MISSING RESOURCES IN A RESOURCE-SENSITIVE SEMANTICS

Gianluca Giorgolo and Ash Asudeh
King's College, London & University of Oxford
Carleton University & University of Oxford

Proceedings of the LFG12 Conference

Miriam Butt and Tracy Holloway King (Editors)

2012

CSLI Publications

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

Abstract

In this paper, we present an investigation of the argument/adjunct distinction in the context of LFG. We focus on those cases where certain grammatical functions that qualify as arguments according to all standard tests (Needham and Toivonen, 2011) are only optionally realized. We argue for an analysis first proposed by Blom et al. (2012), and we show how we can make it work within the machinery of LFG. Our second contribution regards how we propose to interpret a specific case of optional arguments, optional objects. In this case we propose to generalize the distinction between transitive and intransitive verbs to a continuum. Purely transitive and intransitive verbs represent the extremes of the continuum. Other verbs, while leaning towards one or the other end of this spectrum, show an alternating behavior between the two extremes. We show how our first contribution is capable of accounting for these cases in terms of exceptional behavior. The key insight we present is that the verbs that exhibit the alternating behavior can best be understood as being capable of dealing with an exceptional context. In other words they display some sort of control on the way they compose with their context. This will prompt us also to rethink the place of the notion of subcategorization in the LFG architecture

1 Introduction

The distinction between *arguments* and *adjuncts* is central for the LFG architecture as it influences the way in which representations of linguistic expressions are generated both at the functional and the semantic level. At the functional level the distinction between arguments and adjuncts is crucial for the definition of the notion of *completeness* and *coherence* of an f-structure, which is in turn one of the parameters that determines the grammaticality of an expression. Similarly, at the level of semantics, the distinction between arguments and adjuncts has important consequences on the semantic representations we choose and on the way we control the composition of these meanings. Arguments are in fact usually represented as resources that are consumed by predicates, while adjuncts tend to be represented as functions that consume predicates to generate modified versions of them. The choice of whether a particular grammatical function is an argument or an adjunct requires particular attention from a semantic perspective, as it determines important properties such as the scopal relations between quantified expressions and the way in which these relations are captured by our semantic theory. While the importance of this distinction for the f-structural level has been recognized by many in the LFG literature (Bresnan, 1978; Dalrymple, 2001), we think that its effects

[†]This research was supported by an Early Researcher Award from the Ontario Ministry of Research and Innovation, NSERC Discovery Grant #371969 and by a grant from the John Fell Oxford University Press Research Fund. The authors thank Avery Andrews, Doug Arnold, Boban Arsenijević, Miriam Butt, Mary Dalrymple, Ron Kaplan, Dag Haug, Stefano Quaglia, Louisa Sadler, Sebastian Sulger and the audience at LFG12 for their comments and questions. Any remaining errors are our own.

at the compositional semantics level are understudied, and an overview of these effects can help clarify this important grammatical issue.

Traditionally the distinction between arguments and adjuncts has been made on the basis of a mixture of ontological and syntactic tests. Needham and Toivonen (2011) provide an overview of these tests in an LFG perspective. At the same time, they point out that there are cases where the tests fall short of providing a clear distinction between the two classes. In this paper we will analyze some of these examples under the perspective of compositional semantics. We are particularly interested in cases where grammatical functions that are usually considered arguments become *optional*, a property typical of adjuncts. We motivate the discussion on the basis of the equivalences in (1-6).

(1) Alice ate yesterday afternoon. ⇔ Alice ate something yesterday afternoon.

(2) Bob drank last night. ⇔ Bob drank something last night / Bob drank something alcoholic last night.

(3) (?) Bob loves drinking. ⇔ Bob loves drinking alcohol.

(4) Yesterday, Alice debugged for three hours. ⇔ Yesterday, Alice debugged some code/some programs for three hours. *(in a context in which it is known that Alice finds debugging annoying)*

(5) Silvio was accused of tax fraud. ⇔ Silvio was accused of tax fraud by someone. / Someone accused Silvio of tax fraud.

(6) Silvio was accused. ⇔ Silvio was accused of something by someone.

The lefthand side of the equivalences we have in (1), (2), (3) and (4) show that we can omit the object with certain transitive verbs. However the righthand side of the same equivalences show that the argument is not deleted from the semantic representation of the verb, but rather it is filled by some default value. In most cases the argument slot is bound by an existentially bound variable, but this is not always true. For example, in (8) the omitted object is interpreted as a universally bound variable, while in (9) the intuitive reading for the omitted destination of the arriving event is a deictic or indexical one (the origin seems to be interpreted as an existentially bound variable).[1]

(8) W.H.O. warns against homeopathy use. ⇔ W.H.O. warns everyone against homeopathy use.

[1]However notice that Stanley (2000) proposes an analysis of cases like (9) in which the unexpressed arguments are considered bound by a linguistic operator. Stanley bases his analysis on examples like (7) where the raining event location co-varies with the locations quantified over by "everywhere".

(7) Everywhere Bob goes, it rains.

(9) Bob arrived yesterday. ⇔ Bob arrived from somewhere yesterday to the contextually relevant location.

Another important aspect of this phenomenon is that it seems to be lexically specified. Not all transitive verbs can in fact be constructed with an implicit object, and what is even more interesting is that related verbs may present opposite behaviors. For instance the verb *eat* can be constructed without an explicit object but the intensified form *devour* can not.

The equivalences in (5) and (6) show a similar situation for agents in passive constructions. The by-phrase is always optional, but the described events are always understood as requiring an agent. Notice that in this case there is no lexically specified preference for this construction. All transitive verbs allow for an implicit agentive by-phrase.

These examples challenge a resource sensitive semantics, such as Glue Semantics, in two ways. First of all we have to clarify whether these optionally realized semantic roles should be considered arguments or adjuncts. This decision will determine how we represent them in terms of semantic resources. Given that they seem to contribute to the semantic content of an utterance even when they are not present, we are inclined to consider them core arguments of their predicates. This choice motivates the second challenge to a resource sensitive semantics. We need in fact to clarify how these default resources are introduced in the semantic derivation despite the fact that they are not apparently introduced by any linguistic item.

