DUAL FRAMES: A NEW TOOL FOR SEMANTIC PARSING

Jean-Louis Binot[1]
IBM Thomas J. Watson Research Center, Hawthorne,
P.O. Box 218, Yorktown Heights, N.Y. 10598.

Daniel Ribbens
Service d'Informatique, Institut Montefiore, B28,
University of Liege, B4000 Liege, Belgium.

ABSTRACT

The dual frames method is a new tool for specifying and establishing semantic dependencies, which has been implemented in a parser of French called SABA. This method offers solutions to some typical problems of semantic parsing strategies - such as the difficulty of coping with different types of sentence structures and the amount of work needed to specify the vocabulary of a new domain - by providing a general and flexible tool which can handle all the kinds of meaningful terms which can appear in a sentence.

1. INTRODUCTION

Attempts at semantic parsing without the support of a full syntactic component have given rise to different methods, among which those of Schank and his group (Schank et al., 1980), Wilks (1975) and Hayes and Carbonell (1981) are well known. These methods have generally been successful at processing simple declarative sentences, but are less suited to process other kinds of - or more complex - sentence structures. Among the main problems are the difficulty of coping with different word orders and the fact that the burden of untangling complex structures falls on individual semantic information, rather than general syntactic rules, thus increasing the amount of specification needed for the vocabulary of a given domain. Thus, other authors, such as Heidorn (1972), Sowa and Way (1986) and Boguraev and Sparck Jones (1983) have preferred to add a semantic component to a syntactic parser. However, purely semantic parsers have other advantages, notably the potential for greater robustness, which justify their further study.

This paper presents a new method for semantic parsing called "dual frames", which attempts to solve or lessen the above problems. This method has been implemented in a semantic parser of French called SABA. The main advantages of dual frames, as we see them, are the following:

- a good distribution of semantic information in dictionary entries, in a way which makes specification easier and avoids redundancy;
- the capability of processing dependencies between all kinds of meaningful terms (verbs, nouns, pronouns, adjectives, adverbs, coordinate structures) in a uniform way;
- the capability of handling in the same way different kinds of sentence structures (such as active and passive voices, interrogative, declarative and imperative forms), and also of processing semantically symmetrical sentences;
- the introduction of operations for computing new semantic frames during a parse, and the definition of a powerful inheritance mechanism.

The next section provides an outline of the SABA parser. The rest of the paper introduces the dual frames method and details some of its aspects.

2. OVERVIEW OF THE SABA SYSTEM

SABA ("Semantic Analyser, Backward Approach", (Binot, 1985), (Binot et al., 1986)) is a **robust** and **portable semantic** parser of written French sentences developed at the University of Liege, Belgium. A prototype of this parser is running in MACLISP and in ZETALISP; it has been tested successfully on a corpus of about 125 French sentences.

While it is possible to account to some extent for ill-formedness in a syntactic parser (see for example the "parse fitting" method developed by Jensen et al. (1983) for the PLNLP system (Heidorn, 1972)), we believe that robustness can better and more easily be achieved through a semantic parsing strategy. The SABA parser is not based on a French grammar, but on semantic procedures which build directly a semantic dependency graph from the natural language input. These procedures are helped by a fragmentation mechanism which allows the system to process complex sentences by splitting them into clauses. The following example is typical of the level of complexity that can be handled by the system:

(1) *Le gros chien noir aboie furieusement quand des enfants qu'il ne connait pas jouent dans le jardin du voisin.*
(The big black dog barks furiously when children that he doesn't know are playing in the garden of the neighbour.)

To allow for portability, the SABA parser translates its natural language input into an "intermediate" semantic network formalism called SF (for "Sentence Formalism"), the details of which have already been covered elsewhere (Binot, 1984, 1985). The main point of interest here is that before generating the SF output, SABA builds a simplified semantic graph expressing all the semantic dependencies established between the meaningful terms of the sentence. The graph established for sentence (1) is shown in (2).

