AFIA

Association Française pour l'Intelligence Artificielle

WOCFAI '91

First World Conference on the Fundamentals of Artificial Intelligence

1-5 July 1991, Paris, France

Proceedings

Edited by

Michel De Glas and Dov Gabbay

MINISTERE
DE LA RECHERCHE
ET DE LA TECHNOLOGIE

CNRS

A DEPENDENCY SYNTAX FOR THE SURFACE STRUCTURE OF SENTENCES

Leonardo Lesmo, Vincenzo Lombardo

Dipartimento di Informatica - Universita' di Torino
C.so Svizzera 185 - 10149 Torino - Italy
Tel. 39 - 11 - 7712002
e-mail: lesmo@di.unito.it, vincenzo@di.unito.it

Abstract

The paper deals with a dependency-based formalism for representing the syntax of natural languages. The dependency tree adopted for describing the structure of sentences is introduced. The grammar is expressed via ID/LP tables, that are particularly suitable for languages that present a high degree of word order freedom (as Italian). They specify the constraints on the grammaticality of sentences, by modelling linguistic phenomena with the use of features.

Keywords: natural language, syntax, dependency and constituency grammars, grammatical relations.

1 Introduction

The concept of "head" has been introduced by recent formalisms, like GPSG and Head Grammars [13], into constituency syntax. Even if far from the head&modifier approach stated by dependency grammars [10], such formalisms, based upon constituents, have acknowledged the importance of the head inside a phrase, a sort of elected symbol on the right hand side of a rewriting rule.

In dependency grammars, the structure of the sentence is represented via a tree that differs from the more usual parse trees appearing in consituency grammars, since every node in the tree (and not only the leaves) is associated with the words in the sentence. A "constituent" is identified with an entire subtree, and the prominence of the head is emphasized by having it as the root of the subtree. The dependency trees that have been proposed in the literature have been tied to the underlying representation of sentences, as in the Prague School ([14] and more recent formulations [11] [12]), or to the surface structure, as in Word Grammar [3]. We are closer to the latter view: the dependency grammar, that will be described by rules of Immediate Dominance (ID) and Linear Precedence (LP), generates trees that represent the surface structure of sentences.

The necessity of individuating a head in dependency syntax unfortunately rises some problems when some aggregations of words, that seem to feature more than one head, are dealt with. Coordinations (as well as comparative constructions), that are difficult to treat both for dependency and for phrase structure grammars, forced some researchers (see [12]) to linearize complex dependency structures in order to introduce a special component for coordination and apposition and not to violate the condition of projectivity[1] [15]. Even some simpler phenomena, like the assignment of grammatical relations and the treatment of long distance dependencies, seem to be related to another level of linguistic knowledge, that is not purely syntactic. Such an intuition is supported, for example, by the great variety of coordinations that are possible in the language. The categories to conjunct can be very different and it is very hard to find some regularities of behaviour. Consider, for example, the sentences

- **He did it without malice and since he needed a revenge [15]**
- **John asked the time of the train and how late it was**
- **This land extends beyond the hills and further on**

Both from a constituency and dependency points of view the situation is very difficult to analyze: the conjuncts can be different constituents (i.e. different non-terminal symbols) and also different heads can be involved.

The two levels of the grammatical description can be seen as two distinct modules, the first of which builds up basic dependency structures, while the second combines two or more heads (and the related subtrees) into more complex aggregates. Conversely we think that, with an appropriate typing of arcs in the dependency structure, even complex constructions, like coordinations and comparatives, or long distance dependencies, can be represented by a (non-linearized) graph, without violating the condition of projectivity.[2]

The paper deals with the basic module of the grammar in a dependency formalism. The next section defines a dependency structure, on which the constraints on the features, that state the grammaticality of a sentence, are expressed. The third section introduces the grammar, which is described via ID/LP tables (the grammar of a fragment of Italian is presented).

