PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT.

PROJECTIONS AND SEMANTIC DESCRIPTION IN LEXICAL-FUNCTIONAL GRAMMAR

Per-Kristian Halvorsen
Ronald M. Kaplan
Xerox Palo Alto Research Center
and
Center for the Study of Language and Information
Stanford University

ABSTRACT

In this paper we show how the rule language of lexical-functional grammar (LFG) can be extended to permit the statement of semantic rules. This extension permits semantic structure and functional structure to be simultaneously characterized without requiring that the f-structures themselves be taken as input to the semantic component. This makes possible a simplification of functional representations as well as novel semantic analyses, such as a treatment of quantifier scope ambiguity based on functional uncertainty that avoids the need for any quantifier-storage mechanism. The proposals are based on a theory of projections that exploits the notion of structural correspondences to capture the informational dependencies between levels of linguistic form and meaning.

1 Introduction

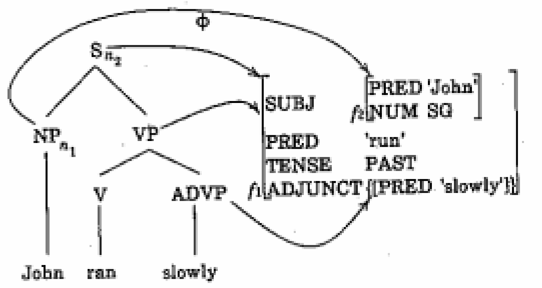
The equality- and description-based organization of LFG (Kaplan and Bresnan 1982) and related unification-based formalisms (DCG (Pereira and Warren 1980), FUG (Kay 1978), PATR (Shieber 1983), and HPSG (Pollard and Sag 1987)) have had easily discernible effects on syntactic theory and the practice of syntactic characterization. But the implications of this organization on architectures for semantic interpretation have not yet been carefully examined. As it turns out, the nature of semantic rules have to be radically altered in this new description-based paradigm as compared to Montague Grammar (Montague 1970). Indeed, some of the most appealing results of Montague's theory do not carry over, for example, compositionality of interpretation and the completeness of the interpretation process (see Halvorsen 1987, 1988). In this paper we show this by way of a case study of semantic interpretation in LFGs based on the notion of *structural correspondences* or *projections* (Kaplan 1987). We first consider what informational dependencies exist between constituent structure (c-structure), functional structure (f-structure), and semantic structure (s-structure). We present a version of the theory of projections where the s-structure is a direct projection of the c-structure and an indirect projection of the f-structure. The specification of these relations is formulated in an extension to the rule language for LFGs that accommodates the notion of multiple projections. In particular, this notation permits the statement of semantic rules on a par with rules for the characterization of functional structures. This theory does not take functional structures to be the sole input to semantic interpretation, and consequently, all semantically relevant information does not have to be funneled through the f-structure. Yet, it allows the dependencies between functional and semantic information to be captured by means of *co-description*—the association of functional and semantic annotations with nodes of the c-structure.

2 Structural Correspondences

A fundamental problem for linguistic theory is to account for the connection between the surface form of an utterance and its meaning. On our view of grammatical theory, this relationship is explicated in terms of *correspondences* between representation structures (Kaplan 1987, Kaplan and Bresnan 1982), rather than *derivation* (i.e. step-wise transformation of structures). We include in the grammar a statement of the informational dependencies between aspects of the linguistic form of the utterance and its meaning, rather than the prescription of an algorithm for step-wise derivation of the meaning of the utterance from its form.

We assume that the various levels of linguistic analysis (syntax, semantics, prosodic structure etc.) are autonomous and obey their own well-formedness conditions. Each level may also employ representations with different mathematical properties (e.g. trees for syntactic structure, finite functions or directed graphs for functional and semantic structures). Even though structures at two different levels may be of different types, we can set them in correspondence using a piece-wise function from elements of one into elements of the other.

Figure 1: C-structure to f-structure correspondence



The original theory of lexical-functional grammar focussed on the correspondence between c-structure trees, exhibiting the hierarchical organization of phrases, and f-structures, representing the grammatical relations holding in the sentence. This correspondence is given by the mapping ϕ, a piece-wise function which may be many-to-one (as illustrated in Figure 1 where the S and VP node are both mapped to the same functional structure, f_1). Moreover, ϕ is not required to be onto (there may be elements of the f-structure which are not in the range of ϕ). This possibility has been used, for example, to provide an intuitive account of unexpressed pronouns (so-called null- or zero-anaphors) (see, for example, Kameyama 1988).

