

EXTENDED ACCESS TO THE LEFT CONTEXT IN AN ATN PARSER

Irina Prodanof and Giacomo Ferrari
Istituto di Linguistica Computazionale
Via della Faggiola 32
I-56100 Pisa, Italy

ABSTRACT

Some Italian sentences related to linguistic phenomena largely known and recently discussed by many computational linguists are discussed in the framework of ATN. They offer certain difficulties which seem to suggest a substantial revision of the ATN formalism. The theoretical assumptions and an experimental implementation of such a revision are presented, together with examples. Many related theoretical points such as some psycholinguistic implications and the relationship between deterministic and non-deterministic hypothesis are also briefly discussed.

I. INTRODUCTION

Certain types of sentences seem to defy the abilities of several parsers, and some of them are being now discussed by many computational linguists, mostly within the deterministic hypothesis.

An examination of their treatment within the traditional ATN paradigm seems to suggest that the real discussion is about how to access the left context and what its form should be.

II. ACCESS TO THE LEFT CONTEXT

A. ATN Grammars

An ATN grammar is a set of networks formed by labelled states and directed arcs connecting them. The arcs can recognize terminal (words) and non-terminal (lexical categories) symbols or recursively call for a network identified by the label of an initial state. When such a call

occurs, the parsing algorithm suspends the current computation and starts a new level of computation. Usually, each network recognizes some linguistic unit such as Noun Phrase (NP), Prepositional Phrase (PP), and Sentence (S) and any recursively embedded call to one of them corresponds to a level of computation.

The parsed parts of the input string are stored (SETRed) into registers as they are recognized. At the end of the network these registers are combined (BUILDeD) into a parse node and returned (POPeD) to the calling net. Appropriate functions can return the content of a register (GETR) or transfer it to another register (combination of SETR with a GETR). This last operation is equivalent to i) the renaming of a register, if the source register is successively set to a different value, ii) the initialization of a register at a lower or higher level, if SENDR or LIFTR are used.

Initialization is commonly used to i) raise lexical features to a higher level where they are used for tests (ex.: subject-verb agreement), ii) pass possible antecedents to lower levels where a gap may be detected in an embedded clause.

B. Difficult access to registers

1. Filler-gap linking

The antecedent passing may cause a theoretically unlimited increase in storage load. By the standard procedure, the analysis of the ambiguous sentence(1)

(1) Giovanni disse che aveva mentito
John said that (he) had lied

"Giovanni" is always SENDRed as possible SUBJECT of a complement, as soon as "disse" is recognized as an STRANS verb. As no subject NP is met after "che", an interpretation is yielded with "Giovanni" in subject position. The second interpretation is produced simply by successively setting the SURJ register to a dummy node, which

(1) The ambiguity of this sentence is the same as its English translation where "he" can be bound either to "John" or to someone else mentioned in a previous sentence. Italian has a gap instead of a pronoun.

remains unfilled. The same treatment is recursively applied to sentences like

- (2) Giovanni pensava che _____ avrebbe raccontato
John thought that (he) would have told
a tutti che _____ aveva fatto una
to everybody that (he) had done a
scoperta
discovery

where "Giovanni" must serve as subject of both the first and the second (linearly) complement.

Instead, in the sentence

- (3) Giovanni disse che i suoi colleghi avevano
John said that his colleagues had
mentito
lied

as the NP "i suoi colleghi" is analysed, it replaces the SENDRed "Giovanni" in the SUBJ register and the correct interpretation is popped.

