Stanford Heuristic Programming Project
Memo HPP-78-10

September 1978

Computer Science Department
Report No. STAN-CS-78-668

BAOBAB, A PARSER FOR A RULE-BASED SYSTEM USING A SEMANTIC GRAMMAR

by

Alain Bonnet

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

BAOBAB, a parser for a rule-based system using a semantic grammar.

STAN-CS-78-668

Heuristic Programming Project Memo 78-10

Alain Bonnet

ABSTRACT

Until a recent knowledge-based system is able to learn by itself, it must acquire new knowledge and new heuristics from human experts. This is traditionally done with the aid of a computer programmer acting as intermediary. The direct transfer of knowledge from an expert to the system requires a natural-language processor capable of handling a substantial subset of English. The development of such a natural-language processor is a long-term goal of automating knowledge acquisition; facilitating the interface between the expert and the system is a first step toward this goal.

This paper describes BAOBAB, a program designed and implemented for MYCIN (Shortliffe 1974), a medical consultation system for infectious disease diagnosis and therapy selection. BAOBAB is concerned with the problem of parsing - recognizing natural language sentences and encoding them into MYCIN's internal representation. For this purpose, it uses a semantic grammar in which the non-terminal symbols denote semantic categories (e.g., infections and symptons), or conceptual categories which are common tools of knowledge representation in artificial intelligence (e.g., attributes, objects, values and predicate functions). This differs from a syntactic grammar in which non-terminal symbols are syntactic elements such as nouns or verbs.

KEY WORDS

NATURAL-LANGUAGE, PROCESSING, PARSING, SEMANTIC GRAMMAR, KNOWLEDGE-BASED SYSTEM, KNOWLEDGE ACQUISITION.

This research was supported in part by the National Science Foundation under Contract No. MCS 77-02712, and by a fellowship from “Direction Generale des Relations Culturelles Scientifiques et Techniques,” France.

TABLE OF CONTENTS

ABSTRACT.

Until a knowledge-based system is able to learn by itself, it must acquire new knowledge and new heuristics from human experts. This is traditionally done with the aid 'of a computer programmer acting as intermediary. The direct transfer of knowledge from an expert to the system requires a natural-language processor capable of handling a substantial subset of English. The development of such a natural-language processor is a long-term goal of automating knowledge acquisition; faciliting the interface between the expert rind the system is a first step toward this goal.

This paper describes BAOBAB, a program designed and implemented for MYCIN (Shortliffe 1974), a medical consultation system for infectious disease diagnosis and therapy selection. BAOBAB is concerned with the problem of parsing - recognizing natural language sentences and encoding tnem into MYCIN's internal representation. For this purpose, it uses a semantic grammar in which the non-terminal symbols denote semantic categories (e.g., infections and symptoms), or conceptual categories which are common tools of knowledge representation in artificial intelligence (e.g. attributes, objects, values and predicate functions). This differs from a syntactic grammar in which non-terminal symbols are syntactic elements such as nouns or verbs.

I. OVERVIEW.

1.1. Introduction.

'whatever formalism is used for parsing: context-free grammar, context-sensitive grammar, or augmented transition networks, most syntax-based parsers focus mainly on criteria of acceptability on syntactic grounds of the input strings. Although this undoubtedly is of linguistic interest, a different approach has been used in the work presented here. The reasons for this are several.

For convenient and philosophical reasons, we do not object to accepting ungrammatical inputs. In addition to this, syntax-based parsers usually accumulate much information which is useless for our purpose. For example, "The patient has a fever" and "The patient is febrile" lead to the same internal representation despite the fact that "fever" is a noun and "febrile" an adjective. Syntactic analysis is also time consuming and does not avoid semantic checks before building a representation of the input string. Therefore, if it is possible to determine the meaning of a statement without using syntactic analysis, we prefer to do so.

A two-part grammar has been designed, choosing efficiency in the inevitable uniformity/efficiency tradeoff. If certain key-words have been encountered during tne preprocessing phase, the specific rules associated are tried, otherwise the general grammar alone is applied.

The general grammar is, to a certain extent only, domain independent. Its rules recognize the format of a legal statement without concern for the meaning of the individual elements. For example, one legal format is "the <attribute> of the <object> <predicate function> <value>". This same rule can apply to "The morphology of the organism is coccus" in the domain of infectious diseases, as well as to "The landscape of the country is mountainous" in the domain of physical geography. The requirement for the general grammar to be applicable is that the systems for the two different domains must be organized in similar fashion. One system must have "coccus" as a value of "morphology", an attribute of the object "organism; the other must have "mountainous" as a value of "landscape", an attribute of the object "country".

As it is difficult to recognize any input by as general structures as those dealing with attributes, objects and values, more specific rules have been incorporated, allowing the presence of specialized terms such as symptoms, infections, which are typically of no use in another domain. This part of the complete grammar will be referred to as the specific grammar.

I .2. Environment.

When one speaks of understanding by a program, one usually defines a test that must be passed in order to claim that the program has understood. The aim of this program is to transform a piece of medical knowledge expressed as a rule (set of premises/set of actions), or as text, into an internal format. When the program achieves this goal (the judge is the expert who must agree with the proposed interpretation), we will say that it has understood or properly interpreted the rule or the submitted piece of text.

BAOBAB therefore will deal with natural language in a specific domain. We expect the interlocutor to be an expert in the medical field. This means that his expressions are "naturally fairly precise", i .e. , he should not have to abandon a usual way of speaking to fit a special jargon to which he would not be accustomed. Let us here point out the difference between such a program and programs primarily concerned with carrying on a dialogue with casual users. For example, Rendezvous (Codd 1974, Codd et al 1978) focussed on clarification dialog strategies systematically used to make sure that the system correctly understood the user's request, providing him with facilities to break down his or her request into several steps if necessary.

A first demand of the expert sitting at a terminal is to get fast answers from the system. We must also take into consideration the situation in which a new rule will be entered. Most of the time, this will occur when an expert detects a missing or erroneous rule while running the consultation (Davis 1976). Accordingly, after adding a rule, the expert will want to test its expected effect as soon as possible. A conventional natural-language processor includes a syntactic treatment followed by a semantics component converting the linguistic structure into an internal representation. Here, some of the grammar rules explicitly contain semantic information and thus do not require any other semantic processing. On the other hand, general rules do need a semantic treatment, for example in order to determine whether an -object-attribute couple makes sense. However, non terminal categories being more restrictive than nouns or verbs, the amount of work necessary to check their mutual coherence is lessened.

Another legitimate demand of the expert, closely linked to the necessity for speed, that he be allowed to express statements in a terse form, such as using mathematical symbols when that seems to be a convenient short-cut. For example, "WBC < 80 " is as acceptable as " the white blood count is less than 80".

The expert must approve the system's interpretatior of a rule in order to avoid adding incorrect rules. For this purpose, BAOBAE uses a generator called PROSE to translate the internal format back into stylized natural language. The expert is then asked if the interpretation was correct. If the parser has failed to find a correct interpretation, it must guide the user toward the reason for failure. For instance, by displaying words that were not recognized, and by telling what expectations were not fulfilled (grammar rules which were only partially successful), the system can help the expert to rephrase the statement , or it can indicate that new objects or attributes have to be taught before proceeding with incorporating new rules.

I.3 MYCIN background.

MYCIN's judgmental knowledge consists of a set of rules. A rule is internally represented by a CONDITION part and an ACTION part. Each of these is a set of clauses linked by the logical operator AND. For example:

```
        ($AND (SAME CNTXT COMPHOMISED)
              (GREATERP* (VAL1 CNTXT PROTEIN) 40))
```

is the internal representation of:

The patient is a compromised host, and
the CSF protein is more than 40.

An internal clause can roughly be viewed as a quadruple:

<predicate function> <object> <attribute> <value>.

The last three elements constitute the usual triple which is a basic representation formalism within the AI community, <attribute> usually denoting a property of <object> and having <value> as a possible value. From now on, <object> might as well be referred to by <context> which is the MYCIN version actually used. Similarly, <attribute> is often named <clinical parameter> or simply <parameter>. In reality, there are several variants of this generic form (e.g., <value> could be missing, or replaced by a list of values, etc.); but at this point, this simplification allows easier comprehension.

a) The 'predicate functions are usually indicated by verbs (e.g., "is", "known"). The verbs also may be accompanied by appropriate modifiers, such as negations or, more generally, adverbs which add information about the certainty factors associated with the current information (e.g., definitely). Example: KNOWN is the predicate function associated with the following statements: * The morphology of the organism has been determined. * We know the genus of the organism. * The duration of the neurologic signs is known.

b) There are 5 objects considered here, organized into a context tree: PATIENT, INFECTION, CULTURE, ORGANISM, THERAPY. The PATIENT presents a possible INFECTION for which a CULTURE is obtained. ORGANISMs are likely to be isolated from this culture and a THERAPY will be recommanded to fight the organisms.

c) A clinical parameter is a characteristic of one of the contexts of the context tree.

Example: The SITE of a culture.
The NAME of the therapy.
The GENUS of an organism.
Tne AGE of a patient.

d) A value is one of the possible values of a clinical parameter: YES or NO for binary parameters often termed yes/no parameters, otherwise a member of the list of possible values. For example: PTCONTRA is a parameter indicating whether "there is contraindication of the current therapy for the patient" its value is simply TRUE or FALSE. On the other hand, SITE is a multiple-valued parameter and has a large list of possible values: BLOOD, NOSE, URINE, THROAT, . . . Although I use the term MULTIPLE PARAMETER for any non-yes/no parameter, this last category is further divided into SINGLE-VALUED PARAMETERS like SITE, which can have a single correct value (excluding all others), and MULTI-VALUED PARAMETERS like SYMPTOMOF and ALLERGY which can have several values at a time. (Each culture was taken from a single site, however, the patient might simultaneously have symptoms of pyuria and frequency, or be allergic to several drugs.)

e) In action clauses, the predicate functions which are dealt with are CONCLUDE and CONCLUDE* (a variant of CONCLUDE with more than 1 value specified). A positive statement is indicated by a positive certainty factor, the negation by a negative one.