In what follows we present our solution. In a nutshell we propose to consider verbs that support implicit objects and constructions like passive as being capable of actively operating on their context during the semantic derivation. We will reject the hypothesis that these verbs and constructions are in some way ambiguous. Instead we will associate a single core meaning to them, but give them the power to operate on their context in *exceptional* cases. For the case of verbs that allow implicit object, this notion will prompt us to reconsider the standard distinction between transitive and intransitive verbs. We will suggest that the distinction between transitive and intransitive verbs is not binary. Instead we propose a continuum of verbal behaviors with certain verbs leaning more clearly towards the transitive end of the continuum, other more towards the intransitive end, and still others presenting less marked uses. This shift in perspective has the effect of changing the way in which we look at the issue of how to distinguish arguments from adjuncts. By showing that the richer categorization we propose helps to clarify these notions in the context of the transitive / intransitive divide, we show that looking at compositional semantics may be crucial to better understanding the notion of *argument*.

The paper is structured as follows. Section 2 discusses prior analyses of the kind of data we are interested in. We will survey a number of proposals and identify a recent one by Blom et al. (2012) as the most promising. In section 3 we show how this proposal can be adapted to work in the context of LFG, and how it relates to a previous extension of Glue Semantics that we have proposed in Giorgolo and Asudeh (2011). In section 4 we provide a detailed analysis of some examples. In section 5 we discuss the consequences of our proposal for the distinction between transitive and intransitive verbs and, more in general, for the notion of subcategorization. We conclude in section 6 with some final remarks.

2 Prior work

We start our review of the literature on the topic with Bresnan's (1978) analysis of transitive verbs constructed without an explicit object. Bresnan proposes a solution based on *lexical ambiguity*. The verbs that can be constructed without an explicit object therefore have two entries in the lexicon: the first one correspond to the standard transitive construction whose meaning is represented by a binary function, while the second specifies an intransitive syntactic structure coupled with a unary predicate, constructed from the original binary relation by *existentially* binding the object argument. For example, Bresnan (1978) gives the following lexical entries for the verb *eat*:

eat:	V,	[___ NP],	NP_1 $\mathbf{eat}$ NP_2
		[___],	$(\exists\ y)$ NP_1 $\mathbf{eat}$ y

The second entry for *eat* is obtained from the first one by removing the syntactic requirement of an object, and by existentially binding the object position of the predicate $\mathbf{eat}$. We will see that the idea that treating verbs like *eat* as ambiguous is a common analysis in the literature.

Dowty (1982) proposes a similar analysis in the context of Montague Grammar by introducing so-called *Relation-Reducing Rules*. One such rule transforms a transitive verb like *eat* into an intransitive one and at the same time changes the semantics of the verb by binding its second argument to an existential quantifier. This approach is completely equivalent to the one of Bresnan (1978), with the additional complication that we have to introduce a device in the lexicon that controls the applicability of the Relation-Reducing Rules. Without such a device the grammar would over-generate as it would allow us to derive ungrammatical sentences such as **Yesterday, John devoured*.

The proposal in Bresnan (1978) does not specify how the implicit quantifier should behave with respect to other quantificational elements in the sentence. Fodor and Fodor (1980) tackled this question by first noticing that, in the case of a quantifier in subject position, the existential binding the second argument of $\mathbf{eat}$ must take narrow scope:

(10) Every boy ate.

The intuitive reading for (10) is the one were there are (possibly) different entities that are eaten by the boys.[2]

To capture this generalization, and to explain it in terms of general logical properties, Fodor and Fodor (1980) resort to an approach based on *meaning postulates*. Their first assumption is that the ambiguity that in Bresnan's proposal

[2]The strongest reading were a single entity is shared by the boys is of course available, as it is entailed by the weaker reading.

$$\begin{tikzcd}
\forall x(P_1(x)) \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \forall x \exists y(P_2(x,y)) \arrow[d, leftrightarrow] \\
P_1(c_1) \wedge \ldots \wedge P_1(c_n) \wedge \top \arrow[r, leftrightarrow] & \exists y(P_2(c_1,y)) \wedge \ldots \wedge \exists y(P_2(c_n,y)) \wedge \top
\end{tikzcd}$$

Figure 1: Equivalences between first-order formulae justifying the narrow scope of implicitly introduced existential quantifiers.

was restricted to the syntactic component is extended to the semantics. They assume that at the semantic representation level there are two predicates for transitive verbs that allow for implicit objects: a standard binary version, and a unary one. For instance, for a verb like *eat* we have a binary predicate $\mathbf{eat}_2$ that represents its meaning when used with an overt object, and a unary $\mathbf{eat}_1$ that corresponds to the meaning of a use like the one in (10). The two versions of the predicates are then related through meaning postulates like those in (11) and (12).

(11) $\mathbf{eat_1}(c) \leftrightarrow \exists y(\mathbf{eat_2}(c,y))$ with c a constant term

(12) $Qx(\mathbf{eat_1}(x)) \leftrightarrow Qx\exists y(\mathbf{eat_2}(x,y))$ with Q a quantifier

In words, the unary version of the predicate is required to be equivalent to the binary one, where the second argument is bound to a an existential quantifier. The relative order between the implicit existential quantifier and other quantificational operators in postulates like (12) is fixed, with the implicit quantifier having narrow scope. This order is not arbitrary but is explained by Fodor and Fodor on the basis of postulates of the kind in (11) and general logical equivalences. For example, in the case of a universal quantifier the relative order of the two quantifiers is determined by the equivalences illustrated in the diagram in Figure 1. The assumption here is that there is at least one constant naming each element in the domain of quantification. The equivalence on the top row is justified by the equivalence in the bottom row, which results from the repeated application of a postulate of the form of (11), and the two "vertical" equivalences which are general logical equivalences.

This solution is therefore a mixture of lexical ambiguity, also extended to the semantic representation language, and general logical axioms. This approach requires in any case a lexical specification that controls when the unary predicates are available (for example restricting them to passive constructions). Notice that Fodor and Fodor (1980)'s observation is not in contrast with the solution of Bresnan (1978): in Asudeh and Giorgolo (2012) we show how we can control the relative scope of explicitly and implicitly quantified arguments in an LFG setting without resorting to meaning postulates.

However this solution presents some drawbacks, mainly connected to the fact that we may have to list a large number of postulates in cases of verbs constructed with more than two arguments. Consider (13), for example.

(13) Most politicians were accused of at least two crimes.

