \downarrow = (\uparrow \textsc{obj}) \mid \downarrow=\uparrow$

The above analysis assumes that, in the case of the verb FRY, the Beneficiary is optional, but it does not say anything about its status as an argument or adjunct – on the approach assumed here (i.e. that of Przepiórkowski 2016), which rejects the argument/adjunct distinction, this issue simply does not arise. For this reason the single BENEFICIARY template in (21) may be used whenever a Beneficiary combines with a predicate, whatever the perceived argument/adjunct status of the Beneficiary. As discussed in Przepiórkowski 2017a, two different templates (or at least a more complex definition of this single template) would be needed on the standard assumption of the argument/adjunct distinction, one for an element of ADJ, and another for $\textsc{obl}_{\textsc{ben}}$ (for the benefit of those predicates which require a *for*-beneficiary).

Each of the above templates for Agent, Patient and Beneficiary contributes the same predicate to the meaning representation (agent, $\mathit{patient}$ and $\mathit{beneficiary}$, respectively), regardless of the lexical nature of the dependent (*Marge*, *marshmallows* and *Homer*, in the example above). We assume here that in the case of many other semantic roles the predicate will depend on the head of the phrase realising the role. For example, in the case of the manner adverb *slowly*, the corresponding con-

[11] The external reviewer proposes to simplify the treatment of some adjuncts, including Beneficiary, by removing them from the template system and instead specifying relevant constraints in appropriate lexical entries, e.g., in the lexical entry of *for*, with the help of inside-out functional uncertainty. This simplified treatment would lead to the same representations as those proposed in this paper. Since – as noted in the review – only some adjuncts could be treated outside of the template system this way, we retain here the more uniform system where all dependents are added via hierarchically organised templates, and leave more detailed investigation of this suggestion for future research.

junct in the meaning representation will be something like $slowly(e)$ (rather than $manner(e, slowly)$), and in the case of the locative *in Springfield* the corresponding conjunct will be $in(e, springfield)$ (rather than $place(e, in, springfield)$). So, for the example sentence *Marge slowly fried marshmallows for Homer in Springfield*, we expect the final meaning representation to be as in (24) (simplifying the analysis of *marshmallows*), with f- and s-structures as in Figure 3.

(24) $\exists e\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) \wedge$
$beneficiary(e, homer) \wedge slowly(e) \wedge in(e, springfield) \wedge past(e)$

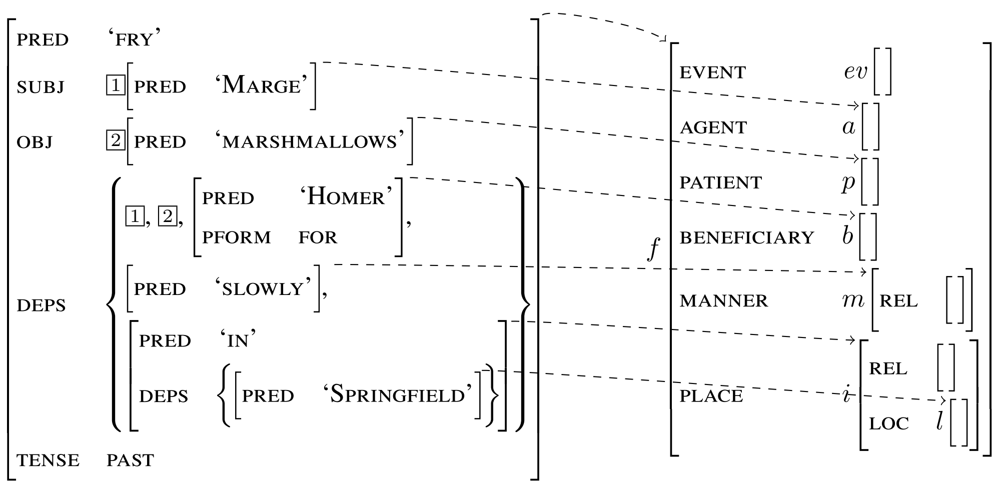


Figure 3: Functional and semantic structures for *Marge slowly fried marshmallows for Homer in Springfield*