A more complex treatment is required for the sentence

- (4) Giovanni pensava che i suoi colleghi avrebbero
John thought that his colleagues would
raccontato a tutti che _____ aveva fatto
have told to everybody that (he) had done
una scoperta
a discovery

where "Giovanni" must get through the first complement and reappear in the second (embedded) one. If SENDR is used, a transit register R1 with the same content as the initialized SUBJ register is to be passed down together with it. When the subject of the first complement ("i suoi colleghi") is found, it replaces "Giovanni" in SUBJ but not in R1. The new SUBJ "i suoi colleghi" and R1 "Giovanni" are again SENDRed to the embedded complement where the agreement with the verb correctly selects, as subject, the content of R1. Now, as the number of levels which to "jump over" is in principle unlimited and each one may have its own subject to be SENDRed, a transit register for each new subject is needed. Thus, for the sentence

- (5) Giovanni era sicuro che i suoi nemici
avrebbero rivelato alla stampa che sua moglie
aveva detto un giorno che l'aveva picchiata(2)

when processing the last complement three registers containing the three possible subjects will be available and shall be visited in order to find the right one(3).

(2) "John was sure that his enemies would have disclosed to the press that his wife had once told that (he) had beaten her". We give here, for clarity, the parenthesized form of this example:

(Giovanni era sicuro (che i suoi nemici avrebbero rivelato alla stampa (che sua moglie aveva detto un giorno (che _____ l'aveva picchiata))).

Notice that in this example the subject-verb agreement is sufficient to select the right antecedent, but this is not always the case.

(3) A possible alternative, equally tricky, is the use of the HOLD-VIRT couple.

2. Lexical features raising

A storage overload may also be caused by the need for raising lexical features. Morphological features are necessary at the least in the test of subject-verb agreement. This is done by LIFTRing in ad hoc registers gender and number from the NP level to the S level. If the NP is popped in a possible subject position the test for agreement may take place by comparing the content of those registers with the corresponding features of the verb. However, there are cases such as ex.(1) in which such information must be used again in the course of the analysis for another (agreement) test. Those features must be, therefore, copied in appropriately labelled registers in order to i) flag their relation to the subject and ii) prevent them from being erased when the same features are LIFTRed from the following NP.

The same need for ad hoc storing may be shown for the object and few PP complements. For example, in the sentence

- (6) Il capitano disse ai _____ soldati che
The captain said to the soldiers that
_____ marciavano pietosamente
(they) were marching peacefully

as the NP "i soldati" embedded in the "a-PP" is the subject of the complement, its gender and number must travel up through PP and S and down to the complement, in order the subject-verb agreement to be satisfied. This implies that at a given moment other registers containing morphological features coexist at the S level with those of the subject.

An obvious remedy to this proliferation of registers consists in adding these features to the NP or PP structure, POPping them, and extracting them from those structures when needed. But classical ATN does not provide a suitable function, a form that should return only a part of a stored tree after having searched through it. Now, although such functions are not particularly complex to design [5, 7], they are not perfectly consistent with the ATN general formalism which does not take into consideration the extraction of pieces of already processed information from the structure they have been inserted in.

C. A functional perspective

1. A generalized retrieving action

If we look at the discussed examples from an entirely functional viewpoint we can describe them as having in common the need for retrieving

information somewhere back in the already built structures; the tricky solutions presented above are, in fact, a way of accessing parts of the left context. These sometimes correspond to the entire content of a register and sometimes to a fragment of it.

We will assume, then, that the left context is stored in a space of memory, equally accessible from any level and that retrieving always concerns fragments of it. At any point of the process this structure contains the current hypothesis about the analysis of the parsed segment of the input from the beginning; hence we will refer to it as Current Global Hypothesis (CGH).

The retrieving action will have two participants, a symbol that triggers the action (trigger) and the information to be retrieved (the target of the action).

In this frame all the different procedures discussed above may be reduced to a single general algorithm of three steps,
 i) identification of a trigger (a gap to be filled, a verb which demands for the subject-verb agreement test)
 ii) extraction of constraints which must guide the search for the target, and
 iii) retrieving of the required information.

