Neo-Davidsonian Semantics in Lexicalized Grammars

Petr Homola
Codesign, s.r.o.
Czech Republic
phomola@codesign.cz

Abstract

We present a method of semantic parsing within lexicalized grammars that generates neo-Davidsonian logical forms. We augment an existing LFG grammar with predicate logic formulae in the lexicon and logical annotations in the context-free rules to obtain logical forms of a sequence of sentences that can be used to construct a model, i.e., formulae with resolved anaphoric and indexical expressions, by skolemization and discourse resolution based on salience.

1 Introduction

Semantic parsing is important for many NLP applications. We present results of experiments with semantic parsing within the framework of Lexical-Functional Grammar (LFG), a well-researched unification-based formalism (Kaplan and Bresnan, 1982; Dalrymple et al., 1995a; Butt et al., 1999; Bresnan, 2001; Dalrymple, 2001; Nordlinger and Bresnan, 2011) for which wide-coverage grammars for a variety of typologically different languages are available.

The method we have developed can be used with any semantic formalism that can be expressed in first-order logic with equality (FOL). The approach of Parsons (1990) is usually called "neo-Davidsonian" because it is a variation of the event-based formal representation of Davidson (1967). The formalism used in this paper is a combination of that of Parsons and Hobbs (1985; 2003) who extends Davidson's approach to all predications. We show how simple and complex predicates are represented, how semantic representations can be incrementally created in LFG via codescription and how discourse and context (anaphoric and indexical expressions) can be treated. While we used LFG in the experiments, the method is flexible enough to be used in any rule-based grammar formalism based on context-free or categorial grammars such as (Uszkoreit, 1986) or (Kay, 1979; Kay, 1984).

In the next section we survey recent approaches to semantics in LFG. In Section 3 the neo-Davidsonian approach to semantics is presented. Section 4 expounds the way how neo-Davidsonian logical forms can be incrementally built up using LFG machinery. Section 5 focuses on how discourse models can be constructed from logical forms by resolving anaphoric and indexical expressions. Finally, we conclude in Section 6.

2 Related Work

Virtually all approaches to formal semantics assume the Principle of Compositionality, formally formulated by Partee (1995) as follows: "The meaning of a whole is a function of the meanings of the parts and of the way they are syntactically combined." This means that semantic representation can be incrementally built up when constituents are put together during parsing. Since c(onstituent)-structure expresses sentence topology rather than grammatical relations, the rules that combine the meanings of subphrases frequently refer to the underlying syntactic structure, that is, f(unctional)-structure in LFG. Indeed, Halvorsen and Kaplan (1995) in their account of semantics within LFG define the s(emantic)-structure as a projection of the c-structure (through the correspondence function σ) but they refer to grammatical functions (GFs) by means of the compound function

$\sigma\phi^{-1}$,[1] as in the following example:

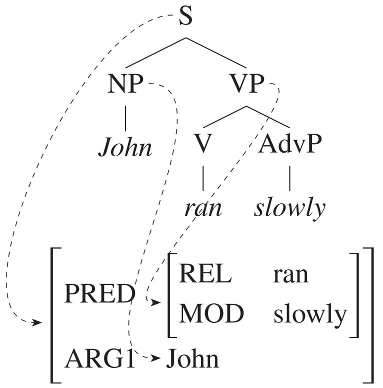
(1) John ran slowly.

The corresponding lexical entry for the verb is

$$(2)\quad \begin{aligned} &(\sigma\mathcal{M} * \text{REL}) = \text{ran} \\ &(\sigma\mathcal{M} * \text{ARG1}) = \sigma\phi^{-1}(\uparrow \text{SUBJ}) \\ &(\sigma\mathcal{M} * \text{ARG2}) = \sigma\phi^{-1}(\uparrow \text{OBJ}) \end{aligned}$$

and the resulting correspondence between the c-structure and the s-structure is

(3)



Note that Halvorsen and Kaplan (1995) represent s-structures as feature structures (since they use functional annotations to construct them). (3) can be more conventionally expressed as $slowly(ran)(John)$.

