UNDERSTANDING NATURAL LANGUAGE THROUGH PARALLEL PROCESSING OF SYNTACTIC AND SEMANTIC KNOWLEDGE: AN APPLICATION TO DATA BASE QUERY

R. COMINO (*), R. GEMELLO (*), G. GUIDA (**),
C. RULLENT (•), L. SISTO (*), M. SOMALVICO (**)

(*) CSELT - Centro Studi e Laboratori Telecomunicazioni S.p.A. - Via G. Reiss Romoli, 274 - 10148 Torino (Italy)
(**) Milan Polytechnic Artificial Intelligence Project, Milano (Italy)

ABSTRACT

This paper describes the main features of the PARNAX system for natural language access (in Italian) to an ADABAS data base. The core of the system is constituted by the analyzer that includes parallel processing of syntactic and semantic knowledge. It is argued that this feature (together with the new macro and micro-analysis technique which is only shortly mentioned in this paper) allowed the system to reach a good linguistic coverage, still ensuring an acceptable degree of efficiency. After the basic architecture and operation of PARNAX have been described, attention is focused on the parallel syntactic/semantic analyzer which is illustrated in detail. The advantages obtained through parallelism are also shortly discussed. Examples of PARNAX operation are presented. References to related works are mentioned, and directions for future research are outlined.

1. INTRODUCTION

This paper discusses a research project devoted to the design of a robust and effective parser for Italian language, that can support a large linguistic coverage, still ensuring an acceptable degree of efficiency. The project has been partially based on previous results by the authors (Guida and Somalvico, 1979, 1980; Guida and Tasso, 1982) and presents several original contributions. Among these are: a two-level analysis strategy that includes a macro-analysis and a micro-analysis phase; a model for semantic processing that is made up of two sequential phases, namely, a nondeterministic part that validates and completes the activity of a syntactic analyzer, followed by a deterministic part that constructs the output internal representation; a parallel algorithm that manages the two cooperating processes of syntactic analysis and the nondeterministic part of semantic analysis.

This research project is supported by the implementation of an experimental system, called *PARNAX,* which is presently running in an INTERLISP version on Siemens 7748 at CSELT (Torino, Italy). PARNAX is a natural language interface to an ADABAS data base containing information on the staff of a sample company.

In this paper, attention is focused on the parallel strategy developed for syntactic and semantic processing. At each step during the parsing of an utterance, the syntactic analyzer proposes candidate syntactic structures for a component of that utterance that fit the given set of syntactic rules. Similarly, the nondeterministic semantic analyzer constructs candidate semantic structures according to a given set of semantic rules. Among these only those structures that can be associated to a corresponding syntactic structure are validated and will be further considered in the following steps of the processing. All other candidate structures are discarded, thus considerably limiting the search space which is actually expanded during the analysis. This enables the system to reduce the nondeterminism of the analysis process, and, therefore, to operate with improved efficiency.

2. BASIC SYSTEM ARCHITECTURE

PARNAX allows casual users to access an ADABAS data base in Italian. It translates natural language requests into NATURAL programs (the ADABAS formal query language) in two steps. First the natural language query is processed by the *ANALYZER,* that generates a semantic representation expressed in an internal formalism, called METANATURAL. METANATURAL is an intermediate language which should allow a sufficient degree of independence of the understanding process of the details of the data base logical schema. Then, the *FORMALIZER* transforms the METANATURAL query into a full NATURAL program. This is eventually supplied to the DBMS which provides the user with the desired answer. System tailoring and updating is ensured by a *KNOWLEDGE BASE MANAGEMENT SYSTEM*

Figure 1 shows the basic architecture and mode of operation of the ANALYZER, that constitutes the core of the system. The analysis is performed at two levels: the upper level, called *macro-analysis,* takes into account the outer sentence structure and suggests possible splitting and normalization of complex or syntactically unusual sentences into simpler fragments; the lower level, called *microanalysis,* parses the sentence fragments and returns the obtained METANATURAL subtrees to the upper level that will compose them into the final METANATURAL tree.