I.4 Sample of a rule acquisition.

In the example that follows, BAOBAB's questions and statements appear in lower case letters, and it displays the internal format of its interpretation in upper case letters. The user enters the rules in upper case letters after the double star. A "carriage-return" is typed to indicate that there are no more clauses in the current part of the rule. A trace of successful grammar rules is shown so that the reader can look ahead at the grammar described in III.4.3.

****

```
Would you like to enter a rule?
** YES
If:     1 - THE ORGANISM IS ROD FACUL
    and 2 - IT IS GRAMNEG
    and 3- IT DID NOT ACQUIRE THE INFECTION IN THE HOSPITAL
    and 4 - THE INFECTION REQUIRING THERAPY IS NOT MENINGITIS
    and 5 -
Then:   1 - FOR SURE 1.0 THE ORGANISM IS A PSEUDOMNAS-AERUGINOSA
    and 2 -
```

```
Grammar rule R025 was successful (not to be confused with KYCIN rules).
Grammar rule R005 was successful
Grammar rule R031 was successful
I don't understand IT DID NOT ACQUIRE THE INFECTION IN THE HOSPITAL even
though all the words were recognized.
Do you want to rephrase it?
++**YES

             THE PATIENT DID NOT GET THE INFECTION IN THE HOSPITAL

Grammar rule R126 was successful
Grammar rule R020 was successful
=PSEUDOMONAS-AERUGINOSA
Grammar rule R031 was successful
the internal format of the rule is:

***********
  PREMISE:  ($AND (SAME CNTXT AIR FACUL)
                  (SAME CNTXT MORPH ROD)
                  (SAME CNTXT GRAM GRAMNEG)
                  (NOTSAME CNTXT NOSOCOMIAL)
                  (NOTSAME CNTXT TREATINF MENINGITIS))
  ACTION:  (CONCLUDE CNTXT IDENT PSEUDOMONAS-AERUGINOSA TALLY 1000)

This is my understanding of your rule:

--------
  If:  1) The aerobicity of the organism is facul, and
       2) The morpnology of the organism is rod, and
       3) The stain of the organism is gramneg, and
       4) The infection was not acquired while the patient was
          hospitalized, and
       5) The infection which requires therapy is not meningitis
  Then:  It is definite (1.0) that the identity of the organism is
            'pseudomonas-aeruginosa

 do you agree with my interpretation?
** YES
 good.
```

COMMENTS.

1) The parser could figure out that the first premise actually contained 2 properties (aerobicity and morphology). Consequently, it split this premise into 2 clauses, which explains the shift in the numbers of premises. Also, these parameters were not mentioned explicitly, but were deduced from tneir respective values.

2) In premise 2 (which became clause 3), the analyzer resolved the pronoun reference by organism (from the previous premise) and thus actually analyzed THE ORGANISM IS GRAMNEG.

3) In premise 3 (which became clause 4), the pronoun reference resolution led to parse: THE ORGANISM DID NOT ACQUIRE THE INFECTION IN HOSPITAL and thus failed. The rephrasing was unambiguous and was successful. The system always indicates which words, if any, were unrecognized in order to guide the user in rephrasing the statement.

4) The objects ORGANISM, PATIENT, INFECTION are always represented by the standard word "cntxt" in clauses but were very important during tne analysis process. A check for consistency between the object and the parameter is always performed before generating any clause.

5) SAME, NOTSAME, CONCLUDE are predicate functions.

6) In premise 4 (which became clause 5), TREATINF is a clinical parameter (attribute) and MENINGITIS is one of its legal values. Likewise, IDENT is a clinical parameter and PSEUDOMONAS-AERUGINOSA is one of its possible values. Notice that it was respelled using the INTERLISP spelling corrector (Teitelman 1975).

1.5. Scope of the language accepted.

Interpreting English sentences consists of finding one or several consistent function-object-attribute-value quadruples. There are various ways to express any natural language statement (surface level) with only one internal representation (deeper level). If we do not want to frustrate the user by the casual computer response: "I do not understand, please rephrase your statement", the program must achieve this several-to-one correspondence.

The expert is not given any constraints concerning his phrasing of sentences. Be is simply advised to express himself in the most precise way he can (avoiding poetics), and use appropriate medical words as Often as possible. This should not be a severe constraint since it is supposed to be his natural way of expression in his professional life. For instance, "The site of the culture is nose" will be preferable to: "a culture was taken from the nose" and obviously to: "A nasal specimen was obtained and sent to the lab". The second statement is still explicit enough, unlike the last one. The program would need strong general Knowledge outside the medical field to understand the last statement. This has not been the concern of BAOBAB or MYCIN thus far. The program has not yet been tested with respect to the "habitability feature" (Watt 1968), that is to say the ease with which the user can learn the conventions of the language accepted in order to avoid going too often beyond the possibilities.

Another characteristic of MYCIN is to deal with non-precise statements or with incomplete information. Consequently, the predicate functions associated with a medical fact are not merely TRUE or FALSE but KNOWN or UNKNOWN, etc. The current program can handle 16 different predicate functions which are briefly described in III.3.

Objects and values are rarely ambiguous. The main difficulty is to find the relevant clinical parameter, which plays the dominant role in the sentence. Some parameters are described with one or two words, like MORPHOLOGY, FEBRILE, GROWTH CONFORMATION, and their recognition is fairly straightforward. On the other hand, some are commonly described by means of a complex sentence, like NOSOCOKIAL indicating whether "The patient acquired the infection while in the hospital" or SPECSTAIN indicating whether "organisms were found on the stain of the culture".

II. RELATION TO OTHER WORKS.

I want to distinguish here between works oriented toward "general natural language understanding" and those oriented toward specific application-s, usually concerned with building interfaces between a user and a program which is an expert in a domain. A major distinction between them is that the first category usually handles a more limited vocabulary than the second, but attempts to analyze inputs more completely, drawing non- trivial inferences based on psychological models or behavior. They usually have ambitious goals, such as buildinq a theory of language understanding (Schank 1973)(Wilks [illegible] . Interesting surveys of these works can be found in (Wilks 1974) and (kinograd 1974). Fundamental works also include the development of various tools such as efficient algorithms to parse sentences (ATN of Woods 1970) (Earley 1970), or how to embed semantics during the analysis process (Procedural semantics, Winograd 1972). Such devices are now used to a large extent by task-oriented systems which thus are an essential contribution which can be used to verify the generality and power of the theoretical tools mentioned above. Question-answering or querying systems and computer-aided instructional systems are functionally similar in the sense that they use roughly similar techniques. The difference lies in the fact that more emphasis is placed on retrieving the relevant information in one, and on carrying on a dialogue in the other. The following is a brief description of some recent systems with which the present work shares some basic features.

In Sophie (Brown 1975), a student is presented with a problem of troubleshooting an electical circuit. A semantic grammar (Brown 1976) is used to analyze the English sentences that the student uses to communicate with the system about the problem. An interesting comparison between a Lisp version (semantic grammar encoded as Interlisp procedures) and an ATN compiled version is drawn, showing that the Lisp version is about twice as fast. On the other hand, three advantages of the ATN formalism are pointed out: (a) conciseness, i.e facility to write, change and communicate the grammar, (b) conceptual effectiveness, which is mainly the coherence between the rule representation (BNF, for instance) and its actual implementation, (c) flexibility for postponing decisions about a path to take during the analysis process.

However, the perspicuity of context-free grammars representation i.e the possibility of telling whether a construction is permitted just by looking at a rule is not maintained in the ATN formalism if it is not implemented on a computer providing graphics facility for displaying the network. The ATN compiled version, witn a compiler similar to Kaplan's GPS (Kaplan 1973) is described in detail in (Burton and Woods 1976) and shown to be about 10 times faster than the interpreted version used in LUNAR (Woods e t al,1972).

PLANES (Waltz 1975,Waltz 8 Goodman 1977) is a system currently aeveloped for answering user's requests from a large data base dealing with aircraft maintenance and flight information. The parser is based on the notion of semantic grammar in which the concepts to recognize are, for example, "plane type" or "aircraft component". An example of a request handled by the system is: "Please tell me if Phantom A5544 had any engine maintenance during April 1974." The program matches the request against pre-stored schemas and, if successful, displays its understanding and asks whether the user agrees. If so, the program retrieves an answer by filling the slots of the relevant answer template.

G. Hendrix developed a number of convenient devices for rapidly creating natural-language interfaces between systems and users (Hendrix 1977). This comprises facilities for dealing with incomplete inputs (ellipsis), and for allowing users to extend the language accepted by the system through paraphrasing facilities. A spelling corrector as well is a grammar editor make the system more habitable. A first system called I&LAND (informal natural language access to navy data) has been built, using these techniques, described in (Sacerdoti 1977). Examples of sentences nandled by the current system are: "What is the speed of the -Kennedy?" then "Its length?", the ellipsis routine leading it to actually parse: "What is the length of the Kennedy??

The primary purpose of MYCIN is the Consultation system. This program does not contain any natural-language capabilities, since the questions are asked by the system. Consequently, little emphasis has been put on the "language-understanding" aspect. However, the necessity to make the system credible to physicians led to design an explanation system (Scott 1977) and thus to the development of a program capable of answering a limited set of questions that physicians might ask concerning:

(a) The status of current knowledge about the patient,
(b) How the system reached its conclusions,
(c) General knowledge contained in Mycin's judgmental rules.

This program uses a key-word approach combined with pattern-matching methods similar to (Colby 1974) and a "scoring technique" to determine which kind of question is asked or which parameter is relevant to the question. Examples of questions handled by the program are:

Is blood a sterile site?
How do you treat meningococcal bacteremia ?
Is oganisrn-1 a streptococcus?