In this case we would have to control the meaning of the predicate $\mathbf{accuse}$ by listing postulates covering all the possible allowed combinations of implicit and explicit quantifiers and constant terms (in the case of a ternary predicate like $\mathbf{accuse}$ this would amount to 12 (non-equivalent) postulates).

The analyses we have considered so far are based on the idea that the alternation between constructions where all the arguments of predicates are expressed and the cases where some specific arguments are left implicit can be best captured in terms of lexical ambiguity. Carlson (1984) and Lasersohn (1993) depart from this assumption and instead propose two similar analyses that explain the alternation on ontological grounds. Here we focus on the analysis of Lasersohn (1993) as it is motivated on an interesting problem that solutions based on lexical ambiguity encounter when dealing with distributive readings of certain predicates. Lasersohn considers sentences like the one in (14) which is usually interpreted to mean something along the lines of (15). The core intuition here is that the unexpressed agent is not necessarily the same for all atomic grading events. Therefore the existential quantifier binding the agent variable in the predicate needs to have narrow scope with respect to the universal quantifier that enumerates the atomic grading events.

(14) The papers were graded.

(15) $\forall y \exists x (y \in \mathbf{paper}^* \rightarrow \mathbf{grade}(x, y))$

While this seems in line with the observation of Fodor and Fodor (1980), this is not the case. The problem is in the way in which the universal quantifier is introduced in the semantic representation. The lexical solution of Fodor and Fodor (1980) can in fact control the scope of the quantifiers associated with implicit arguments only with respect to quantifiers that are introduced by other lexical resources. However, the standard assumption is that the universal quantifier in (15) is part of the lexical entry of the verb *grade*, specifically of its distributive reading that we can represent as (18).[3]

(18) $\lambda y \lambda x \forall z (z \in y \rightarrow \mathbf{grade}(x, z))$

It is easy to see that an approach based on existentially binding the variable corre-

[3]Lasersohn (1993) uses a meaning postulate similar to the one in (16) to construct the distributive reading of a verb.

(16) $\alpha(X) \leftrightarrow \forall y (y \in X \rightarrow \alpha(y))$

This approach makes his argument less compelling as it would be easy to modify this postulate in the case of a binary (or in general n-ary predicate) predicate to obtain the desired result:

(17) $Qx(R(x, Y)) \leftrightarrow \forall y Qx (y \in Y \rightarrow R(x, y))$ where Q is a quantifier and Y a plural entity.

This approach would however suffer the same drawbacks described for Fodor and Fodor (1980).

sponding to the agent role would lead to the wrong interpretation:

(19) $$(\lambda y \lambda x \forall z(z \in y \rightarrow \mathbf{grade}(x,z)))\,\mathbf{paper}^* \rightsquigarrow$$
$$\exists x \forall z(z \in \mathbf{paper}^* \rightarrow \mathbf{grade}(x,z))$$

The solution proposed by Lasersohn (1993) is radically different from those we have seen so far and is based on a flat semantic representation centered around *events*. According to Lasersohn (1993), a sentence like (14) is interpreted as (20). In this representation the event is considered central and the various roles are introduced by specific predicates that link the individuals involved with the event. Most importantly, in a case like (14) we are not required to specify an agent.

(20) $\exists e(\mathbf{grade}(e) \wedge \textsc{patient}(\mathbf{paper}^*, e))$

To explain the equivalences of the kind we saw in (1-6) Lasersohn (1993) introduces ontological postulates that require that every *atomic* event (at least those showing the behavior under discussion) must have an agent. For instance, to capture the intuition behind the interpretation of (14) we can introduce the postulate in (22).[4] In words, postulate (22) says that every atomic grading event must have at least one agent. The restriction to atomic events is crucial to obtain the correct reading. In fact, by restricting the applicability of the postulates to atomic events, Lasersohn (1993) is able to obtain the correct relative scope between the universal quantifier introduced by the distributive reading of the verb (which is nothing but a collection of atomic events), and the existential quantifier introduced by the ontological rules that Lasersohn assumes govern the meaning of verbs.

(22) $\forall e(\textsc{atom}(\mathbf{grade}, e) \rightarrow \exists x(\textsc{agent}(x,e)))$

While this analysis has some attractive features, like the ability to explain in a simple and general way the interactions between implicit arguments and distributive readings, it also has some major drawbacks. The first problem is connected to the fact that implicit arguments are analyzed as being introduced at a very general level. This gives the wrong predictions for certain cases. Consider for example the verb *break*. This verb supports distributive readings, as attested by (23), which is usually interpreted as equivalent to *each mirror was broken by someone/something*.

(23) The mirrors were broken

However, the ontological postulate that would explain this equivalence, would also give us the wrong prediction that in the case of the atomic event introduced by the unaccusative use of *break* in (24) there is an unexpressed agent. This interpretation does not seem to be supported by intuition, as sentence (24) is not necessarily equivalent to *Someone/something broke my TV*.

[4] ATOM is defined by Lasersohn (1993) as follows:

(21) $\textsc{atom}(\alpha, e) \leftrightarrow (\alpha(e) \wedge \neg\exists e'(e' < e \wedge \alpha(e')))$

where $<$ is an ordering relation on events (e.g. a mereological one).

(24) My TV broke.

In general, unaccusative uses seem to contradict the ontological requirement that atomic events must have an agent.[5] Ontological postulates are applied to all cases that involve the predicates they control, and there is no way to limit their application to predicates introduced by specific linguistic expressions without shifting to a lexical perspective.[6] This suggests that the ontological requirement is too strong. Natural language has the expressive means (through specific constructions or specific lexical items) to refer to atomic events that may lack an agent, so we should assume a more flexible semantic model that allows both cases.[7]

Another problem with the analysis of Lasersohn (1993) is in the assumption that the existential quantifier that binds an implicit agent in a passive construction with a distributive reading must always have narrowest scope. There are cases like (26) where this generalization does not seem to hold.

(26) The numbers were summed.

The preferred reading for sentence (26) is one where a single entity sums the numbers (or where at least this entity performs the final addition that gives the final result). In this case, the problem again seems to be the strength of the ontological postulates, more specifically in the fact that they focus on atomic events.

In our opinion, the drawbacks of Lasersohn's event-based approach far outweigh its benefits. At the same time we recognize that in the case of lexical approaches to implicit arguments, the problem of how to derive the correct scope between quantifiers binding implicit arguments and quantifiers controlling distributive readings remains open. A naive lexical solution to this problem would amount to hard coding the relative order of the quantifiers in the lexical entries. (27) would give us the correct interpretation in a case of a sentence like (14).