Functional structures (e.g. f_1 and f_2 in Figure 1) are finite monadic functions. Thus f_1 is the function which, when applied to the argument SUBJ, yields the function f_2; when applied to the argument TENSE its value is PAST, and so on. In LFG the information about the properties of f-structures that correspond to c-structure nodes is expressed in simple constraints expressing equality, set inclusion, or existence. A Boolean combination of such constraints is referred to as a *functional description*. Thus the simple functional description

$$(f_1 \text{ SUBJ}) = f_2 \tag{1}$$

states that the result of applying the function characterized by the f-structure corresponding to the S node n_1 (i.e. $f_1 = \phi(n_1)$) to the argument SUBJ is the function f_2 corresponding to the NP node, since $f_2 = \phi(n_2)$. Indeed, given the correspondence ϕ illustrated in the figure, the information in this equation can be equivalently formulated as

$$(\phi(n_1) \text{ SUBJ}) = \phi(n_2). \tag{2}$$

The structural correspondence view separates the statement of informational dependencies between levels of analysis from the process of computing one representation based on information about another. This perspective also opens up the possibility of there being equi-potent informational dependencies between *several* structures, as when viewing the language as consisting of a number of mutually constraining modules. This is the outlook invited by the observation underlying situation semantics (Barwise and Perry 1983) that the interpretation of an utterance depends not only on its syntactic form, but also on the circumstances in which it was uttered, the intonation with which it was uttered, etc..

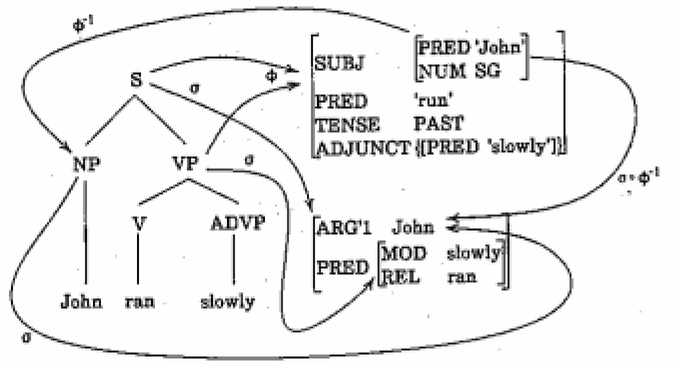
From the linguistic point of view, the crucial question is which subset of all the conceivable informational dependencies do in fact obtain between the different levels of analysis of an utterance. Formally, a correspondence can be defined between any two levels, but a particular linguistic theory may assign meaning to only some of these possibilities. Thus, the theory of projections specializes the notion of structural correspondence and embodies particular claims about what levels of representation (e.g. c-structure and s-structure) are directly related through functional mappings, which levels are related through composite mappings, and which levels are related only by accident.

Figure 2 shows a semantic structure in addition to the functional structure associated with the syntactic tree. The s-structure is viewed as a projection of the c-structure tree through the correspondence function σ. With this configuration the s- and f-structures are not directly related to each other, but informational dependencies holding between them can still be expressed; albeit in an indirect fashion. In particular, informational dependencies between f-structures and s-structure can be specified through inversion and composition of the mappings ϕ and σ. Thus, given σ and ϕ both having a domain of c-structure nodes, we can define $\sigma' = \sigma \circ \phi^{-1}$ as a correspondence between functional and semantic structures. As we illustrate below, the fact that functional subjects (e.g. *John*) supply the semantic information associated with the agent-role of active agentive verbs (e.g. *run*) can be expressed in terms of σ'.[1]

Although grammatical relations represented in the f-structure are commonly aligned with semantic argument roles as in this example, this is not always the case. One instance of divergence between f- and s-structures arises in the analysis of adverbs. In order to provide an elegant account of subject-verb agreement as well as other phenomena, ϕ associates the same f-structure to both the S and VP nodes (as in Figure 2). However, there is a clear semantic distinction between adverbial phrases which

[1]Since the f-structure projection ϕ is many-to-one, its inverse is a more general relation, not a single-valued function. This feature among others moves us outside the standard bounds of unification towards more general constraint-programming (Jaffar and Lassez 1987).