While we have not done so, it should be possible to write a similar semantic grammar for the explanation system. Its adaptation would include new concepts like <type-of question> (how, why, what...) and <predicate function> would be replaced by <topic-of-question> such as "conclude", "treat", or "rule out". For example : Why did you conclude. . . ? " , "How did you treat the infection?", Why did you rule out the possibility.. .?".

All these systems were designed for operating on specific domains. As a consequence, they do not need to dig out subtleties which would not be taken into consideration by the knowledge base, nor they have to perform such delicate tasks as disambiguating between multiple-meanings words, since, most of the time, the meaning relevant to the domain is the only one considered in the dictionary. The first point can be illustrated by the following example. Suppose there is no distinction made between "the patient has a fever" and "the patient has a bad fever", a single parameter FEBRILE existing, it is then clear that "bad" can just be ignored without affecting the resulting representation of the input string. This introduction of "fuzziness" is indeed a characterization of a "shallower level of understanding", which is sufficient for such systems, compared to the general understanders outlined at the beginning of this section.

The second point can be illustrated by tne following example. In a general "idealistic" understander, a word like "patient" might be considered as an adjective (showing patience) as well as " a person under medical treatment" which is the only sense considered here. Let us note that the same problem was actually encountered by Winograd in Shrdlu (Winograd 1972), where other possible meanings of "block" -psychological inhibition for example- were not taken into consideration.

A common feature of the systems which have been described is that they do not use an explicit syntactic analyzer. Note however that some general understanders do not have either (Charniak 1972, Wilks 1973). Instead, the parsing is achieved on semantic basis. The key-word approach is an ex treine position which has the advantage to be unsensitive to the transformational paraphrasings of a same statement. In active and passive mode, the same words need to be recognized, regardless of their order in the string. Consequently, it allows a larger freedom in the tray of expressing oneself. On the other hand, the incapacity of capturing any structure of sentences causes that meaningless statements are easily accepted - "the febrile is patient" -. Further, when a conflict appears between two candidates competing for tne most likely interpretation, it is difficult to decide whether the conflict must be resolved (cnoice to make), or whether the two candidates must be kept because several ideas were expressed in the sentence.

In fact, analyzers based on a semantic grammar have adopted an intermediate position (between syntactic and key-word based parsers) with respect to the two points previously mentioned. The semantic grammar rules carry an implicit structure of inout strings although tnere is no explicit check on grammatical agreement.

Tne main difference between BAOBAB and the other three seems to lie, on one nand, in the choice of a context-free grammar versus an ATM formalism, and on the other nand, in the fact that BAOBAB's general grammar is only constituted of' conceptual entities (no surface words), but this distinction presumably depends on the amount of bottom-up preprocessing achieved before actually using the grammar, thus replacing groups of words by their underlying concept. This nart of the grammar is transportable to other domains that also use general categories such as objects, attributes and values.

A significant difference between PLANES and the others is that PLANES makes little use of the constructions of sentences -order of words are only taken into consideration in special cases -. Rather, "concept case frames are utilized to assign a meaning to the input strings by looking at the registers that have been set during the analysis.

III.THE ANALYZER.

This cnapter tiescribes tne analyzer. We first describe how the dictionary is organized and now the preprocessing phase is achieved. Then, the subgrammar for predicate functions, used in a bottom-up manner, is shown. Finally, the main grammar used (top-down) to parse the input strings is described in detail.

III.1 The dictionary.

In order to avoid repeating information common to several words which are close in meaning, it is convenient that one of them be considered as a terminal word. It will then be the only one of the group to be integrated into the semantic network describing the relationships existing between the different concepts which are dealt with. The notion of closeness of meaning is partly explained by the TXTSYN pointer and completed in the description of the preprocessing phase. 'The pointers described here are only the ones used by the parser.

III.l.l TXTSYN.

The pointer from a word to its terminal f'orm is called TXTSYN (a terminal word points to itself). The words related by a txtsyn pointer may be synonyms in the usual sense, but also abbreviations, root words or infinitive forms tnat cannot be founo by Winograd's root extraction algorithm as outlined in III.2.

Example: TXTSYN(eschericnea-coli) = e.coli
TXTSYN(e-coli) = e.coli

which means that e.coli has been arbitrarily chosen as the terminal word.

Example: TXTSYN(began) = begin (infinitive is terminal)

III.1.2 INCONCEPT.

When a word suggests the presence of one or several clinical parameters, it points to it (them) by The INCONCEPT pointer.

Example: INCONCEPT(Morphology) = (Morph)
INCONCEPT(pregnant) = (Motherhood)
INCONCEPT(abnormal) = (Abnormal Cxrab Lensign)

The word ABNORMAL might suggest the 3 parameters ABNORMAL (an organism is not normally found at a certain site), CXRAE (The patient's x-ray is abnormal), LENSIGN (The patient had recent abnormal neurologic signs).

III.1.3 VALUESYN.

This gives the value (in the sense of value of a parameter) that the word might imply in certain contexts.

Example: VALUESYN(negative) = gramneg (in context of stain)
VALUESYN(white) = Caucasian (in context of race)

Examining tne context allows the system to decide whether such a value is correct. For instance, in "white blood count", "Caucasian" will be discarded as a meaning for "wnite".

III.1.4 EXPECT/EXPECTED.

These are the pointers between parameters and their possible values, EXPECT defining the valid set of values for a parameter (NIL for a yes/no parameter), EXPECTED giving the possible parameters implied by the Value.

```
     Example: EXPECT(Morph) = (Rod Coccus . ..)
              EXPECTED(Rod) = EXPECTED(Coccus) = (Morph)
              EXPECT(Site) = (Blood Nose Throat Urine . ..)
              EXPECTED(Blood) = EXPECTED(Urine) = (Site Portal
                                                        Infsite)
     (Urine might be the site of a culture, of an infection or the
             portal of entryj
              EXPECT(Whensym)= date
              EXPECT(Contaminant) = NIL
              EXPECT(AGE) = Number
```

111.1.5 COMPOUND and HYPART.

These pointers enable the recognition of groups of words as a whole, for example *'streptococcus group a" which will be replaced by "streptococcus-group-a",

III.1.6 NOCONTENT.

This indicates that a word has no medical meaning. However, the word might be important to figure out the structure of the sentence. In this category, "or", "of", "and", etc.

III.1.7 TEMPLATE.

This gives the template of the internal clause according to the predicate function.

```
     Example: TEMPLATE(<known>) = (cntxt parm)
              TEMPLATE(<same>) = (cntxt parm valu)
              TEMPLATE(<greaterp*>) = ((val1 cntxt parm) num)
```

III.1.8 PICKGROUP.

It gives the object of a parameter and is used in the procedures which check for semantic coherence inside the rules.

111.2 Preprocessing.

The preprocessing phase is very similar (from paragraph a to f) to the one used by tne explanation system from where it was extracted (Scott 1977).

The following heuristics are used to accomplish this task.

a) Standard reduction to a canonical form: An implementation of Winograd's root extraction algorithm (Winograd 1972) replaces the plural form of a noun by its singular and replaces a conjugated form of a verb by its infinitive. For instance "acquires" and "acquired" are both replaced by "acquire". Also, it can extract the root of certain substantives ("combination" giving "combine").

b) Resolution of certain irregular forms which cannot be found by a general method is then given by the TXTSYN pointer, for instance "began" is replaced by "begin".

c) Croups of words which have special meaning when they appear together are taken into account by creating a single hyphenated word, like "streptococcus-group-a". In this particular case, "a" might otherwise be considered as an article which would lead to a misinterpretation.

d) Conventional words have been cnosen to represent several words which might be synonyms in context (therapy, treatment) or abbreviations (echerichea-coli, e-coli).

e) If a word has not been found in the dictionary are, an attembt is made to respel it using the DWIM Interlisp routine (organisn --> organism). (Teitelman 1975).

f) Punctuation is currently ignored. This also includes detaching punctuation marks from the end of words, since it is most common not to type a space between the end of a word and the following punctuation.

g) Some no-content words are ignored (the, a, this, that, an,...) because nuances between articles (e.g., definite versus indefinite) are not handled by the system. On the other hand, words like "of" or "from" are used to recognize certain structures of sentences. Words that are still unknown after using the spelling corrector are also kept to let the user know where a failure occured during the analysis.

h) Some yes/no parameters can only be described by using a complex piece of' sentence. In order to facilitate the grammar's task, bottom-up recognition of' some of these parameters is carried out before invoking general rules. For example, <angrow> and <airgrow> might be recognized after the success of some of the following patterns:

```
<aerobically> --> in aerobic plate
<aerobically> --> in aerobic bottle
<anaerobically> --> without oxygen
<airgrow> --> grow <aerobically>
<angrow> --> grow <anaerobically>
<airgrow> --> able to <airgrow>
<angrow> --> able to <angrow>
```

These patterns permit the recognition of the following 6 sentences defining the parameter <airgrow>:

1. (Tne organism) was able to grow aerobically.
2. (The organism,) was aole to grow in the aerobic plate.
3. (The organism) was able to grow in the aerobic bottle.
4. (Tne organism) could grow aerobically.
5. (The organism) could grow in the aerobic plate.
6. (The organism) could grow in the aerobic bottle.

as well as 4 similar sentences for <angrow>.

III.3 Subgrammar for predicate functions.

Tne preaicate function of any clause is indicated by verbs (mainly auxiliaries), negations and some appropriate modifiers. Their recognition is accomplisned independently of the general grammar for the following reasons:

a) Auxiliary verbs (is, has, etc.) are used by tne Q-A program (part of tne explanation system) in a manner incompatible with the analyzer uescribed here. A convenient way to circumvent this problem Without modifying the dictionary was to combine verbs before invoking tne previously uescribed procedure f'or producing a canonical form of the statement.

b) It seems to be worthwhile in the tradeoff between bottom-up and top-down process; for instance (see the grammar rules described in III.4), <faculfun> appears in many rules and it seems to be efficient to recognize it once and for all before using the general grammar.

c) Predicate functions may be modified by operators, as described in 111.3.2, wnich appear in non-adjacent part of the sentence. This phenosenon is difficult to handle with general grammar rules.