(27) $\lambda y \forall z \exists x (z \in y \rightarrow \mathbf{grade}(x, z))$

The price to pay to get the correct reading would be to say that the meaning of a passive construction is not derived by general principles from the meaning we associate with the active form of the verb, but rather that it is separately specified in the lexicon. In turn the lexicon should be enriched with postulates that govern the relation between the meaning of active and passive forms. We believe that this

[5]Some languages use a reflexive-like construction in place of unaccusative constructions. This could suggest an implicit agent co-referring with the patient. However this is does not seem to be the case as the following Italian example shows:

(25) Ieri, alle tre, la porta si è chiusa. È stato Marco. (*Yesterday, at three, the door closed. Marco did it.*)

[6]A variant of this approach is implemented in the event-based lexical solution we discuss in Asudeh and Giorgolo (2012).

[7]To be fair, any lexical solution that does not employ a flat semantics like the one used by Lasersohn (1993) will have difficulties in explaining unaccusative constructions.

is a rather high price to pay. A more promising approach would probably first tackle the question of how distributive readings are represented in the lexicon. We leave this line of research for future work.

We conclude this review of the literature with a recent proposal by Blom et al. (2012), that approaches the problem from a lexicalist perspective but avoids the problematic assumption that implicit arguments are evidence of an underlying ambiguity. In a nutshell, Blom et al. (2012) assume that the interpretation of predicates that allow for implicit arguments corresponds to functions that can take a variable number of arguments. The implicit arguments are marked at the type level as belonging to special *option* or *sum types*, which are derived from standard types by the addition to the domain of a distinguished element that is meant to represent the absence of any other usable value. The introduction of sum types is then coupled with an extension of the meaning language, the simply-typed λ-calculus, that includes equality predicates for the sum types and a *choice* term constructor, e.g. the familiar `if-then-else` construction used in many programming languages. Blom et al. (2012) then represent the semantics of verb like *eat* as a function that takes arguments both of standard types and of sum types. The function performs checks on the sum type argument and in case it does not find a usable value (i.e. the function has been passed the distinguished additional element) it proceeds to bind the missing argument with an existential quantifier. The result is a meaning capable of adapting to the context in which it is used.

We believe that this solution is superior to the other presented so far for a number of reasons. First of all, being a pure lexical solution, it avoids the difficulties encountered by the approaches based on meaning postulates (i.e. the explosion of the number of postulates) and those based on ontological restrictions. Lexical solutions are capable of a much finer grained control on the distribution of implicit arguments, also capturing the intuitive idea that these constructions are somehow tied to lexical knowledge, as the opposite behavior of two related verbs like *eat* and *devour* so clearly suggests. At the same time, this solution avoids one of the weak points of lexical solutions: the idea that constructions with implicit arguments are in some way connected to ambiguous lexical entries.

We see two problems with analyses based on the use of ambiguous lexical entries. First of all the two entries are not effectively expressing two different meanings, but rather two different patterns of composition. Our intuition is that the difference between *Yesterday, John ate* and *Yesterday, John ate something* is not related to the meaning of the verb *ate*, but rather to pragmatic considerations, such as the availability of certain information or the focus on certain aspects of the message transmitted. The second problem, which is connected to the first one, is that the use of ambiguous lexical entries introduces a form of non-determinism that, again, is not reflected in our linguistic intuitions.

These considerations lead us to think that solutions based on lexical ambiguity are at best approximations of the phenomenon we want to model. The use of ambiguous lexical entries seems justified only because of the limitations of traditional formal semantic tools. Instead, the data we consider in this paper suggest the need of more refined formal tools, capable of dealing in this case with flexible forms of composition. The analysis of Blom et al. (2012) gives us exactly this form of flexibility.

In what follows we will specify how this solution can be integrated in the LFG architecture. We will show that the analysis of Blom et al. (2012) is actually an instance of a more general framework for an enriched form of composition that we have already presented in Giorgolo and Asudeh (2011), where it was justified on the basis of the unrelated phenomenon of conventional implicatures.

3 Adapting Blom et al. (2012) to the LFG framework

Blom et al. (2012) develop their proposal in the context of Abstract Categorial Grammar, a framework that shares many similarities with LFG-Glue but that also has some differences. The most important difference is the higher importance that Glue Semantics assigns to the resource sensitive nature of semantics. We show here how Blom et al. (2012)'s ideas can be integrated in Glue Semantics and more generally in the LFG framework.

3.1 Monads for implicit arguments

We present here the essential technical details of the analysis of Blom et al. (2012) and how they are related to the extension of Glue Semantics we presented in Giorgolo and Asudeh (2011).

We have already briefly said that Blom et al.'s (2012) solution is based on two parallel extensions. At the level of semantic types, Blom et al. introduce option or sum types, constructed by generating (inductively) for each type τ a new type τ^o, whose domain is the domain of τ with the addition of a distinguished element $*_\tau$. At the level of the meaning language, they add an equality relation $=$ for each new optional type and a special polymorphic function `option`, shown in (28), and defined on the basis of a choice construction, here represented by the familiar mathematical notation for functions defined by cases.

(28) $$\texttt{option}(x, f, d) = \begin{cases} d & \text{if } x = *_\tau \\ f(x) & \text{otherwise} \end{cases} : \tau^o \rightarrow (\tau \rightarrow \sigma) \rightarrow \sigma$$

`option` works by inspecting its first argument: if it is the distinguished element of the optional type τ^o the function returns its third argument, otherwise the function returns the application of its second argument to its first argument.[8] These extensions are put to use in the entries of lexical items that allow implicit arguments. For instance, (29) is the interpretation proposed by Blom et al. (2012) for the verb *eat* (where o is the object argument and s is the subject argument). The verb denotes a

[8]The assumption here is that every value in τ^o that is not $*_\tau$ is also in τ. The fact that τ is a subtype of τ^o would not be a sufficient condition in this case, given that the second argument of `option` has a negative occurrence of τ.

function of two arguments, the first of type "optional" e and the second a pure e. In case the first argument is a regular object its value is used by applying the function $\lambda u(\mathbf{eat}(s,u))$ to it, otherwise the object argument is bound in the predicate $\mathbf{eat}$ by an existential quantifier.