The following MANNER and PLACE templates take care of the appropriate inclusion of dependents such as *slowly* and *in Springfield* into the analysis:[12]

(25) MANNER $:=$ $(\uparrow_\sigma \text{ MANNER}) = (\uparrow \text{ DEPS} \in)_\sigma$
$\lambda P \lambda Q \lambda e.\ P(e) \wedge Q(e)$:
$[(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap$
$[(\uparrow_\sigma \text{ MANNER REL}) \multimap (\uparrow_\sigma \text{ MANNER})] \multimap$
$(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$

(26) PLACE $:=$ $(\uparrow_\sigma \text{ PLACE}) = (\uparrow \text{ DEPS} \in)_\sigma$
$\lambda P \lambda Q \lambda e.\ P(e) \wedge Q(e)$:
$[(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma] \multimap$
$[(\uparrow_\sigma \text{ PLACE REL}) \multimap (\uparrow_\sigma \text{ PLACE})] \multimap$
$(\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$

The property Q in the above templates is provided by specific lexemes, such as the following:

[12]A more general parameterised template could be defined instead which would take semantic attribute names such as MANNER and PLACE as arguments.

(27) *slowly* Adv ($\uparrow$ PRED) = 'SLOWLY'
$\lambda e.\ slowly(e)\ :\ (\uparrow_\sigma \text{ REL}) \multimap \uparrow_\sigma$

(28) *in* P ($\uparrow$ PRED) = 'IN'
($\uparrow_\sigma$ LOC) = ($\uparrow$ DEPS $\in)_\sigma$
$\lambda x \lambda e.\ in(e, x)\ :\ (\uparrow_\sigma \text{ LOC}) \multimap (\uparrow_\sigma \text{ REL}) \multimap \uparrow_\sigma$

In the case of the manner adverb, the property in question is $\lambda e.\ slowly(e)$. In the case of the locative preposition *in*, the complete property will be specified once the preposition combines with the following NP, e.g., *Springfield*, yielding $\lambda e.\ in(e, springfield)$.

In both (and other similar) cases additional constraints may be added to lexical entries to the effect that, say, *slowly* always expresses Manner and the locative *in* always realises the role Place. The following respective constraints would do the trick: (MANNER $\uparrow_\sigma$) and (PLACE $\uparrow_\sigma$). Such constraints would encode the intuition that typical adjuncts determine their semantic role in the sentence (while the semantic role of typical arguments is determined by the predicate). However, at least in the case of locative and temporal PPs this seems to be too strong, as such phrases do not necessarily express the place or time of the event denoted by the verb, as the following examples (from Jaworska 1986: 355–356) illustrate:

(29) The campaigners planned *until Christmas* in detail.

(30) The new tenants are reclaiming *behind the garage*.

Clearly, the planning activity of the time from now until Christmas does not have to last the whole time until Christmas, and similarly reclaiming some location can be performed at a different location. For this reason we do not include such constraints in the above lexical entries, until this matter is further investigated.

Let us now illustrate this analysis with a simple derivation of the desired meaning representation (24), assuming the analysis of *Marge slowly fried marshmallows for Homer in Springfield* which results in the functional and semantic representations in Figure 3. The following instantiated meaning constructors are initially in the set of premises:

(31)
1. $marge : a$
2. $marshmallows : p$
3. $homer : b$
4. $springfield : l$
5. $\lambda e.\ fry(e) : ev \multimap f$
6. $\lambda P.\ \exists e\ P(e) \wedge past(e) : [ev \multimap f] \multimap f$
7. $\lambda P \lambda x \lambda e.\ P(e) \wedge agent(e, x) : [ev \multimap f] \multimap a \multimap ev \multimap f$
8. $\lambda P \lambda x \lambda e.\ P(e) \wedge patient(e, x) : [ev \multimap f] \multimap p \multimap ev \multimap f$
9. $\lambda P \lambda x \lambda e.\ P(e) \wedge beneficiary(e, x) : [ev \multimap f] \multimap b \multimap ev \multimap f$
10. $\lambda P \lambda Q \lambda e.\ P(e) \wedge Q(e) : [ev \multimap f] \multimap [(m \text{ REL}) \multimap m] \multimap ev \multimap f$
11. $\lambda P \lambda Q \lambda e.\ P(e) \wedge Q(e) : [ev \multimap f] \multimap [(i \text{ REL}) \multimap i] \multimap ev \multimap f$