Such a graph is a uniform structure made from oriented binary dependencies, where each dependency points from a complement to the term qualified by this complement (which we shall call for short a "complementee").[2] The graph is built by applying a bottom-up strategy based on a repetitive fragmentation mechanism:

Parsing strategy:

Repeat the following until success or dead end:

1. Fragment the sentence into clauses;
2. Select the innermost clause;
3. Establish relevant semantic dependencies for that clause;
4. Replace the clause, in the text of the sentence, by a special non-terminal symbol.

The fragmentation procedure extends to the left and to the right of each verb until it finds words which identify the limits of a clause; then heuristic rules based on the nature of these limits determine the

[1] This research was performed at the University of Liege, Belgium.

[2] A relative pronoun is not processed as other pronouns, but as a complement of its reference, to which it is tied by a special dependency LR ("Liaison Relative").

(2)

```
                 SIZE         AGENT          MANNER
              *----0----> * ------------> 0 <--------- *
             gros       /|\ chien        /|\ aboie  furieusement
                   COLOR 0                |
                         |         MOMENT |
                         *                |  SITUATION      POSSESSIVE
                        noir          ----> 0 <--------- * <-----0----- *
                                     |      jouent        jardin          voisin
                              AGENT  |
                                     |  LR       OBJECT        BENEFICIARY
                                     * <--- * ---------> 0 <----------- *
                               enfants     que         connait            il
                                                        (NEG)
```

innermost clause.[3] Step 3 is accomplished by applying the dual frames method described hereafter. Lastly, step 4, combined with the repetitive nature of the fragmentation mechanism, enables the parser to reconstruct correctly the content of higher level clauses once embedded inner clauses have been parsed. The successive states of the input for example (1) are shown below with the result of each fragmentation step. Each step leads to the building of that part of the graph shown in (2) that corresponds to the selected clause.

(3) *Le gros chien noir aboie furieusement quand des enfants qu'il ne connait pas jouent dans le jardin du voisin.*

Le gros chien noir aboie furieusement quand des enfants PR jouent dans le jardin du voisin.

Le gros chien noir aboie furieusement PC.

PP[4]

3. DUAL FRAMES: BASIC CONCEPTS

Dual frames is a new method for the specification and the establishment of semantic dependencies, which has been designed to handle all possible kinds of constituents in a uniform way. Basically, this method consists of using a dual system of semantic property lists respectively called "Elists" and "Tlists".[5]

3.1. Elists

Elists can be viewed as kinds of case frames, but are assigned to every meaningful term, not only to verbs. Different meanings of the same term cân have different Elists. As an illustration, let us consider a single clause example:

(4) *Le chien aboie furieusement dans le jardin.*
(The dog barks furiously in the garden.)

The Elist of the verb "aboyer" ("to bark") looks like this:[6]

```
(5) ABOYER: ELIST:
     ((AGENT NOUN_CLASS_RESTRICTION (CANINE))
      (SITUATION NOUN_CLASS_RESTRICTION (PLACE))
      (MOMENT NOUN_CLASS_RESTRICTION (TIME))
      (MANNER))
```

This Elist states that "aboyer" can have an AGENT argument (which must be a canine), and arguments of MOMENT, SITUATION and MANNER. NOUN__CLASS__RESTRICTION introduces a restriction on the semantic category of nominal arguments.

[3] Situations of choice arising during the two first steps are handled by a backtracking mechanism.

[4] PR, PC and PP denote respectively a relative clause, a conjunctive clause and a main clause.

[5] These names stand simply for "LIST of the complementeE" and "LIST of the complemenT".

[6] All semantic specifications shown in this paper are only meant as illustrations of the concepts described, and not as parts of some universal model. In practice, separate specifications can be provided for each application domain.

3.2. Tlists

Semantic restrictions alone are not sufficient to obtain correct parses except in very simple cases. The possible roles of a term depend also on the way this term is used in a sentence. The basic idea of Tlists is to list explicitly the possible roles of every meaningful term or construct processed during a parse.