A last remark: dependency syntax is based exclusively on words [Hudson 84]; larger aggregates can nevertheless be named, by following the same terminology of the constituency grammars, in order to describe the formalism itself. Such names are possibly introduced only for the sake of clarity, but are not a part of the formalism.

2 The Dependency Structure

2.1 The formalism

The dependency tree that we build represents the surface structure of a sentence. A **node** is a triple whose first component is the Node Label, the second is the Node Type and the third is an ordered list of elements:

$$n = < NL, NT, <e_1\ e_2 \ldots e_i \ldots e_m > >_{m \geq 1}$$

NT belongs to a predefined set of node types, that is:

$$NT \in NTypes = \{TOP, VERB, REF, ADJ, ADV, DET\}$$

The elements of the set NTypes can be considered as the representatives of classes of grammatical categories, that are also organized into hierarchies of subclasses. The categories that belong to the same type share a similar semantic treatment and, usually, the same syntactic behaviour. Of course, the syntactic similarity between two categories that have been introduced within the grammar can be only partial, otherwise the two categories should collapse into one. For example, the class of adjectives (of which the symbol ADJ is the representative) features two different behaviours that can be associated with two subclasses, a class of (common) qualifying adjectives and a class of *relational* adjectives. Consider the two adjectives piacevole (pleasant) and renale (renal), which are two examples of a common qualifying and a relational adjective respectively. Both words will be associated with nodes of the same type ADJ in the dependency structure, even if their respective categories behave differently: a relational adjective, so called because of its

[1] The condition of projectivity has been stated in many equivalent forms across different theories: [15] gives a formal description of it, for the case of the underlying structure of sentences; Hudson [3] formulates it as the *Adjacency principle* on surface structures, which he excerpts from Robinson: "If A depends directly on B (i.e. is a modifier), and some other element C intervenes between them (in linear order of strings), then C depends directly on A or on B or on some other intervening element". The name "condition of projectivity" comes from the analogy of the principle with the operation of projecting on a line ("linear order of strings") the sentence represented in the dependency tree.

[2] Moreover, such a typing is necessary for the treatment of long-distance-dependencies and of the sharing of nodes within a dependency framework.

derivation from a noun (rene - kidney) that results to be put in relation with the noun specified, can only follow it, as in

- *** una renale malattia** vs **una malattia renale**
  a renal desease,

while qualifying adjectives can stand on both sides of it

- **una piacevole conversazione** vs **una conversazione piacevole**
  a pleasant conversation

This difference of behaviour can be accounted for in the grammar, by an appropriate use of the features' values: the members of the category ADJ, whose feature *subtype* has the value *relational,* can only follow the noun.

Much more articulated is the class of categories underlying the REFerent type: a REF node is associated with words belonging to the high-level categories NOUN and PRONOUN. Such words refer to an entity in the world where the sentence is interpreted, and so the related categories are joined together into the REF type. In the sentences

- **il cuoco cucino' un pesce**
  the chef cooked a fish

- **il cuoco lo cucino'**
  the chef it cooked
  "the chef cooked it"

both **un pesce** (a fish) and **lo** (it) refer to an entity in the world, but we can notice that they behave differently from a syntactic point of view. **un pesce** follows the verb (but can precede **it** too), while **lo** must absolutely precede **it**. Rules are even more complicated if the verb is untensed. Nevertheless the splitting of the REF class into NOUN and PRONOUN is not sufficient to account for the syntactic behaviours of referents. For example, relative pronouns (RELPRON) and personal pronouns (PERSPRON) appear in different constructions, and even inside the RELPRON category, almost every pronoun can belong to an autonomous category. Let us consider the examples:

- **La ragazza che viene in autobus vive in citta'**
  The girl who comes by bus lives in town

- **La ragazza il cui abito rosso viene dall'India vive in citta'**
  The girl whose red dress comes from India lives in town