Recent approaches to semantics in LFG are based on the so-called "glue semantics" (Dalrymple et al., 1993; Dalrymple et al., 1995b; Dalrymple, 2001). Consider the sentence

(4) Bill obviously kissed Hillary

Its semantic form is, according to Dalrymple et al. (1993), $obviously(kiss(Bill, Hillary))$. Glue semantics uses linear logic; lexical entries are assigned "meaning constructors" that consist of a logical expression and instructions for how the meaning is put together. For *to kiss*, for example, we have

$$(5)\quad \begin{aligned} &\forall X, Y.(f \text{ SUBJ})_\sigma \rightsquigarrow X \otimes (f \text{ OBJ})_\sigma \rightsquigarrow Y \\ &\quad \multimap f_\sigma \rightsquigarrow kiss(X, Y) \end{aligned}$$

In words, (5) means that if the meaning of $(f \text{ SUBJ})_\sigma$ is X and the meaning of $(f \text{ OBJ})_\sigma$ is Y, then the meaning of f_σ is $kiss(X, Y)$. For brevity, meaning constructors are sometimes written as $[\![\textbf{word}]\!]$. For (4), then, we get

$$(6)\quad \begin{aligned} &[\![\textbf{Bill}]\!] \otimes [\![\textbf{obviously}]\!] \\ &\otimes [\![\textbf{kissed}]\!] \otimes [\![\textbf{Hillary}]\!] \\ &\vdash f_\sigma \rightsquigarrow obviously(kiss(Bill, Hillary)) \end{aligned}$$

The idea behind glue semantics is that the lexicon and the rules for syntactic assembly provide meaning constructors that are interpreted as soon as all expressions on the left-hand side of the linear implication ($\multimap$) are available.[2]

Note that both Halvorsen and Kaplan (1995) and glue semantics use higher-order logic. In the next section we go on to outline an account of semantics that, while using codescription, relies on pure FOL for representation and on conjunction as the means of meaning assembly, as advocated on Minimalist grounds by Pietroski (2005).

3 Neo-Davidsonian Logical Representation

The sentence *John loves Mary* can be logically expressed (disregarding tense for the sake of simplicity) using a binary predicate for the verb and constants for its arguments:

$$(7)\quad love(John, Mary)$$

Davidson (1967) has introduced "events" into the description of logical forms of sentences to be able to refer to "actions" by means of FOL (i.e., events are treated as individuals). We use the notation and terminology of Hobbs (1985; 2003) who introduced the "nominalization operator" and the term "eventuality" to refer to "possible events". The predicate $love$ in (7) can be "nominalized" and we assume that the following equivalence holds:

$$(8)\quad love(x, y) \equiv \exists e.love'(e, x, y) \wedge Rexist(e)$$

The newly introduced variable e is the eventuality of John's loving Mary and Hobbs' predicate $Rexist$

[1] ϕ is the correspondence function from c-structures to f-structures.

[2] More recent work on glue semantics uses a slightly different notation. (5) would be written as

$$\lambda X.\lambda Y.kiss(X, Y) : (\uparrow \text{SUBJ})_\sigma \multimap [(\uparrow \text{OBJ})_\sigma \multimap \uparrow_\sigma]$$

expresses that the eventuality is realized (this predicate is discussed in (Hobbs, 1985; Hobbs, 2003), we do not use it in the remainder of the paper).

Parsons (1990), too, uses events but he proposes unary predicates for actions and special predicates for their arguments. In this spirit, we use the following notation:

(9)
$$\begin{aligned} &love'(e, x, y) \equiv \\ &\equiv love''(e) \wedge actor(e, x) \wedge patient(e, y) \end{aligned}$$

Rather than θ-roles, we use what is called "protoroles" (Dowty, 1991), "tectogrammatical roles" (Sgall et al., 1986) or "roles on the action tier" (Jackendoff, 1990) in the literature on formal semantic analysis of natural languages. Thus $actor(e, x)$ means that x has the role "actor" in the eventuality e.