III.3.1 The subgrammar.

The grammar is described in a BNF-like context-free formalism, square brackets [ ] enclosing optional elements, a slash / separating alternatives of the expansions of the rules, and angle brackets < > enclosing non terminal elements. Tne top-level rule is: <Pred-function> := <Novalfun>/ <Faculfun> / <Num1fun> / <Num2fun>.

These 4 types of predicate functions are now going to be explained in terms of values expected. Another classification in terms of certainty factors can be found in (Snortliffe 1976, pp 102 to 105).

111.3.1.1 <Faculfun> functions.

<Faculfun> stands for facultative values functions, which means that the template expected by the function may contain values but this is not compulsory. The two functions of this category are <same> and <notsame>.

Example: (NOTSAME CNTXT FEBRILE) represents "The patient is not febrile".
(SAKE CNTXT SITE NOSE) represents "The site of the culture is nose".

```
<Faculfun> := <Same> / <Notsame>
<Same>     := <Same> <Same>
<Notsame>  := <Same><Notsame> / <Same><Notsame><Same>
<Same>     := is / nas / was / had / been ...
<Notsame>  := not / never / no . . .
```

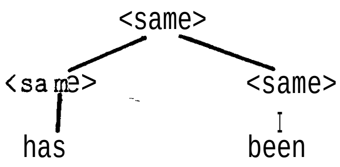

Figure 1.

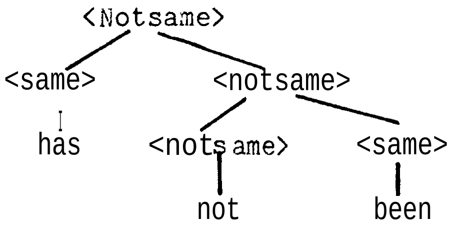

Figure 2.

111.3.1.2 <Novalfun> f-unctions.

<Novalfun> stands for no-value functions, which means that the template expected by the function must not contain values.

Example: (KNOWN CNTXT MORPH) --> "The morphology of the patient is known."

```
<Novalfun> :=<pronoun> <novalfun>
<novalfun> := <known> / <notknown> / <definite> / <notdefinite>
<known> := <same> <known>
<known> := know / known/ knew/ determined
<notknown> := <notsame><known> / <same> <notknown>
<definite> := <known> with certainty / definite
<notdefinite> := <notknown> with certainty / <notsame> <definite>
```

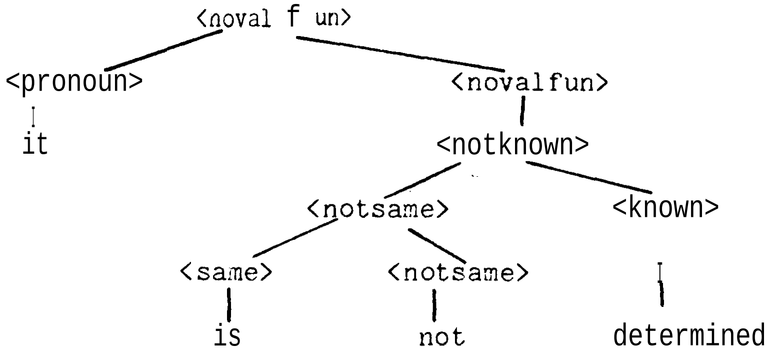

Figure 3.

III.3.1 .3 <Num1fun> functions.

<Num1fun> stands for functions expecting one numeric value.

Example: (LESSP* (VAL1 CNTXT CSFCELLCOUNT) 10) --> "The white blood count from the cerebro spinal fluid is less than 10."

```
<Num1 fun>    := <greaterp*>/<greateq*>/<lessp*>/<lesseq*>
<greaterp*> := <same> greater than / >
<greateq*>   := <greaterp*> or equal to / <greaterp*> =
<lessp*>      := <same> less than / <
<lesseq*>     := <lessp*> or equal to / <lessp*> =
```

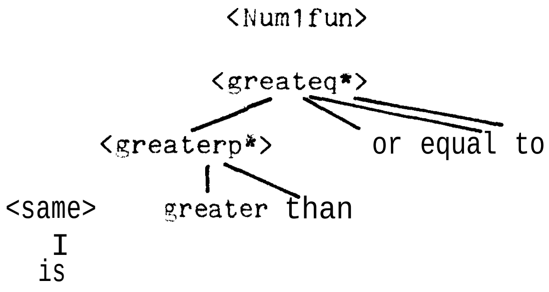

Figure 4.

11X.3.1.4 <Num2fun> function.

<Num2fun> stands f'or functions expecting two numeric values. Actually, there is only one function called <between*> in this category.

Example: (BETWEEN* (VAL1 CNTXT AGE) 0.25 0.50)) --> "The age of the patient is between 3 and 6 months.″

```
< N um2 fun>  := <Between*>
<Between*> := <same>< between>
```

III.3.2 Modification of the function by an operator.

An operator is a prefix of a statement which can generate itself a clause. It can refine or modify the first predicate function found. For example :
It is definite that the morphology of the organism is coccus

```
<def>                    (SAME CNTXT MORPH COCCUS>
```

Figure 5.

Thus, <def> applied to <same> gives <defis>, and finally, the clause to build up is: (DEFIS CNTXT MORPH COCCUS>. Five new predicate functions can thus be built:

```
<Defis>       := <def> <same>
<defnot>      := <def> <notsame>
<notknown>    := <notknown> <anyfunction>
<tnoughtnot>  := there is evidence that <notsame>
<mightbe>     := tnere is no evidence that <notsame>
<couldbe>     := there is no evidence that <same>
```

Notice that in <known> <statement with function>, <known> will be simply ignored because it does not add any information (redundant).
Ex: "It is known that the patient is alcoholic" is absolutely equivalent to "The patient is alcoholic."

III.4 GENERAL GRAMMAR .

III.4.1 Grammar as procedures.

The grammar has been encoded as INTERLISP procedures, as opposed to, say a BNF-like f'orm. This allows a better efficiency (see footnote) due to the possibility to compile it. Such a choice naturally leads to blend (but not necessarily) several types of knowledge, semantic and structural for instance. By looking at a rule like:
<R011> := <object> <faculfun> <ynparm>,
one might think that "The culture is febrile" will be accepted.

Note: The average time for parsing inputs is 1.5 second on the Stanford PDP-10 time-sharing system, mediumly loaded and with the dictionary on a hash-file.

Actually, an implicit check f’or coherence between the object and the parameter is performed inside the rule, thus allowing the system to refuse such a statement. Coherence between parameter and value can also be checked, tnus allowing the system to accept "The infection is cystitis", but to reject "The morphology is cystitis".

If’ one encodes a grammar as procedures, the "tightness" of the rule can be specified dynamically that is, the ability to ignore some words at specific points during parsing. The "loose" form of the above rule can be described as:

<R011>' := <object> <faculfun> [skip] <ynparm>.

The role of tne SKIP procedure is to permit words like "high" or "bad" to be skipped ("The patient has a high fever”) because such nuances are not currently handled by the system. The procedure for ignoring words is explained in more detail in IV.?.

On the other hand, having the grammar as a data structure interpreted by a program allows:

Easier modifications since it is more convenient to add a piece of data than a piece of’ code.

There is no distorsion between the BNF used to display the grammar and its actual procedural implementation, although the LISP encoding, as shown in the appendix is fairly straightforward.

III.4.2 Taxonomy of’ grammar rules.

In terms of the actions that are triggered when a rule is successful, grammar rules can be divided into two subsets. Generation rules are used to build up internal clauses. Conjunction rules are used to rebuild a piece of the sentence to be analyzed after resolving anaphoric references. At the moment, this includes pronoun references and elliptic resolutions such as when the verb has not been repeated.

In terms of generality/specificity, certain rules are general in that they are not associated with a particular type of information; in addition, they never include a surface word in their expansion part. Other rules are specific in that they are associated with a particular piece of information; also, they often include surface words in their expansion part.

A table specifies what actions have to be carried out whenever a rule is successful. For any generation rule, it consists of one or several templates to be filled in. For conjunction rules, it consists of rebuilding the input to give it to parse to the grammar.

Furthermore, a rule can lead to different actions being undertaken depending on whether it is parsed as a premise or as an action. This distinction disappears if the text to be parsed is not a conventional Mycin rule. khen analyzing text, all sentences are treated as premises. Conversely, several rules can lead to the same action. For instance:

<R020> := <multparmcntxt> .<faculfun> <value>
with: <multparmcntxt> := <multparm> of <object>
which allows parsing of: "The identity of the organism is e.coli.", and
<R031> := <object> <faculfun> <value>
which allows parsing of: "The organism is an e.coli."
lead to the same clause, as they evidently correspond to two surface structures for the same meaning.

111.4.3 General grammar.

The parsinq process is top-down and left-to-right. The usual problem of left recursion for left-to-right parsings is handled by creating an intermediate symbol when necessary, and by duplicating the corresponding rule. basically, a successful rule extracts the matching part of the sentence and returns the rest, if any, in case the sentence would lead to several clauses.

III.4.3.1 General generation rules.

Example 1: The morphology of the organism is not known.