On this functional ground, the same description fits to the binding of an anaphoric pronoun to its antecedent [7] as in

- (7) a) Giovanni disse a Maria che voleva
 John said to Mary that (he) wanted
 sposarl-la
 marry her
 as opposed to
 b) Giovanni disse a Maria che voleva
 John said to Mary that (he) wanted
 sposarsi
 to marry

The function that searches back may be constrained by several types of restrictions, including i) morphological features, i.e. the gender and number of the pronoun or those required for agreement by the syntactic environment (e.g. the verb), ii) syntactic idiosyncrasies of some lexical items such as the STRANS verbs that determine which of their arguments is to be the subject of the complement, iii) semantic features that may be introduced in the process, and iv) syntactic determination of the scope of the search.

2. Manipulations of the left context

Retrieving of the antecedent may actually correspond to two different operations depending upon whether the antecedent to be bound linearly precedes or follows the symbol it is to be bound to.

In the cases discussed above the gap or the pronoun always follow their antecedent and it is therefore possible to immediately insert the

binding pointer. Moreover, in many common sentences the antecedent linearly follows its dependent, as in

- (8) Quando si arrabbia, Giovanni diventa rosso
 When (he) gets angry, John becomes red
 (9) Se lo vedi, saluta Giovanni
 If (you) see him, say hello to John
 da parte mia
 on my behalf

In this case, the binding should take place in two steps, the flagging of the need for a forward binding and the moving of the pointer from the antecedent, once detected, to the flag. Notice that this class of sentences entirely lies beyond the abilities of the classical ATN framework.

Relative pronouns also need to be bound to an antecedent and, besides, are the surface signal of an embedding. No special processing difficulty is proposed by sentences like

- (10) Il ragazzo che corre
 The boy who runs

where the relative pronoun occurs exactly where the embedding begins. In this case a scope restriction can limit the search for an antecedent to the immediately preceding NP.

But in the sentences

- (11) Il ragazzo del quale ti parlavo
 The boy about whom I was talking; to you
 (12) Il ragazzo del cui padre ti parlavo
 The boy about whose father I was talking
 to you

the relative clause boundary is set one word before the relative pronoun, and in

- (13) Il ragazzo del padre del quale
 The boy about the father of whom
 ti parlavo
 I was talking to you

where the phenomenon known as pied-piping occurs, such a boundary may be located several words before the relative pronoun.

In an ATN these cases meet the initial set of arcs which recognize a PP embedded in an NP, as in

- (14) il ragazzo dell'ascensore
 the boy of the lift

and the correct interpretation is obtained only by backtracking (for 11 and 12) and again using a set of transit registers (for 12). No solution is known for sentence (13).

In the framework we have been discussing, an action which structurally modifies the left context can be proposed. It should embed the component(s) being processed in a relative clause as the relative pronoun in set.

A third type of access to the left context is the relabelling of a processed component, already used for the passive transformation.

III. EXPERIMENTAL IMPLEMENTATION

A. General assumptions

Most of the above discussion is based on our own experience with an ATN parser, although valid also for other parsers. Some of the linguistic phenomena we are trying to functionally classify have been recently discussed in the frame of the deterministic parsing and some progress in the treatment of those sentences has been done. However we prefer to stick to the non-deterministic hypothesis, because no valuable argument seems to stand against the idea of ATN as a collection of alternative heuristic strategies representing those used in the human sentence processing. On the contrary it is possible to introduce in an ATN parser some mechanism for selecting strategies in function of a text or a sublanguage [11].

Our assumption is, then, that the possibility of backtracking must be kept, but restricted by

- i. introducing linguistic constraints to guide the choice of alternatives;
- ii. manipulating the left context in some cases in which backtracking can be intuitively charged on inadequacies of the parser rather than on points of real linguistic ambiguity.

A second assumption more related to the current implementation of the system refers to the grammatical formalism. A functional representation a la McKay [14] is used. A detailed explanation of this formalism is not necessary to the understanding of this paper. It is enough to know that its basic unit is the attribute-value pair, where an attribute is a symbol (label) and a value is a symbol or another functional description. In the sentence

(15) He killed her

a possible pair is SUBJ=he, or SUBJ=HEAD=he together with SUBJ=CAT=PRON etc. Any sequence of symbols is a path leading to a value; thus the value of SUBJ=HEAD is "he".