Macro-analysis operation is mainly rule-based: a set of *structural rules* of the type < pattern, fragmentation-normaliza-

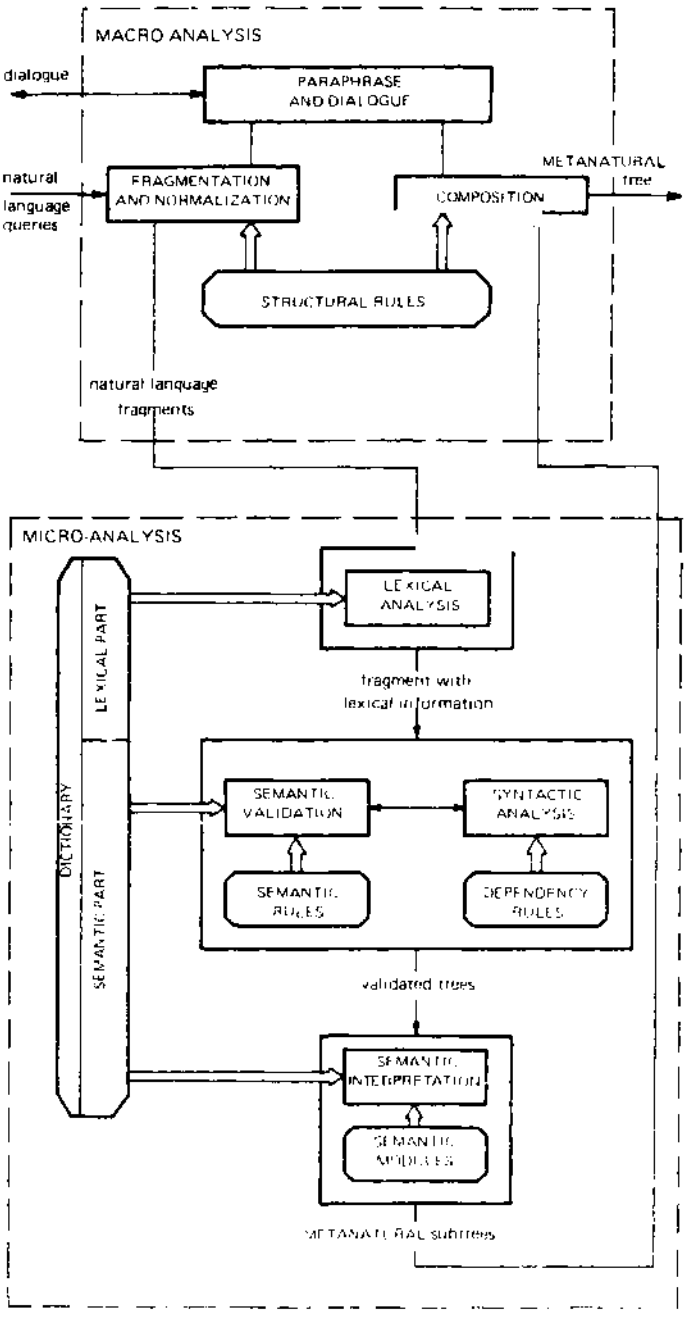

Fig. 1 - Basic mode of operation of the ANAL YZER

tion action, composition action > is used both by *the fragmentation and normalization* module for analysing the outer surface structure of the input sentence, and by the *composition* module for building up the output METANATURAL tree. Macro-analysis also includes a *paraphrase and dialogue* module that manages the user-system interaction and generates echoing paraphrases of the input requests.