- **La ragazza con la quale ho pranzato ieri vive in citta'**
  The girl with whom I had the lunch yesterday lives in town

che cannot be preceded by determiners or followed by prepositions, **cui** can behave as an adjective too and be preceded by determiners, **la quale** has a gender (depending on the article which it requires) and a number. Moreover **cui** (as well as **quale**, but not **che**) can introduce a long-distance-dependency, as in

- **Il libro di cui hai letto l'introduzione tratta della guerra chimica**
  The book of which you have read the introduction deals with (the) chemical war

Such differences have led us to introduce very refined categories in order to deal with relative pronouns: CHEREL (i.e. RELative CHE, to distinguish it from the other categories to which **che** belongs: conjunction, comparative particle, ...), CUIREL, QUALIREL.

The types VERB, ADVerb and DETerminer are self-explanatory, while a few words are required to introduce the type TOP. A TOP node is associated with the so-called function words [15], that can be identified intuitively as words that cannot have dependents. Categories like conjunctions, prepositions and comparative particles will be associated to TOP nodes: they share the common feature that their semantic interpretation cannot be worked out apart from other words in the sentence that are syntactically related to them and that can be considered as functions' arguments. A PREPosition connects the noun (that is the entity) which modifies it to the noun or to a verb which it modifies; conjunctions and comparative particles connect several

conjuncts or two comparison terms respectively. As we will see in the discussion about links, the modifiers of a TOP node do not stand in a real dependency relationship with it. The TOP node is the root of coordinated constructions (where its name derives from), even if only from a structural point of view.

2.2 Heads and Links

The elements of the sequence which is the third component of a node are of two kinds: **heads** and **links**. Each node contains one or more heads and zero or more links. The presence of more than one head is possible only within the TOP nodes, when prepositional conjuncts occur, while the absence of links, which are links to possible modifiers, identifies a leaf node in the dependency tree. A head has the following form:

<HEAD, <lexical-data> >

where HEAD is a marker, that individuates the type of the element, and "lexical-data" includes the information for the word associated with this node and that is retrieved from the corresponding lexical entry. Such information is represented via pairs <feature,value>. The set of features depends on the category of the word: a verbal form contains, among others, the feature "tense" with a different value "+" or "-", depending on if it is tensed or not respectively, while a noun will contain the features gender and number. Consider, for example, the head element of a REF node associated with the word **ragazzo** (boy):

```
<HEAD,
      < < cat, noun>
        < lemma , ragazzo >
        < gender , masculine >
        < number , singular >
        < proper , - > > >
```

"Lemma" indicates the normalized (i.e. not inflected) form of the word, while "proper" is a feature that states whether a noun is a name or not.

A link has the following form:

<LINK, Link-Type, NL, <features> >

LINK is a marker, NL is the label that identifies the linked node (a dependent), <features> are information about the linked substructure (i.e. the substructure whose root is NL), gathered by means of percolation of the features upwards. Link-Type is the most notable characteristic of a link. The set of link types is

{DEP, STR, D&S}

Dependencies embody two different kinds of information: the surface attachment of a node to another at a higher level and the deep connection between a modifier and its governing node. The three types in the set represent all the possible situations. D&S arcs connect two nodes in both the surface and deep relations: for example, a determiner is always connected to a noun both structurally, i.e. superficially attached, and deeply, in the sense that a determiner governed by a noun, is logically dependent from it, and not from other nodes in the structure. These two types of connections are taken apart in the other two elements of the set. STR is the type of purely superficial attachments: a clear example of this is how are structurally linked nodes that are involved in long-distance-dependency phenomena. If we consider again the sentence