The functions that access the data structure are specifically designed to treat this type of representation but we think that they could be easily generalized. The term "component" will be used to identify the set of paths starting from the same label (radix).

The reasons of this choice are as follows:

- i. addressing fragments of information is more easily done by following a path of labels than by visiting a unlabelled tree;

- ii. functional grammar allows any depth of linguistic representation through the left adjunction of labels;
- iii. functional syntactic representation and lexical features (also expressed in terms of attribute-value pairs) may be treated in a uniform way.

B. The parser

The basic features of the ATN parser used in our previous experiments [19], i.e. the network formalism and the parsing algorithm, are kept, while the data structure and the whole set of actions and forms have been modified. Also the use of the push-down mechanism has been modified to some extent.

The data structure is a list which is mainly accessed with a typical LIFO stack policy. It represents a unique memory space non splitted into registers. The state saving mechanism necessary for the treatment of non-determinism is provided by ND-LISP [2,17], the dialect of LISP in which the system is written. At any point in which non-determinism is called, the previous context, in particular the data, is saved and only the new values are set in the current context. Therefore, there is no difference between the use of the traditional register table and this special list since both of them are handled in the same way. This (LIFO) list contains at any point of the process the CGH, i.e. the entire left context literally represented in terms of attribute-value pairs.

We give hereafter a list in Backus notation of the functions which access the CGH.

1. Actions

- a. <storing actions>::=
ADD pair location |
ASSIGN label path
<location>::= NIL | <form>
<label>::= any label
<pair>::= label value
<value>::= * | <form>
- b. <list manipulation>::=
PUSH |
POP |
INSERT data item
<data>::= any data
<item>::= <form>

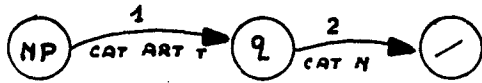
2. Forms

- FIND path test level dtype |
- FINDVAL path test level dtype |
- LOCATE path test level dtype
<path>::= <label+>
<test>::= T | any test
<level>::= T | CL
<dtype>::= T | ND | L

The basic storing action is ADD which is used to store any incoming piece of structure. The string

(16) il cane
the dog;

recognized by the network



is stored by the following actions

1. (ADD (DET *))
2. (ADD (HEAD *))

If location is NIL, the current component is meant, otherwise the form LOCATE specifies the path leading to the radix to which the new pair is to be ADDED.

Relabelling of a component is done by the action ASSIGN. In the sentences

- (16a) Il cane mangia
the dog eats
- (16b) Il cane e' mangiato
the dog is eaten

the NP 'il cane' will be first labelled FOCUS or FIRSTNP. Then, after having recognized the verb, the action

```
(ASSIGN SUBJ (LOCATE FOCUS T CL T)) or
(ASSIGN OBJ (LOCATE FOCUS T CL T))
```

will properly classify the NP as

```
SUBJ or OBJ = FOCUS = DET = IL
              =HEAD = N = CANE
```

Extraction of information is done by the forms FIND, which returns a pair, and FINDVAL, which returns only the value of a pair. LOCATE works exactly in the same way, but returns a pointer to a given radix. All the three functions can work in different modes. They can search either only the current level (CL) or through the entire list (T). In this latter case the current level is excluded and, if no further options are specified, the lower (the nearest to the top) occurrence is returned. Another option (dtype) returns all the occurrences either appended in a list (L) or one by one, non-deterministically (ND). A third option evaluates conditions in order to select the component identified by the specified path.

In sentence (4) the antecedent retrieving is performed by the form

```
(FINDVAL (SUBJ) (AND (EQ (FINDVAL (SUBJ NUM) T T
ND)
(FINDVAL (HEAD NUM) T CL
T))
(EQ (FINDVAL (SUBJ GEN) T T
ND)
(FINDVAL (HEAD GEN) T CL
T))) T ND)
```

which searches for a subject through all the levels non deterministically. Such an NP must agree in number and gender with the current level head, i.e. the verb(4). If this expression is embedded in the function

```
(ADD SUBJ _____)
```

the correct subject(s) is (are) copied in the complement (5).