Micro-analysis consists of three phases. The first phase, namely *lexical analysis,* classifies the elementary components of the input fragment (words or simple constructs and idioms) according to their lexical type. Its operation is based on a nondeterministic finite-state recognizer that utilizes the *lexical part of the dictionary,* which contains lexical roots and models (Courtin, 1977). The second phase includes two parallel pro cesses: the *syntactic analysis,* which is based on the dependency grammar model (Hays, 1964, Courtin, 1977), and the first part of the semantic analysis, called *semantic validation* The knowl edge bases utilized in this phase are the *dependency rules,* the *semantic rules,* and the *semantic part of the dictionary.* The result of the syntactic analysis and semantic validation is a set of validated trees. A validated tree is a modified dependency tree, where semantically meaningless components have been pruned and appropriate semantic variables have been added Each node of the tree is therefore associated both to a semantic variable that expresses the kind of information o1 the cor responding sentence fragment, and to a pointer to a procedure (semantic module), which will be later used to assign the correct value to the variable. The third phase, namely *semantic interpretation.* is a deterministic process devoted to construct the output METANATURAL tree. It operates through a bot-tom-up activation of the *semantic modules* that are referred to in validated trees, and assignes the appropriate values to the corresponding semantic variables.

3. IMPROVING NONDETERMINISTIC ANALYSIS THROUGH PARALLELISM

Parallelism between syntactic analysis and the first part of the semantic analysis (semantic validation) is used in PAR NAX to improve the efficiency o1 the parsing process. Both syntactic analysis and semantic validation are nondeterministic processes, that can be viewed as problems to be solved by a reduction-to-subproblems mechanism As far as syntactic analysis is concerned the problem is to construct all the correct dependency trees for a given sentence. This problem, once a word in the sentence has been chosen as a candidate root o1 the tree, can be split ted into subproblems. Each o1 them consists in finding all the dependency subtrees whose roots (determined through dependency rules) are just one step below the root of the dependency tree corresponding to the basic problem. Many groups of alternative subproblems may arise, since many groups of roots can be generated according to the dependency rules. A similar situation arises in the case of semantic validation, where the problem consists in finding all the correct semantic trees, and the subproblems in finding the semantic subtrees whose possible sequences of roots are determined by the seman-tic rules. Both semantic and syntactic problems could be separately solved by two distinct search processes in their respective search spaces (AND-OR graphs), which are generally very large.

Our choice is to solve both problems in a joined way, using two parallel, step-by-step search processes in the two search spaces. To this purpose a correlation between the two search spaces is necessary. This is defined in terms of an association criterion among problems in the two spaces (which is generally a many-to-many relation). According to this, for each problem P in one search space only those reductions of P are taken into account which have at least one association among the possible sets of successor problems of the problem P' associated, in the other space, with P Thus, most of the search for non existing solutions (semantic trees without corresponding syntactic trees, and vicoversa) is avoided, with the desired result of constructing, with considerably minor effort, only the set of the semantic trees which are also syntactically valid (validated trees).

In the current implementation of PARNAX the parallel processing strategy above outlined is realized in a sequential way in which syntactic and semantic steps alternate in the analysis, due to the limitations of the available INTERLISP system.

4. AN EXAMPLE

A simplified example of parallel syntactic/semantic processing is shown in Figure 3. The focus is centered on one step of the analysis concerning the segment "NATI A TORINO PRIMA DLL 1958" (for the english translation refer to the Appendix)

Lexical and semantic types are extracted from the dictionary and are self explanatory. For what concerns the structure of semantic rules, taking rule I as an example, its meaning is the following one. a segment conveying the intorma tion *(A)* SELLCION (i.e., how to select an object from a set) must contain a word of lexical type "verb" and semantic type "AT" inside that segment the information (VALUE must be found (it is underlined) while ("'DATE and ("'COMPARATIVE may possibly be found, in any order. F 13 is the pointer to the semantic module used during the bottom-up activation of the semantic interpretation phase. The structure of dependency rules is evident considering rule 1 as an example; an adverb may be the dependent of a verb; the weights (-20, +10) determine the possible positions of the dependent (Courtin, 1977).

At the point of the analysis shown in Figure 3, two associated problems are considered; the former (P) consists in looking for a dependency tree rooted in "NATI", the latter (M) consists in looking for a semantic tree rooted in ©SELECTION. They are reduced to five and four subproblems respectively. Only subproblems SP1 and SP3 can be associated to subproblems SM1 and SM3 respectively. All other subproblems are pruned since no problem in { SM2, SM4 }can be associated to some problem in { SP2, SP4, SP5 }. The analysis continues with problems SP1, SP3, SM1, and SM3, but only SPI and SMI will lead to the solution (boldface in Figure), while SP3 or SM3 will fail in following steps.