In the remainder of the paper, we refer to "neo-Davidsonian" formulae (with actions denoted by double-primed predicates) as *logical forms* (LF).[3]

4 Logical Forms in LFG

In LFG, parsing is driven by a context-free grammar that conforms to the X′-theory (Jackendoff, 1977). In this section we show how LFs can be incorporated formally within LFG. Notationally, we follow Kaplan (1995).[4]

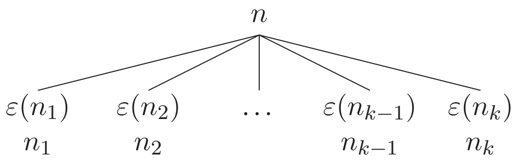
In the formal architecture of LFG, N is the set of nodes, F is the set of f-structures, by Φ we denote the set of formulae (LFs) and V denotes the set of variables that may occur in LFs. In standard LFG, the mapping $M : N \to N$ maps nodes to their mother node and $\phi : N \to F$ maps nodes to f-structures. We introduce $\xi : N \to \Phi$ that maps nodes to formulae and $\tau : N \to V$ that maps nodes to variables.

For terminal nodes, ξ and τ are defined in the lexicon. For example, the morpholexical entry for the proper name *John* contains, besides the usual information, i.e., its category (N) and f-structure [PRED 'John'], a formula ($\xi(n) \overset{df}{=} x = \text{John}$) and a designated variable ($\tau(n) \overset{df}{=} x$ where x is the variable from $\xi(n)$). For a noun, such as *dog*, we have $\xi(n) \overset{df}{=} dog(x)$ and $\tau(n) \overset{df}{=} x$. Likewise, the morpholexical entry for the finite form of the verb *to see* contains its category, its f-structure [PRED 'see⟨(↑ SUBJ), (↑ OBJ)⟩', . . .], a formula ($\xi(n) \overset{df}{=} see''(e)$) and a designated variable ($\tau(n) \overset{df}{=} e$ where e is the variable from $\xi(n)$).

The formula of a nonterminal node is composed from the formulae of its daughter nodes. In LFG, context-free rules are augmented with functional annotations. Likewise, we augment them with logical annotations. Since Hobbs' (1985; 2003) "ontologically promiscuous" formulae are conjunctions of atomic formulae, we combine the formulae of the daughter nodes using the logical connective $\wedge$.

In (10), n_i are the daughter nodes of n and $\varepsilon(n_i)$ are the corresponding logical annotations. Note that $\{n_1, \ldots, n_k\} = M^{-1}(n)$.

(10)



Obviously, $M^{-1}(n) \neq \emptyset$ holds for n, for it is a nonterminal node. We define $\xi(n)$ for nonterminal nodes as follows:

(11)
$$\xi(n) = \bigwedge_{m \in M^{-1}(n)} \xi(m) \wedge \varepsilon(m)$$

We also introduce a new variable, $\tau(n)$. For ease of exposition, we give all formulae in an equivalent prenex normal form, i.e., $\mathrm{Q}_1 x_1 \ldots \mathrm{Q}_n x_n.\varphi$ where Q_i are quantifiers over variables x_i and φ is open (i.e., it contains no quantifiers).

To refer to terms in the formulae associated with nodes in the subtree induced by a rule, we use two metavariables in the logical annotations defined in the context of $\varepsilon(n_i)$ as follows:

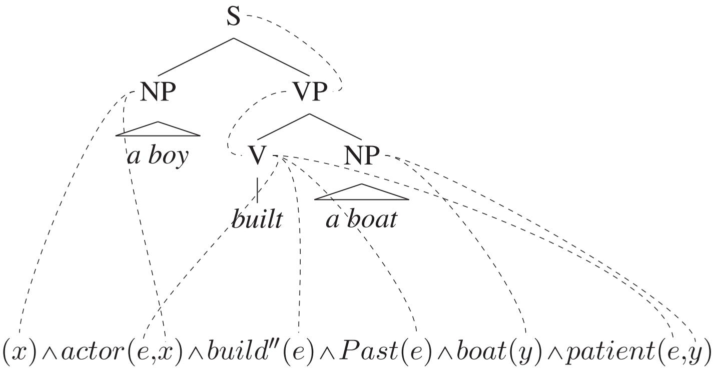
(12)
$$\begin{aligned} \triangle &= \tau(M(n_i)) \\ \bigtriangledown &= \tau(n_i) \end{aligned}$$

