Categorial Semantics for LFG

Mary Dalrymple
Xerox PARC
Palo Alto, CA 94304 USA
dalrymple@parc.xerox.com

1 Introduction

A categorial semantics for Lexical-Functional Grammar provides a means for semantic interpretation of sentences of natural language that is appropriately constrained both syntactically and semantically. The f-structure of LFG provides a cross-linguistically uniform format for representing syntactic information; constraining a derivation with respect to the f-structure rather than a phrase structure tree allows reference to relevant functional syntactic information without requiring construction of a phrase structure tree whose form is (often dubiously) motivated on semantic grounds. Additionally, a categorial semantics constrains semantic derivations appropriately, obviating the need for an appeal to well-formedness conditions on the resulting semantic representation.

2 Previous Work

Most semantic analyses appeal to syntactic constraints on semantic derivations. In particular, many analyses assume that such syntactic constraints are statable in terms of phrase structure tree configurations (Montague, 1974). However, it is well-known that a variety of phrase structure configurations can express the same syntactic predicate-argument relations within and across languages (Kaplan and Bresnan, 1982); thus, syntactic constraints on semantic derivations are better expressed at a level at which the relevant syntactic information is expressed more uniformly. Such a level is the *f-structure* of LFG.

Halvorsen (1983) first provided a theory of semantic interpretation for LFG in which semantic interpretation rules are related to the f-structure. His system involves an intermediate level of representation, the 'semantic structure', which is represented as a directed graph (like the f-structure). Translation rules map from f-structures to semantic structures, and these structures are then interpreted (or translated into a formula of intensional logic).

The approach to be presented here also relies on f-structure configurations to provide syntactic constraints on categorial semantic derivations. However, an intermediate level of semantic representation such as Halvorsen's semantic structure is not introduced. In the categorial semantic framework developed by Fernando Pereira (Pereira, 1990; Pereira and Pollack, 1991; Pereira, 1991), syntactic structures are directly associated with interpretations (or their types), and syntactic configurations license the combination of these interpretations in a semantic derivation. On this approach, 'logical forms' are not viewed as manipulable syntactic objects; instead, a logical formula is simply a graphical representation of a meaning that is lexically provided or that is the outcome of a semantically justified derivation. In this, the approach differs from other recent approaches to semantic interpretation in LFG (Halvorsen and Kaplan, 1988), in which the interpretation of an f-structure is represented as a directed graph, and semantic derivation proceeds principally by unification of semantic representations. As a consequence, these approaches require constraints on semantic derivations to be stated as well-formedness conditions on semantic representations, contrary to the commonly-held goal of dispensability of logical form.

To illustrate a categorial semantic analysis within LFG, I will provide a small fragment of syntactic and semantic rules of English; the fragment contains rules for quantified noun phrases, nominal modification, and clauses headed by transitive and intransitive verbs. Many of these rules are modifications and extensions of rules originally described in Pereira (1990), though Pereira's system appeals to phrase structure configurations rather than f-structures to constrain semantic derivations; in particular, the rules Pereira provides for quantifiers and relative clauses have direct counterparts in the set of rules to be described below.

3 Sentence Interpretation

A sentence such as (1) has the interpretation given in (2):[1]

(1) John crashed.

(2) $crash(john)$

This interpretation is the outcome of a derivation according to a set of rules to be described below. Some of the rules must be licensed by particular f-structure configurations, while some are unrestricted in their applicability. Example 1 has the following f-structure:

(3) $\begin{bmatrix} \text{PRED} & \text{`crash}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & [\text{PRED `John'}] \end{bmatrix}$

Annotated phrase structure rules like the following are assumed: [2]

$$\begin{array}{lccc} \text{S} & \longrightarrow & \text{NP} & \text{VP} \\ & & (\uparrow \text{SUBJ}) = \downarrow & \uparrow = \downarrow \end{array}$$

$$\begin{array}{lccc} \text{VP} & \longrightarrow & \text{V} & (\text{NP}) \\ & & \uparrow = \downarrow & (\uparrow \text{OBJ}) = \downarrow \end{array}$$

Notice that these phrase structure rules encode only syntactic information. No semantic information or constraints are required.