5. THE POWER OF PARALLELISM

Some empirical experimentations have been done to estimate the benefits obtained through parallel execution of syntactic and semantic processing. Two samples A and B of about 50 and 30 sentences respectively have been considered. A has been chosen from a set of sentences proposed by casual users; B has been constructed to contain a broad spectrum of different ways of expressing the same request. Fach sentence in the two samples has been parsed first using the syntactic analyzer alone (I), and then using the parallel syntactic/semantic processor (II) thus generating the complete validated trees as well. The following parameters have been estimated, cardinality of the syntactic search space expanded in both cases I and II, and analysis time for syntactic processing alone in case I and parallel syntactic/semantic processing in case II. The mean values of the ratios of the above parameters in cases I and II (case I/case II), tor both samples A and B, have been computed. 1 he table of Figure 2 illustrates the results obtained.

Note that the sum of the time required for both syntactic and semantic analysis (performed in parallel) is less than the time needed for the syntactic analysis alone. We also outline that the benefits of parallel processing increase with the length of the sentences.

	Sample A	Sample B
CARDINALITY RA1 I()	181	*2.61*
ANALYSIS TIME RATIO	1.36	2.03

Fig. 2 - Experimental results of parallel syntactic/semantic processing.

6. EXPERIMENTAL ACTIVITY

An experimental version of PARNAX has been fully implemented at CSELT (Torino, Italy) in the last two years. The system is connected to an ADABAS data base containing information on the staff of a sample company. It supports free Italian language interaction (including single sentences as well as sequences of interrelated utterances), and is able to deal with ungrammatical sentences, ellipsis and telegraphic: forms, a class of anaphoric references, and multiple queries. PARNAX can also engage the user in a limited dialogue, that includes echoing user's request with a system generated paraphrase, presenting alternative interpretations when reference ambiguities occur, and asking the user tor validation of the proposed interpretation in the cases where other interpretations could exist.

PARNAX is written in INTERLISP and is running on Siemens 7748. PARNAX dimensions are about 300 kbytes for algorithms and 350 kbytes tor knowledge bases (the present version includes about one thousand lexical roots, 9 structural rules, 130 dependency rules, 70 semantic rules, and 27 semantic

sentence: ELENCA I DIPENDENTI NATI A TORINO PRIMA DEL 1958

lexical type: verb prep prop-name adverb prep num

semantic type: AT – VA CO – DT

(attribute) (value) (comparative) (date)

semantic rules: I) (@ SELECTION [AT,verb] ➡ { @ DATE, @ COMPARATIVE, @ VALUE }, F-13 III) (@ COMPARATIVE [CO, adverb] ➡ { }, F-12

II) (@ DATE [DT, num] ➡ { @ COMPARATIVE, @ DATE-M/D }, F-25 IV) (@ VALUE [VA, prop-name] ➡ { }, F-1

dependency rules:

1) verb * adverb: -20, +10
2) verb * prop-name: +20
3) verb * num: -10, +5, +25
4) prop-name * prep: -20
5) num * prep: -15
6) num * adverb: -25, +25