[3] We could go even further and reify the double-primed predicates along the scheme

$$P(e) \equiv act(p, e)$$

where p is an "action constant" that corresponds to the predicate P. While such a notation would surely be useful in formulating an axiomatic theory for reasoning, the difference between the notations has no impact on the presently described compositional mechanism.

[4] In the remainder of the paper we assume that we have a LFG grammar that produces c- and f-structures and rules out ill-formed sentences.



$$\psi = \exists e, x, y.boy(x) \wedge actor(e,x) \wedge build''(e) \wedge Past(e) \wedge boat(y) \wedge patient(e,y)$$

$$\varphi = \exists e_1, e_2, e_3, x, y.boy(x) \wedge actor(e_3,x) \wedge build''(e_1) \wedge Past(e_1) \wedge boat(y) \wedge patient(e_2,y) \wedge e_1{=}e_2 \wedge e_2{=}e_3$$

Figure 1: Logical form of *A boy built a boat*.

Thus $\triangle$ and $\triangledown$ are to logical annotations what ↑ and ↓ are to functional annotations.

Let us consider the sentence *A boy built a boat* (slightly adapted from (Hobbs, 1985)). The logical part of the lexicon, used to build up the LF of the sentece, is given in (13).

(13)

word	ξ	τ
a	$\top$	z
boy	$boy(x)$	x
boat	$boat(y)$	y
built	$build''(e_1) \wedge Past(e_1)$	e_1

Because of the definition (11) of ξ for nonterminal nodes, every node must have a formula (i.e., ξ is defined for all nodes), thus we use $\top$ as the formula of the article *a*.

The context-free rules and the corresponding functional and logical annotations are given in (14) (we omit the rule 'NP → Det N' as it is irrelevant for the present discussion).

(14)

$$\begin{array}{llcc} \text{S} & \rightarrow & \text{NP} & \text{VP} \\ & & (\uparrow \text{SUBJ}) = \downarrow & \uparrow = \downarrow \\ & & actor(\triangle, \triangledown) & \triangle = \triangledown \\ \\ \text{VP} & \rightarrow & \text{V} & \text{NP} \\ & & \uparrow = \downarrow & (\uparrow \text{OBJ}) = \downarrow \\ & & \triangle = \triangledown & patient(\triangle, \triangledown) \end{array}$$

The c-structure and the corresponding formula induced by the logical annotations (more precisely, its existential closure) are given in Figure 1. The prenex normal form we obtain is the formula φ but since $e_1 = e_2$ and $e_2 = e_3$, we can use $\psi = \varphi[e_1/e, e_2/e, e_3/e]$ where e is a newly introduced variable that embraces the eventualities e_1, e_2, e_3.

4.1 Complex Predicates

Complex predicates have been subject to research within LFG since the 1990s (Alsina, 1996; Alsina, 1997; Alsina et al., 1997; Broadwell, 2003). Consider the sentence *John made Mary cry* with a syntactically formed causative. The sentence is represented by one f-structure (i.e., the f-structures of *made* and *cry* are unified and the corresponding nodes are coheads) with a complex (hierarchically organized) PRED value:

(15) cause⟨(↑SUBJ),cry⟨(↑OBJ)⟩⟩

The f-structure of *John made Mary cry* is syntactically monoclausal:

(16)
$$\begin{bmatrix} \text{PRED} & \text{cause}\langle(\uparrow\text{SUBJ}),\text{cry}\langle(\uparrow\text{OBJ})\rangle\rangle \\ \text{SUBJ} & [\text{“John”}] \\ \text{OBJ} & [\text{“Mary”}] \end{bmatrix}$$

The LF of *John made Mary cry* is given in (17). Note that two eventualities are introduced (e_1, e_2) as the expression, although syntactically monoclausal, is semantically complex.