(29) $\lambda o \lambda s(\texttt{option}(o, \lambda u(\mathbf{eat}(s,u)), \exists x(\mathbf{eat}(s,x)))) : e^o \to e \to t$

Given that the type of the denotation of *eat* is $e^o \to e \to t$, the framework requires also an *optionalization* operation of type $\tau \to \tau^o$, that embeds any type into its optional extension.[9] In this way the entry for a verb like *eat* can be combined with an object of type e by lifting the last one to the type e^o. At this point we cannot combine the verb with a quantifier object. In his master's thesis, Blom shows that, in a system that includes functional abstraction and application (as even the simply-typed λ-calculus does) and the optionalization operator, a type like $e^o \to e \to t$ can be "lowered" to $e \to e \to t$, the standard type for a transitive verb. We now show that the system described by Blom et al. (2012) is (almost) a monad, the core mathematical object of the extension of Glue Semantics we proposed in Giorgolo and Asudeh (2011).

We start with a brief introduction to monads (the interested reader may find a more thorough introduction aimed at linguists in Shan (2001) and Giorgolo and Asudeh (2012)). A monad can be defined as a triple $\langle M, \eta, \star \rangle$. M is a type constructor that maps any type τ to a new type $M\ \tau$. η is a function of type $\tau \to M\ \tau$ that lifts values of type τ into values of type $M\ \tau$. η must satisfy certain rules with respect to the third object of the triple, $\star$, so that it functions as a sort of "identity" function that simply embeds values of τ in the new type $M \tau$. $\star$ is a binary function of type $M\ \alpha \to (\alpha \to M\ \beta) \to M\ \beta$ that allows us to bind a value contained in its first monadic argument to a standard name to be used in the body of a function generating a new monadic object. Intuitively $\star$ plays the role of a special functional application that mediates between monadic and non-monadic values.

There are different ways in which we can intuitively understand how monadic values are different from traditional ones. The metaphor we will use in this paper sees monadic values as *computations* that produce values. The idea is that lexical resources that have a monadic type require some effort to be unpacked. What is most important is that these computations may have side effects besides producing a value, much like computations run by a computer may have other effects besides returning a value (e.g. writing some intermediate results to a log file, access a database or printing warnings to a console). In this paper we will be using monads to model computations that possibly fail without returning a value.

In Giorgolo and Asudeh (2011), we presented a system to integrate monads in Glue Semantics. Here we present a superior system suggested to us by Avery Andrews (p.c.) and based on the logic presented in Benton et al. (1998). The present system is at the same time simpler and more elegant and all the analyses we discussed in Giorgolo and Asudeh (2011) can be easily translated in the new

[9]The operator is explicitly introduced and discussed in Blom (2012).

formalism. The idea is to extend the set of linear connectives with a unary connective $\Diamond$ — used to mark monadic resources. In natural deduction format, the proof theory of this new connective is captured by the usual *introduction* and *elimination* pair of rules, shown here respectively in (30) and (31) with the corresponding Curry-Howard-like correspondence.

(30) $$\frac{x : a}{\eta(x) : \Diamond a}\,\Diamond I$$

(31) $$\frac{m : \Diamond a \qquad \begin{array}{c}[x : a]_i \\ \vdots \\ n : \Diamond b\end{array}}{m \star \lambda x.n : \Diamond b}\,\Diamond E_i$$

The introduction rule states that when we have a resource we are always free to lift it to a monadic level. This is reflected in the λ-calculus side by embedding the value in the monadic type by using the “innocuous” η map. The elimination rule is better understood if we consider first the λ-terms that encode the proof step. What rule (31) says is that if we are able to produce a monadic value n by assuming some value that we call x associated with a resource a and we have a proof of a computation that generates such a resource, then we can use the $\star$ operator to extract the value from the computation and plug it into the body of n. At the level of Glue Logic, we go from a situation with two monadic resources and an open hypothesis, to one without the hypothesis and with only one monadic connective.

We can now show that the system of Blom et al. (2012) is a monad, more specifically what is known in the functional programming tradition as the $Option$ or $Maybe$ monad. The type constructor M is represented in this case by the operation $\cdot^o$ which generates a new type τ^o for each type τ by adding a distinguished element $*_\tau$ to its domain. The *optionalization* operator corresponds to η, as it maps values of type τ into values of type τ^o. η defined in this way can be proven to satisfy the rules we mentioned above with respect to $\star$, which is not used by Blom et al. (2012), but which would be defined as in (32).

(32) $$m \star k = \begin{cases} * & m = * \\ k(m) & \text{otherwise} \end{cases}$$

Despite the similarity between (32) and the definition for the `option` operation given in (28), the two functions are quite different and operate at different levels, as $\star$ is used to combine different monadic resources, while `option` is used only internally in lexical entries. This means that we will keep this last operation as a primitive addition to the language used to specify lexical entries. Finally, the *de-optionalization* operation that is needed for the analysis of the composition of verbs taking monadic arguments with quantified arguments can also be shown to be derivable in this system. We give the proof in (33), where we show that a resource $\Diamond\alpha \multimap \beta$ can be “lowered” to a resource $\alpha \multimap \beta$ without monadic subformulae in negative contexts.

(33)
$$\cfrac{\cfrac{\cfrac{[x:\alpha]^1}{\eta(x):\Diamond\alpha}\,\Diamond I \qquad f:\Diamond\alpha\multimap\beta}{f(\eta(x)):\beta}\multimap E}{\lambda x.f(\eta(x)):\alpha\multimap\beta}\multimap I_1$$

In this framework verbs like *eat*, *read* or *drink* subcategorize for an object but they consume it only when wrapped in a monad. For example, (34) would be the lexical entry for the verb *eat*.[10]

(34)
$$\begin{array}{lll} \text{eat} & \text{V} & (\uparrow \textsc{pred}) = \text{‘eat’} \\ & & \lambda o \lambda s(\texttt{option}(o, \lambda u(\mathbf{eat}(s,u)), \exists x(\mathbf{eat}(s,x)))) \\ & & \Diamond(\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma \end{array}$$

The λ-term in (34) represents the semantic contribution of *eat*. It is a function of two arguments, o and s, the former a *computation* returning a value of type e while the latter is a pure value of type e (possibly produced by a computation at a different level). The body of the function uses the `option` procedure to test the result of o: if it is a value of type e then the term $\lambda u(\mathbf{eat}(s,u))$ is applied to the result and the result is used as the second argument of the relation $\mathbf{eat}$, otherwise `option` returns its third argument $\exists x(\mathbf{eat}(s,x))$ were the second argument of $\mathbf{eat}$ is bound by the existential quantifier.