12. $\lambda e.\ slowly(e)\ :\ (m\ \textsc{rel}) \multimap m$

13. $\lambda x \lambda e.\ in(e,x)\ :\ l \multimap (i\ \textsc{rel}) \multimap i$

Meaning constructors 1–4 come from the lexical entries of nouns (with the simplified treatment of *marshmallows* as a proper name), 5 comes directly from the lexical entry of *fried*, 6 – from the PAST template called there, 7–11 – from the AGENT, PATIENT, BENEFICIARY, MANNER and PLACE templates called there, 12 – from the lexical entry of *slowly* and 13 – from the lexical entry of *in*.

It is easy to see that these constructors give rise to the expected meaning representation for this sentence:

(32) 14. $\lambda e.\ in(e, springfield)\ :\ (i\ \textsc{rel}) \multimap i$ (from 13 and 4)

15. $\lambda x \lambda e.\ fry(e) \wedge agent(e,x) : a \multimap ev \multimap f$ (from 7 and 5)

16. $\lambda e.\ fry(e) \wedge agent(e, marge) : ev \multimap f$ (from 15 and 1)

17. $\lambda x \lambda e.\ fry(e) \wedge agent(e, marge) \wedge patient(e,x) : p \multimap ev \multimap f$ (from 8 and 16)

18. $\lambda e.\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) : ev \multimap f$ (from 17 and 2)

19. $\lambda x \lambda e.\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) \wedge beneficiary(e,x) : b \multimap ev \multimap f$ (from 9 and 18)

20. $\lambda e.\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) \wedge beneficiary(e, homer) : ev \multimap f$ (from 19 and 3)

21. $\lambda Q \lambda e.\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) \wedge beneficiary(e, homer) \wedge Q(e) : [(m\ \textsc{rel}) \multimap m] \multimap ev \multimap f$ (from 10 and 20)

22. $\lambda e.\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) \wedge beneficiary(e, homer) \wedge slowly(e) : ev \multimap f$ (from 21 and 12)

23. $\lambda Q \lambda e.\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) \wedge beneficiary(e, homer) \wedge slowly(e) \wedge Q(e) : [(i\ \textsc{rel}) \multimap i] \multimap ev \multimap f$ (from 11 and 22)

24. $\lambda e.\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) \wedge beneficiary(e, homer) \wedge slowly(e) \wedge in(e, springfield) : ev \multimap f$ (from 23 and 14)

25. $\exists e\ fry(e) \wedge agent(e, marge) \wedge patient(e, marshmallows) \wedge beneficiary(e, homer) \wedge slowly(e) \wedge in(e, springfield) \wedge past(e) : f$ (from 6 and 24)

4.2 Frame inheritance via template inheritance

In FrameNet, the verb FRY evokes the Apply_heat frame. FrameNet lists 15 semantic roles of this frame, of which only 5 are mentioned above (Agent – called Cook in the Apply_heat frame, Patient – called Food there, Beneficiary, Manner and Place). Many of these roles are also present in many other frames. However, an inheritance hierarchy makes it possible to avoid redundancy and it is trivial to formalise in LFG with the use of templates. Following FrameNet, we assume that the maximally general frame Event introduces the following 7 roles[13] and marks them as optional: Place, Time, Duration, Explanation, Frequency, Manner and Timespan. Correspondingly, we define the @EVENT_FRAME template which optionally calls templates such as @TIME, etc., which in turn introduce particular dependents in a way illustrated in the previous subsection; cf. (33).

(33) EVENT_FRAME := (@PLACE) (@TIME) (@DURATION) (@EXPLANATION)
(@FREQUENCY) (@MANNER) (@TIMESPAN)

An immediately subordinate frame, Intentionally_act, introduces a few additional roles, including the obligatory Agent and the optional Domain, and inherits most of the roles introduced by Event, apart from Timespan.[14] This may be represented via the template in (34).

(34) INTENTIONALLY_ACT_FRAME := @EVENT_FRAME
$\neg(\uparrow_\sigma$ TIMESPAN$)$ @AGENT (@DOMAIN) ...