Let us consider the rule: <R011> := <mult parmcntxt><novalfun>
with: <multparmcntxt> := <multparm>of<object> / <object><multparm>

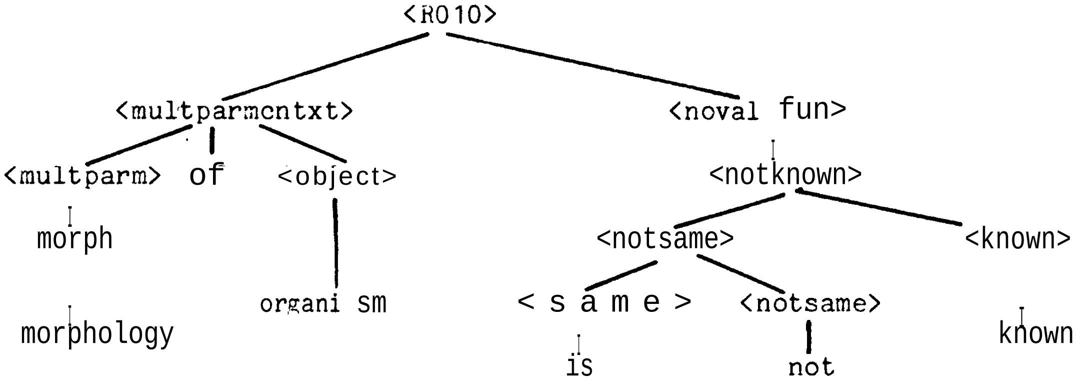

Figure 6.

The internal clause built up after the success of this rule is: (NOTKNOWN CNTXT MORPH). Top-down and bottom-up arrows respectively show the role of each process. Actually, the preprocessed sentence was: "morph of organism <notknown>", having ignored articles, replaced "morphology" b y "morph" (INCONCEPT pointer), and "is not known" by the predicate function <notknown>. Notice that the same rule would also be successful on the following sentences:

The growth conformation is known with certainty.
The infection site is definite.
The aerobicity of the organism is known.

Example 2: The patient has a high fever.
Let us consider the rule:
<R011> := <object> <faculfun> [skip] <ynparm>
The tight rule will fail ("high" causing the failure), and if no other rule has been successful, <R011> will succeed in the second pass (loose form) as shown in Figure 7.

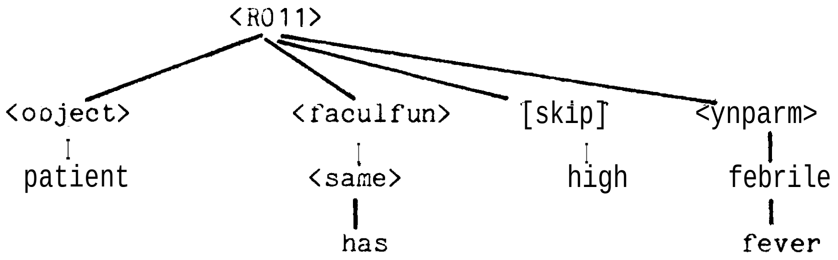

Figure 7.

An interesting point may be discussed here. There is no absolute criterion for deciding whether a word may be skipped or not. If "high" were ignored in the first pass, it might prevent the success of another rule in which " high" is a necessary element. The section devoted to the control structure explains in more detail the two-pass process presently used.

Let us notice that <R011> also allows the system to parse:
The patient is not a compromised host.
The organism was able to grow anaerobically.

Example 3: A lumbar puncture has been performed on the patient.

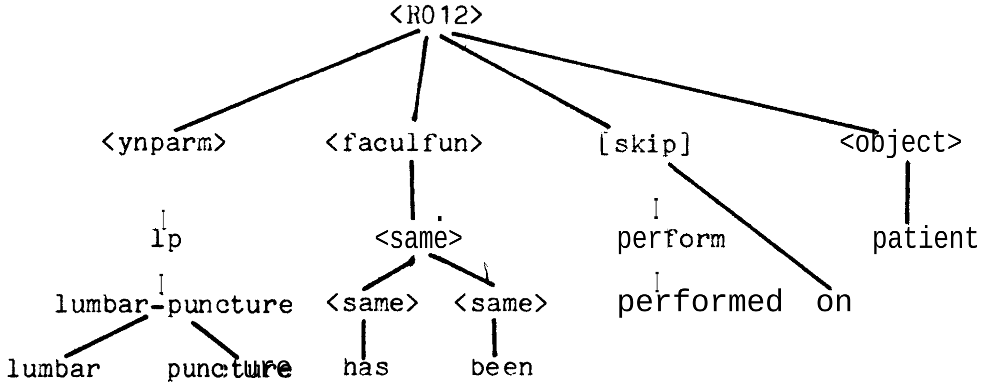

Figure 8.

Note that places in sentences where "skips" are allowed are usually where many expressions could be used. For instance, in Example 3, after "has been", many different expressions might be used. This feature makes it unnecessary to foresee all of the possible phrasings, e.g., "done on", "undertaken on", etc. Tnis is evidently a step toward a key-word approach, but only no-content words (in a medical sense) can be skipped. This minimizes risks of misinterpretation. Also, a rephrased statement such as "The patient has received a lumbar puncture" would be parsed by the previous rule <R011>.

Example 4: The morphology, aerobicity and growth conformation of the organism are known. Several properties are stated and this should lead to a split of the sentence into several internal clauses. Let us consider the rule <R015> := <multparml+cntxt> <novalfun>. The recognition will be performed as shown in Figure 9. This leads to the following 3 clauses:

```
(AND (KNOWN CMTXT MORPH)
     (KNOWN AIR MORPH)
     (KNOWN CNTXT CONFORM))
```

<R015> would also be successful on: "the age, sex and weight of the patient are definite?

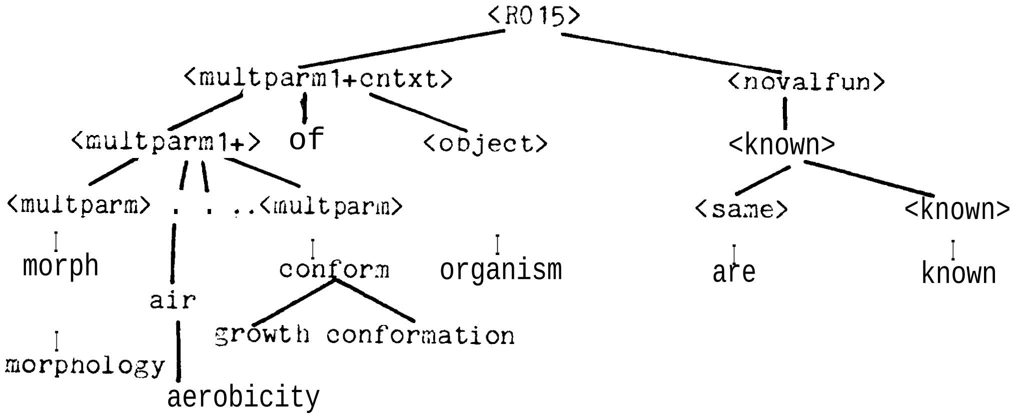

Figure 9.

Example 5: The site of the culture is one of: nose ear throat. Such a sentence is parsed by the rule R018 as shown in figure 10. <homovall+> stands for more than one homogeneous (corresponding to the same parameter) value. In this case, the internal clause generated is a variant of the standard template and is actually:

(SAME CNTXT SITE (ONEOF NOSE EAR THROAT)). The following sentence is parsed similarly: "The infection requiring therapy is among: cystitis pyelonephritis."

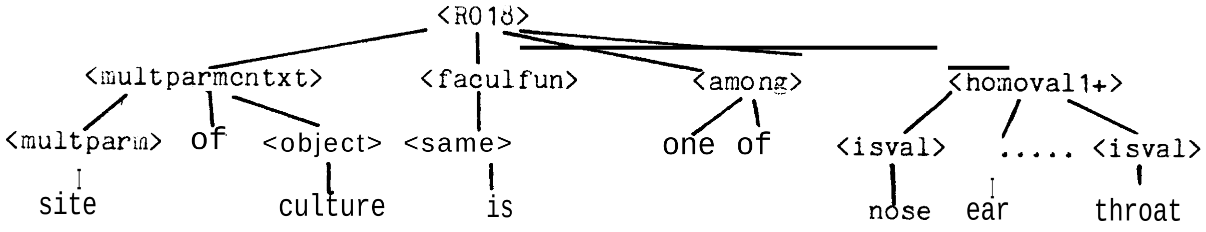

Figure 10.

Example 6: The organism is rod anaerobic and grampos. Such a sentence is processed by tne rule R025 as shown in Figure 11 in which <heteroval1+> stands for more than one heterogeneous (different parameters) value. "The patient is a male Caucasian" is parsed similarly.

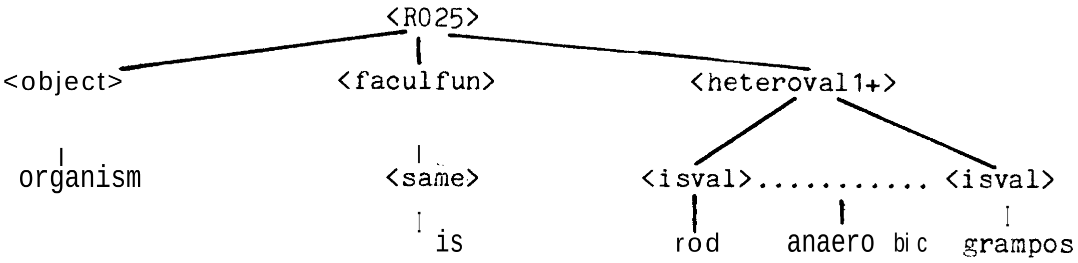

Figure 11.

Example 7 : The organism is a pseudomonas aeruqinosa.

This sentence is parsed by the rule R031 as shown in figure 12. Let us note here that the name of an organism is indicated in the sentence not only through the explicit mention of "identity" . When no parameter is mentioned in the sentence but a value that could belong to "identity" as well as to other parameters is present, an inference process is invoked which decides that "identity" is the probable parameter. Similarly, in "the patient is <number>", the parameter is assumed to be "age" (common sense inference).

