Lars Ahrenberg
Department of Computer and Information Science
Linköping university
S-581 83 LINKÖPING
SWEDEN

A system for object-oriented dialogue in Swedish

1. Introduction

Two models for semantic interpretation that are currently being developed are constraint-based models (e.g. Fenstad et al. 1985, Halvorsen 1987) and models employing object-oriented knowledge representation formalisms such as frame systems or semantic networks (e.g. Bobrow&Webber, 1980; Sondheimer et al. 1984, Hirst 1987). This paper describes a dialogue system for Swedish in which I wish to combine features of both models. A large part of its linguistic knowledge, including semantic and pragmatic knowledge, is expressed as constraints. The semantic objects associated with linguistic expressions in the interpretation process are elements of a semantic network. Moreover, constraints and object descriptions play a major role also in the treatment of context.

The system, called FALIN, is being developed with the following purposes in mind: First, I want to investigate and demonstrate the possibilities of integrating syntactic, semantic and pragmatic knowledge in the interpretation process while still having that knowledge in separate modules. Second, I want to investigate the possibilities of treating dialogue phenomena such as indexicality and coherence within such a system. The results will be used in the design of a larger and more general system, LINLIN (the Linköping Natutral Language Interface; see Ahrenberg et al., 1986; Ahrenberg 1987).

As application I have chosen a simple drawing system where the human partner can draw, manipulate and ask questions about geometrical figures on a screen. The reason for this choice is that a visible domain makes it quite obvious whether the system is interpreting inputs correctly or not.

The system is still under construction. The morphological and syntactic components are in operation while the semantic components are still to be integrated in the system and the pragmatic components do not yet exist. In this paper I therefore concentrate on the problem of expressing and distributing semantic constraints, i.e. the rules that express the contributions of lexical and grammatical elements to the interpretation of the expressions of which they are part. First, I give a short overview of the system's architecture.

A system for object-oriented dialog in Swedish
Lars Ahrenberg
Proceedings of NODALIDA 1987, pages 96-106

2. System overview

The interaction with FALIN is restricted to simple sequences of the kind that can be expressed by finite automata. The basic sequences are, with the user's moves first: Question/Answer, Instruction/Execution and Assertion/Acceptance. The system may also ask questions of the user in the process of interpretation and inform him/her of problems with the input.

The system will always try to classify an input in terms of the illocutionary categories that are allowed. This classification to a large extent determines what actions the system will execute and what information it will present to the user.

The analyzer and the knowledge bases that it has access to are illustrated in figure 1.

The ***morph dictionary*** consists of a stem dictionary and a set of affix dictionaries, all of them compiled into letter trees. All entries are in their surface form (cf. Karlsson, 1986). Fixed expressions comprising more than one graphical word such as ***i dag***