Note that the way Timespan is not inherited, via the negative constraint $\neg(\uparrow_\sigma$ TIMESPAN$)$,[15] preserves the monotonicity of inheritance, as this semantic role is defined on the higher frame as optional. Similarly, the obligatoriness of a role which is specified as optional on a superordinate frame may be ensured by an existential constraint such as $(\uparrow_\sigma$ MANNER$)$, as in the case of frames evoked by verbs such as BEHAVE or TREAT, for which the expression of Manner is obligatory.

Another frame inheriting from Event and adding a few more semantic roles, including the obligatory Agent,[16] is Objective_influence, which is a superordinate frame of Transitive_action, which in turn introduces the obligatory Patient role; cf. (35)–(36).

(35) OBJECTIVE_INFLUENCE_FRAME := @EVENT_FRAME @AGENT
(@CIRCUMSTANCES) ...

(36) TRANSITIVE_ACTION_FRAME := @OBJECTIVE_INFLUENCE_FRAME
@PATIENT ...

Both Transitive_action and Intentionally_act are immediately superordinate frames of Intentionally_affect, as formalised in (37).

[13]We ignore here another type of FrameNet role, ‘Core Unexpressed’, not expressed by dependents.

[14]Also Duration is not mentioned in the description of Intentionally_act in the November 2016 release of FrameNet. It is not clear to me whether these omissions are intentional, but we will use the lack of Timespan here to illustrate how such an apparently non-monotonic aspect may be modelled in LFG.

[15]As noted by an anonymous reviewer of the accompanying paper Przepiórkowski 2017a, it would be more elegant not to ‘unpack’ the TIMESPAN template this way, but rather have a constraint such as ¬@TIMESPAN. However, this would have the effect of negating a sequence – normally interpreted as conjunction – of two statements, one of which is a constructor. As I am not sure what this would mean (but see Przepiórkowski 2017b for a suggestion), I propose this clearer – even if less elegant – encoding.

[16]In FrameNet, this role is called Influencing_entity here, but it is renamed to Agent in the subordinate frame Transitive_action.

(37) $\textsc{intentionally\_affect\_frame} := @\textsc{intentionally\_act\_frame}$
$@\textsc{transitive\_action\_frame} \ldots$

The technically unfortunate effect of this is that the template call @AGENT is inherited twice. This is a potential problem as this template includes a meaning constructor, so two copies of this constructor will be present whenever the @INTENTIONALLY_AFFECT_FRAME template is called. There is a straightforward solution to this problem, though, consisting in making the content of the previous definition of the AGENT template optional and adding an obligatory constraint to the effect that the semantic attribute AGENT be defined:

(38) $$\begin{array}{ll} \textsc{agent} := & \big(\,(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{agent}) \quad (\uparrow \textsc{subj}) \in (\uparrow \textsc{deps}) \\ & \quad \lambda P \lambda x \lambda e.\ P(e) \wedge agent(e,x) : \\ & \qquad [(\uparrow_\sigma\ \textsc{event}) \multimap \uparrow_\sigma] \multimap (\uparrow_\sigma\ \textsc{agent}) \multimap (\uparrow_\sigma\ \textsc{event}) \multimap \uparrow_\sigma \big) \\ & (\uparrow_\sigma\ \textsc{agent}) \end{array}$$

Note that multiple occurrences of the $(\uparrow_\sigma\ \textsc{agent})$ constraint have exactly the same effect as a single occurrence, so multiple inheritance of this part of the template is not harmful. The only place in the grammar that assigns a value to the AGENT feature is the optional part of the new AGENT template in (38), so – in the case of multiple calls to this template – at least one copy of this optional part must be used. On the other hand, at most one may be used, as more would introduce multiple copies of the meaning constructor within the optional part. Each such constructor causes a consumption of the glue resource introduced by the subject, corresponding to the value of the AGENT attribute, and only one such resource is introduced by the subject. Hence, exactly one of the multiple copies of the optional part of the AGENT template will actually be used.