- **Il libro di cui hai letto l'introduzione ___ tratta della guerra chimica,**
  The book of which you have read the introduction ___ deals with (the) chemical war

the prepositional phrase **di cui,** which introduces the relative sentence (that must necessarily begin with a substructure containing a relative pronoun) and, hence, structurally depends on the verb **hai letto**, is logically dependent on **introduzione**, and the underscore individuates its natural position (after a noun). The structural dependency of **di cui** from **hai letto** is represented via a STR arc. Conversely, the logical dependency of **di cui** from **introduzione** is specified via a DEP arc that connects them, even if in very distant positions in the structure. Such a connection is determined on the basis of syntactic and semantic data in the example at hand. In the sentence above, **di cui** is a "logical" modifier of the direct object **l'introduzione**, as it results from the application of the following rule: if the "displaced" substructure X is not a complement of the verb that governs the relative sentence and this verb is transitive; then X is a "logical" modifier of the direct object. This rule obviously belongs to the second module of the grammar (not described in the paper) that deals with the grammatical relations. In the example

- **Il professore di cui parlavi a Mario insegna greco**
  The professor of whom you were talking to Mario teaches Greek

di cui is a complement of **parlavi** (*you were talking*) and the dependency relationship does not involve a long distance. On the contrary, in the example

- **Il libro che stai scrivendo e di cui hai dato il primo capitolo a Giorgio non sara' mai pubblicato,**
  The book you are writing and of which you have given the first chapter to Giorgio will never be published,

we can notice how in the second relative sentence the long-distance-dependency is governed by the rule cited above, even if **dare** (*give*) also requires an indirect object as complement.

A long-distance-dependency is then represented in the formalism by allowing a node to be shared (that is governed) by two different nodes via two different arcs. The "shared" nodes described in [3] are used to cope with "equi" constructions, like

- **Un ragazzo persuase Luisa a comprare una enciclopedia.**
  A boy persuaded Luisa to buy an encyclopedia.

Luisa depends on **persuase** via a D&S arc and on **comprare**, the verb in the subordinate sentence, via a DEP arc.[3] The introduction of such a new type in the representation of long-distance-dependencies and multiple governors allows the satisfaction of the condition of projectivity [12], that must be applied only to structural arcs (i.e. STR and D&S). Structural arcs represent the superficial links of the nodes in the dependency tree. Let us consider the complex example

- **Il ragazzo che viaggio' con Luisa la persuase a comprare un'enciclopedia**
  The boy who travelled with Luisa persuaded her to buy an encyclopedia

where the accusative pronoun **la** (referring to **Luisa**) is the direct object of **persuase** and the subject of **comprare**. The corresponding dependency tree in the formalism we have just described is shown in figure 1. The non-violation of the condition of projectivity is guaranteed by the strict ordering of the elements in every node and the introduction of a DEP arc. Such an ordering reveals to be very important in both the phases of generation and analysis of a sentence: during the generation, it preserves the satisfiability of the condition of projectivity by the structure which is being built; during the analysis it is fundamental in the process of finding the mapping between the grammatical relations involved and the nodes in the dependency tree [5].

3 ID/LP tables

The idea of representing a grammar via ID/LP rules was introduced by Gazdar [2] in constituency syntax within the GPSG framework. ID rules specify the relationship of Immediate Dominance, while LP rules represent the Linear Precedence, in order to split the two components, that are usually represented together by the rewriting rules of a phrase-structure grammar. Many redundancies are avoided especially in the case of free-word order languages [16]. Italian shows a so high degree of freedom in the order of constituents [9], that the choice of representing a grammar by means of ID/LP rules seems appealing.

[3] This sharing is determined via a lexical rule, associated to the lexical entry for persuade. Such a kind of rules and, generally, rules that act upon grammatical relations, such as subj, dir-obj, indir-obj, are employed by the upper level of the grammar (see [5] for details).

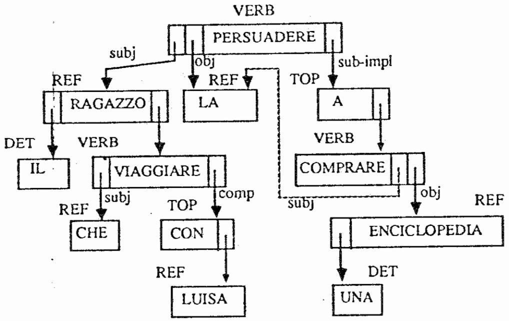

fig.1 An example of a dependency tree. Bold arrows represent standard dependency relation, simple arrows are only structural arcs that do not involve dependency but propagate it from upper levels, dashed arrows are relations of only dependency that account for unbounded links and shared modifiers. The labels on links are assigned by the grammatical relations module.