The lexical entries involved in the derivation of sentence (1) are:

$$\begin{array}{lll} \textit{John} & \text{NP} & (\uparrow \text{PRED}) = \text{`John'} \\ & & \uparrow_\sigma = [\emptyset \vdash j] \\ \textit{crashed} & \text{V} & (\uparrow \text{PRED}) = \text{`crash}\langle\text{SUBJ}\rangle\text{'} \\ & & (\uparrow \text{TENSE}) = \text{PAST} \\ & & (\uparrow \text{PRED})_\sigma = [\emptyset \vdash \lambda x.crash(x)] \end{array}$$

The notation f_σ stands for the interpretation of an f-structure f, often referred to as the semantic *projection* of f (Kaplan, 1987; Halvorsen and Kaplan, 1988). The interpretation for any f-structure f is a *sequent*:

(4) $f_\sigma = [a \vdash M]$

The sequent '$[a \vdash M]$' is a pair consisting of a set of *assumptions* a, somewhat analogous to a 'quantifier store' (Cooper, 1983), and a matrix term M in which free variables introduced by the assumptions in a may occur (Pereira, 1990; Pereira, 1991; Dalrymple et al., 1991). In the following, I will speak of such expressions as introducing the meaning M under the assumptions in a.

I assume a fixed order of application of the meaning of a verb to its semantic arguments, with the order determined by the syntax (though this assumption is not crucial to the analysis). Arguments are applied in the following order:[3]

(1) Obliques
(2) OBJ2
(3) OBJ
(4) SUBJ

The PRED of the f-structure of an active verb such as *own* will, then, be associated via the σ mapping with the following interpretation:

(5) $\lambda y.\lambda x.own(x,y)$

Notice that the verb is required to combine with the object first, and then the subject, in accordance with the argument ordering given above. For a passive verb, the ordering will be reversed. For the passive verb *(be) owned*, the order will be:

(6) $\lambda x.\lambda y.own(x,y)$

Here, the verb combines first with the oblique by-phrase, then with the subject.

The rule for interpreting an f-structure for a clause headed by an intransitive verb is:[4]

(7) Clause with intransitive verb:

[1] I will ignore tense and aspect in the representation of sentence meanings.

[2] See Bresnan (1982) for an explication of the relation between c-structure and f-structure and the notation commonly used to represent that relation.

[3] This order of application was also proposed by Dowty (1982), and is reminiscent of the obliqueness ordering for arguments in HPSG (Pollard and Sag, 1987).

[4] This rule should apply when f has a PRED and a SUBJ, but no other governable grammatical functions; it should not apply if the verb is transitive and there is a SUBJ and an OBJ, although f is unifiable with the f-structure of a transitive verb as well as an intransitive one. There are several ways of ensuring the needed result: the valence of the verb can be reflected in its semantic type; f-structures can be typed, with this rule applying only to intransitive f-structures (Zajac and Emele, 1990); or the PRED and its arguments can be separately specified, with the arguments of the PRED specified as a list which can be matched against, as in recent work by John Maxwell, Ron Kaplan, and others.

$$\left.\begin{array}{l} f: \begin{bmatrix}\text{PRED} & \text{P}\\ \text{SUBJ} & \text{S}\end{bmatrix}\\ \text{P}_\sigma = [a_P \vdash m_P]\\ \text{S}_\sigma = [a_S \vdash m_S]\end{array}\right\} \Rightarrow$$

$$f_\sigma = [a_P \cup a_S \vdash m_P\,(m_S)]$$

The derivation of the meaning f_σ of an f-structure f with a PRED and SUBJ proceeds by applying the meaning of the PRED to the meaning of the SUBJ. The associated assumption set is the union of the assumptions from the PRED and the SUBJ. The f-structure for sentence 1 licenses the following derivation and provides the expected meaning (under a null assumption set):

(8) $$f_1: \begin{bmatrix}\text{PRED } f_2\text{:`crash}\,\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ } f_3\text{:}\big[\text{PRED `John'}\big]\end{bmatrix}$$

Lexically specified meanings:

$(f_2)_\sigma = [\emptyset \vdash \lambda x.crash\,(x)]$
$(f_3)_\sigma = [\emptyset \vdash j]$

By rule 7:

$(f_1)_\sigma = [\emptyset \cup \emptyset \vdash \lambda x.crash\,(x)(j)]$
$\quad = [\emptyset \vdash crash\,(j)]$

4 Quantification

Sentence 9 contains a quantified noun phrase and has the meaning represented in (10):

(9) Every car crashed.