An important part of the above reasoning is that only one element of DEPS may be the value of SUBJ and, hence, only one value of DEPS may correspond to the AGENT (in the active voice). There is no corresponding restriction on, say, BENEFICIARY, MANNER or PLACE. This means that, in case templates corresponding to such roles are inherited multiple times, they may pick out multiple elements of DEPS, that is, there may be multiple realisations of such roles. This property, called iterability, is one of the tests supposedly distinguishing arguments from adjuncts: only adjuncts, and not arguments, may iterate (see, e.g., Panevová 1974, Bresnan 1982b, Williams 2015, and the discussion in Przepiórkowski 2016). However, according to the current analysis, apart from the subject and the object, any kind of dependent may in principle (subject to multiple inheritance of the corresponding template) be iterated, whether it is realised as a supposed argument, or as a supposed adjunct. This, in fact, seems to better describe the data, given the possible iterability of OBLS (Zaenen and Crouch 2009) or instrumental phrases (Goldberg 2002), sometimes analysed in LFG as arguments (Bresnan 1982b: 164–165).[17]

[17] Obviously, in order to more fully model the iterability of such dependents, the corresponding templates should be explicitly defined as iterable (e.g., replacing optionality with a Kleene star), rather than relying on accidental (and limited) multiple inheritance.

Returning to the running example, the Apply_heat frame also inherits from two superordinate frames, Intentionally_affect and Activity (the latter not discussed here), and also introduces a few specific roles such as Beneficiary,[18] Container and Medium; cf. (39).

(39) APPLY_HEAT_FRAME := @INTENTIONALLY_AFFECT_FRAME
@ACTIVITY_FRAME (@BENEFICIARY)
(@CONTAINER) (@MEDIUM) ...

With such a hierarchy of templates, the lexical entry for *fried* boils down to (40).

(40) lexical entry for *fried*, 3rd version:

fried V ($\uparrow$ PRED) = 'FRY'
$\lambda e.fry(e) : (\uparrow_\sigma \text{ EVENT}) \multimap \uparrow_\sigma$
@APPLY_HEAT_FRAME
@PAST

In practice, different verbs belonging to the same frame may additionally introduce specific – possibly different – morphosyntactic constraints on the realisation of the same role, as is the case with the verbs BEGIN and ENTER evoking the Activity_start frame: only BEGIN may realise the Activity role as an infinitival phrase (*begin to negotiate*) and only ENTER may realise it as an *into*-PP (*enter into negotiations*).

5 Conclusion

There is hardly any LFG work on types of adjuncts; instead, different kinds of adjuncts are lumped into the (X)ADJ set. There are a couple of other frameworks which make a categorisation of all types of dependents, arguments and adjuncts alike. Perhaps the most prominent of these is FrameNet, but there is also Functional Generative Description (Sgall et al. 1986), supported (just like FrameNet) by extensive corpus annotation (Hajič et al. 2006), as well as the mainstream cartographic approach (Cinque 1999, 2010). In this paper we propose to carry over the main ideas of FrameNet to LFG: introduce all types of dependents lexically (which does not preclude constructional analyses like that of Asudeh et al. 2013), and organise such extended valency information hierarchically, so as to avoid redundancy and capture generalisations. As shown in Przepiórkowski 2017a, this approach is broadly compatible with standard LFG: templates such as @TIME or @PLACE may add adjuncts directly to the ADJ set, instead of DEPS as assumed here. However, this approach also lends support to the proposal to get rid of the argument/adjunct distinction altogether (Przepiórkowski 2016): both types of dependents are introduced via the same mechanism, with typical adjuncts introduced higher in the hierarchy and inherited by many different frames, and typical arguments introduced closer to the leaves of the hierarchy and, hence, inherited by fewer frames. This approach also models the fact that certain prototypical adjuncts, introduced high in the hierarchy,

[18]This role is introduced so low in the FrameNet hierarchy probably by mistake; it should rather be introduced at the level of Intentionally_act.

may behave more like arguments on some lower frames: this is the case with the Manner role mentioned above. But regardless of its natural place in the version of LFG that does not assume the argument/adjunct dichotomy, we believe that any version of LFG will benefit from the adoption of a more structured approach to the lexicon in general and to valency in particular.