Tlists are obtained by the parser in different ways. Tlists of terms such as adjectives and adverbs are specified in the dictionary as **intrinsic** properties of these terms. Thus the specification shown below states that the adverb "furieusement" ("furiously") can only fill the MANNER role:

```
(6) Furieusement: TLIST: ((MANNER))
```

A Tlist can also be **inherited** from another word. The Tlist of a noun, for example, is inherited from the preposition leading the nominal group. To each preposition is assigned a specific Tlist. Thus "jardin" ("garden") in (4) will inherit the Tlist assigned to the French preposition "dans":

```
(7) Dans: TLIST: ((SITUATION))
```

A noun without a preposition, like "chien" ("dog") in (4), will be said to be introduced by a special dummy preposition called PHI, from which it will inherit its Tlist:

```
(8) PHI: TLIST:
     ((AGENT VOICE_RESTRICTION (VA))
      (OBJECT VERB_CLASS_RESTRICTION  (TRANSITIVE))
      (BENEFICIARY VERB_CLASS_RESTRICTION
                  (STATE EVENT))
      (MOMENT) (INSTRUMENT) (NAME) )
```

Tlists can also express restrictions, which bear on possible complementees. In fact, Elists and Tlists have exactly the same structure and will hereafter be referred to by the generic name of **semantic frames**. The above Tlist states that each noun without a preposition can play the following roles: AGENT of an active verb, OBJECT of a transitive verb, BENEFICIARY of a state or an event, MOMENT, INSTRUMENT or NAME.

Subordinate clauses are processed like noun groups, except that the Tlist of the subordinated verb is inherited from the conjunction leading the clause. Thus, "jouent" ("play") in (1) will inherit from the conjunction "quand" ("when") a Tlist containing only the role MOMENT.

3.3. Establishing dependencies

The dependencies that can be established between two terms are determined by an "Elist/Tlist intersection mechanism" described here:

A1: Consider only the dependencies which are mentioned both in the Tlist of the complement and in the Elist of the complementee;

A2: Among these, retain only the dependencies for which all restrictions mentioned in the Elist and in the Tlist are satisfied and agreement rules, if any, are also satisfied.

The set of dependencies established between the terms of a given clause must furthermore satisfy global constraints: a same term cannot be tied by two dependencies of the same name, and the set of all dependencies established for any given structure (clause or group) must form a connected graph.

We shall apply these rules to example (4). The French word "chien" has at least two possible meanings: "dog" (which belongs to the class of canines) and "gun hammer", which belongs, say, to the material objects. "Chien" inherits the Tlist of PHI shown in (8). Comparing this Tlist with the Elist of "aboyer" in (5), rule A1 yields 2 possible dependencies: AGENT and MOMENT. For the first meaning of "chien", rule A2 will keep AGENT and discard MOMENT. The second meaning of "chien" doesn't satisfy any of the restrictions in the Elist of "aboyer" and will be discarded because it remains unconnected. Knowing furthermore that "jardin" denotes a place, the system finds easily that the only admissible result for the whole clause is:

```
(9)          AGENT            MANNER
         * ----------> 0 <---------- *
     chien            /|\ aboie     furieusement
     (dog)  SITUATION |
                      |
               jardin *
```

The distinction between Elists and Tlists, illustrated in the above example, helps to reduce redundancy and ease specification. While Elists are intrinsic properties, Tlists can be inherited from prepositions and conjunctions; the Tlist of PHI shown in (8), which expresses in a few lines the possible semantic roles of prepositionless nouns in French, is a good example of the conciseness that can thus be achieved.

It should be noted that unlike Wilks' paraplates (Wilks, 1975), Tlists are not exclusively related to prepositions. A Tlist is assigned to every meaningful term or construct processed by the system; moreover, and again unlike paraplates, new Tlists can be computed from old ones, as we shall show in section 5.

It can also be noted than Elists and Tlists offer some similarities to Sowa's conceptual graphs (Sowa, 1984); however, while conceptual graphs are basically a representation formalism for concepts, Elists and Tlists were especially designed as a tool for parsing without the support of a grammar, and, as such, can include syntactic restrictions such as the voice restriction in (8).