In the case of a passive construction we can derive its denotation from the active form using the function `passivize` defined in (36) that takes as argument a function of type $e \to e \to t$ and returns a new function of type $\mathit{Option}\ e \to e \to t$.[11]

(36) $$\texttt{passivize}(f) = \lambda a \lambda p(\texttt{option}(a, \lambda a(f(a,p)), \exists x(f(x,p))))$$

At the level of Glue terms, this corresponds to remapping the template on the left in (37) to the one on the right.

(37)
$$\begin{tikzcd}[column sep=tiny] (\uparrow & \textsc{obj})_\sigma \multimap (\uparrow \arrow[rrr, bend left] & \textsc{subj})_\sigma \multimap \uparrow_\sigma \qquad \Diamond(\uparrow \arrow[r, bend left] & \textsc{obl})_\sigma \multimap (\uparrow & \textsc{subj})_\sigma \multimap \uparrow_\sigma \end{tikzcd}$$

[10]For the sake of simplicity we do not require anything of the implicit object but to exist. A more realistic lexical entry would require the bound variable to be something that is food for the referent. For example if Dr. McCoy from Star Trek utters “Every subject ate” referring to a group of alien beings in his lab, we expect that each subject ate something compatible with its biology. Notice that the lexical entry in (34) allows us to make the value of the bound variable x dependent on the value of the variable s as required. Also see Asudeh and Giorgolo (2012).

[11]Alternatively we could move the `option` outside the second lambda abstraction:

(35) $$\texttt{passivize}(f) = \lambda a(\texttt{option}(a, \lambda a \lambda p(f(a,p)), \lambda p \exists x(f(x,p))))$$

The two definitions are equivalent.

3.2 Implicit arguments and projections

The analysis based on the idea of optional arguments can be easily integrated in Glue Semantics as we just described. However, the projective nature of LFG forces us to be more precise about the way in which the absence of an argument is specified in the derivation. In fact, although Blom et al. (2012) speak about optional arguments, the absence of an argument is only implicitly signalled by the use of the distinguished value $*$ which still counts as a resource that is regularly consumed. The fact that LFG allows us to have access to all the linguistic structures computed at earlier stages gives us a way to actually indetify the contexts in which the introduction in the semantic derivation of the special value $*$ is necessary. In this way we are able to eliminate any source of non-determinism that may stem from the uncertainty connected to introduction of the "silent" $*$ value.

The solution we propose is based on the idea that the $Option$ monad can be interpreted as representing a computation that may fail. In the case of implicit arguments the computation that may fail is the one that constructs the semantic resources out of the actual linguistic elements of the sentence. In LFG the f-structure projects a semantic structure (s-structure) that is used to construct the premises for the glue proof. In Glue Semantics the s-structure is then used together with the lexicon as the input for the procedure that generates *resources* (i.e. premises) for the semantic derivation. This procedure is normally understood as producing a set of resources/premises. What we make explicit is the possibility that this procedure encounters an exceptional situation, such as when attempting to instantiate the linear formula template for the verb *eat* (cf. (34)). In that case, there is no linking with the s-structure projected by the OBJ feature, as no such feature is present. We assume that the procedure signals this error and links it to the rest of the template formula which can instead be instantiated. The error therefore becomes a (faulty) premise for the semantics derivation.[12]

Alternatively we can reuse some of the intuitions of the second analysis we presented in Asudeh and Giorgolo (2012). If we posit that the lexical entry of a verb like *eat* introduces in the s-structure both an AGENT and a PATIENT feature whose values are determined on the basis of the f-structure by the σ projection we can understand the presence in our derivations of an error premise in two (roughly equivalent) ways:[13]

1. the values of the features of the s-structure may all be initialized to $*$ signaling by default that no resource, corresponding to that semantic feature, has been explicitly introduced yet. The σ projection fills the values of the features that have a corresponding f-structural counterpart. In the case of an

[12]A fundamental assumption of this analysis is that grammatical functions are subctegorized in the semantic representation of lexical resources rather than at the syntactico-functional level. We elaborate more on this idea in section 5.

[13]In Asudeh and Giorgolo (2012), we use features like ARG$_1$ and ARG$_2$ instead, since AGENT and PATIENT are redundant with predicates in the event semantics in the meaning language, but here we do not assume an event semantics.

Word	Category	Constraints
John	N	$(\uparrow \textsc{pred}) = $ 'John' $\mathbf{john} : \uparrow_\sigma$
ate	V	$(\uparrow \textsc{pred}) = $ 'eat' $\lambda o \lambda s(\texttt{option}(o, \lambda u(\mathbf{eat}(s,u)), \exists x(\mathbf{eat}(s,x))))$ $\Diamond(\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$
something	N	$(\uparrow \textsc{pred}) = $ 'some' $\lambda P \exists x(P(x))$ $(\uparrow_\sigma \multimap X) \multimap X$
kiss	V	$(\uparrow \textsc{pred}) = $ 'kiss' $\lambda o \lambda s(\mathbf{kiss}(s,o))$ $(\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$
kissed$_{\text{pass}}$	V	$(\uparrow \textsc{pred}) = $ 'kiss' $\texttt{passivize}(\lambda o \lambda s(\mathbf{kiss}(s,o))) \rightsquigarrow$ $\lambda a \lambda p(\texttt{option}(a, \lambda a(\mathbf{kiss}(a,p)), \exists x(\mathbf{kiss}(x,p))))$ $\Diamond(\uparrow \textsc{obl})_\sigma \multimap (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$

Table 1: Toy lexicon

implicit object the PATIENT feature receives no value.

2. σ by default attempts to fill the values of all s-structural features. If a feature cannot be assigned a value an error is raised and registered in the s-structure using the special value $*$.

If we choose this second approach we have to change the lexicon accordingly to make direct reference to the s-structural features. The changes are straightforward. However, in this paper we choose the first implementation of the monadic approach.

To make our proposal clearer, we now present a detailed analysis of some interesting cases.

4 Analysis

In all the analyses, we assume the toy lexicon in Table 1.