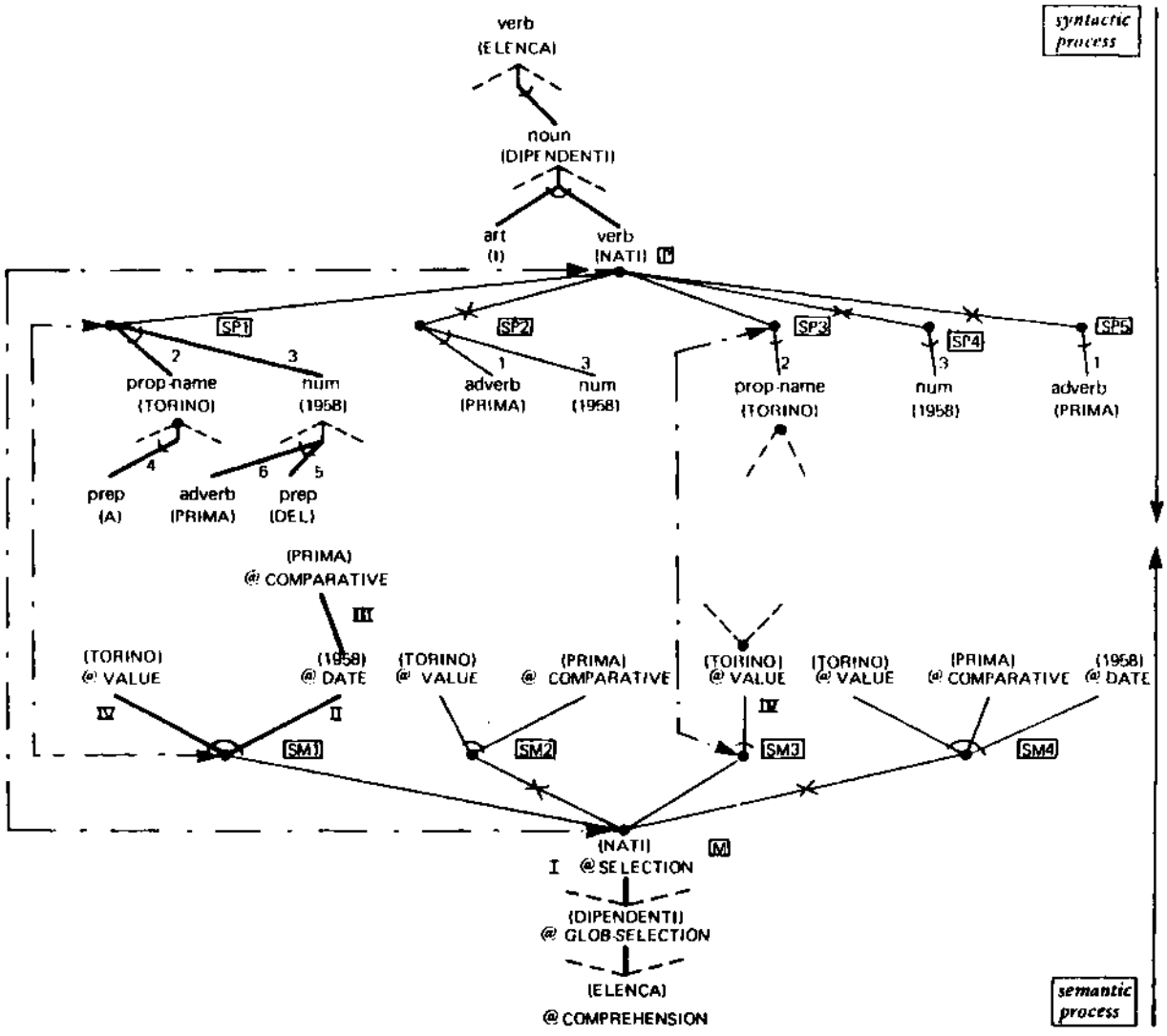

Fig. 3 - A simplified example of parallel syntactic/semantic processing

modules). The average time needed to parse an average-length sentence (9 words) is approximately 3 CPU seconds (without accessing the DBMS).

Figure 4 reports a short dialogue with PARNAX. F igure b shows a small sample of different ways of expressing the same request (all ot which are correctly understood by PARNAX) that should give an idea of the linguistic coverage of the system. For the English translations of the examples refer to the Appendix.

> CONTA I DIPENDENTI DELLA DIVISIONE INFORMALICAL NATI DOPO IL 1949 E COMUNICAMI LA LORO CATEGORIA

* la richiesta contienc due distinte domande

1. numero dei dipendenti con divisione Informatica e anno di nascita maggiore di 1949
   14
2. categoria professionals dei dipendenti con Divisione Inforrnatica e anno di nascita maggiore di 1949

COGNOMF	NOME	CATEGORIA
Rossi	Mario	I
Verdi	Claudio	D

> OUAL E' IL LIVELLO DI OUFLLI CHE ERA LORO SONO LAUREATE

* livello dei dipendenti con divisione inforrnatica, anno di nascita maggiore di 1949 e titolo di studio laurea

COGNOME	NOME	LIVELLO
Rossi	Mario	8

Fig. 4 - A sample dialogue.