Figure 2: Functional and semantic structures



modify sentences (e.g. *On several occasions*) and adverbials that modify verb-phrases (e.g. *slowly*) (see Stalnaker and Thomason 1973). If s-structures are projected directly from the f-structure, then the usual alignment of functional and semantic relations together with the conflation of information used for subject-verb agreement would obliterate the distinctions needed to characterize these differences. By projecting s-structure from the c-structure, the adverb distinctions can be maintained, while the convergence of f-structural and semantic information can still be captured through the composite projection σ' defined above.

3 Projections and Co-Description

Previous proposals for semantic interpretation of LFGs took functional structures as input to the semantic component. The semantic representations were described based on an analysis of the level of functional structure (what we now call *description by analysis*). The first examples of this approach were provided by Halvorsen (1982) and Halvorsen (1983). There, LFGs were interpreted with four interpretation principles which applied in any order to f-structure configurations that matched the pattern specified in the interpretation principle. The patterns picked out semantically significant aspects of the f-structure. These interpretation principles licensed the introduction of a set of semantic equations. The complete set of semantic equations had been found when all the semantically significant f-structure configurations had been matched by an interpretation principle. The semantic equations could be solved using the same unification algorithms as in the construction of the functional structure itself. Other examples of semantic interpretation using the description-by-analysis approach are given by Frey and Reyle (1983) and Reyle (1987). They defined a set of *transfer rules* which mapped functional structures into semantic representations by means of a modified lambda-calculus. The mapping from c-structure to f-structure is the prototypical example of description-by-analysis—the functional description is produced by matching the context-free rules against the node-configurations in the c-structure.

Every interpretation scheme based on description-by-analysis requires that all semantically relevant information be encoded in the functional structure. This assumption rules out the possibility of writing semantic interpretation rules triggered by specific c-structure configurations unless otherwise unmotivated "diacritical" features are included in the f-structure. There are two reasons for this: (1) The connection between syntactic rules and interpretation principles is severed by stating the interpretation rules on f-structures; (2) No separate semantic rule language is provided.

Our proposal for semantic interpretation is not based on an analysis of the f-structure. Rather, it depends on *co-description*, on simultaneously associating functional and semantic constraints through a single analysis of the c-structure tree.

The objective of the theory of projections is to focus attention on the relationship between various aspects of linguistic form and the interpretation of utterances. It is useful to compare our prosposal to other approaches that use *distinguished attributes* in a single level of representation to combine syntactic and semantic information (e.g. FSTRUC and SITSCHEMA as in Fenstad et al (1985, 1987), or SYNTAX and SEMANTICS as in Karttunen (1986) and Pollard and Sag (1987)). Although equality over monadic functions or attribute-value unification are expressive enough formally to encode quite complex informational dependencies, they may not provide the best conceptual framework for understanding the connection between relatively independent modules. This requires a framework where modularity and interaction can comfortably coexist. The theory of projections is an attempt at providing such a framework.

Thus the present proposal is not a formal refutation of the approach to the specification of semantic structures in unification grammars using one or more distinguished attributes. The distinguished attributes approach can instead be viewed as an implementation technique for simple projection theories which do not utilize inversion and composition of functional mappings as we propose here.

4 Notational conventions

The point of departure for our semantic rule notation is the syntactic rule formalism of LFG. As originally formulated by Kaplan and Bresnan (1982), context-free phrase-structure rules are annotated with constraints that are instantiated to form functional descriptions, as illustrated in this simple S rule:[2]

$$\begin{array}{lll} S \rightarrow & NP & VP \\ & (\uparrow \texttt{SUBJ}){=}\downarrow & \uparrow{=}\downarrow \end{array}$$

In these annotations, $\uparrow$ and $\downarrow$ refer to functional structures corresponding to specific nodes of the phrase-structure tree. For a tree configuration matching the rule, $\uparrow$ denotes the f-structure that directly corresponds to the mother of the node that matches the rule-category it is annotated to, and $\downarrow$ denotes the f-structure corresponding to that rule-category. The annotation on the NP, for example, indicates that the SUBJ of the f-structure corresponding to the NP's mother, namely the S node, is the f-structure corresponding to the NP node. Kaplan (1987) gave a precise explication of this arrow notation in terms of the structural correspondence ϕ, the function $\mathcal{M}$ on c-structure nodes that takes a node into its mother, and a single special symbol $*$ that denotes the node matching the rule-element it is annotated to. With these primitives the symbol $\uparrow$ can be seen as a convenient abbreviation for the specification $\phi\mathcal{M}*$, $\downarrow$ abbreviates $\phi*$, and an equivalent formulation of the SUBJ constraint above is $(\phi\mathcal{M}* \texttt{ SUBJ}){=}\phi*$.