The constituents of an Italian sentence can be rearranged in many ways, without changing its semantic interpretation , i.e. its "meaning" as defined in [15], even if changes may occur in its TOPIC/FOCUS articulation. Let us consider an example:

- **I bambini vanno a scuola in inverno**
  The children go to school in winter

that can be rearranged as

- **A scuola i bambini vanno in inverno**
- **In inverno i bambini vanno a scuola**
- **Vanno a scuola in inverno i bambini**

(and so on).

It is very simple to write down the rules for this fragment of the language by simply stating that:

S -->$_{id}$ {NP, V, PP}
NP -->$_{id}$ {Det, N}   Det $<_{lp}$ N
PP -->$_{id}$ {P, NP}   P $<_{lp}$ NP

An ID-rule states that the symbol at left immediately dominates the symbols at right, while nothing is specified about the order of the dominated symbols. The precedence relationships are given by the LP-rules. The alternative to such a representation should be the usual way of indicating explicitly all the rewriting rules that account for all of the possible orderings of the constituents in the example above.

Dependency grammars, that have been widely employed for dealing with free word order languages, can be easily described via ID/LP rules. However, in the dependency formalisms it is not sufficient to express the linear order of subconstituents: the order constraints must also include the precedence relationships between the head and each of its dependents. ID/LP relations between any pair of categories can be represented via the entries of two tables, as shown in the figures 2 and 3 for a fragment of Italian.

Each entry in the ID-table states the relation of immediate dominance between the categories Cat$_1$ on the row and Cat$_2$ on the column respectively. The entry is empty (i.e.

Dominated / Dominant	Verb	Noun	Det	Prep	Adj	Aadv	Cherel
Verb	> (↑ head.tense,+) < (↓ head.tense,-)	= (↑ head.tense,+) < (↑ head.tense,-)	●	= (↑ head.tense,+) < (↑ head.tense,-)	> (↑ head.tense,+) < (↑ head.copula,+)	●	> (↑ head.tense,+)
Noun	< (↓ head.tense,+) (↓ el.relative,+) (↑ head.proper,-)	●	>	< (↑ head.proper,-)	=	●	●
Det	●	●	●	●	●	●	●
Prep	< (↓ head.tense,-)	<	●	●	●	●	●
Adj	●	●	●	<	●	> <	●
Aadv	●	●	●	●	●	●	●
Cherel	●	●	●	●	●	●	●

Fig.2- The Immediate Dominance Table.

● immediate dominance is not allowed
< : the dominant category must precede the dominated one
> : the dominant category must follow the dominated one
= : both orders are allowed simultaneously
Constraints related to an order sign apply only if that order is realized
Constraints appearing apart apply independently of the order
Constraints appearing in the same entry are intended to be in XOR relation.

	Verb	Noun	Det	Prep	Adj	Aadv	Cherel
Verb	●	= up: Verb	●	= up: Verb > up: Noun	= up: Verb > up: Noun	●	> up: Verb
Noun		= up: Verb	●	= up: Verb	= up: Verb	●	> up: Verb
Det			●	●	< up: Noun	●	●
Prep				= up: Noun/ Verb	= up: Noun/ Verb	●	> up: Verb
Adj					= up: Noun	●	> up: Verb
Aadv						●	●
Cherel							●

Fig.3 - The Linear Precedence Table.