(10) $every\,(\lambda y.car\,(y),\ \lambda x.crash\,(x))$

This sentence has the f-structure shown in (11), constructed on the basis of the lexical entries below:

(11) $$\begin{bmatrix}\text{PRED `crash}\,\langle\text{SUBJ}\rangle\text{'} & \\ \text{TENSE PAST} & \\ \text{SUBJ} & \begin{bmatrix}\text{SPEC} & \big[\text{PRED `every'}\big]\\ \text{PRED `car'} & \end{bmatrix}\end{bmatrix}$$

every	DET	$(\uparrow \text{PRED}) = \text{`every'}$
		$\uparrow_\sigma = [\emptyset \vdash every]$
car	N	$(\uparrow \text{PRED}) = \text{`car'}$
		$(\uparrow \text{PRED})_\sigma = [\emptyset \vdash \lambda y.car\,(y)]$

The type of the quantifier *every* is the familiar generalized quantifier type $(e \rightarrow t) \rightarrow (e \rightarrow t) \rightarrow t$: quantifiers are functions from properties to properties, yielding a truth value. The following schematic rule is necessary to interpret quantified noun phrases:

(12) Quantified noun phrase, preliminary version (handles unmodified nominals only):

$$\left.\begin{array}{l} f: \begin{bmatrix}\text{SPEC} & \text{S}\\ \text{PRED} & \text{P}\end{bmatrix}\\ \text{S}_\sigma = [a_S \vdash m_S]\\ \text{P}_\sigma = [a_P \vdash m_P]\end{array}\right\} \Rightarrow$$

$$f_\sigma = [a_S \cup a_P, quant\,(m_S, x, m_P) \vdash x]$$

The notation 'a, A' represents the set a plus the singleton A. By this rule, a *quant* assumption is added to the assumption set for the noun phrase. The *quant* assumption acts like an element in a Cooper store, keeping together the information associated with the quantified noun phrase. m_S in the *quant* assumption is the meaning of the specifier (here, *every*); x is the variable introduced by the quantifier as the meaning of the quantified noun phrase; and m_P is the meaning of the PRED, which will form the first argument of the generalized quantifier *every* when quantifier discharge takes place. The derivation of the meaning of sentence 9 according to the rules given thus far proceeds as follows:

(13) $$f_1: \begin{bmatrix}\text{PRED } f_2\text{ :`crash}\,\langle\text{SUBJ}\rangle\text{'} & \\ \text{TENSE PAST} & \\ \text{SUBJ } f_3: & \begin{bmatrix}\text{SPEC } f_4: \big[\text{PRED `every'}\big]\\ \text{PRED } f_5\text{ :`car'}\end{bmatrix}\end{bmatrix}$$

Lexically specified meanings:

$(f_2)_\sigma = [\emptyset \vdash \lambda x.crash\,(x)]$
$(f_4)_\sigma = [\emptyset \vdash every]$
$(f_5)_\sigma = [\emptyset \vdash \lambda y.car\,(y)]$

By rule 12, 'Quantified noun phrase':

$(f_3)_\sigma = [\{quant\,(every, x, \lambda y.car\,(y))\} \vdash x]$

By rule 7, 'Clause with intransitive verb':

$(f_1)_\sigma =$
$\quad [\{quant\,(every, x, \lambda y.car\,(y))\} \vdash crash\,(x)]$

According to these rules, the meaning for f-structure f_1 is a meaning under an assumption about the variable x. The meaning of f_1 without assumptions is obtained by discharging the (sole) quantifier assumption in the assumption set. The quantifier discharge rule relates a sequent and a syntactic licensing environment to a new sequent:

(14) Quantifier discharge:

$disch(f,\ [a, quant\,(m_S, x, m_P) \vdash SCOPE\!:\!t]) = [a \vdash m_S\,(m_P, \lambda x.SCOPE)]$

Conditions on f: none

By this discharge rule, the *quant* assumption is removed from the assumption set, the variable x introduced by the quantifier assumption is abstracted out of the scope $SCOPE$ (required to be of type t), and the quantifier is applied to its scope. The syntactic licensing environment is the f-structure f. In this rule, f is unconstrained; there are no conditions on f. This means that the quantifier discharge rule has an unrestricted syntactic licensing condition. A quantifier may scope over any syntactic constituent, as long as it is of the correct semantic type.[5]

To interpret sentence 9, *disch* can now be applied to the sequent $(f_1)_\sigma$ associated with the f-structure f_1:

$disch\,(f_1, [\{quant(every, x, \lambda y.car(y))\} \vdash crash(x)]) = [\emptyset \vdash every\,(\lambda y.car\,(y), \lambda x.crash(x))]$

The result is the meaning of f_1 with all assumptions discharged. I will assume that what is generally referred to as the 'meaning' of an utterance is the meaning obtained when all assumptions have been discharged.