(17)
$$\begin{array}{l} \exists e_1, e_2.cause''(e_1) \wedge cry''(e_2) \wedge \\ \wedge actor(e_1, John) \wedge patient(e_1, e_2) \wedge \\ \wedge actor(e_2, Mary) \end{array}$$

Alsina (1997) proposes the creation of complex PRED values using the so-called restriction operator (Kaplan and Wedekind, 1991;

Kaplan and Wedekind, 1993; Wedekind and Kaplan, 1993). Note that as for LFs, no special machinery is needed, syntactically formed complex predicates can by modelled by rules using ordinary logical annotations.

4.2 Syntactic Pivots

The sentences

(18) a. John sold a car to Paul.
b. Paul bought a car from John.

have the same content if interpreted according to a theory that models the common relationship between *to sell* and *to buy*. However

(19) a. I made John sell a car to Paul.
b. I made Paul buy a car from John.

differ in their content although both entail (18a) and (18b). To express the different perspectives of (18a) and (18b) we use the predicate $Pivot$ that distinguishes the individual around which a clause revolves syntactically ($Pivot(e, John)$ and $Pivot(e, Paul)$ in (18a) and (18b), respectively).[5]

Similarly, adverbial modifiers are pivot-sensitive. Consider the following two sentences:

(20) a. John painted Mary.
b. Mary was painted by John.

Using the approach sketched above, we arrive at the following LF:

(21) $$\exists e.paint''(e) \wedge Past(e) \wedge \\ \wedge actor(e, John) \wedge patient(e, Mary)$$

While (21) represents the meaning of both (20a) and (20b), note that the meanings of the two sentences differ if we add the phrase *with great pleasure* because the corresponding predicate $with\_great\_pleasure$ modifies not only the eventuality but also the entity which is the syntactic pivot of the sentence. In English, the syntactic pivot is the subject, i.e., $John$ in (20a) and $Mary$ in (20b). To express the difference between the two sentences, we add the conjunction $Pivot(e, x) \wedge x = y$ to the formulae where x is a newly introduced variable and y is the variable associated with the subject.[6]

5 From Logical Forms to Models

To obtain a model from a sequence of LFs, we have to resolve all anaphoric and indexical expressions and replace all existentially quantified variables with constants. In contrast to parsing, the conversion of LFs into a model was implemented procedurally in our experiments.

Anaphoric expressions refer to an entity in the same sentence or in a preceding one. Anaphora resolution uses morphological (agreement), syntactic (switch-reference), semantic (logical acceptability) or pragmatic (topic/focus articulation) information provided by c-structures, f-structures and i-structures (King, 1997).

According to Kaplan (1979; 1989), the character (linguistic meaning) of an indexical expression is a function from contexts that delivers the expression's content at each context. We assume that the information about the current speaker, hearer, time, and place is available to the system and this contextual information is used to resolve expressions such as *I, you, my, here, now, Past*, etc.

Consider the following two sentences that represent a simple mini-discourse (a coherent sequence of utterances):

(22) a. I saw a dog.
b. The dog barked.

The corresponding LFs, without any intersentential relations, are given in (23).

(23) a. $\exists e_1, x.see''(e_1) \wedge Past(e_1) \wedge$
$\wedge actor(e_1, I) \wedge patient(e_1, x) \wedge$
$\wedge dog(x)$
b. $\exists e_2, y.bark''(e_2) \wedge Past(e_2) \wedge$
$\wedge actor(e_2, y) \wedge dog(y)$

To convert the LFs to a model, we instantiate the existentially bound variables with constants that satisfy the LFs using the information provided by f-structures (morphological, syntactic, semantic, and pragmatic). For e_1 and e_2, we introduce constants c_{e_1} and c_{e_2}. The constant I represents the indexical pronoun *I*. We use contextual information to re-

[5] For a description of pivots at the level of f-structure see (Falk, 1998; Falk, 2000).