Lastly, let us note that the dual frames method supports preferences a la Wilks as well as more traditional mandatory restrictions, and this without needed additional specifications such as Wilks' exhaustive list of bare templates. We only require a slight modification of the rules of section 3.3. In the "preference mode", all dependencies which pass rule A1 will be considered as acceptable by the system, and rule A2 will be used to prefer, among them, the ones satisfying the greatest number of restrictions. This other mode has also been implemented in the SABA system, which can run either in restriction mode or in preference mode.

3.4. Classes and hierarchies

The SABA system offers the possibility of specifying, for each given domain, a hierarchy of classes, which will then be taken into account by the restriction checking mechanism. The system accepts also heterarchies (thus, knives, for example, could be classified both as cutting tools and as piercing tools).

To each individual concept is assigned a property CLASSLIST, which enumerates entry points for that term in the hierarchy. If several entry points are given, they are interpreted as a **disjunction** of classes. This rule is useful to specify different aspects of the same concept. Thus the word "departement" can denote a place, say in a store, or an animate collective (the set of employees working in the corresponding place), but does not always denote both concepts simultaneously. It will be simply specified like this:

```
(10) departement: CLASSLIST: (PLACE EH)
```

3.5. Robustness

As Carbonell and Hayes (1983) noted, case instantiation systems have some inherent robustness, stemming from the fact that they are more or less insensitive to the order of arguments. A French speaking person, for example, could say "*The big dog black*" in a literal translation of (1), since adjectives, in French, can be placed behind the qualified noun. This kind of mistake is easily handled by the SABA parser.

However, the main advantage of semantic parsers does not lie in the insensitivity to some specific kind of mistake, but in the more fundamental fact that such parsers do not require an exhaustive specification of all syntactically admissible constructs. This makes it easier to define procedures searching for specific features in a flexible way. Sentence (11), where the negation marker is misplaced, and (12), where the interrogative construct is incorrect, illustrate some other kinds of mistakes that can be handled by SABA.

(11) *Il n'a travaille jamais.* (He worked never.)

(12) *Tu aimes Marie?* (You love Mary?)

4. SUBJECT IDENTIFICATION RULES

There are cases in which the basic Elist/Tlist intersection mechanism fails. One of these cases arises with semantically symmetrical structures, such as:

(13) *John loves Mary.*

where the AGENT of the loving action cannot be determined by semantic restrictions alone. A similar problem arises with passive voice: thus no amount of semantic restrictions could allow the parser to choose John as an AGENT in (14) and as a semantic OBJECT in (15).

(14) *John has cheated.*

(15) *John was cheated.*

Other semantic systems have faced these problems by introducing, in some way, **positional restrictions**. These restrictions are implicit in Riesbeck's expectation mechanism (Riesbeck, 1974), which is tied to a left-to-right parsing order, and in Wilks' preference system (Wilks, 1975), where everything crucially depends on the template matching order. They appear explicitly in Hayes and Carbonell's system (Hayes and Carbonell, 1981) as positional markers. A general inconvenience of positional restrictions, however, is that they are strongly related to a specific word order (usually the one used in active declarative sentences).

In fact, what the above approaches are trying to do, in ad hoc and not fully satisfying ways, is to get around the crucial notion of syntactic subject. We believe that even in a semantic system, the notion of **subject** is necessary to solve cleanly the problems mentioned above: John is the AGENT in (13) because he is the subject of an action in the active voice; he is the OBJECT in (15) because he is the subject of an action in the passive voice. We shall propose below a general and semantic way of determining and using the notion of subject in a semantic parser.

Our solution is based on Fillmore's "subject selection rule" (Fillmore, 1968). The key idea is to define "Subject restriction rules", which are kinds of inverted selection rules, and then to use these rules to identify the subject:[7]

Subject restriction rules:

1. The AGENT of an active action verb must be the subject if it is introduced by PHI;

[7] These rules do not cover the problem, which appears in English but not in French, of distinguishing between direct and indirect object.