The three last actions, PUSH, POP, and INSERT, manipulate the items in the list. PUSH adds a new (empty) item in front of the list. The elements of the component being analysed (phrases or sentences) are ADDED in this top item, which has been therefore referred to as current level. POP removes the current top-item and embeds it into the new top-item, possibly assigning a label to the corresponding component. Finally INSERT inserts an item, corresponding to a new level, somewhere back between 'item' and the front part of the list, and fills it with 'data'.

List manipulation takes place independently from the starting or the ending of the process expressed in a subnet. Thus a component can be POPed after the end of its recognition procedure, when also its function is clarified.

The arc recognizing an object, for ex., can be expressed as follows

```
(START NP T
(COND (FIND (SUBJ) T CL T)
(POP OBJ))
(TO q1))
```

which means that if there already is a subject, the current component must be popped with the label OBJ.

The use of the INSERT function is primarily motivated by the treatment of certain relative clauses. Relative pronouns are surface signals that trigger the embedding into a relative clause of the currently processed component(s). In the sentence

- (17) Il libro della trama del quale parlavamo
The book about the plot of which we talked

such an embedding takes place immediately after 'libro', thus producing

(4) An "anaphoric" facility is also implemented not to repeat an embedded form with the same argument as the embedding one.

(5) We do not intend to suggest that the correct mechanism of trace/antecedent binding is the copying of the antecedent in the trace position. A slightly modified version of this function might produce the insertion of the antecedent path, as in the orthodox functional grammar. The procedure, however, does not substantially change.

(18) (il libro (RELCL (della trama (del quale) parlavamo))

The general rule may be formulated as follows: "a new level labelled RELATIVECLAUSE is to be inserted immediately after the antecedent of the relative pronoun". Analysis of (17) will therefore proceed as follows;

- when the relative pronoun "quale" is encountered, the form

```
(FIND (HEAD) (AND (EQ (FINDVAL (HEAD GEN) T T ND)
                    (FINDVAL (DET GEN) T CL T))
 (EQ (FINDVAL (HEAD NUR) T T ND)
     (FINDVAL (DET NUR) T CL T)))
T T)
```

returns the lower head which agrees in number and gender with the determiner of "quale" ("quale" is both masculine and feminine), i.e. "libro". This is the antecedent.

- The function

```
(INSERT RELCL (LOCATE...as for FIND))
```

inserts a new item with label RELCL.

- On the same arc the function (POP DI-ARG) embeds "del quale" in "della trama" and a second POP embeds (della trama (del quale)) in the recently inserted relative clause component.

- The recognition of a relative clause is continued by a (START S....) arc. The control is finally then returned to the NP process with the complex NP "il libro..." as the current component.

IV. ADVANTAGES

A. Efficiency

The parser we have been presenting is based on the core algorithm of the ATN. Our modifications affect the set of forms and actions and the data structure. The parsing algorithm, therefore, keeps the efficiency of traditional ATN. We have already shown that the storing of the data structure does not present any special difference from the traditional registers system, even in relation to the treatment of non-determinism. The memory load is, therefore, strictly a function of the length of the parsed segment of the input and no overhead determined by manipulations of structures is added as in the case of transit registers.

The actions and forms are equivalent to the traditional ones, but for the fact that most of them must visit the whole left context for every access. Anyway this effect hardly balances the setting of transit registers. In fact, it is worth noting that in the majority of common sentences such accesses are very reduced, so that

no substantial difference exists in comparison with the traditional register access. In the discussed complex cases the access to the CGH is a known function of the length of the list, i.e. of the depth of embedding of the current level. Within any item search proceeds linearly as for any ordinary pattern-matching.

The only substantially new fact is the possibility of embedding the current component; this eliminates the need for backtracking, at least for some sentences.