As discussed above, ϕ is a structural correspondence relating two syntactic levels of representation. The function σ is another correspondence, and in the present theory it maps between the set of c-structure nodes and the units of s-structure that characterize the content of an utterance, or, following Barwise and Perry (1983), the *described situation*. Along with the names of other correspondences mentioned by Kaplan (1987), the symbol σ is introduced into the vocabulary of our constraint language so that descriptions of semantic structure can be specified.[3]

If the f-structure and semantic structure are both considered projections of the c-structure, either ϕ or σ can be prefixed to any expression denoting a c-structure node to denote the corresponding f- or s-structure unit. The following are all well-formed expressions of the extended rule language: $\phi\mathcal{M}*$, $\phi*$, $\sigma\mathcal{M}*$, and $\sigma*$. $\mathcal{M}$, ϕ, and σ are all right-associative. Consequently, $(\sigma\mathcal{M} * \texttt{ ARG1})$ denotes the value of the ARG1 attribute in the semantic structure corresponding to the mother of the current node.

The inverse of the c-structure to f-structure projector, ϕ^{-1}, gives us the c-structure node, or set of c-structure nodes, corresponding to a given f-structure. ϕ^{-1} can be prefixed to any expression denoting a unit of f-structure. Thus $\phi^{-1}(\phi\mathcal{M} * \texttt{ SUBJ})$ denotes the set of nodes that map to the SUBJ function in the f-structure corresponding to the mother of the matching node. Reverting to the abbreviatory arrow convention, this expression can be simplified to $\phi^{-1}(\uparrow \texttt{ SUBJ})$.

The composition of the σ projection with the inverse of the ϕ correspondence can now be used to express the fact that the functional subject and the first argument (or agent) in an active sentence coincide even if the information about the subject/agent is scattered throughout the sentence, as is the case in sentences with extraposed relative clauses (*A man entered who was limping*). This is accomplished by letting the semantic structure of the agent correspond to the full set of c-structure nodes that map to the functional subject, e.g. $\phi^{-1}(\uparrow \texttt{ SUBJ})$ if $\uparrow$ denotes the f-structure of the main clause. The semantic structure corresponding to the subject of the mother's f-structure is then denoted by $\sigma\phi^{-1}(\uparrow \texttt{ SUBJ})$ or, using the definition of σ' above, $\sigma'(\uparrow \texttt{ SUBJ})$.

The name of a projection can occur in any equation whether in the lexicon or on phrase-structure rules. Where a particular equation occurs depends on the nature of the semantic generalization it expresses. The following is a typical lexical item with semantic equations:

$$\begin{array}{lll} \texttt{KICK} & \texttt{V} & (\uparrow \texttt{PRED}){=}\texttt{'KICK'} \\ & & (\sigma\mathcal{M}* \texttt{ REL}){=}\texttt{KICK} \\ & & (\sigma\mathcal{M}* \texttt{ ARG1}){=}\sigma'(\uparrow \texttt{SUBJ}) \\ & & (\sigma\mathcal{M}* \texttt{ ARG2}){=}\sigma'(\uparrow \texttt{OBJ}) \end{array}$$

This lexical entry contains two kinds of equations. First, there is a pure functional description:

[2] In the following we will refer to these constraints as equations. This should not obscure the fact that LFG permits constraints other than purely equational ones to be expressed.

[3] We emphasize that we could also have chosen another formal configuration in which semantic structures correspond to some other level of representation (e.g. f-structure, as suggested by Halvorsen 1987 and Kaplan 1987). Which configuration offers the best account of syntax/semantics interactions depends in part on what the linguistic facts are. The adverb facts briefly discussed above seem to favor σ as a mapping from c-structure.

$(\uparrow \texttt{ PRED}) = \texttt{'KICK'}$

Second, there are inter-module equations constraining the relationship between the semantic interpretation and the functional properties of the phrase. The inter-modular constraint

$(\sigma\mathcal{M}* \texttt{ ARG1})=\sigma'(\uparrow \texttt{ SUBJ})$

asserts that the agent argument role (labeled ARG1) of the *kick* relation is filled by the interpretation which, by force of other equations, is associated (indirectly through the nodes given by ϕ^{-1}) with the functional subject.