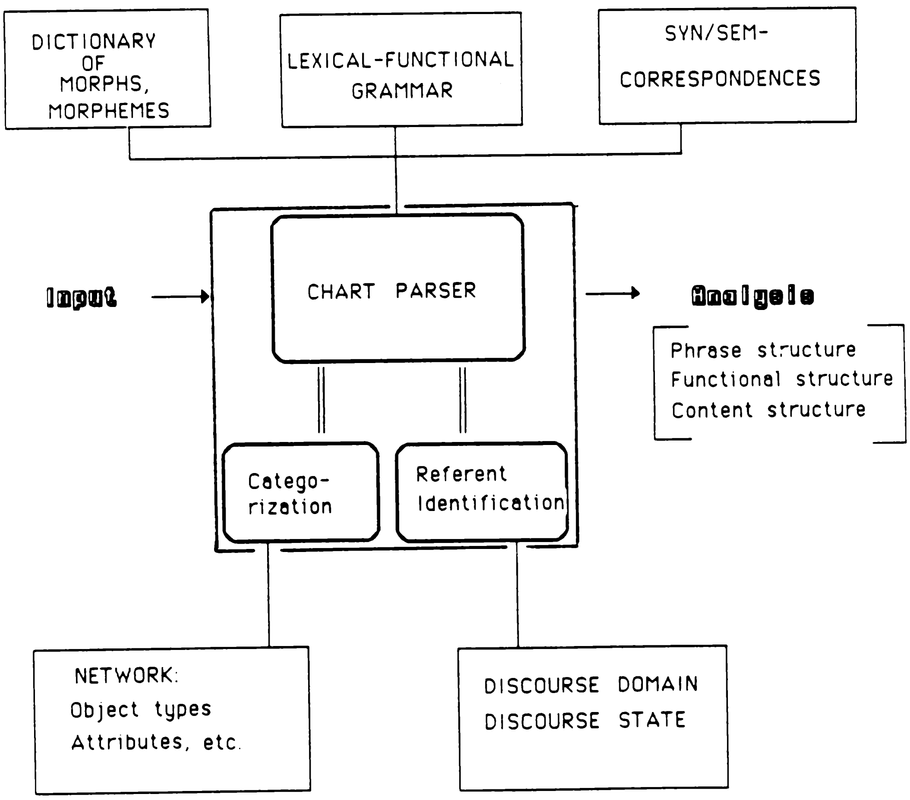

Figure 1: An overview of FALIN's analyzer.

(today) or ***hur många*** (how many) are included in the stem dictionary. The morph dictionary can be searched in different modes, e.g. one may choose to look for only one analysis of a given string, or all of them, or one may include or exclude the possibility of analyzing a word as a compound.

A morph in the dictionary is associated with a set of morphemes. With each morpheme there are associated a continuation class of suffix lexicons and, optionally, a flag guiding the continued search. A morpheme is either a stem or an affix. A stem morpheme carries information about syntactic category, morphosyntactic features and meaning. The meanings of a stem morpheme are collected in a *lexeme set*, where a lexeme identifies a unique semantic object as value of a semantic attribute. Basically, there is one lexeme for each sense of the morpheme. An affix morpheme is associated with morphosyntactic features and, possibly, information about category changes that it induces.

Given a string such as ***cirklarna*** (the circles) the dictionary search will result in the structure (1a). The first element of this structure, N, indicates syntactic category and the second element, !Cirkel, identifies a lexeme set. The content of the lexeme set may be (1b) where each different item identifies a node in the network. At that node further information about this sense of the morpheme can be found. For instance, &Circle#1 may represent the geometrical concept of a circle whereas &Circle#2 may represent the sense of "study circle".

```
(1a) (N  (!Cirkel) ((GENDER   Utral)
                    (NUMBER   Plural)
                    (SPEC     Definite)
                    (CASE     Unmarked))))
```

```
(1b) !Cirkel  =  ((TYPE  &Circle#1)  (TYPE  &Circle#2))
```

The ***Lexical-Functional Grammar*** is a phrase-structure grammar with annotated functional schemata in the style of Kaplan&Bresnan (1982). It deviates in several respects from the current theory and practice of LFG, however. There are no semantic forms and no attribute PRED. Instead of PRED an attribute LEX is used. The value of LEX is a lexeme set. An important difference between LEX and PRED is that LEX is not obligatory. Consequently properties such as coherence and completeness of functional structures are not determined by functional information, but are induced from semantic constraints associated with object type definitions.

In the interpretation process an input sentence is assigned three structures: a constituent structure (c-structure), a functional structure (f-structure) and a semantic structure (s-structure). The c-structure is a phrase-structure tree whereas the other two structurus are descriptor structures encoding information in terms of attributes and values. The f-structure encodes grammatical information, in particular information about grammatical relations and morphosyntactic features. The s-structure encodes information about the input sentence regarded as a message. Thus, it is not a semantic structure in a strict sense, since it represents a contextually adequate interpretation of the input and contextual factors are used in its construction. Partial structures for sentence (2) are shown in figures 2a-2c.

(2) Rita en cirkel i övre högra hörnet.
(Draw a circle in the upper right corner.)