[6] Note that LFs need not be purely semantic (for this reason we called them *logical* rather then *semantic*). Pivots are a syntactic notion but they affect semantic interpretation. This is consonant with the approach of Hobbs (2003) who uses, for example, a syntactic predicate (nn) to represent noun-noun compounds in LFs.

place it with a proper name, say *Peter*. The predicate $Past$ is indexical, too. We replace $Past(c_{e_i})$ with $Timepoint(c_{e_i}, t_i) \wedge Before(t_i, n)$ where t_i is a newly introduced constant which represents a time point, and n is the constant for “now” (taken from the context). x is replaced with c_x, a newly introduced constant.

In the second sentence, y is recognized as referentially anchored (we sketch the algorithm for anaphora resolution in the following subsection) and replaced by c_x. The resolved formulae are given in (24) and the situation described in (22) is their conjunction. Since Hobbs (1985; 2003) does not use the connective $\neg$, the formulae are always satisfiable, i.e., a model always exists.

(24)
a. $see''(c_{e_1}) \wedge Timepoint(c_{e_1}, t_1) \wedge$ $Before(t_1, n) \wedge actor(c_{e_1}, Peter) \wedge$ $\wedge patient(c_{e_1}, c_x) \wedge dog(c_x)$
b. $bark''(c_{e_2}) \wedge Timepoint(c_{e_2}, t_2) \wedge$ $\wedge Before(t_2, n) \wedge actor(c_{e_2}, c_x) \wedge$ $\wedge dog(c_x)$

5.1 Anaphora Resolution

The parser operates on isolated sentences, disregarding intersentential coreferences. To resolve intrasentential and intersentential anaphora in coherent sequences of utterances, we formalize salience of entities at the level of pragmatics.

For the purposes of this subsection we assume that we have a discourse that consists of sentences $s_1, \dots, s_n$ and the corresponding feature structures $f_1, \dots, f_n$. An entity is a feature structure that represents a person, an object or an event. Every entity has a special attribute, INDEX, to represent coreferences.

The discourse context is formally a list of indices (values of the INDEX attribute) $C = \langle i_1, \dots, i_m \rangle$. The sentences are processed one by one. We start with $C = \varnothing$. For every f-structure f_i, we proceed as follows:

1. For every entity in f_i, we try to find its referent in C (using available morphological, syntactic, semantic, and pragmatic information). If a referent was found for an entity, its index in C is moved to the front of the list. Otherwise, a new index is assigned to the entity and prepended to the list.
2. The indices of entities that belong to the focus are moved to the front of C because of their high salience.[7]

6 Conclusions

The paper presents a method of integrating neo-Davidsonian event semantics within the formal framework of LFG. We showed how logical forms can be obtained from LFG rules augmented with logical annotations. Further, we showed how anaphoric and indexical expressions can be resolved so that we can construct a (possibly partial) model of the processed text.

Our approach differs from glue semantics in using conjunction instead of linear logic for meaning assembly. Moreover, unlike glue semantics we do not use higher-order logic in semantic forms for practical reasons (mainly in order to be able to use FOL-based theorem provers and model builders for reasoning over LFs).

In further work we will extend our grammar to account for more phenomena and in the next step we will focus on inferential entailment in models.

References

Alex Alsina, Joan Bresnan, and Peter Sells. 1997. Complex Predicates: Structure and Theory. In Alex Alsina, Joan Bresnan, and Peter Sells, editors, *Complex Predicates*, pages 1–12.

Alex Alsina. 1996. *The Role of Argument Structure in Grammar: Evidence from Romance*. CSLI Publications.

Alex Alsina. 1997. A Theory of Complex Predicates: Evidence from Causatives in Bantu and Romance. In Alex Alsina, Joan Bresnan, and Peter Sells, editors, *Complex Predicates*, pages 203–246.

Joan Bresnan. 2001. *Lexical-Functional Syntax*. Blackwell Textbooks in Linguistics, New York.