> dimmi l'anno di nascita dei dipendenti laureati
> vorrei sapere l'anno di nascita di tutti i laureati
> dimmi quando sono nati i dipendenti chc sono laureati
> anno di nascita laureati
> vorrei sapere qual e l'anno in cui sono nati tutti quei dipendenti che hanno conseguito la laurea
> vorrei sapere l'anno di nascita dei dipendenti con titolo di studio laurea
> dimmi in che anno sono nati coloro che sono laureati
> dimmi l'anno in cui sono nati i dipendenti in possesso di laurea
> dimmi l'anno di nascita di chi ha conseguito la laurea
> dei laureati dimmi l'anno di nascita
> dei dipendenti che sono laureati voglio sapere l'anno in cui sono nati
> cerca i dipendenti il cui titolo di studio e laurea e comunicami l'anno in cui sono nati
> qual e la data di nascita dei dipendenti che hanno la laurea?
> qual e l'anno di nascita di quelli che si sono laureati
> qual e l'anno di nascita dei dipendenti che sono in possesso di laurea?
> laureati anno nascita

Fig. 5 - A sample of different ways of expressing the same request.

7. CONCLUSION

The parallel syntactic/semantic analyzer developed within the PARNAX project shares some features with the RUS/PSI-KLONE System (Bobrow and Webber, 1980). Although an in-depth comparison of the two approaches is difficult, two major differences seem worth being outlined:

- The way in which syntactic and semantic processing interact, which is strictly step-by-step sequential in RUS/PSI-KLONE and far more oriented towards a real parallelism in PARNAX;
- The structure of semantic analysis, which is intended as a single process in RUS/PSI-KLONE while it is splitted in two bases, namely nondeterministic validation and deterministic interpretation in PARNAX.

Among most promising research directions that deserve further attention we mention the refinement of both syntactic and semantic models used by PARNAX to better fit the issues of parallel processing, and the study of improved parallel strategies.

As a concluding remark we note that the use of parallel search algorithms that utilize several and diverse sources of knowledge is not limited to the analysis of natural language but represents a core topic in many AI problems and could be of interest in several applications.

APPENDIX

Figure 3 List the employees born in Turin before 1958

Figure 4.

> count the employees in the computer science department born after 1949 and let me know their professional category

* the request contains two queries.

1. number of employees having department computer science and birth year greater than 1949.
2. professional category of the employees having department computer science and birth year greater than 1949

> what is the level of those among them who received a doctor degree

* level of employees having department computer science, birth year greater than 1949, and doctor degree.

Figure 5

> tell me the birth year of the employees with a doctor degree.

REFERENCES

[1] Bobrow R.J., Webber B.L 1980. Knowledge representation for syntactic/semantic processing. *Proc. 1st Annual Nat. Conf. on Artificial Intelligence,* AAA I, Stan ford.

[2] Courtin J. 1977. *Algorithmes pour le traitement interactif des languages naturelles.* These, University Scientifique et Medicale de Grenoble. Grenoble, France.

[3] Guida G., Sornalvico M. 1979. A two level modular system for natural language understanding. *Proc. Int. Joint Conf. on Artificial Intelligence* Tokyo, Japan, 34(3-347.

[4] Guida G., Sornalvico M. 1980. Interacting in natural language with artificial systems, the DONAU project. *Information Systems 5,* 333-344.

[b] Guida G., Tasso C. 1982. NLI. A robust interface for natural language person-machine communication. *Int. Journal of Man-Machine Studies* 7 7,417-433.

[6] Hays D.G. 1964. Dependency Theory: a formalism and some observations. Memorandum RM4087 P.R., The Rand Corporation.