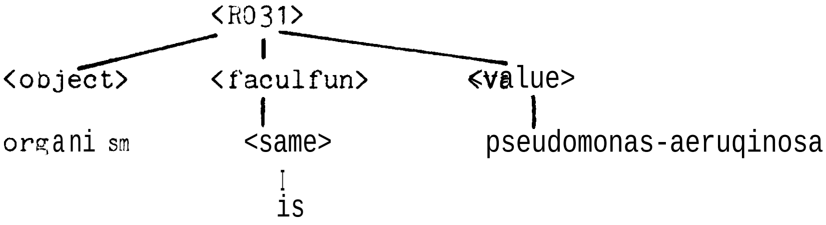

Figure 12.

Such a rule deduces the parameter from an ambiguous value. The parameter is then included explicitly in the restatement of the sentence, in order to verify the user's agreement. For instance here, the above sentence will be rephrased "the identity of the organism is pseudomonas aeruginosa" . The same rule would also be successful with: "A culture was taken from blood" on the loose form after skipping "taken from" , as well as "the patient is male", or "the drug is penicillin",

Example 8: The white blood count from the cerebro spinal fluid is inferior to 10.

Here, the preprocessing phase perf'orms a lot of work in order to facilitate the grammar's task. The preprocessor mainly recoqni zes groups of words that make sense together and substitutes a single abbreviation for tne group of words.

```
        WBC <-- white blood count
        CSF <-- cerebro spinal fluid . .
```

Then, the rule <R055> := <numparmcntxt> <numlfun> <isnum> is successful as snown in the Figure 13.

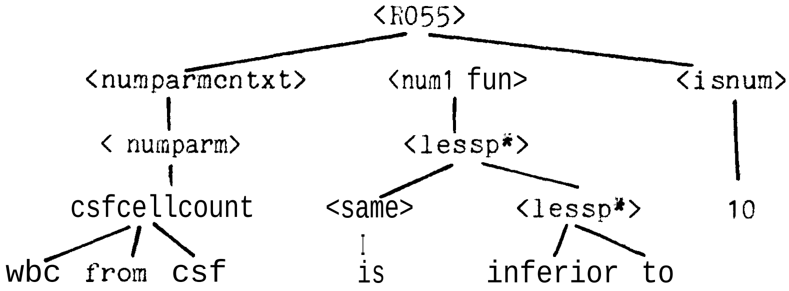

Figure 13.

The same rule can be used for:

The age of the patient is less than 3 months.
The csf' protein is less than 40
The patient's creatinine clearance is greater than 80

III.4.3.2. Conjunction rules.

Example 1: The patient is jaundiced and is not a compromised nost. The beginning of the sentence is easy to parse and gives the internal clause (SAME CNTXT JAUNDICED). The role of a conjunction rule is to recognize that the subject is missinq and to trigger the search f'or it. The action invoked will then be to find the subject oy a very straightforward method (last clause analyzed) thus leading to add PATIENT onto the remaining part of the sentence. It will then return "the patient is not a compromised host" as the remaining sentence to be analyzed. The referent chosen is the object of the last cla-use generated. It relies on the assumption that, when several properties are stated in the same input, they probably refer to the same object.

The first conjunction rule is described below:

```
    <R005> := [<pronoun>] <faculfun> <isval> /
              [<pronoun>] <faculfun> <ynparm> /
              [<pronoun>] <numlfun> <isnum> /
              [<pronoun>] <faculfun> <isnum> /
            [<pronoun>] <num2fun> <isnum> <isnum>
```

This rule also recognizes:

"and it is gramneg" (expansion 1)
"and has neurologic signs" (expansion 2)
"and is less than 10 years old" (expansion 3)
"is 56" (expansion 4)
"is between 7 and 77 years old" (expansion 5)

Notice here that the diversity of the forms to recognize for similar structures is merely due to arbitrary decisions about the representation of knowledge, imprecision in the formalism analogous to the one occuring in the meaning of links in semantic networks (Woods1975). For example, "the patient has fever" is represented by a binary parameter FEBRILE, but might as well be represented by a multiple parameter SYMPTOM which would have FEBRILE as a possible value. A second conjunction rule recognizes incomplete structures of sentences similar to the previous one except that the function is also missing. When the rule is successful, it will then trigger the execution of a routine whose purpose is to retrieve the subject and the verb (or predicate function). This rule can be described as:

<R006> := <isval> / <ynparm> / <isnum>.

111.5. Specific grammar.

Specific grammar rules are used to recognize parameters which can be described only by fairly complex sentences. These rules usually have a loose form, which avoids foreseeing all possible phrasings. The risks of a strictly key-word approach (as outlined in the discussion ending the section II) are decreased by the presence of common sentence structures associated with parameters of this type in addition to indispensable key-words.

Example 1: The concept of' "symptom" is of primary importance for establishing a medical diagnosis. Consequently, it is semantically very rich. Some symptoms are referred to by binary parameters as MUMPSYM, indicating whether "the patient has shown symptoms of mumps++, or VAGDIS, indicating whether "the patient has increasing vaginal discharge". Others are represented by a multiple parameter termed SYMPTOMOF as in "the patient has symptoms of dysuria" in which case DYSURIA is a legal value for SYMPTOMOF. The following rules express the semantics of "symptom".

```
<symptomof> := patient <faculfun> symptom of <uria>
               <uria> <faculfun> <among> symptom of patient
<uria>      := dysuria / frequency / hesitancy / suprapubic-
               discomfort / urgency...
<ynsymptom> := patient <faculfun> symptom <otheruti>
<otheruti>  := of mumps / of increasing vaginal discharge /
               [skip] lower urinary tract
```

Let us note that the "[skip]" appearing in the last line allows accepting Various phrasings as:

The patient has symptoms referable to
concerning the
associated with
located in the lower urinary tract.

Example 2 : NOSOCOMIAL is a very important parameter which can have a complex description. Two rules enable its recognition, as shown in Figure 14.

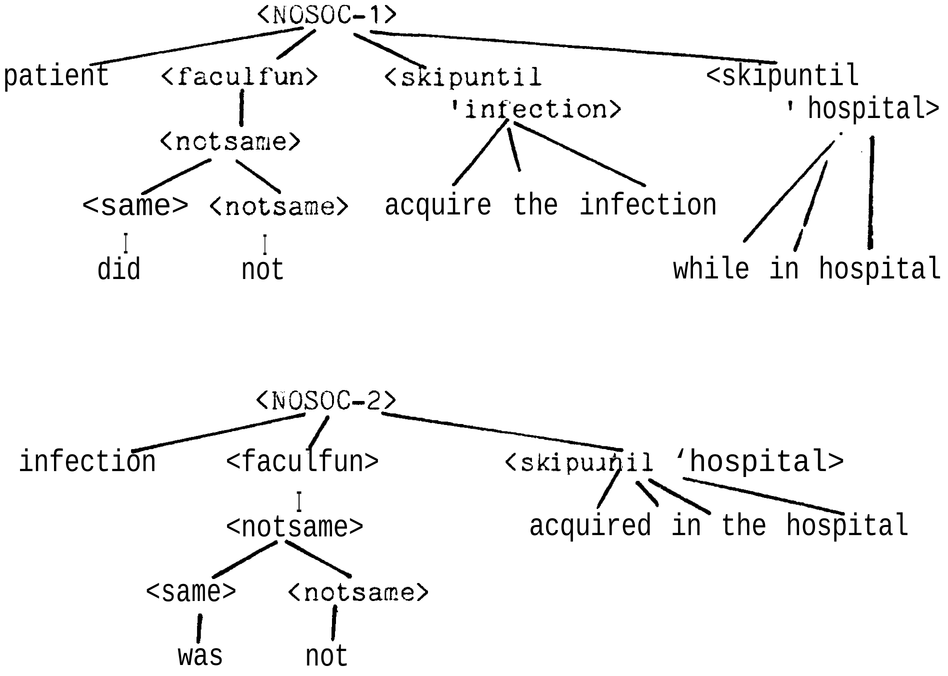

Figure 14.

Example 3 : Age is another important concept taken into consideration in many rules (for the dose of a drug for instance) and it is rarely stated explicitly in most sentences that use the concept. The following rule is used to recognize it:

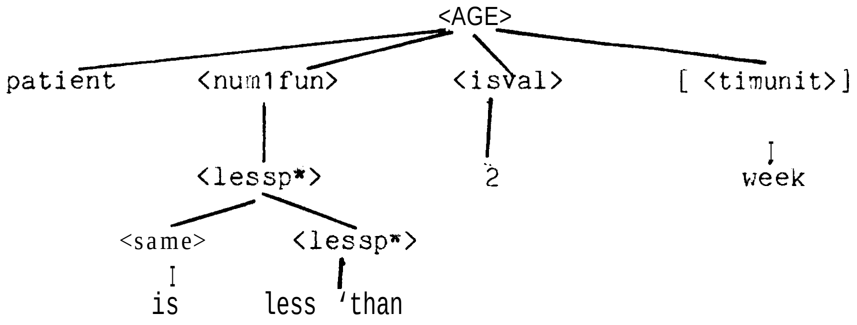

Figure 15.

IV . IMPLEMEN TATION .

This chapter gives details on the current implementation. It describes the control structure and certain strategies (tightness/looseness for example), how the semantic categories are assigned during the analysis and finally displays an interactive session of acquisition of′ rules.

IV.1 Control structure.

A depth-first control structure has been chosen because the first parse usually leads to the only interpretation possi ble, which is primarily due to the precision of the statements in the medical field.

A specific rule is always associated with a key-word which can be the possible parameter itself or an indispensable word such as "hospital" for <nosocomial>. Consequently, once the right key-word has been encountered, chances of success for a specific rule are higher than for a general rule (all are tried in sequence without preference).

Let us notice that, ii′ the beginning of a sentence cannot be mapped into an internal representation, no further attempt is aade to analyze the rest of the sentence (except after an "and" which usually introduces another property). Analyzing any part of a sentence after a piece which was not understood might lead to misinterpretations as shown in the following example : "the sister of the patient is febrile″ would indeed be interpreted as "the patient is febrile″ if "the sister** has not been understood, unless a fragmentation technique similar to Wilks' is used to attempt filling the templates with the different pieces.