2. The INSTRUMENT of an active action verb must be the subject if it is introduced by PHI;

3. The OBJECT of a passive action verb must be the subject if it is introduced by PHI;

4. The BENEFICIARY of an event or a state must be the subject if it is introduced by PHI.

Subject identification rules:

1. If, in an attempted parse, an argument must satisfy a subject restriction rule in order to fill some case of a verb, then check if another subject has already been identified for that parse. If so, the restriction fails. If not, the argument will be chosen as the subject of the verb and the restriction will be satisfied.

2. In a successful parse, a subject must have been identified for every verb other than an imperative or infinitive.

3. In a successful parse, the subject must precede the verb, except in interrogative sentences when the subject is a personal pronoun.[8]

Let us look again at example (13). Assuming some typical action Elist for "love", the parser will match this EList with the Tlist of PHI and find two possible interpretations:

```
(16)     AGENT(love,John).OBJECT(love,Mary)
         OBJECT(love,John).AGENT(love,Mary)
```

In the first case, the subject will be identified as John, and in the second case as Mary. The second interpretation will be discarded because of rule 3. In example (15), "John was cheated", the interpretation taking John as AGENT will not allow the parser to identify a subject because no subject restriction rule will be activated; it will thus be discarded by rule 2 above.

5. COMPUTING SEMANTIC FRAMES

The usefulness of dual frames has been enhanced by defining operations for computing new semantic frames (Elists or Tlists) from existing ones. We distinguish two basic kinds of situations in which such computations are useful.

5.1. Semantic frames Union

In ambiguous situations, where different semantic frames of the same type (Elist or Tlist) could be used for the same term and choosing between them would be impossible or too difficult, it is possible to compute a resulting semantic frame as a kind of "union" of the given frames. The intuitive idea of **semantic frame union** is to keep all possible dependencies with the weakest restrictions. More precisely, the rules are:

Semantic frames union rules:

1. Every dependency mentioned in any one of the argument frames belongs in the resulting frame;

2. A restriction assigned to a dependency will belong to the resulting frame if and only if it belongs to all the argument frames;

3. If a restriction belongs to the resulting frame, its set of acceptable values is formed as the union of the sets of acceptable values of that restriction in all arguments.

An example of the use of semantic frames union may be found in the processing of pronouns. The roles that prepositionless pronouns can fill can be at least partially determined by taking into account the surface form of the pronoun itself. Thus "whom" can obviously not be the AGENT of an action, nor the BENEFICIARY of a state, while for "he" these two roles are allowed.

These distinctions can be introduced by grouping pronouns into different classes and by assigning to each class a Tlist which will be inherited by the pronouns of that class when they are introduced by PHI. Four classes of pronouns are used in the SABA system: S12 (pronouns of the two first persons that can be subject), S3 (pronouns of the third person that can be subject), OD (pronouns that can be direct object) and OI (pronouns that can be indirect object). The Tlists assigned to S12, OD and OI are:

```
(17) S12: TLIST:
     ((BENEFICIARY VERB_CLASS_RESTRICTION
                   (STATE EVENT))
      (OBJECT VOICE_RESTRICTION (VP))
      (AGENT VOICE_RESTRICTION (VA)))
     OD:  TLIST: ((OBJECT VOICE_RESTRICTION (VA)))
     OI:  TLIST: ((BENEFICIARY))
```

S3 is the same as S12 but with an additional INSTRUMENT role.

The only problem with the above method is that a pronoun may belong to several classes, as illustrated by the following examples:

(18) *Nous mangeons.* (We eat)

(19) *Il nous voit.* (He sees us)

(20) *Il nous parle.* (He talks to us)

The French pronoun "nous" belongs to S12, OD and OI! The Tlist of such pronouns will be determined by applying the "semantic frames union" operation to the Tlists of the different classes. For the Tlists of (17), this operation yields the following result:

```
(21) nous: resulting TLIST:
     ((BENEFICIARY)
      (OBJECT VOICE_RESTRICTION (VP VA))
      (AGENT VOICE_RESTRICTION (VA)))
```

5.2. Semantic frames Intersection

In constraint situations, where different semantic frames of the same kind (Elists or Tlists) should simultaneously be taken into account for the same term, a resulting semantic frame can be computed as a kind of "intersection" of the given frames.