In conclusion, it seems that if there is a difference from the traditional ATN it is in favour of the version presented here.

B. Generalization and modularity

The set of actions and forms presented seem to provide a functional description of many linguistic phenomena. They can be regarded as linguistic (procedural) generalizations, at least on the functional ground. This supports our claim that linguistic phenomena can be described, independently from the formalism that expresses them (the grammar), in terms of general operations. This set of operations is open-ended and can, therefore, be increased with functions designed for the treatment of new phenomena, as they are discussed and described. Furthermore, those actions can be taken to represent mental operations of the language user, thus providing a valuable frame for psycholinguistic experiments.

It is obvious that this view strongly inclines towards the idea of parser as a collection of heuristic strategies and processes and also offers a symmetric alternative to the HOLD hypothesis. According to this hypothesis there are points in a sentence in which comprehension needs a heavier memory load; instead in our view an overhead of operations is suggested. Anyway the distinguished phenomena coincide, thus keeping the integrity of the experimental data(6).

C. Naturality

Our hypothesis seems more natural in two ways. It embeds into a non-deterministic frame some operations very similar to some of those designed and discussed in the deterministic hypothesis [3, 4, 15, 16, 19]. The result is a strong limitation of the effects of non-determinism, at least for those cases they are designed to treat. It is interesting that starting from two opposed viewpoints comparable results are obtained. Nevertheless, as stated above, we think that imposing constraints to a non-deterministic model is more natural than being imposed global constraints by the assumption of determinism. In the first hypothesis, in fact, a deterministic behaviour of the parser may be ultimately

(6) In this paragraph we refer to the ideas and the experiments presented by Kaplan [12, 13] and Warner & Maratsos [21].

obtained, in some points, as a result of observation of real linguistic restrictions while those phenomena such as ambiguity which can be adequately treated only in a non-deterministic frame, are not "a priori" ruled out. Then, a model such as the non-deterministic one, in which there is place for the study of human heuristic constraints, seems more attractive and natural.

Our hypothesis seems intuitively natural also in so much as it tries to propose a "theory of guess". During the comprehension of a sentence guesses (CGH's) are progressively enriched and stored in a space of memory. During this process errors may be done. For some of them it is enough to modify the previous guess while for others a real backtracking and reanalysis is necessary. Although the distinction between the two types of errors is unclear, it provides a valuable frame for further research in the domain of computational linguistics as well as psycholinguistics. In particular it seems to distinguish in the activity of sentence comprehension a phase of structuring from a phase of perception. Errors occurring in the former are remedied by modifying a guess, while those occurring in the latter need backtracking and the choice of another strategy.

V. PERSPECTIVES

A more serious systematization of the proposed functions, as well as the extension of the model to more and more linguistic phenomena are obvious extensions of the present project.

Another direction where investigation seems to be particularly fruitful is the relation between syntax and semantics. On one hand, the fact that the result of the analysis is progressively stored in a unique space of memory do not impose special constraints on the structure of the analyzed string. On the other hand, many of the presented functions include parameter slots for conditions which may be filled with any kind of test. This model seems, therefore, to avoid "physiological" boundaries between syntax and semantics. The stored structure can be a semantic one and the tests can also incorporate semantic descriptions. This seems to eventually lead to an easier integration of the two levels. We will present shortly [10] a first approximation to a frame into which such an integration can be realized.

ACKNOWLEDGEMENTS

We would like to thank Pierpaolo Degano, Luigi Rizzi, Franco Turini and Oliviero Stock who read

various early versions of this paper, and contributed to it with their valuable comments.