4.1 Implicit objects

The first example we analyze is the case of an implicit object (38).

(38) John ate

The simplified f-structure associated with (38) is shown in (39).

(39)
$$e\begin{bmatrix}\textsc{pred} & \text{‘eat’}\\ \textsc{subj} & j\begin{bmatrix}\textsc{pred} & \text{‘John’}\end{bmatrix}\end{bmatrix}$$

When instantiating the Glue Term for the verb *ate*, the parser / interpreter tries to access the OBJ function in (39). This leads to an error given that no such function is represented in (39). The error is signaled and propagated to the rest of the interpretation process. The error is explicitly introduced in the semantic derivation by the premise $\Diamond n$, marked by $\Diamond$ as it is not a pure value but a computational object, and associated with the special value $*$. The resulting proof is shown below and consists of two simple functional applications/$\multimap$-eliminations.

$$\dfrac{\begin{array}{c}\text{John}\\ [\![\text{John}]\!] : j\end{array} \qquad \dfrac{\begin{array}{c}\text{ate}\\ [\![\text{ate}]\!] : \Diamond n \multimap j \multimap e\end{array} \qquad \begin{array}{c}\texttt{error}\\ * : \Diamond n\end{array}}{\lambda s \exists x(\mathbf{eat}(s,x)) : j \multimap e}\multimap E}{\exists x(\mathbf{eat}(\mathbf{john},x)) : e}\multimap E$$

The error is detected by the meaning component of *ate* and a default interpretation for the object (the existentially bound variable) is used. In this way, the error is neutralized and the process is successful, leading to the expected interpretation.

4.2 Explicit objects

The second example we consider shows how the same lexical entry for the verb *ate* generates the correct interpretation when the object is explicitly realized as in sentence (40).

(40) John ate something

Based on the f-structure in (41) we associate with the sentence the semantic derivation in Figure 2.

(41)
$$e\begin{bmatrix}\textsc{pred} & \text{‘eat’}\\ \textsc{subj} & j\begin{bmatrix}\textsc{pred} & \text{‘John’}\end{bmatrix}\\ \textsc{obj} & st\begin{bmatrix}\textsc{pred} & \text{‘some’}\end{bmatrix}\end{bmatrix}$$

The crucial step in the proof is the “lowering” of the type of the denotation of *ate* from the type $Option\ e \rightarrow e \rightarrow t$ to the type $e \rightarrow e \rightarrow t$. This corresponds to the de-optionalization proof we presented in (33). At the level of meaning terms, we simply create a new function that wraps its first argument in a monad using η, therefore generating a computation that does nothing besides returning the value passed as an argument.

$$
\dfrac{\dfrac{\begin{array}{c}\text{John}\\ [\![\text{John}]\!] : j\end{array} \quad \dfrac{\dfrac{\dfrac{\begin{array}{c}\text{ate}\\ [\![\text{ate}]\!] : \Diamond t \multimap j \multimap e\end{array} \quad \dfrac{[z : t]^1}{\eta(z) : \Diamond t}\,\Diamond I}{\lambda s(\mathbf{eat}(s,z)) : j \multimap e}\,\multimap E}{\lambda z \lambda s(\mathbf{eat}(s,z)) : t \multimap j \multimap e}\,\multimap I_1 \quad [w : t]^2}{\lambda s(\mathbf{eat}(s,w)) : j \multimap e}\,\multimap E}{\mathbf{eat}(John, w) : e}\,\multimap E}{\boxed{1} = \lambda w(\mathbf{eat}(John, w)) : t \multimap e}\,\multimap I_2
$$

$$
\dfrac{\boxed{1} \qquad \begin{array}{c}\text{something}\\ [\![\text{something}]\!] : (t \multimap X) \multimap X\end{array}}{\exists y(\mathbf{eat}(John, y)) : e}\,\multimap E
$$

Figure 2: Proof for *John ate something*

4.3 Passives

Finally, we show how a passive construction without a *by*-phrase gets an existential interpretation. The example sentence and the associated f-structure are shown respectively in (42) and (43). The proof has exactly the same shape as the one for the case of an implicit object. What is interesting is how the Glue and meaning terms for the passive form of *kiss* are constructed on the basis of their active counterparts (see Table 1). The resulting denotation corresponds to a function that is capable of providing an existential closure in case the *agent* is not expressed phonologically.

As it was the case for the analysis of implicit objects, the procedure that instantiates the linear formula governing the compositional behavior of *kissed*$_{\textit{pass}}$ fails as there is no projection of an OBL feature in the s-structure. The error is added to the premises that guide the semantic composition reasoning and is linked to the resource corresponding to the passive verb.

(42) John was kissed

(43)
$$
k \begin{bmatrix} \text{PRED} & \text{‘kiss’} \\ \text{SUBJ} & j \begin{bmatrix} \text{PRED} & \text{‘John’} \end{bmatrix} \end{bmatrix}
$$

$$
\dfrac{\begin{array}{c}\text{John}\\ [\![\text{John}]\!] : j\end{array} \quad \dfrac{\begin{array}{c}\text{kissed}_{\text{pass}}\\ [\![\text{kissed}_{\text{pass}}]\!] : \Diamond n \multimap j \multimap k\end{array} \quad \begin{array}{c}\texttt{error}\\ * : \Diamond n\end{array}}{\lambda p \exists x(\mathbf{kiss}(x,p)) : j \multimap k}\,\multimap E}{\exists x(\mathbf{kiss}(x,\mathbf{john})) : e}\,\multimap E
$$

5 The transitive/intransitive continuum and subcategorization

Before concluding, we would like to briefly discuss some ideas that emerged in the analysis of the data discussed in this paper. We first focus on the specific phenomenon of implicit objects and see how it may be related to other related phenomena and what it can tell us about the traditional transitive/intransitive distinction. We then extend the discussion to the more general notion of subcategorization and its position in LFG.

The fact that verbs like *eat* and *warn* can be constructed either with an explicit or an implicit object blurs the standard distinction between transitivity and intransitivity. This distinction is usually considered to cut across levels of analysis, as transitivity is normally explained as both a syntactic and a semantic property. Yet in the cases we discussed there seem to be a misalignment between syntax and semantics, given that *Yesterday, John ate* represents a syntactically intransitive structure that still retains a semantic interpretation constructed around a binary predicate. These verbs therefore seem to be a sort of in between case, showing a preference for being constructed transitively, but also allowing an intransitive use while retaining their transitive meaning.