Semantic frames intersection is used in the SABA system for the processing of coordinate structures. We apply a generalized version of one of Fillmore's rules (Fillmore, 1968), stating that meaningful terms or structures can only be coordinated if they can play the same semantic role with respect to the rest of the sentence. In the dual frames method, this amounts to saying that the intersection of the Tlists of the conjuncts should not be empty. This intersection will then be taken as the resulting Tlist of the coordinate structure. Thus, in (22):

(22) *Jean viendra ce soir ou demain.*
(John will come this evening or tomorrow.)

the Tlist of "ce soir" ("this evening") will be inherited from PHI (as shown in (8)), and the Tlist of the adverb "demain" ("tomorrow"), which is an intrinsic property of that adverb, is shown in (23); the resulting Tlist of the coordinate structure "ce soir ou demain" will then be computed as shown in (24)

```
(23) Demain: Tlist: ((MOMENT))

(24) ce soir ou demain: resulting TLIST: ((MOMENT))
```

The above example is deceptively simple. In the general case, the dependencies belonging to the intersection have associated restrictions which must be taken into account, as well as the hierarchies of admissible values for these restrictions. The general rules of semantic frames intersection are:

Semantic frames intersection rules:

1. A given dependency will belong to the resulting semantic frame if and only if it belongs to all argument frames;

[8] This rule handles verb/subject inversion in French interrogative sentences.

2. If a restriction is assigned to a dependency in at least one of the argument frames, it will appear in the resulting frame;

3. The set of acceptable values for any restriction in the resulting frame is the set (possibly empty) of all nearest common successors of the sets of acceptable values for the restriction in all argument frames.

6. SEMANTIC FRAMES INHERITANCE

The idea of case inheritance was already expressed by Charniak (1981). However, inheriting case names, as was suggested, is not sufficient. One must also account for possible restrictions or preferences, with their sets of acceptable values, and for the hierarchies, or even heterarchies, associated with these sets. The operations defined in the previous section allow us to define a powerful Elist inheritance mechanism which can combine Elists at different levels of the hierarchy. As an example, declaration (25) states that every action usually has an animate (EA) AGENT and arguments of SITUATION and of MOMENT, while (26) indicates that the AGENT of the specific action of "programmer" ("programming") must be a human (EH). These Elists will be combined in order to produce (27).

```
(25) (ADD_CLASS_PROPERTY ACTION
      ELIST
      ((AGENT NOUN_CLASS_RESTRICTION (EA))
       (MOMENT NOUN_CLASS_RESTRICTION (TIME))
       (SITUATION NOUN_CLASS_RESTRICTION (PLACE)))

(26) programmer: ELIST:
      ((AGENT NOUN_CLASS_RESTRICTION (EH)))

(27) programmer: resulting ELIST:
      ((AGENT NOUN_CLASS_RESTRICTION (EH))
       (SITUATION NOUN_CLASS_RESTRICTION (PLACE))
       (MOMENT NOUN_CLASS_RESTRICTION (TIME)))
```

The rules of semantic frames inheritance are the following:

Semantic frames inheritance rules:

1. Elists inherited from different predecessors in the hierarchy will be combined by using semantic frames intersection;

2. Elists inherited by different classes in the CLASSLIST of a term will be combined by using semantic frames union;

3. Elists at different levels of the hierarchy will be combined by using an operation called *semantic frames merging*, which keeps all possible dependencies but with the strongest restrictions. This merging combines rule 1 of semantic frames union and rules 2 and 3 of semantic frames intersection.

The effect of the third rule can be observed in example (27): all dependencies from (25) and (26) were kept, and the AGENT dependency retained the strongest restriction ("human" being considered as the nearest common successor of "human" (EH) and "animate" (EA) in the hierarchy).