Some of′ the general rules contain a flag which allows them to be "sloppy" i.e., to ignore a word and proceed with the analysis. Such rules are always tried in the tight form first. If the rule fails but was partially successful, it will be tried again in the loose form in the second pass if no rules successed during the first pass. The additional set of rules to be tried in loose form will thus be very small.

Example : Suppose the word "obese" is not known to the system, and the sentence to be parsed is "The patient is an obese male ". After the failure of the tight rule, the loose version will skip "obese" and succeed, but the expert will be notified that "obese" was not understood. He will then be able to teach the program about this new concept and will add a premise "the patient is obese" to complete the meaning of' the partially-understood previous statement.

So far, there is no mechanism to handle possibilities of multiple interpretations of an input string. Disambiguations are made during the preprocessing phase, for example, "csf" (cerebro spinal fluid) might be a possible value of tne site of a culture but, if associated with white blood count as in "the white blood count from the csf", it actually refers to the parameter "csfcellcount" and is recognized as such before the grammar is evoked. As a consequence, the parsing is very deterministic and no back-up mechanism has been implemented yet. This might be a weakness as the extension of the grammar proceeds.

The above mechanism to build up a clause is mostly a loop on the set of' rules to try. An action is carried out whenever a success occurs. An important feature is that the list of rules to try is set during the preprocessing phase, specific rules being appended at the head of the list whenever a triggering word is encountered. The analysis process is stopped when the remaining sentence is NIL (success> or when a whole iteration of loose rules has been performed without any success.

A shortcoming of the current implementation is that the same task may be perf'ormed several times. The merging of common parts of different rules as used in transition networks implementations would certainly permit a more compact representation and a better efficiency.

IV.2. Example of' parsing.

Let us consider "The morphology of the organism is rod". The preprocessed sentence is: "Morph of organism <same> rod". How the appropriate variables are set during tne processing of the sentence is shown below with the successful rule:

<R020> := <multparmcntxt> <faculfun> <isval>

WORD is the word currently being analyzed.
How the variables PARAM, TYPE-P, TYPE-OEJ, VALIUM are set as the analysis proceeds is shown below on the top-down parse tree.

```
WORD <-- morph

     <multparmcntxt>
        <multparm>
           <isparm>      PARAM <-- morph
                          TYPE-P <-- prop-orq
                          Returns True .
           Checks that PARAM expects a value
                          Returns True.
```

```
WORD <-- of
        <surf 'of> matches the right surface word.
                           Returns True.
WORD <-- organism
        <object>  sets: TYPE-OEJ <--prop-org
                           CONT <-- organism
                           Returns True.
WORD <-- <same>
    <faculfun>
          Sets  FUN <-- <same>
         Returns True
WORD G-rod
    <isval>
             Sets VALIUM <-- rod
                  PFVALIUM <-- morph (parameter expected from value. )
         <check1> PARAM subset of PFVALIUM: True
         <check2> TYPE-OEJ = TYPE-P   : True
WORD <-- NIL
Returns True.
```

The variables PARAM, FUN and VALIUM are bound to values that allow building the appropriate internal representation.

1v.3. Sample of a rule acquisition session.

In the examples that follow, the BAOBAB program utterances are in lower case letters and it shows its internal lisp inter- pretations in upper case letters The user enters the rules in upper case letters after the double asterisk; every time he has finished with a set of premises or actions, he types a "carriage-return" which explains the lines of blanks. Comments have been inserted by nand in order to point out a few f'eatures of the parsing. Note that in all the examples shown below, the interpretation is correct. What can be done if the expert disagrees is part of future directions. Note that the kind of inputs that are not currently handled deals with complicated inputs expressing meta-knowledge (how to use medical knowledge) rather than pure medical knowledge. Examples are :

(1) You have examined the list of members associated with the category of the organism and found some that agree with respect to the following symptoms : air and conform.

(2) This drug is one of the treatments which have contributed to an improvement in the patient's symptoms of the infection.

Another kind of non-understood statement includes "non medical ways" of expressing oneself that would require a lot of knowledge about world in general like:

(3) She drinks 4 bottles of Mouthon-Rothschild every morning, to express that "the patient is alcoholic".

EXAMPLE I.

```
Would you like to enter a rule?
** YES
If:      1 - THE CREATININE CLEARANCE IS <= 80
    and 2 -
Tnen:  1  - CONCLUDE THAT THE PATIENT HAS DECREASED RENAL FUNCTION
    and 2 -

the internal format of' the rule is :

LISPRULE
--------
  PREMISE:  ($AND (LESSEQ* (VAL1 CNTXT CCR)
                           80) )
  ACTION:  (CONCLUDE CNTXT RENALFAILURE YES TALLY 1000)

This is my understanding of your rule:

--------
  If:  The patient's most recent creatinine clearance (in ml/min) is less
           than or equal to 30
  Then:  It is def'inite (1.0) that the patient does have decreased renal
           function

 Comments: In the premise, the object PATIENT was not stated, but was
   inferred from the parameter. As no certainty factor was stated,
   it was supposed to be 1.

do you agree with my interpretation?
** YES
good.
```

EXAMPLE II.

```
Would you like to enter a rule?
** YES
If:      1 - THE CULTURE SITE IS CSF
    and 2 - THE PATIENT HAS NOT HAD RECENT NEUROLOGIC SYMPTOMS
    and 3 - HE HAS NOT HAD RECENT NEUHOLOGIC SIGNS
    and.4  - THE WHITE BLOOD COUNT FROM THE SPINAL FLUID IS < 10
    and-5 - THE CSF PROTEIN IS LESS THAN 40
    and 6 -
Then:  1 - IT IS LIKELY (.8) THAT THE INFECTION REQUIRING THERAPY IS NOT MENINGITIS
    and 2 -

the internal format of' the rule is :
```

```
LISPRULE
--------
  PREMISE:  ($AND (SAME CNTXT SITE CSF)
                  (NOTSAME CNTXT NEUROSYMP)
                  (NOTSAME CNTXT NEUROSIGN)
                  (LESSP* (VAL1 CNTXT CSFCELLCOUNT)
                          10)
                  (LESSP* (VAL1 CNTXT PROTEIN)
                          40))
  ACTION:  (CONCLUDE CNTXT TREATINF MENINGITIS TALLY -800)

This is my understandinp of your rule:

--------
  If:  1) The site of the culture is csf, and
       2) The patient has not had recent neurologic symptoms, and
       3) The patient has not had recent neurologic signs, and
       4) The WBC from the CSF is less than 10, and
       5) The CSF protein is less than 40
  Then:  There is strongly suqgestive evidence (.8) that the infection which
             requires therapy is not meningitis
do you agree with my interpretation?
** YES
good.

  COMMENTS: The pronoun resolution procedure replaced "he" by
            "patient?
```

EXAMPLE III.

```
Would you like to enter a rule?
** YES
If':     1 - THE CULTURE SITE IS NOT ELOOD
    and 2 - WE'DON'T KNOW WHETHER THE ORGANISM WAS ABLE TO GROW AEROBICALLY
    and 3 - THE LAEORATORY DID NOT ATTEMPT TO GROW THE ORGANISM
            ANAEROBICALLY
    and 4 -
Tnen:   1 - THE AEROEICITY OF THE ORGANISM COULD BE FACUL (.5) ANAEROBIC (.2)
            OR OBLIGATE-AERCB (.3)
    and 2 -
the internal format of tne rule is :

LISPRULE
--------
  PREMISE:  ($AND (NOTSAME CNTXT SITE ELOOD)
                  (NOTKNOWN CNTXT AIRGROW)
                  (NOTSAME CNTXT ANTRY))
  ACTION:  (CONCLUDE* CNTXT AIR TALLY ((FACUL 500)
                         (ANAEROBIC 200)
                         (OBLIGATE-AEROB 300)))
```

```
This is my understanding of your rule:

--------
  If:  1) The site of the culture is not blood, and
      2) It is not known whether the organism was able to grow aerobically,
         and
      3) The lab has not attempted to grow the organism anaerobically
  Then:  There is evidence that the aerobicity of the organism is facul (.5)
           anaerobic (.2) obligate-aerob (.3)

do you agree with my interpretion?
** YES
good.
```

EXAMPLE IV.

```
Would you like to enter a rule?
** YES
If':      1 - THE INFECTION WHICH REQUIRES THERAPY IS PERITONITIS OR
              PELVIC-ABSCESS
    and 2 - THE SITE OF THE CULTURE IS ELOGD
    and 3 - THE PORTAL OF ENTRY IS GI
    and 4 -
Then:  1 - THE ORGANISMS WHICH MIGHT EE CAUSING THE INFECTION COULD BE
           BACTEROIDES (.7) E-COLI (.53) KLEBSIELLA-PNEUMONIAE (.18)
           PROTEUS (. 17) ENTEROBACTER (.11)
    and 2 -

the internal format of the rule is :

LISPRULE
--------
  PREMISE:  ($AND (SAME CNTXT TREATINF (ONEOF PERITONITIS PELVIC-ABSCESS))
                  (SAME CNTXT SITE BLOOD)
                  (SAME CNTXT PORTAL GI))
  ACTION:  (CONCLUDE* CNTXT COVERFOR TALLY ((BACTEROIDES 700)
                         (E.COLI 530)
                         (KLEBSIELLA-PNEUMONIAE 180)
                         (PROTEUS 170)
                         (ENTEROBACTER 110)))

Tnis is my understanding of' your rule:

  If:  1) The infection which requires therapy is one of: peritonitis pelvic-
          abscess, and
       2) The site of the culture is blood, and
       3) The portal of entry of the organism is gi
  Then:  There is evidence that the organisms (other than those seen on
           cultures or smears) which might be causing the infection is
           bacteroides (.7) e.coli (.53) klebsiella-pneumoniae (.18) proteus
           (.17) enterobacter (.11)
```

```
do you agree with my interpretation?
** YES
good.
```