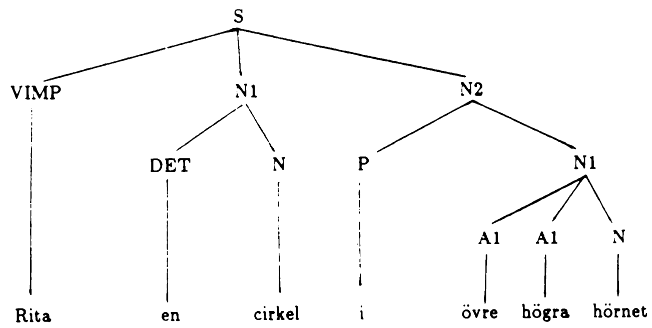

Figure 2a: ***A constituent structure.***

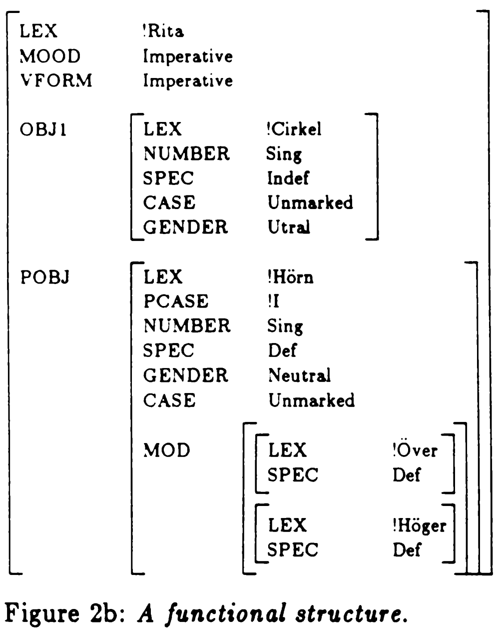

Figure 2b: ***A functional structure.***

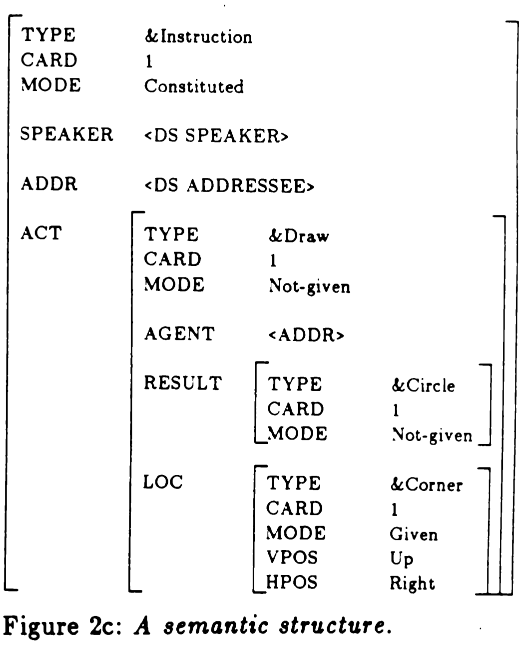

Figure 2c: ***A semantic structure.***

To be well-formed the three structures must be in a relation of *proper correspondence*. The constraints on proper correspondences between c-structure and f-structure are stated in the lexical-functional grammar whereas the constraints on proper correspondences between f-structure and s-structure are included in the definitions of individual object types and attributes. Also functional attributes are assigned such constraints. I refer to these latter rules collectively as Syntactic/Semantic correspondences, or Syn/Sem-correspondences for short.

The domain knowledge of the system is encoded in a semantic network with data structures representing object types, object instances and attributes. The object types represent concepts such as "circle", "line" and "instruction" and carry information about supertypes and subtypes, part-whole relationships and "prototypes". A prototype expresses constraints on the values of attributes that are allowed for instances of the type. As said above they also carry linguistic information specific to the type. For instance, the object type for "circle" will contain the information that it is included in the lexeme set !Cirkel. The object type for "instruction" will contain the information that an instruction can be constituted by means of an imperative utterance. Similarly, attributes representing semantic roles contain information about how they are expressed linguistically, whether by lexemes or grammatical relations.

An object instance has a unique internal name and a description. An illustration is given in (3).

```
(3) Circle29:    ((TYPE        &Circle#1)
                  (CENTRE      Point13)
                  (RADIUS      6)
                  (COLOUR      Black)
                  (RESULT-OF   Draw4))
```

The discourse domain basically consists of all the objects that exist, i.e. are part of the network at any given stage in the discourse. However, without imposing some kind of stratification on the discourse domain it will not be possible to handle anaphoric or implicit reference. There have been various suggestions how this should be done (e.g. Grosz, 1977; Alshawi, 1987). The first method that will be explored in this system is to introduce an object representing the system's view of "a dialogue state" at any given moment. The description of this object, which will comprise context factors such as speaker, addressee, current topics, current visible objects etc, will then be updated for each new utterance.

The processor consists of a chart parser communicating with modules that classify descriptions and determine their referents, if any. The chart parser presently works in a bottom-up mode building c-structure and f-structure in parallel. Thus, the consistency of functional information is checked whenever a task is executed. The parser has certain deterministic traits, which I will not describe here, but it will always find an analysis if there is one.