References

Alsina, Alex. 1996. *The Role of Argument Structure in Grammar*. CSLI Lecture Notes, No. 62, Stanford, CA: CSLI Publications.

Arnold, Doug, Butt, Miriam, Crysmann, Berthold, King, Tracy Holloway and Müller, Stefan (eds.). 2016. *Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar*, Stanford, CA, CSLI Publications.

Asudeh, Ash, Dalrymple, Mary and Toivonen, Ida. 2008. Constructions with Lexical Integrity: Templates as the Lexicon–Syntax Interface. In Miriam Butt and Tracy Holloway King (eds.), *The Proceedings of the LFG'08 Conference*, pages 68–88, University of Sydney, Australia: CSLI Publications.

Asudeh, Ash, Dalrymple, Mary and Toivonen, Ida. 2013. Constructions with Lexical Integrity. *Journal of Language Modelling* 1(1), 1–54.

Asudeh, Ash and Giorgolo, Gianluca. 2012. Flexible Composition for Optional and Derived Arguments. In Miriam Butt and Tracy Holloway King (eds.), *The Proceedings of the LFG'12 Conference*, pages 64–84, Stanford, CA: CSLI Publications.

Asudeh, Ash, Giorgolo, Gianluca and Toivonen, Ida. 2014. Meaning and Valency. In Miriam Butt and Tracy Holloway King (eds.), *The Proceedings of the LFG'14 Conference*, pages 68–88, Stanford, CA: CSLI Publications.

Bresnan, Joan (ed.). 1982a. *The Mental Representation of Grammatical Relations*. Cambridge, MA: The MIT Press.

Bresnan, Joan. 1982b. Polyadicity. In Bresnan (1982a), pages 149–172.

Bresnan, Joan and Kanerva, Jonni M. 1989. Locative Inversion in Chicheŵa: A Case Study of Factorization in Grammar. *Linguistic Inquiry* 20(1), 1–50.

Butt, Miriam, Dalrymple, Mary and Frank, Anette. 1997. An Architecture for Linking Theory in LFG. In Miriam Butt and Tracy Holloway King (eds.), *The Proceedings of the LFG'97 Conference*, University of California, San Diego: CSLI Publications.

Cinque, Guglielmo. 1999. *Adverbs and Functional Heads: A Cross-Linguistic Perspective*. New York: Oxford University Press.

Cinque, Guglielmo. 2010. *The Syntax of Adjectives: A Comparative Study*. Cambridge, MA: The MIT Press.

Corbett, Greville G. and Fraser, Norman M. 1993. Network Morphology: a DATR Account of Russian Nominal Inflection. *Journal of Linguistics* 29, 113–142.

Daelemans, Walter, Smedt, Koenraad De and Gazdar, Gerald. 1992. Inhertance in Natural Language Processing. *Computational Linguistics* 18(2), 205–218.

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*. San Diego, CA: Academic Press.

Dalrymple, Mary, Hinrichs, Angie, Lamping, John and Saraswat, Vijay. 1993. The Resource Logic of Complex Predicate Interpretation. In *Proceedings of ROCLING 1993*, pages 3–21.

Dalrymple, Mary, Kaplan, Ronald M. and King, Tracy Holloway. 2004. Linguistic Generalizations over Descriptions. In Miriam Butt and Tracy Holloway King (eds.), *The Proceedings of the LFG'04 Conference*, pages 199–208, Stanford, CA: CSLI Publications.

Davis, Anthony R. 2001. *Linking by Types in the Hierarchical Lexicon*. Stanford, CA: CSLI Publications.

Evans, Roger and Gazdar, Gerald. 1996. DATR: A Language for Lexical Knowledge Representation. *Computational Linguistics* 22(2), 167–216.

Fellbaum, Christiane (ed.). 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: The MIT Press.

Fillmore, Charles J. and Baker, Collin. 2015. A Frames Approach to Semantic Analysis. In Bernd Heine and Heiko Narrog (eds.), *The Oxford Handbook of Linguistic Analysis*, pages 791–816, Oxford: Oxford University Press, second edition.

Fillmore, Charles J., Johnson, Christopher R. and Petruck, Miriam R.L. 2003. Background to FrameNet. *International Journal of Lexicography* 16(3), 235–250.