EXAMPLE V.

```
Would you like to enter a rule?
** YES
If:      1 - THE INE'ECTION REQUIRING THERAPY IS NOT MENINGITIS
    and 2 - ORGANISMS WERE NOT SEEN ON THE STAIN OF THE CULTURE
    and 3 - THERE IS NO EVIDENCE THAT THE TYPE OF THE INFECTION IS NOT TB
    and 4 - THE CRYPTOCCCCAL ANTIGEN IN THE CSF WAS POSITIVE OR
               THE CSF COCCIDIOIDES SEROLOGY WAS POSITIVE
    and 5 -
Then: 1 - THE TYPE OF THE INFECTION IS PROEAELY NOT TB (.8)
    and 2 -

Tne internal format of the rule is:


LISPHULE
--------
  PREMISE:  ($AND (NOTSAME CNTXT TREATINF MENINGITIS)
                  (NOTSAME CNTXT SPECSTAIN)
                  (MIGHTBE CNTXT TYPE TB)
                  ($OR(SAME CNTXT CRYPTO-SEROLOGY)
                      (SAME CNTXT COCCI-SEROLOGY)))
  ACTION:  (CONCLUDE CNTXT TYPE TB TALLY -800)

This is my understanding of your rule:

--------
  If:  1) The infection which requires therapy is not meningitis, and
       2) Organisms were not seen on the stain of the culture, and
       3) There is no evidence that the type of the infection is not tb, and
       4) The cryptococcal antigen in the csf was positive, or
          The csf coccidioides serology was positive
  Then:  There is strongly suggestive evidence (.8) that the type of the
            infection is not tb

do you agree with my interpretation?
** YES
good.
Would you like to enter a rule?
** NO
Ok; good bye.
```

CONCLUSION.

This paper nas described a technique to translate English inputs expressed by an expert of a specialized domain into underlying structures used by a knowledge-based system. The trade-off between freedom of expression and reliability in the interpretation lead us to use a method that selectively ignore certain phrases which might not be crucial f'or the translation. However, this method may cause ambiguities in interpretation, hence we emphasize the importance of the expert's agreement with an interpretation. The current grammar produces adequate parses on about 90% of the statements contained in MYCIN rules. On a sample of 208 sentences chosen from current MYCIN rules and various repnrasings obtained from physicians, 55% were parsed by the general cornain independent grammar, 35% by a grammar specific to the infection diseases aomain and 10% were failed to be parsed, due to paraphrasings tnat would require a tremendous amount of' knowledge outside the medical field. Parsing these sentences with this semantic grammar is fairly f'ast (between 1 and 2 seconds). The level of understanding currently provided seems to be sufficient f'or the acquisition of' new rules.

FUTURE DIRECTIONS.

I see future improvements and extensions of this work in the following major directions. First, the grammar must be enlarged in order to make the system more "habitable" that is to enable the expert to remain within the limits of' the language accepted without his being conscious of these limits.

Second, the technique described here, along with strateqies for interpreting dialogs and texts should be suitable for building a general purpose interface for use by different tasks within a knowledge-based system such 'as question-answerinq, volunteered data via summaries describing clinical nistory and current status of patients.

ACKNOWLEDGEMENT

*

*   *

I wish to express my gratitude for interest, advice and/or helpful comments on drafts of the paper to: Bruce Buchanan, Jan Aikins, Jim Bennett, Bill Clancey, Handy Davis, Larry Fagan, Ted Snortliffe, Carli Scott, Helen Tognetti and Bill Van Melle.

*

*   *

APPENDIX : A sample of grammar rules.

Numbered rules are general, rules mentioning a concept between brackets are specific rules. Procedures having a nnemotechnic name such as YNPARMCNTXT, which stands for yes-no-parameter-context, are primitives allowing the recognition of non-terminal elements of the grammar rules. Most of the rules shown here have been already explained in section 111.4. The code is clearly very close to the BNF form used before. SK denotes the flag used for "tightness", always nil at the first pass, set to TRUE at the second if any.

```
(R010
  [LAMBDA (SK) **COMMENT**
    (AND (MULTPARMCNTXT)
         (SKIP SK)
         (NOVALFUN])

(R011
  [LAMBDA (SK) ** COMMENT**
    (AND (CONTEXT)
         (FACULFUN)
         (SKI? SK)
         (YNPARM)
         (CHECK2 TYPE-OBJ TYPE-P)
         (SETQ LASTCLAUSE (LDIFF REST CUR])
             (* saved for further pronoun resolution)

(R020
  [LAMBDA NIL ** COMMENT**
    (AND (MULTPARMCNTXT)
         (FACULFUN)
         (CHECK1 PARAM (ISVAL))
         (CHECK2 TYPE-P TYPE-OBJ])
```

```
(<YNSYMPTOM>
  [LAMBDA NIL  ** yes/no symptoms of uti **
    (AND (SURF 'PATIENT)
         (FACULFUN)
         (SURF 'SYMPTOM)
         (COND
           ((SURFL '(OF MUMPS) )
             (SETQQ PARAM (MUMPSYM)))
           ((AND (SURF 'OF)
                 (SKIPWORD 'NEW)
                 (SKIPWORD 'OR)
                 (SKIPWORD 'INCREASING)
                 (SURE' 'VAGINAL-DISCHARGE))
             (SETQQ PARAM (VAGDIS)))
           (( AND (SKIPUNTIL 'LOWER)
                 (SURF 'URINARY-TRACT))
             (SETQQ PARAM (LOWER-UTI-SX])
```

The previous rule enables the recognition of sentences as shown below:

```
                                          of mumps
                has                       of [new] [or] [increasing]
 [The] patient  has no         symptom[ s ]        vaginal discharge
                does not have             referable to the
                                          concerning the    lower urinary
                                          of the                tract
```

```
(YNPARMCNTXT
  [LAMBDA NIL  ** True if matches <ynparm> of <context> or
         ( COND                    <context> <ynparm> **
           ((YNPARM)
             (SURF 'OF)
             '(CONTEXT)
                   T)
           ((CONTEXT)
             (COND
               ( (YNPARM)
                   T]
```

REFERENCES

J.Brown, R.Burton: Multiple representations of knowledge for tutorial reasoning. In D.Bobrow and A.Collins (Eds.), Representation and understanding, Academic press, New York, 1975.

R.Burton: Semantic grammar, an engineering technique for constructing natural language understanding systems, BBN report n0 3453, december 1976.

R. Burton, W. Woods : A compiling system for augmented transition networks, Proc. of Conference on Computational Linguistics, Cttawa 1976.

E.Charniak: Toward a model of children's story comprehension, A.1 TR-266 M.I.T, Artificial intelligence laboratory, Cambridge, Mass, 1972.

E.Codd: Seven steps to Rendezvous with the casual user, Proc. IFIP TC-2 Working conference on data base management systems, Cargese, Corsica, April 1-5, 1974.

E.Codd, R.Arnold, JM.Cadiou, C.Chang, N.Roussopoulos: Rendezvous version 1, An experimental English-language query formulation system for casual users of' relational data bases, IBM research laboratory, San Jose California, January 1978.

K.Colby, R.Parkinson, B.Faught : Pattern-matching rules for the recognition of natural language dialogue expressions. A.1 memo 234, Stanford atificial intelligence project, 1974.

R.Davis: Applications of meta-level knowledge to the construction, maintenance and use of large knowledge bases, Stanford Artificial Intelligence Laboratory, Memo AIM-283, July 1976.

J.Earley: An efficient context-free parsing algorithm. Corn. of the ACM, 13 (1970).

G.Hendrix: The Lifer manual, A guide to building practical natural language interfaces. Technical note 138, Artificial intelligence center, Stanford research institute, Menlo park, California 1976.

R.Kaplan : A general syntactic processor, in Matural Language processing Ed. Randall Rustin, New York, Algorithmics press, 1973.

E.Sacerdoti: Language access to distributed data with error recovery, Proc. of 5th IJCAI, Cambridge, Mass, August 1977.

R.Schank: Identification of conceptualizations underlying natural language, in.Schank and Colby (Eds.) Computer models of thought and language, San Francisco, Freeman & Co. 1973.

C.Scott, W.Clancey, R.Davis, E.Shortliffe : Knowledge-based consultation systems groups, Stanford heuristic programming project memo MPP-77-1, february 1977.

E.Shortliffe: Mycin, a rule-based computer program for advising artificial intelligence laboratory, Memo AIM-251, October 1974.

E.Shortliffe : Computer-based medical consultations : MYCIN , Artificial intelligence serie n0 2, Elsevier computer science library, 1976.

W.Teitelman, Interlisp reference manual, Xerox Palo Alto research center, Palo Alto, California 1975.

D.Waltz: Natural language access to a large data base: an engineering approach, Proc. of the 4th IJCAI, Tbilisi, USSR, 1975.

D.Waltz, B.Goodman: Writing a natural language data base system, Proc. of 5th IJCAI, Cambridge, Mass, August 1977.

W.Watt: "Habitability", American documentation, vol.19 n0 3, July 1968.

Y.Wilks: An artificial intelligence approach to machine translation, in R.Schank and K.Colby (Eds.), Computer models of thouqht and language, San Francisco Freeman & Co, 1973.

Y.Wilks: Natural language understanding systems within the A.1 paradigm a survey and some comparisons, Stanford artificial intelligence laboratory, Memo AIM-237, december 1974.

T.Winograd: Understanding natural language. New York academic press [illegible] .

T.Winograd: Five lectures on artificial intelligence, Stanford AI-Memo-246, Stanford University, 1974.

W.Woods: Transition network grammars for natural language analysis, CACM vol.13,10, October 1970.

W.Woods, R.Kaplan, B.Nash-Webber, The Lunar sciences Natural language system, final report, report n0 2378, BBN Inc.,Cambridge, Mass 1972.

W.Woods : What's in a link, in D.Bobrow and A.Collins (eds.), Representation and understanding, Academic press, New York, 1975.