REFERENCES

- (1) Aho A., Ullmann J.D., The theory of parsing, translation and compiling, Prentice Hall, New York, 1972.
- (2) Asirelli P., Lami C., Montangero C., Pacini G., Simi M., Turini F., MACMA-LISP Reference Manual, NT C75-13, IEI CNR, Pisa, 1975.
- (3) Berwick R.C., Weinberg A.S., Syntactic Constraints and efficient Parsability, in Proceedings of the 21st Annual Meeting of ACL, Cambridge MA, 15-17 June 1983, pp.119-122.
- (4) Berwick R., A deterministic parser with broad coverage, in Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe 8-12 August 1983, pp. 710-712.
- (5) Cappelli A., Ferrari G., Moretti L., Prodanof I., Stock O., Parsing an Italian Text with an ATN parser, NT ILC CNR, Pisa, 1978.
- (6) Cappelli A., Ferrari G., Moretti L., Prodanof I., Stock O., Automatic Analysis of Italian, in Proceedings of the AISB-80 Conference on Artificial Intelligence, Amsterdam, 1 - 4 July, 1980.
- (7) Cappelli A., Ferrari G., Moretti L., Prodanof I., Stock O., Il trattamento di alcuni fenomeni anaforici mediante un ATN, in Atti del seminario "Sull'anafora", Firenze 14-16/12/1981, pp.275-289.
- (8) Cappelli A., Ferrari G., Moretti L., Prodanof I., Stock O., Costruzione, sperimentazione ed estensione di un ATN come modello di analisi del linguaggio naturale, in Ricerche di Psicologia, VII, 25, 1983, pp. 159-164.
- (9) Cappelli A., Ferrari G., Moretti L., Prodanof I., Towards an integrated model of sentence comprehension, in A.Cappelli (ed.) Research in Natural Language Processing in Italy (1981), Giardini, Pisa, 1983, pp. 45-58.
- (10) Cappelli A., Ferrari G., Moretti L., Prodanof I., A framework for integrating syntax and semantics, in Bara B. & Guida G. (eds.) Natural Language Processing, North Holland, (to appear).
- (11) Ferrari G., Stock O., Strategic selection for an ATN Syntactic Parser, in Proceedings of the 18th Annual Meeting of ACL, Philadelphia, June 19-22 1980, pp. 113-115.

- (12) Kaplan R., Augmented Transition Networks as psychological Models for Sentence Comprehension, *Artificial Intelligence* 3, 1972, pp. 77-100.
- (13) Kaplan R., On process Models for sentence Analysis, in Norman D. & Rumelhart D. (eds.) *Explorations in Cognition*, Freeman, S.Francisco 1975, pp. 117-135.
- (14) Kay H., Functional Grammar, in *Proceedings of the 5th Meeting of the Berkeley Linguistic Society*, Berkeley 1979, pp. 142-158.
- (15) Marcus M., *A Theory of Syntactic Recognition for Natural Language*, MIT Press, Cambridge MA, 1980.
- (16) Marcus M.P., Hindle D., Fleck G., D-Theory: Talking about Talking about Trees, in *Proceedings of the 21st Annual Meeting of ACL*, Cambridge MA, 15-17 June 1983, pp. 129-138.
- (17) Montanero C., Pacini G., Turini F., *MD-LISP Reference manual*, NT C76-3, IEI CRR, Pisa, 1976.
- (18) Prodanof I., Ferrari G., Revising and ATN parser, in *COLING82, Proceedings of the 9th International Conference on Computational Linguistics*, Prague, 5-10 July 1982, pp. 101-105.
- (19) Shipman W.D., Marcus M., Towards minimal data structure for deterministic parsing, in *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, Tokyo, August 20-23 1979, pp.815-817.
- (20) Stock O., *ATMSYS: un sistema per l'analisi grammaticale automatica delle lingue naturali*, NI B76-29, IEI CRR, Pisa, 1976.
- (21) Wanner E., Maratsos M., An ATN approach to comprehension, in H. Halle, J.Presnan, G. Miller (eds.) *Linguistic Theory and Psychological reality*, MIT Press, Cambridge MA, 1978, pp.119-161.
- (22) Winograd T., *Language as a Cognitive Process. Syntax*, Addison-Wesley, Reading MA, 1983.
- (23) Woods W., Transition Network Grammars for Natural Language Analysis, in *CACI 13-10*, 1970, pp. 591-606.