7. CONCLUSION

As illustrated in this paper, the dual frames method offers several advantages with respect to existing semantic systems. The systematic separation of all semantic information between Elists and Tlists provides both flexibility and conciseness of specification; subject identification rules free the method from positional restrictions; and, lastly, a powerful inheritance mechanism based on well defined semantic frames operations eases the task of specifying a new application domain.

The use of well defined list or graph operations in natural language processing has received increasing attention. At first glance, our "semantic frames union" could be likened to Sowa's "join" (Sowa, 1984) or to the concept of unification (Shieber, 1985). However, unlike unification, frames union never fails, since its purpose is to handle ambiguities between possibly conflicting interpretations. In fact, the operation which can be taken as a special form of unification is the "semantic frames merging" of section 6. Let us note however that, while unification is usually presented as a fundamental primitive, frames merging derives from the two basic operations of frames union and frames intersection which, as we have shown, have reasons to exist in their own right.

Many other issues of the SABA parser were not discussed here, including resolution of lexical ambiguities, of attachment ambiguities, of quantifier scope and of pronoun reference. More details can be found in (Binot, 1985).

ACKNOWLEDGMENTS

Thanks are due to Yorick Wilks for his helpful and in-depth comments on the SABA system, to John Sowa, of IBM Systems Research Institute, for an interesting discussion about conceptual graphs, and to George Heidorn, Karen Jensen and Norman Haas, of IBM Research, for many judicious comments on previous versions of this paper.

REFERENCES

1. J-L. Binot, A set-oriented semantic network formalism for the representation of sentence meaning, In *Proc. ECAI84*, Pise, 1984, pp. 147-156.
2. J-L. Binot, *SABA: vers un systeme portable d'analyse du francais ecrit*, Ph.D. dissertation, University of Liege, 1985.
3. J-L. Binot, P-J. Gailly and D. Ribbens, Elements d'une interface portable et robuste pour le francais ecrit, In *Proc. Huitiemes Journees de l'Informatique Francophone*, Grenoble, 1986.
4. B.K. Boguraev and K. Sparck Jones, How to drive a database front-end using general semantic information, In *Proc. Conference on Applied Natural Language Processing*, Santa Monica, 1983.
5. J.G. Carbonell and P.J. Hayes, Recovery strategies for parsing extragrammatical language, *AJCL* 9:3-4, 1983.
6. E. Charniak, The case-slot identity theory, *Cognitive Science* 5, 1981, pp. 285-292.
7. C. J. Fillmore, The case for case, in (Bach and Harms, eds.) *Universals in Linguistic Theory*, Holt, Rinehart and Winston Inc., New-York, 1968.
8. P.J. Hayes and J.G. Carbonell, Multi-strategy construction-specific parsing for flexible data base query and update, In *Proc. IJCAI81*, Vancouver, 1981, pp. 432-459.
9. G.E. Heidorn, Natural language inputs to a simulation programming system, Naval Postgraduate School Report, Monterey, California, 1972.
10. K. Jensen, G.E. Heidorn, L.A. Miller and Y. Ravin, Parse fitting and prose fixing: getting a hold on ill-formedness, *AJCL* 9:3-4, 1983.
11. C. K. Riesbeck, Computational understanding: analysis of sentence and context, Memo AIM-238, Stanford University, 1974.
12. R.C. Schank, M. Lebowitz and L. Birnbaum, An integrated understander, *AJCL* 6:1, 1980.
13. S. Shieber, An introduction to unification-based approaches to grammar, Tutorial session, 23rd Meeting of the ACL, 1985.
14. J.F. Sowa, *Conceptual structures - information processing in mind and machine*, Addison-Wesley, 1984.
15. J.F. Sowa and E.C. Way, Implementing a semantic interpreter using conceptual graphs, *IBM Journal of Research and Development* 30:1, 1986.
16. Y. Wilks, An intelligent analyser and understander of English, *CACM* 18:5, May 1975.