The role of the classifying component is to determine an appropriate object type for an s-structure constituent. Sometimes a TYPE-descriptor can be determined easily from the lexical information, but there are several complications, such as disambiguation and the handling of headless phrases. A general requirement is that, if a lexeme set has been indicated, the value of TYPE must be an element of that set. Other descriptors of the semantic structure are required to be compatible with the TYPE-descriptor according to its prototype.

The task of the referent identification component is to determine referents of the description found in an s-structure constituent. Not all s-structure constituents will refer to an already existing individual, of course. For these there is still a need to determine a mode of application of the description, i.e. the conditions under which a referent will exist.

The semantic structure associated with a constituent will normally not be constructed until the constituent is judged syntactically complete by the parser, i.e. when an inactive edge is proposed for introduction into the chart. Thus, a constituent such as *en svart fråga* (a black question) may be rejected by the analyzer on the grounds that descriptions of questions cannot contain descriptors refering to colour. Similarly, sentences such as (4) and (5) will be disambiguated when semantic constraints are taken into account. For instance, an active edge spanning the words *flytta cirkeln* of (5) and looking for a locative adverbial can combine syntactically with an inactive edge spanning the words *i hörnet*, but the proposed edge will be rejected on semantic grounds, since the location expressed by the latter words won't be of the appropriate type for a movement action.

(4) Rita cirkeln i hörnet.
(Draw the circle in the corner.)

(5) Flytta cirkeln i hörnet.
(Move the circle in the corner.)

3. Rules for syntactic/semantic correspondences

The relation between syntactic structure and semantic structure is perceived in different ways by different theories. Often some form of an isomorphism hypothesis is adopted. In formal semantics and other schools adopting a "rule-to-rule"-principle the correspondence is a derivational correspondence, not a structural one. This approach has also been used in natural language processors, e.g. in the Rosetta project (Appelo et al. 1987). Other natural language processors rely implicitly or explicitly on structural isomorphy between syntactic and semantic structures (e.g. Lytinen, 1987; Danieli et al., 1987). While I believe that simple one-to-one relations between syntactic and semantic elements are sufficient too handle simple language fragments, I also feel that there are limits to such a methodology. There are syntactic constituents that correspond to no semantic object (e.g. formal subjects and objects), there are those that correspond to more than one semantic object (e.g. locutionary and illocutionary contents) and there are cases where several syntactic constituents relate to one and the same semantic object (e.g. idioms, adjectival attributes). Such structural modifications are easily expressed by descriptor schemata. Moreover, semantic schemata can be associated with syntactic objects and, in the other direction, functional schemata can be associated with semantic objects. Also, descriptor schemata can be associated with contextual factors in very much the same way as they are associated with syntactic objects.

Another question is what syntactic constituents should be considered relevant for the correspondence rules. Halvorsen (1983) defines the correspondences in terms of translation rules which associate functional structures with semantic structures. The semantic structures have quite a restricted form, however, (equivalent to formulas of illocutionary logic) and employ only a limited number of attributes.

Halvorsen (1987), on the other hand, states the correspondences already at c-structure level. The correspondences between functional and semantic structures are captured by means of a projection operator, σ. The projection operator takes functional structures as arguments and returns the corresponding semantic structure. A schema associating the subject constituent with the first argument of a verb is written as in (6).

(6) $((\sigma \uparrow) \text{ ARG1}) = (\sigma(\uparrow \text{ SUBJ}))$

Schemas of this kind are attached both to lexical entries and to rules in the grammar. A schema such as (6) would be attached to every verbal stem in the language that allows this correspondence, i.e. the great majority of verbs. The lexical entry for the verbal stem ***kick*** is specified as follows (***ibid.*** p. 9):

(7) KICK V S-ED $((\sigma \uparrow) \text{ REL}) = \text{KICK}$
$(\uparrow \text{ PRED}) = \text{'KICK'}$
$((\sigma \uparrow) \text{ ARG1}) = (\sigma(\uparrow \text{ SUBJ}))$
$((\sigma \uparrow) \text{ ARG2}) = (\sigma(\uparrow \text{ OBJ}))$

There are some disadvantages with this method, however. First, correspondences of the type in (6) are not stated as rules, in particular not as rules about subjects and first arguments, but as specific information about individual words, and, since there are many alternative correspondences, lexical entries tend to be overloaded with information. This is actually a general problem with lexical-functional grammars where lexical entries are fully specified. Second, the role of the functional predicate 'KICK' is unclear. If information about predicate-argument structure is moved from functional structure to semantic structure, as Halvorsen suggests it should, it seems to be of very little significance.