In general, assumptions may be discharged after any application of a functor to an argument, as long as the syntactic environment for assumption discharge has been met. Thus, a predicate *apply* can be defined:

(15) $apply\,(f, [a_F \vdash Fun], [a_A \vdash Arg]) \overset{def}{=} discharge(f, [a_F \cup a_A \vdash Fun\,(Arg)])$

apply operates on sequents in a syntactic licensing environment f. $discharge(f, S)$ is the result of applying any number of discharge (*disch*) rules licensed by the syntactic configuration f to S. (Note that *apply* is not a function, since the result of *apply* depends on the number and the choice of assumptions to be discharged.) By this function application rule for sequents, then, the meaning of the functor is applied to the meaning of the argument; the union of the functor assumptions and the argument assumptions is taken; and some number of discharge rules may be applied. This definition of *apply* will be used in the following to apply predicates to their arguments and to permit subsequent assumption discharge.

Given this new definition of *apply*, interpretation rule 7 for clauses headed by intransitive verbs can be restated:

(16) Clause with intransitive verb:

$$f: \begin{bmatrix} \text{PRED} & \text{P} \\ \text{SUBJ} & \text{S} \end{bmatrix} \Rightarrow f_\sigma = apply\,(f, \text{P}_\sigma, \text{S}_\sigma)$$

The interpretation for an f-structure f, representing an unmodified clause with an intransitive verb, is obtained by applying the PRED P to the SUBJ S in the syntactic licensing environment f. In general, f_σ will constitute an assignment of f to a sequent that satisfies the constraints given by the lexical entries and the rules of interpretation.

It should be noted that rule 16 is incomplete in providing interpretations only for sentences not involving adverbial modification; an analysis of adverbials, though straightforward in this framework, will not be provided here.

5 Nominal modification

Rule 12 for the interpretation of quantified noun phrases is incomplete, since it applies only to unmodified nominals. Consider sentence (17), its f-structure, displayed in Figure 1, and its meaning, (18):

(17) Every car that John owned crashed.

(18)

$$every\,(\lambda x.car\,(x) \wedge own\,(j, x),\ \lambda y.crash\,(y))$$

These lexical entries are necessary:

that	CMP	$(\uparrow \text{PRED}) = \text{PRO}$
		$(\uparrow \text{TYPE}) = \text{REL}$
		$\uparrow_\sigma = [rel\,(x) \vdash x : e]$
owned	V	$(\uparrow \text{PRED}) =$ 'own ⟨SUBJ, OBJ⟩'
		$(\uparrow \text{TENSE}) = \text{PAST}$
		$(\uparrow \text{PRED})_\sigma = [\emptyset \vdash \lambda y.\lambda x.own(x, y)]$

Syntactically, a relative clause contains a fronted constituent (a TOPIC; see Bresnan and Mchombo (1987)) which is related to a gapped position in the sentence. This fronted constituent contains a relative pronoun or *that*. The relative pronoun may be deeply embedded in the fronted

[5]Here I will not discuss conditions on preferred scopes for quantifiers (such as the tendency for the quantifier *each* to outscope other quantifiers, or for quantifiers to scope inside the clause in which they appear).

constituent, as in the case of pied piping. Semantically, the interpretation of a relative clause is the property obtained when the position filled by the relative pronoun is abstracted out. For example, here are some relative clauses with a rough representation of their meanings:

(19) a. (the man) that I saw: $\lambda x.saw(I, x)$

b. (the man) whose brother's car I drove: $\lambda x.drove(I, x\text{'s } brother\text{'s } car)$

I assume that relative pronouns such as *that* or *whose* introduce a variable under a *rel* assumption which is abstracted out in the course of the derivation. The interpretation of a relative clause is obtained by a rule allowing the discharge of the *rel* assumption associated with the relative pronoun (and possibly other assumptions as well):

(20) Relative clause interpretation:

$$f : \begin{bmatrix} \text{TOPIC} & \text{TOP} \\ \text{REL} & \text{R} \end{bmatrix} \Rightarrow f_\sigma = discharge(f, \text{R}_\sigma)$$

The *rel* discharge rule applies only under syntactically licensed conditions:

(21) Relative clause assumption discharge:

$$disch(f,\ [a_{REL}, rel(x) \vdash REL{:}t]) = [a_{REL} \vdash \lambda x.REL]$$

Conditions:

$$f : \begin{bmatrix} \text{TOPIC} & \text{TOP} \\ \text{REL} & \text{R} \end{bmatrix}$$

$$(f\ \text{TOPIC GF}*) = relp : \begin{bmatrix} \text{PRED} & \text{PRO} \\ \text{TYPE} & \text{REL} \end{bmatrix}$$

$$relp_\sigma = [rel(x) \vdash x : e]$$

The relative pronoun must appear in the fronted TOPIC constituent. This is indicated in the second condition by the regular expression TOPIC GF*; this expression involves *functional uncertainty* (Kaplan and Maxwell, 1988) and requires that the relative pronoun *relp* must appear at the end of a path that is a member of the language described by the regular expression. Here, the path expression does not constrain where the relative pronoun may be found within the fronted constituent (GF* is a sequence of zero or more grammatical functions); a more complete syntactic analysis of relative clauses would constrain the path appropriately. The result of the application of rule 21 is that the variable introduced by the relative pronoun is abstracted out.

The value of the MODS attribute is a set of f-structures, interpreted according to the following rule:

(22) The semantic value of a set of f-structures is the set of corresponding sequents. If F is a set of f-structures:

$$F_\sigma = \{f_\sigma \mid f \in F\}$$

The rule for the interpretation of quantified noun phrases with nominal modification is given in Figure 2. According to this rule, the derivation of the meaning of a quantified noun phrase proceeds by introducing a variable (x in Figure 2) under a *quant* assumption, consisting of the meaning of the specifier of the noun phrase, the variable, and the quantifier restriction REST.

Recall the definition of *apply* given in 15:

(23) $$apply(f, [a_F \vdash Fun], [a_A \vdash Arg]) \overset{def}{=} discharge(f, [a_F \cup a_A \vdash Fun(Arg)])$$

This definition will now be extended so *apply* can take a set of sequents as its argument. The result is a set of sequents:

(24) $$apply(f, Set, Arg) \overset{def}{=} \{s \mid Fun \in Set \wedge s = apply(f, Fun, Arg)\}$$

The function *conj* is defined as the conjunction of a set of sequents; the matrices of the sequents are conjoined and the union of the assumptions is taken:

(25) $$conj(S) \overset{def}{=} [\bigcup_{[a \vdash M] \in S} a \vdash \bigwedge_{[a \vdash M] \in S} M]$$

REST, then, is the conjunction of the result of applying the PRED meaning and the meaning of each of the modifiers in MODS to the variable x.

Finally, a rule for the interpretation of a clause containing a transitive verb is also needed:

(26) Clause with transitive verb:

$$f : \begin{bmatrix} \text{PRED} & \text{P} \\ \text{SUBJ} & \text{S} \\ \text{OBJ} & \text{O} \end{bmatrix} \Rightarrow$$

$$f_\sigma = apply(f, apply(f, \text{P}_\sigma, \text{O}_\sigma), \text{S}_\sigma)$$

The interpretation of sentence 17 can now be derived; the derivation is sketched in Figure 3.

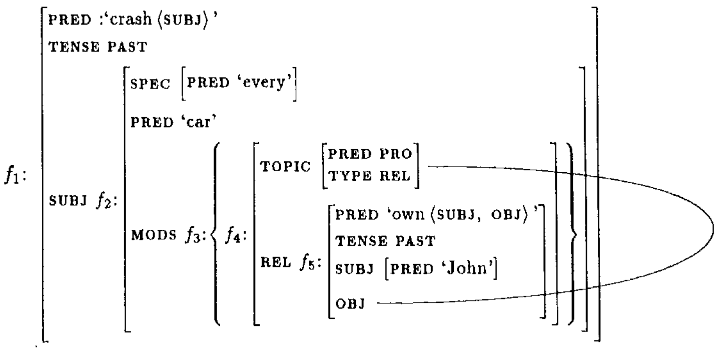

Figure 1: F-structure for *Every car that John owned crashed.*

$$\left.\begin{array}{l} f : \begin{bmatrix} \text{SPEC} & S \\ \text{PRED} & P \\ \text{MODS} & M \end{bmatrix} \\ S_\sigma = [a_S \vdash m_S] \\ [a_{REST} \vdash m_{REST}] = conj\,(apply\,(f, M_\sigma \cup \{P_\sigma\}, [\emptyset \vdash x])) \end{array}\right\} \Rightarrow$$

$$f_\sigma = [a_S \cup a_{REST},\ quant\,(m_S, y, \lambda x.m_{REST}) \vdash y]$$