On the flip side, in the case of cognate objects and similar constructions, we observe that intransitive verbs are used in a transitive way. However, at the level of semantics, their interpretation remains that of unary predicates. For example, sentence (44) is equivalent to *John died horribly* and does not involve a second entity. Similarly in example (45) we are informed of the specific way in which John was dancing, not he was engaging in a certain relation with or he was performing a certain action on a second entity.

(44) John died a horrible death

(45) Yesterday, John danced the waltz all night long

Here we have verbs that are normally used in intransitive constructions but that in some cases allow for a transitive use, but at the semantic level their meaning remains that of unary predicates.

The generalization that emerges from this considerations is that, from the syntactic perspective, the distinction between transitive and intransitive verbs may be too coarse, and instead a sort of graded continuum between verbs that are always constructed as transitive and others that are always intransitive with in between cases in the middle may better capture the reality of things. At the same time, the data suggests that the distinction between transitive and intransitive meanings is maintained in all the in between cases.

More generally this leads us to reconsider the notion of subcategorization in LFG. It has already been proposed that the best place to capture the idea that predicates require certain arguments is not at the level of f-structure (which is too close to the surface syntactic level) but rather at the level of the linear term that control semantic composition (Kuhn, 2001; Asudeh, 2012). In current LFG practice, there is in fact some duplication of information, given that Glue terms also encode the information about subcategorization, and possibly do so in a more refined and controlled way. But most importantly, the considerations about implicit arguments that we developed in this paper suggest that indeed the requirements about arguments are best expressed at the level of semantics, where we can better observe their effects in problematic cases. Therefore, we propose a revision of the standard practice in LFG of checking for predicate arguments early in the interpretation process, a check that instead should take place in the last phase of the process. At the same time, this idea suggests that the kind of syntactic requirements that we impose on the surface form of linguistic expressions should be more flexible and permit us to account also, for example, for the in between cases we just discussed.

6 Conclusion

In this paper we discussed the case of implicit arguments in a number of constructions that range from optional objects to optional by-phrases in passive constructions. We reviewed a number of proposals in the literature, and identified those based on lexical solutions as the best candidates to properly treat the phenomenon under discussion. We focused in particular on the solution of Blom et al. (2012) as the best one, given that it maintains the benefits of lexical solutions without resorting to the idea that these expressions are in any sense ambiguous, a common assumption in the literature that is not supported by our linguistic intuitions. We adapted this solution to the LFG framework. The adaptation makes use of the extension of Glue Semantics with monads, an addition we introduced Giorgolo and Asudeh (2011), to analyse the unrelated phenomenon of conventional implicatures. In this way we managed to extend the analytic capabilities of LFG at no cost, and we have obtained further evidence that supports the idea that we need to extend the traditional semantic toolkit with more powerful mathematical structures. Finally we have discussed how this data prompts us to rethink the place of subcategorization in the LFG architecture. We proposed a view where subcategorization is considered a semantic/compositional property of linguistic expressions, rather than cutting across multiple levels of analysis.

References

Asudeh, Ash. 2012. *The Logic of Pronominal Resumption*. Oxford: Oxford University Press.

Asudeh, Ash and Giorgolo, Gianluca. 2012. Flexible Composition for Optional and Derived Arguments. In *Proceedings of the LFG12 Conference*, Stanford: CSLI Publications, this volume.

Benton, Nick, Bierman, G. M. and de Paiva, Valeria. 1998. Computational Type from a Logical Perspective. *Journal of Functional Programming* 8(2), 177–193.

Blom, Chris. 2012. *Optional Arguments in Abstract Categorial Grammar*. Masters Thesis, Utrecht University.

Blom, Chris, de Groote, Philippe, Winter, Yoad and Zwarts, Joost. 2012. Implicit Arguments: Event Modification or Option Type Categories? In Maria Aloni, Vadim Kimmelman, Floris Roelofsen, Galit Sassoon, Katrin Schulz and Matthijs Westera (eds.), *Logic, Language and Meaning, 18th Amsterdam Colloquium, Amsterdam , The Netherlands, December 19-21*, volume 7218 of *Lecture Notes in Computer Science*, pages 240–250, Springer Berlin / Heidelberg.

Bresnan, Joan. 1978. A Realistic Transformational Grammar. In Morris Halle, Jan Bresnan and George A. Miller (eds.), *Linguistic Theory and Psychological Reality*, MIT Press.

Carlson, Greg N. 1984. Thematic Roles and Their Role in Semantic Interpretation. *Linguistics* 22(3), 259–280.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. San Diego, CA: Academic Press.

Dowty, David. 1982. Grammatical Relations and Montague Grammar. In Pauline I. Jacobson and Geoffrey K. Pullum (eds.), *The Nature of Syntactic Representation*, pages 79–130, D. Reidel.

Fodor, Jerry A. and Fodor, Janet Dean. 1980. Functional Structure, Quantifiers, and Meaning Postulates. *Linguistic Inquiry* 11(4), 759–770.

Giorgolo, Gianluca and Asudeh, Ash. 2011. Multidimensional Semantics with Unidimensional Glue Logic. In Miriam Butt and Tracy Halloway King (eds.), *Proceedings of the LFG11 Conference*, pages 236–256, CSLI Publications.

Giorgolo, Gianluca and Asudeh, Ash. 2012. $\langle M, \eta, \star \rangle$ Monads for Conventional Implicatures. In Ana Aguilar Guevara, Anna Chernilovskaya and Rick Nouwen (eds.), *Proceedings of Sinn und Bedeutung 16*, volume 1, pages 265–278, MIT Working Papers in Linguistics.

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

Lasersohn, Peter Nathan. 1993. Lexical Distributivity and Implicit Arguments. In *Proceedings from Semantics and Linguistic Theory III*, pages 145–161.

Needham, Stephanie and Toivonen, Ida. 2011. Derived Arguments. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG11 Conference*.

Shan, Chung-chieh. 2001. Monads for Natural Language Semantics. In Kristina Striegnitz (ed.), *Proceedings of the ESSLLI-2001 Student Session*, pages 285–298, 13th European Summer School in Logic, Language and Information.

Stanley, Jason. 2000. Context and Logical Form. *Linguistics and Philosophy* 23, 391–434.