George Aaron Broadwell. 2003. Complex predication and parallel structures in optimality-theoretic syntax. Handout from a talk given at the 7th Workshop on Optimality Theory Syntax in Nijmegen, Netherlands.

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

[7] We assume that the parser provides information about topic/focus articulation. We use i-structures (King, 1997).

Mary Dalrymple, John Lamping, and Vijat Saraswat. 1993. LFG Semantics via Constraints. In *Proceedings of Sixth Meeting of the European ACL*.

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

Mary Dalrymple, John Lamping, Fernando Pereira, and Vijat Saraswat. 1995b. Linear logic for meaning assembly. In *Proceedings of Computational Logic for Natural Language Processing*.

Mary Dalrymple. 2001. *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics*. Academic Press.

Donald Davidson. 1967. The logical form of action sentences. In *The logic of decision and action*, pages 81–95. University of Pittsburg Press.

David Dowty. 1991. Thematic Proto-Roles and Argument Selection. *Language*, 67:547–619.

Yehuda N. Falk. 1998. On Pivots and Subjects. MS, Department of English, The Hebrew University of Jerusalem.

Yehuda N. Falk. 2000. Pivots and the Theory of Grammatical Functions. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG Conference*.

Per-Kristian Halvorsen and Ronald M. Kaplan. 1995. Projections and Semantic Description in Lexical-Functional Grammar. In Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell, and Annie Zaenen, editors, *Formal Issues in Lexical-Functional Grammar*, pages 279–292. CSLI.

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

Jerry R. Hobbs. 2003. Discourse and Inference. MS, Information Sciences Institute, University of Southern California, Marina del Rey.

Ray Jackendoff. 1977. *X-Bar Syntax: A Study of Phrase Structure*. MIT Press, Cambridge, Massachusetts.

Ray Jackendoff. 1990. *Semantic Structures*. MIT Press, Cambridge, Massachusetts.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *Mental Representation of Grammatical Relations*. MIT Press, Cambridge.

Ronald M. Kaplan and Jürgen Wedekind. 1991. Restriction and Structure Misalignment. MS, Xerox Palo Alto Research Center.

Ronald M. Kaplan and Jürgen Wedekind. 1993. Restriction and Correspondence-based Translation. In *Proceedings of the 6th EACL Conference*, pages 193–202.

David Kaplan. 1979. On the Logic of Demonstratives. *Journal of Philosophical Logic*, 8(1):81–98.

David Kaplan. 1989. Demonstratives: An Essay on the Semantics, Logic, Metaphysics, and Epistemology of Demonstratives and Other Indexicals. In *Themes from Kaplan*, pages 481–563. Oxford University Press.

Ronald M. Kaplan. 1995. The Formal Architecture of Lexical-Functional Grammar. In *Formal Issues in Lexical-Functional Grammar*, pages 7–28. CSLI Publications.

Martin Kay. 1979. Functional Grammar. In *Proceedings of the 5th meeting of the Berkeley Linguistics Society*.

Martin Kay. 1984. Functional Unification Grammar: a formalism for machine translation. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting of the Association for Computational Linguistics*.

Tracy Holloway King. 1997. Focus Domains and Information Structure. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG Conference*.

Rachel Nordlinger and Joan Bresnan. 2011. Lexical-Functional Grammar: Interactions Between Morphology and Syntax. In Robert D. Borsley and Kersti Börjars, editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, pages 112–140. Wiley-Blackwell Publishing.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press.

Barbara H. Partee. 1995. Lexical Semantics and Compositionality. In Lila R. Gleitman and Mark Liberman, editors, *Invitation to Cognitive Science. Language*, volume 1, pages 311–360. MIT Press.

Paul M. Pietroski. 2005. *Events and Semantic Architecture*. Oxford University Press.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reider Publishing Company.

Hans Uszkoreit. 1986. Categorial Unification Grammars. In *Proceedings of COLING*.

Jürgen Wedekind and Ronald M. Kaplan. 1993. Type-driven Semantic Interpretation of F-Structures. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*.