Figure 2: Quantified noun phrase interpretation rule

By rule 26 for clauses with transitive verbs and the lexical entries given:
$f_{5_\sigma} = [\{rel(x)\} \vdash own(j, x)]$
By rule 20 for relative clauses, allowing the application of rule 21 for the discharge of relative pronoun assumptions:
$f_{4_\sigma} = [\emptyset \vdash \lambda x.own(j, x)]$
By rule 22 for interpreting sets of f-structures:
$f_{3_\sigma} = \{[\emptyset \vdash \lambda x.own(j, x)]\}$
By the rule for quantified noun phrases given in Figure 2 and the lexical entries for *every* and *car*:
$f_{2_\sigma} = [\{quant(every, y, \lambda x.car(x) \wedge own(j, x))\} \vdash y]$
By rule 16 for clauses with intransitive verbs:
$f_{1_\sigma} = [\{quant(every, y, \lambda x.car(x) \wedge own(j, x)\} \vdash crash(y)]$
By quantifier discharge rule 14:
$f_{1_\sigma} = [\emptyset \vdash every\,(\lambda x.car(x) \wedge own(j, x), \lambda y.crash(y))]$

Figure 3: Derivation of the interpretation of *Every car that John owned crashed.*

6 Conclusion

The small fragment of English presented above is easily extensible to handle other semantic phenomena, such as sentential modification. Constraining semantic derivations with respect to f-structures is preferable to the standard approach of using phrase structure trees, since f-structures need not be specifically tailored to solving the interpretation problem, but are motivated on independent grounds. The categorial semantics rules presented above provide an interpretation for f-structures directly, without the need for constructing an intermediate level of 'logical form'.

7 Acknowledgements

For invaluable assistance, I am very grateful to Fernando Pereira and Stuart Shieber. Many thanks are also due to John Lamping and Vijay Saraswat for their help and encouragement, and to Joan Bresnan, Ken Kahn, Ron Kaplan, Lauri Karttunen, Chris Manning, John Maxwell, and Annie Zaenen for helpful comments and discussion.

References

Joan Bresnan and Sam A. Mchombo. 1987. Topic, pronoun, and agreement in Chicheŵa. *Language*, 63(4):741–782.

Joan Bresnan, editor. 1982. *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge, MA.

Robin Cooper. 1983. *Quantification and Syntactic Theory*, volume 21 of *Synthese Language Library*. D. Reidel, Dordrecht.

Mary Dalrymple, Stuart M. Shieber, and Fernando C. N. Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4):399–452.

David R. Dowty. 1982. Grammatical relations and Montague Grammar. In Pauline Jacobson and Geoffrey K. Pullum, editors, *The Nature of Syntactic Representation*, pages 79–130. Reidel, Dordrecht.

Per-Kristian Halvorsen and Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 1116–1122, Tokyo, Japan. Institute for New Generation Systems.

Per-Kristian Halvorsen. 1983. Semantics for Lexical-Functional Grammar. *Linguistic Inquiry*, 14(4):567–615.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA.

Ronald M. Kaplan and John T. Maxwell. 1988. An algorithm for functional uncertainty. In *Proceedings of COLING-88*, volume 1, pages 297–302, Budapest.

Ronald M. Kaplan. 1987. Three seductions of computational psycholinguistics. In Peter Whitelock, Harold Somers, Paul Bennett, Rod Johnson, and Mary McGee Wood, editors, *Linguistic Theory and Computer Applications*, pages 149–188. Academic Press, London.

Richard Montague. 1974. *Formal Philosophy*. Yale University Press, New Haven. Richard Thomason, editor.

Fernando C. N. Pereira and Martha E. Pollack. 1991. Incremental interpretation. *Artificial Intelligence*, 50:37–82.

Fernando C. N. Pereira. 1990. Categorial semantics and scoping. *Computational Linguistics*, 16(1):1–10.

Fernando C. N. Pereira. 1991. Deductive interpretation. In *ESPRIT Symposium on Natural Language and Speech*. Brussels, November 1991.

Carl Pollard and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics*, volume 1. CSLI/University of Chicago Press, Stanford University. CSLI Lecture Notes, Number 13.

Rémi Zajac and Martin Emele. 1990. Typed unification grammars. In *Proceedings of COLING-90*, volume 3, pages 293–298, Helsinki.