In FALIN correspondences of the type (6), although in a slightly different form, are associated directly with the attributes SUBJ and ARG1 as elements of the network. Through inheritance they become available to any relation that accept ARG1 (or one of its subattributes) as an attribute.

Semantic attributes such as ARG1 and ARG2 can be regarded as abstract semantic roles (cf. Wachtel 1987). Roles such as being the agent of an act of drawing or the speaker of an utterance are differentiations of ARG1, whereas the result of a drawing, i.e. the picture, and the message of an utterance are differentiations of ARG2. Although these attributes are not in themselves representing grammatical functions, they allow the formulation of simple rules for the interpretation of grammatical relations.

Rules that induce a different mapping between grammatical relations and semantic arguments, such as rules for passive constructions, will also have their results stated on the descriptions of the attributes involved instead on the descriptions of individual verbs. Individual verbs need only be specified for the kinds of mapping they permit. Thus, if we include both the active and the passive cases in the same rule, we get something of the form of (8). The arrows have their usual interpretations as metavariables for corresponding structures. To distinguish functional and semantic structures the latter are indexed by a lowered 's' and the former by an 'f'. Schemas without arrows state conditions on the structure in which the attribute itself occurs.

(8) SUBJ: $\{$ (PASSIVE YES) $(\uparrow_s \text{ ARG2}) = \downarrow_s$ /
(PASSIVE NO) $(\uparrow_s \text{ ARG1}) = \downarrow_s \}$

Conversely, the description of ARG1 will be as in (9), where (AV OBJ) identifies the agent relation in a passive clause.

(9) ARG1: $\{ (\uparrow_f \text{ PASSIVE YES}) \; (\uparrow_f \text{ AV OBJ}) = \downarrow_f$ /
$(\uparrow_f \text{ PASSIVE NO}) \; (\uparrow_f \text{ SUBJ}) = \downarrow_f \}$

By distributing the functional schemas in the semantic network we reduce much of the lexical overloading in ordinary lexical-functional grammars. Every different sense of a morpheme is given its own entry. Moreover, when a stem is part of an idiom or other polymorphemic item, information about this is not only attached to the stem, but also to the relevant node in the network. For instance, the morpheme *ta* (take) is associated with a LEX-value, !Take, that have a fairly large number of different senses. In this set we would also find the action &Take-away, expressed in Swedish as *ta bort*. This item is distinguished from all the others in the same set by a special condition on functional structures expressing it, i.e. that it contains the two descriptors in (10) at top level. Here, PRT is an attribute representing a verbal particle.

(10) &Take-away

$$(\uparrow_f \text{ LEX}) = \text{!Ta}$$
$$(\uparrow_f \text{ PRT LEX}) = \text{!Bort}$$

A functional structure may correspond to a content structure in two different modes. I distinguish a ***constitutive*** (or ***illocutionary***) mode from a ***strict*** (or ***locutionary***) mode. The utterance of an expression constitutes an illocutionary act, i.e. an object instance of a particular illocutionary type. The description of this object is said to correspond to the functional structure of the expression in the constitutive mode. The descriptions of the objects referred to in the utterance, on the other hand, are said to correspond strictly with the f-structures of their refering expressions. Constitutive correspondence will be indicated by double arrows, $\Uparrow$ and $\Downarrow$, to distinguish it from strict correspondence.

Of the linguistic elements that participate in constitutive schemata I will here only consider mood descriptors. A rule for the imperative mood may be formulated as follows:

(11) (MOOD Imperative):

$$(\Uparrow_s \text{ TYPE}) = \&\text{Instruction}$$
$$(\Uparrow_s \text{ AGENT}) = \langle \text{DS SPEAKER} \rangle$$
$$(\Uparrow_s \text{ PATIENT}) = \langle \text{DS ADDRESSEE} \rangle$$
$$(\Uparrow_s \text{ ACT}) = \uparrow_s$$
$$(\Uparrow_s \text{ ACT ARG1}) = (\Uparrow_s \text{ PATIENT})$$

Here DS is a reference to the description of the discourse state. When an s-structure is constructed by means of (11) the current values for the indicated attributes of the discourse state will be retrieved. The fourth schema relates the two different corresponding s-structures to each other, thus integrating the locutionary meaning into the description of the illocutionary act.