The extended rule language permits a third type of equation as well. This is the pure semantic equation. The lexical entry for the past tense marker illustrates this type of equation (cf. any σ-equation in the entry below, which is adapted from Fenstad et al (1987)).

-ED	AFF	$(\uparrow \texttt{ TENSE})=\texttt{PAST}$
		$(\sigma\mathcal{M}* \texttt{ LOC})=\sigma*$
		$(\sigma* \texttt{ IND ID})=\texttt{IND-loc}$
		$(\sigma* \texttt{ COND RELATION})=\prec$
		$(\sigma* \texttt{ COND ARG}'1)=(\sigma* \texttt{ IND})$
		$(\sigma* \texttt{ ARG}'2)=\texttt{LOC-D}$
		$(\sigma* \texttt{ POL})=1$

The analyses this notation makes possible exhibit several improvements over earlier treatments.

First, we can now explicate in equational terms the mixture of functional and semantic information implicit in LFG's semantic forms, where the associations between functional entities (SUBJ, OBJ etc.) and semantic roles are given by the order of arguments, as in the example below.

$\texttt{'kick} < (\uparrow \texttt{ SUBJ}),(\uparrow \texttt{ OBJ}) > \texttt{'}$

Our equational treatment of the functional and the semantic information that semantic forms encode consigns the different types of information to separate levels of representation while explicitly marking the cross-level dependencies.

Second, the correct assignment of interpretations to roles in passive constructions and other constructions involving lexical rules is also achieved without further stipulations and without modification of the lexical rules. The original version of the passive rule in LFG (Kaplan and Bresnan 1982)

$\texttt{SUBJ} \rightarrow \texttt{BY-OBJ};$
$\texttt{OBJ} \rightarrow \texttt{SUBJ}$

can be applied directly to the lexical form for *kick* with the desired result. This contrasts with the proposals of Cooper (1985; 1986) and Gawron (1986), where more elaborate mechanisms are introduced to cope with the effects of relation changing rules, and with the analysis in Barwise and Perry (1983), where no allowances are made for the effects of such rules.

As a final example of the beneficial effects of our extended rule language, we examine a storage-free analysis of quantifier scope ambiguities.

5 The treatment of scope

The preferred mechanism for scope analysis both in formal linguistic treatments and in natural language systems based on them has long been the so-called Cooper-storage (Cooper 1976).[4] This approach combines a compositional interpretation of a syntax tree with the ability to pass around in the tree a pair consisting of the interpretation of the quantified noun phrase and an indexed variable which is eventually bound by the quantifier in the noun phrase. Different scope interpretations are achieved by discharging the quantifier expression at different points in the process of putting together the interpretation of the rest of the sentence. The theory of interpretation and the rule-language which is presented here makes it possible to handle scope ambiguities without recourse to a storage mechanism.

The generalization underlying the use of Cooper-storage is that the quantifier associated with a quantified noun phrase can take scope at any one of a number of different levels in the sentence. This is exemplified by sentences like (3), which has two interpretations, one non-specific (4) and one specific (5), depending on the scope of the existential quantifier relative to the verb *try*.

$$\text{Bill is trying to find a pretty dog} \tag{3}$$

$$[\text{Bill is trying } \exists x[\text{Bill find a pretty dog}(x)]] \tag{4}$$

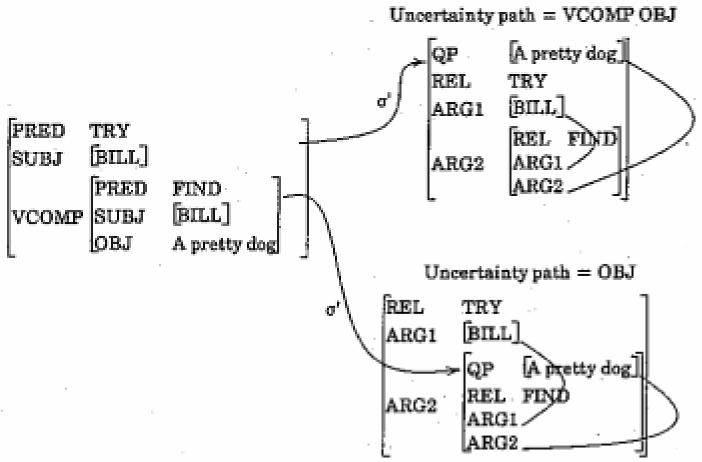
$$\exists x[\text{Bill is trying [Bill find a pretty dog}(x)]] \tag{5}$$

In our analysis, these semantic facts are captured by having a single functional structure associated with two semantic structures, as shown in Figure 3.