< : the category on the row must precede the one on the column
> : the category on the row must follow the one on the column
= : both orders are allowed
● the two categories cannot depend on the same node, on the same side with respect to the head
The table also includes information about the categories that must dominate the pair which linear order relation refers to.

contains a black circle) for pairs of categories that cannot be in a dependency relation, as is the case for the pair <VERB,DET>. The sign < says that the head, of category Cat_1 must precede its dependent, of category Cat_2. Conversely, a head Cat_1 is preceded by its dependents Cat_2 if the sign is >. A sign = is used to represent the possibility of a simultaneous presence of a dependent Cat_2 on both sides of the head Cat_1. The possible alternatives in an ID-entry are intended to be in an exclusive-or relation, hence only one of them can be satisfied. Each alternative is attached with a number of constraints on the features, that must be tested on the nodes of the dependency tree, once they are gathered via the percolation of the values upwards from the substructure. Let us consider few examples: all the categories immediately dominated by VERB, except CHEREL, can stand on both sides of it only if the verbal form is tensed; the constraint (↓e1.RELATIVE,+) in the entry for NOUN-VERB dominance means that the feature RELATIVE must be present in the first element of the VERB node beneath the noun. Such a feature will be propagated up from the subtree dominated by the verb, once it has been originated by the lexical entry of a relative pronoun (belonging to CHEREL or the other categories of the RELPRON class). This constraint is necessary to state that a noun-verb dominance is feasible only in the context of a relative sentence, whose first substructure must include a relative pronoun.

These constraints on features' values are of the same form as the $=_c$ equations in the LFG formalism [1]. This means that such constraints are only check equations on the features' values: they cannot introduce a new value into the node information.

The LP table introduces further constraints on the structures allowed by the ID-table. It describes the precedence constraints among the modifiers at the same level and on the same side with respect to the head. Such information allows to reject, as ill-formed, sequences such as

- ***bianco il cavallo**
  white the horse

where the adjectival modifier precedes the determiner. Each entry in the LP-table specifies the linear precedence between two nodes of the categories given by the row and the column respectively. The conditions "up" specify the governor category (ies) under which the precedence constraint holds. Hence a linear precedence statement between two categories is not absolute, but can change, depending on the governor category. This is somewhat different from the GPSG framework, where the presence of non-terminal symbols allowed to ignore many details. For example, if a noun has an adjective and a verb (i.e. a relative sentence) as modifiers, the adjective must precede the relative sentence, in order to avoid expressions like

- **il gatto che .si arrampica sull'albero bianco**
  the cat that is climbing the tree white

if we want to mean that white is the colour of the cat. On the contrary, if an adjective and a verb are modifying a verb, they can appear in any order, of course satisfying the constraints imposed in the ID-table. In the examples

- **e' bello andare al mare** (a)
  (It) is nice to go to the sea
  (ADJ < dependent VERB)

- **andare al mare e' bello** (b)
  To go to the sea is nice
  (ADJ > dependent VERB)

the copula **e'** *(is)* has the same two modifiers in inverse orders in the two sentences, **bello** *(nice)* and **andare** *(to go)*.[4] The specification of the governing category inside the linear precedence constraints is useful for the lack of different (non-terminal) symbols in dependency grammars. If we used different names for

[4] There exist other constraints that we have not considered here and that state the ungrammaticality of sentences like "bello andare al mare e'", or "e' andare al mare bello". Surely these sentences appear to be not very nice to a native speaker: their ungrammaticality can be asserted with a precedence constraint inserted in the ID-table, that forbids the separation of the copula and its adjective.

the verbs that depend upon a noun and the verbs that depend upon another verb, we would not need to add such a specification about the common parent of two modifiers; phrase structure grammars, that employ non-terminal symbols, call the formers, together with their substructures, Relative sentences, and the latters, with their subtrees, Verbal Complements or VP'.

The inspection of the tree in figure 1 and of the ID/LP tables in figures 2 and 3 makes clear that the tables describe accurately the constraints that a tree structure, as introduced in the section 2, must satisfy.

4 Conclusions

We have presented a dependency grammar that covers some basic phenomena of the Italian language. The grammar is represented via ID/LP tables that state the relations of immediate dominance and linear precedence between pairs of categories.