Findlay, Jamie Y. 2016. Mapping Theory without Argument Structure. *Journal of Language Modelling* 4(2), 245–289.

Flickinger, Daniel. 1987. *Lexical Rules in the Hierarchical Lexicon*. Ph.D. Thesis, Stanford University, Stanford, CA.

Goldberg, Adele E. 2002. Surface Generalizations: An Alternative to Alternations. *Cognitive Linguistics* 13(4), 327–356.

Goldberg, Adele E. and Ackerman, Farrell. 2001. The Pragmatics of Obligatory Adjuncts. *Language* 77(4), 798–814.

Grimshaw, Jane and Vikner, Sten. 1993. Obligatory Adjuncts and the Structure of Events. In Eric Reuland and Werner Abraham (eds.), *Knowledge and Language*, volume II, pages 143–155, Dordrecht: Kluwer.

Hajič, Jan, Panevová, Jarmila, Hajičová, Eva, Sgall, Petr, Pajas, Petr, Štěpánek, Jan, Havelka, Jiří, Mikulová, Marie, Žabokrtský, Zdeněk, Ševčíková Razímová, Magda and Urešová, Zdeňka. 2006. Prague Dependency Treebank 2.0 (PDT 2.0).

Jaworska, Ewa. 1986. Prepositional Phrases as Subjects and Objects. *Journal of*

Linguistics 22, 355–374.

Kaplan, Ronald M. and Bresnan, Joan. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Bresnan (1982a), pages 173–281.

Kuhn, Jonas. 2001. Resource Sensitivity in the Syntax-Semantics Interface: Evidence from the German Split NP Construction. In Detmar Meurers and Tibor Kiss (eds.), *Constraint-Based Approaches to Germanic Syntax*, pages 177–215, Stanford, CA: CSLI Publications.

Miller, George A., Beckwith, Richard, Fellbaum, Christiane, Gross, Derek and Miller, Katherine J. 1990. Introduction to WordNet: An Online Lexical Database. *International Journal of Lexicography* 3(4), 235–244.

Panevová, Jarmila. 1974. On Verbal Frames in Functional Generative Description. Part 1. *The Prague Bulletin of Mathematical Linguistics* 22, 3–40.

Patejuk, Agnieszka and Przepiórkowski, Adam. 2016. Reducing Grammatical Functions in Lexical Functional Grammar. In Arnold et al. (2016), pages 541–559.

Przepiórkowski, Adam. 2016. How *not* to Distinguish Arguments from Adjuncts in LFG. In Arnold et al. (2016), pages 560–580.

Przepiórkowski, Adam. 2017a. A Full-Fledged Hierarchical Lexicon in LFG: The FrameNet Approach. In Victoria Rosén and Koenraad De Smedt (eds.), *The Very Model of a Modern Linguist*, volume 8 of *Bergen Language and Linguistics Studies*, pages 202–219, Bergen: University of Bergen Library.

Przepiórkowski, Adam. 2017b. Some Doubts about Meaning Constructors and Semantic Structures in LFG + Glue, unpublished manuscript.

Ruppenhofer, Josef, Ellsworth, Michael, Petruck, Miriam R. L., Johnson, Christopher R., Baker, Collin F. and Scheffczyk, Jan. 2016. *FrameNet II: Extended Theory and Practice*. Revised November 1, 2016.

Sgall, Petr, Hajičová, Eva and Panevová, Jarmila. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Dordrecht: Reidel.

van der Linden, Erik-Jan. 1992. Incremental Processing and the Hierarchical Lexicon. *Computational Linguistics* 18(2), 219–238.

Vijay-Shanker, K. and Schabes, Yves. 1992. Structure Sharing in Lexicalized Tree-Adjoining Grammars. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992)*, pages 205–211, Nantes.

Williams, Alexander. 2015. *Arguments in Syntax and Semantics*. Cambridge: Cambridge University Press.

Zaenen, Annie and Crouch, Dick. 2009. OBLS Hobble Computations. In Miriam Butt and Tracy Holloway King (eds.), *The Proceedings of the LFG'09 Conference*, pages 644–654, Trinity College, Cambridge, UK: CSLI Publications.