The quantifier phrase, QP, can occur on two levels, either in the semantic structure corresponding to the nodes that map to the f-structure VCOMP, or in the semantic structure corresponding to the top level sentence. Putting these observations in more general terms, we see that a noun phrase NP_k in a complement C_j, where j indicates the level of embedding of C_j, can be quantified

[4] A mechanism similar to Cooper-storage for use in interpretation of quantifiers in the LUNAR system was proposed much earlier by Woods (Woods et al 1972).

Figure 3: Functional-structure and semantic structure for quantifier scope ambiguities



into the semantic structure corresponding to the top-level sentence, S_0, or into the semantic structure corresponding to any of the verbal-complements, VCOMPs, induced by embedded complement-phrases, C_i, where $i < j$. Stated in this fashion one can see that the problem of scope ambiguity can be analyzed by a generalization of the "functional uncertainty" notation (Kaplan and Maxwell 1988).

Kaplan and Maxwell (1988) introduced what we can call outside-in uncertainty. Function-application expressions of the form $(f\ \alpha)$ were used to designate a functional structure reachable from f via any path drawn from the regular set α. For example, $(\uparrow \texttt{COMP}^* \texttt{OBJ})$ could be used in the characterization of object relative clauses. We generalize this notion to allow inside-out uncertainty. We let expressions of the form $(\alpha\ f)$ denote functional structures from which f is reachable over some path in α. Thus the expression $(\texttt{VCOMP}^* \texttt{ OBJ } \uparrow)$ denotes some f-structure from which there is a path to $\uparrow$ consisting of any number of VCOMPs followed by OBJ.

Now the generalization about scope possibilities stated above can be captured by adding the following annotation to any N which is the head of a quantified NP:[5]

$$(\sigma'(\texttt{VCOMP}^* \texttt{ GF } \uparrow)\ \texttt{QP}) = \sigma\mathcal{M}*$$

The uncertainty expression $(\texttt{VCOMP}^* \texttt{ GF } \uparrow)$ denotes functional structures from which the f-structure of the NP containing the N can be reached following a path consisting of zero or more VCOMPs followed by a GF. In other words, for every C_i within which the NP is embedded, the f-structure, f_i, corresponding to this C_i is picked out. The complete annotation states that the semantic structure of the NP, $\sigma\mathcal{M}*$, can be the value of the QP attribute of the semantic structure corresponding to any such f_i through the σ' mapping, $(\sigma' f_i\ \texttt{QP})$.

Since our language for the statement of semantic rules is an extension of the language of functional descriptions, this descriptive possibility is immediately available to us. Functional uncertainty also has an efficient implementation based on an incremental decomposition of finite-state machines (Kaplan and Maxwell 1988).

6 Summary

We have shown how to formulate a semantic interpretation mechanism for LFGs based on structural correspondences and a theory of projections. We have also utilized a simple extension of the language for functional constraints to permit the treatment of multiple projections. While previous proposals have taken functional structures as input to the semantic interpretation component and thus have required all semantically relevant information to be reflected in the functional structure, our proposal uses codescription to coordinate the functional and semantic properties of a construction without imposing this requirement. This allows us to simplify functional representations by eliminating functionally irrelevant but semantically significant material. It also puts at our disposal the full power of the rule language of LFG, including functional uncertainty, and this, in its turn, makes it possible to formulate novel semantic analyses, such as a treatment of quantifier scope ambiguities that avoids the use of any storage mechanism.

[5] Here GF stands for the set of governable grammatical functions (e.g. SUBJ, OBJ).

REFERENCES

[1] J. Barwise and J. Perry. *Situations and Atti-*

tudes. MIT Press/Bradford Books, Cambridge, Mass., 1983.

[2] R. Cooper. Meaning representation in Montague grammar and situation semantics. In *Proceedings from Workshop on Theoretical Issues in Natural Language Processing*, pages 28–30, Halifax, Nova Scotia, May 28–30 1985.