To be properly corresponding an f-structure and an s-structure must meet certain general requirements. The functional attributes and descriptors can be divided into two classes, semantically relevant and semantically irrelevant. The latter descriptors play no role in the correspondence relation, whereas every semantically relevant functional descriptor must correspond to a structure of semantic descriptors according to one of the syn/sem-correspondences defined for it. Both f-structures and s-structures must be consistent and determined. Moreover, the s-structure constituents must be typed, compatible with a prototype and specified as to how they apply as descriptions of objects in the discourse domain. Not all information in s-structures have a counterpart in functional descriptors, however. It may instead be retrieved from the discourse state. All this means that there is no requirement on strict isomorphy, whether derivational or structural, between f-structures and s-structures. Still, the use of schemata and the postulation of only two classes of correspondences make the framework both principled and restricted.

Acknowledgements

This paper reports work in progress of the project "Analysis and Generation of Natural Language Texts" financed by the National Swedish Board for Technichal Development. I am indebted to Nils Dahlbäck, Arne Jönsson, Magnus Merkel and Mats Wirén for valuable discussion and to Ulf Dahlén for much of the programming.

References

Ahrenberg, L. Dahlbäck, N., Jönsson, A., Merkel, M. och Wirén, M. 1986. Mot ett dialogsystem för svenska. *NLPLAB Memo 86-01*. Department of computer and information science, Linköping university: Linköping.

Ahrenberg, L. 1987. Parsing into Discourse Object Descriptions. In *Proceedings, Third Conference of the ACL European Chapter*, Copenhagen 1-3 April, 1987, pp. 140-147.

Alshawi H. 1987. *Memory and context for language interpretation*. Cambridge University Press: Cambridge.

Appelo, L., Fellinger, C. and Landsbergen, J. 1987. Subgrammars, Rule Classes and Control in the Rosetta Translation System. In *Proceedings, Third Conference of the ACL European Chapter*, Copenhagen 1-3 April, 1987, pp. 118-133.

Bobrow, R. J., Webber, B. L. 1980. Knowledge Representation for Syntactic/Semantic Processing. In *Proceedings, First Annual National Conference on Artificial Intelligence*, Stanford, August 1980, pp. 316-323.

Fenstad, J. E., Halvorsen, P-K, Langholm, T. and van Benthem, J. 1985. Equations, Schemata and Situations: A framework for linguistic semantics. Manuscript, CSLI, Stanford University.

Grosz, B. J. 1977. The Representation and Use of Focus in Dialogue Understanding. (PhD thesis) *SRI Technichal Note No. 151*, SRI International: Menlo Park.

Halvorsen, P-K. 1983. Semantics for Lexical-Functional Grammar. *Linguistic Inquiry 14:4*, 567-615.

Halvorsen, P-K. 1987. Situation Semantics and Semantic Interpretation in Constraint-based Grammars. *Technichal Report CSLI-TR-87-101*, Centre for the Study of Language and Information: Stanford.

Kaplan, R. & Bresnan, J. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Bresnan J. (ed.) 1982: *The Mental Representation of Grammatical Relations*, The MIT Press: Cambridge Mass., pp. 173-281.

Karlsson, F. 1986. A paradigm-based morphological analyzer. In Karlsson, F. (ed.) 1986: *Papers from the Fifth Scandinavian Conference of Computational Linguistics, Helsinki, December 11-12 1985*. University of Helsinki: Helsinki, pp. 95-112.

Lytinen, S. L. 1987. Integrating syntax and semantics. In Nirenburg, S. (ed.) 1987: *Machine translation*. Cambridge University Press: Cambridge, pp. 302-316.

Moreno, D., Ferrara, F., Gemello, R. and Rullent, C. 1987. Integrating Semantics and Flexible Syntax by Exploiting Isomorphism between Grammatical and Semantical Relations. In *Proceedings, Third European Chapter ACL Conference*, Copenhagen, April 1-3, 1987, pp. 278-283.

Sondheimer, N. K., Weischedel, R. M. and Bobrow, R. J. 1984. Semantic Interpretation Using KL-ONE. In *Proceedings of Coling '84*, Stanford University, Cal. 2-6 July 1984, pp. 101-107.

Wachtel, T. 1987. Discourse structure in LOQUI. In *Recent Developments and Applications of Natural Language Understanding*. UNICOM Seminars Ltd: London, pp. 161-86.