Many phenomena, as the assignment of grammatical relations to the substructures in the dependency tree or the treatment of coordination are not dealt with in the grammar introduced, since we believe that they belong to a different level of linguistic knowledge which is not purely syntactic.

Some details are left unspecified in the basic module described, because of the space constraints. Let us cite someone. The number of nouns that depend upon a verb is usually restricted to one or two (or three in English, in case of bitransitive verbs): in our formalism such a restriction is posed by the module that deals with grammatical relations, that will reject a greater number of nouns not marked by prepositions. The grammar distinguishes between exactly one and more than one dependents, by allowing a category to be preceded by itself in the LP-table. The case of obligatory modifiers, in the sense of dependents that are not optional, is treated with the introduction of constraints on the features' values, that must correspond to the lexical entries of the obligatory dependents.

The framework introduced represents the basis for an automatic system of language comprehension. In the past years, we have developed a parser that builds dependency structures for Italian sentences [6,7,8]. Although the parser is quite efficient (it operates quasi-deterministically), the main disadvantage is the absence of an explicit grammar. The rules are hand-coded and it is not easy to understand which are the linguistic constraints they embody. This paper has presented a solution to this problem, by introducing a formalism for expressing dependency grammars. The grammar, represented as facts in Prolog, can be translated by a precompilation step into an internal format that is suitable to be interpreted by a simple parsing algorithm (see [4] for details), that builds dependency structures that are subsequently interpreted by a semantic component.

References

[1] J. Bresnan, *The mental representation of grammatical relations*, MIT Press, 1982.

[2] G. Gazdar, E. Klein, G. Pullum, I. Sag, *Generalizad Phrase Structure Grammar*, Basil Blackwell, Oxford, 1985.

[3] R. Hudson, *Word Grammar*, Basil Blackwell, Oxford, 1984.

[4] V. Lombardo, *Parsing Dependency Grammar*, Submitted for publication, 1991.

[5] L. Lesmo, V. Lombardo, *The Assignment of Grammatical Relations in Natural Language Processing*, Technical Report, Dipartimento di Informatica, Universita' di Torino, 1990.

[6] L. Lesmo, P. Torasso, *A Flexible Natural Language Parser based on a two-level Representation of Syntax*, Proc. of the 1st Conference ACL Europe, Pisa, Italy, 1983, pp.114-121.

[7] L. Lesmo, P. Torasso, *Analysis of Conjunctions in a Rule-Based Parser*, Proc. ACL 85, Chicago, USA, 1985, pp.180-187.

[8] L. Lesmo, P. Torasso, *Weighted Interaction of Syntax and Semantics in Natural Language Analysis*, Proc. IJCAI 85, Los Angeles, USA, 1985, pp.772-778.

[9] P.H. Matthews, *Syntax*, Cambridge University Press, 1981.

[10] I. Mel'cuk, *Dependency Syntax: Theory and Practice*, State University of New York Press, Albany, NY, 1987.

[11] V. Petkevic, *A New Dependency Based Specification of Underlying Representations of Sentences*, Theoretical Linguistics, vol.14, No. 2/3, 1987.

12] M. Platek, J. Sgall, P. Sgall, *A Dependency Base for a Linguistic Description* in: Sgall P. (ed.), *Contributions to Functional Syntax, Semantics, and Language Comprehension*. Praha, Academia, 1984.

[13] C.J. Pollard, *Generalized Phrase Structure Grammars, Head Grammars, and Natural Language*, Ph.D. Thesis, Stanford University, 1984.

[14] P. Sgall , E. Haijcova, L. Nebesky, A. Goralcikova, *A Functional Approach to Syntax in Generative Description of Language*, New York, 1969.

[15] P. Sgall , E. Haijcova, J. Panevova, *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*, D. Reidel Publishing Company, 1986.

[16] S.M. Shieber, *Direct Parsing of ID/LP Grammars*, Technical Note 291R, SRI Int., November 1983 (also in Linguistics and Philosophy, vol.7, n.2).