[3] R. Cooper. Verb second – predication or unification. Delivered at the Workshop on Comparative Germanic Syntax, Reykjavik, June 1986.

[4] R. Cooper. *Montague's Semantic Theory and Transformational Syntax.* PhD thesis, Department of Linguistics, University of Massachusetts, 1976.

[5] J. E. Fenstad, P-Kr. Halvorsen, T. Langholm, and J. van Benthem. *Equations, Schemata And Situations: A Framework for Linguistic Semantics.* Technical Report 29, CSLI, Stanford University, 1985.

[6] J. E. Fenstad, P-Kr. Halvorsen, T. Langholm, and J. van Benthem. *Situations, Language and Logic.* Volume 34 of *Studies in Linguistics and Philosophy*, D. Reidel, Dordrecht, 1987.

[7] W. Frey and U. Reyle. A PROLOG implementation of lexical functional grammar. In *Proceedings from the Eighth International Joint Conference on Artificial Intelligence*, Vol. II, Karlsruhe, 1983.

[8] M. Gawron. Types, contents and semantic objects. *Linguistics and Philosophy*, 9(4):427–476, 1986.

[9] P-Kr. Halvorsen. Order-free semantic composition and lexical-functional grammar. In Ján Horecký, editor, *COLING 82: Proceedings of the Ninth International Conference on Computational Linguistics*, pages 115–120, North Holland Publishing Company, Amsterdam, 1982.

[10] P-Kr. Halvorsen. Semantics for lexical-functional grammars. *Linguistic Inquiry*, 14(3):567–613, 1983.

[11] P-Kr. Halvorsen. Situation semantics and semantic interpretation in constraint-based grammars. In *Proceedings from the International Conference on Fifth Generation Computer Systems, FGCS-88*, Tokyo, 1988. (Extended and revised version of [12]).

[12] P-Kr. Halvorsen. Situation semantics and semantic interpretation in constraint-based grammars. Technical report, CSLI-TR-101, Stanford University, 1987.

[13] J. Jaffar and J-L. Lassez. Constraint logic programming. In *Proceedings from the Fourteenth POPL*, pages 111–119, Munich, 1987.

[14] M. Kameyama. Precedence conditions on overt and zero nominals Unpublished paper, Microelectronics and Computer Technology Corporation, Austin, Texas.

[15] R. Kaplan. Three seductions of computational psycholinguistics. In P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, and P. Bennett, editors, *Linguistic Theory and Computer Applications*, Academic Press, 1987.

[16] R. Kaplan and J. Bresnan. Lexical-functional grammar: a formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, Mass.: MIT Press, Cambridge, 1982.

[17] R. M. Kaplan and J. Maxwell III. An algorithm for functional uncertainty. In *Proceedings from COLING88*, Budapest, 1988.

[18] L. Karttunen. *D-PATR: A Development Environment for Unification-Based Grammars.* Report No. CSLI-86-61, Stanford University, 1986.

[19] M. Kay. Functional grammar. In *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Socielty*, University of California, Berkeley, California, 1978.

[20] R. Montague. Universal grammar. *Theoria*, 36:373–398, 1970. Reprinted in R. Thomason. (ed.) Formal Philosophy. Selected Papers of Richard Montague.

[21] F. Pereira and D. Warren. Definite clause grammars for language analysis–a survey of the formalism and a comparison with transition networks. *Artificial Intelligence*, 13:231–278, 1980.

[22] C. Pollard and I. A. Sag. *Information-Based Syntax and Semantics. Vol. I: Fundamentals.* Volume 13 of *CSLI Lecture Note Series*, University of Chicago Press, 1987.

[23] U. Reyle. Compositional semantics for LFG. In C. Rohrer and U. Reyle, eds. *Natural Language Parsing and Grammatical Theory*, Reidel, 1987.

[24] S. Shieber. The formalism and implementation of PATR-II. In Barbara Grosz et al, editor, *Research on Interactive Acquisition and Use of Knowledge. Final Report*, SRI International, 1983.

[25] R. Stalnaker and R. Thomason. A semantic theory of adverbs. *Linguistic Inquiry*, IV(2), 1973.

[26] W. A. Woods, R. M. Kaplan, and B. Nash-Webber. *The Lunar Sciences Natural Language Information System: Final Report.* Technical Report 2378, Bolt, Beranek and Newman Inc., Cambridge, Ma., June 1972.