FONAL

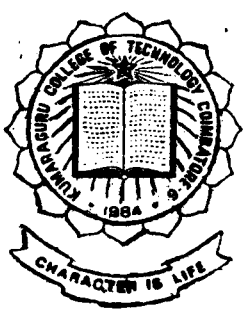
File Operations in NAtural Language

P-152

Project Report

Work of

Rajesh .B
Muruganand .K
Aju Kuriakose Ninan



1991-92

Guided by

Ms. R. Pavai. B.E.,

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING
OF BHARATHIAR UNIVERSITY
COIMBATORE

Department of Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore-641 006

Coimbatore-641 046 during the academic year 1991-92.

ACKNOWLEDGEMENT

We express our sincere gratitude to our Revered **Prof.P.Shanmugam** B.E.,M.Sc.,M.S.,M.I.E.E.E.,M.I.S.T.E.,for allowing us to make use of the available facilities.We thank you Sir, for your valuble suggestions and encouragement you had given us during the course of the project.

We wish to thank our Beloved principal Major **T.S.Ramamurthy** who has provided us the infrastructural facilities for carrying out this project work.

We whole heartedly thank our Guide and class advisor Ms **R.Pavai** B.E,.MISTE.,for her motivating guidance and moral support she provided us .Without her willingness and involvement,this project wouldn't have been completed.

We thank Mrs. **Gomathy** and Mr.**Shanmugha Shankar,** lecturers of CSE department for lending a helping hand during the final stages of this project.

We thank all our friends, in particular **M.Muthu Kumar**, who have helped us in completing this project.

NLP Team

SYNOPSIS

Natural Language Processing is a branch of Artificial Intelligence which aims at making the computer to understand the commands given in the natural language.

FONAL is a natural language software which executes the File operations with the commands given in simple sentences.FONAL accepts three types of simple present tense sentences which should be in active form and non-recursive in manner.FONAL has an efficient and extensive parsing algorithm which parses the input sentences to considerable merit.FONAL has it's own grammar for this application.

FONAL consists of three major modules-the parser,the understander and the generator.The Parser does the work of paraphasing.The Understander actually recognises the meaning of the words.The Generator produces the result of the operation.FONAL has a dynamic database so the facts can be added and deleted.It's a menu-driven and user friendly software in which the operations can be done easily as per the instructions.The introduction part provides the complete details of the software.

CONTENTS

CHAPTER 1

INTRODUCTION TO NATURAL LANGUAGE PROCESSING

Natural Language Processing,abbreviated as NLP,is a branch of Artificial Intelligence.It aims at making the computer understand everyday conversational human language rather than specialized computer languages.The work in this field has been going on from since early 1960's.Many AI professionals believe that the most important task that AI can solve is Natural Language Processing.The reason for this belief is that NLP opens the door for direct human-computer dialogues which would bypass normal programming and operating system protocol.

Natural Language Processing tries to make the computer to understand the commands written in standard human language.By studying and exploring NLP,one can gain a better understanding of how humans store,think about and communicate knowledge.NLP's goal is to design computer systems that can understand Natural Language.Natural Language is a form of agreement among members of a culture or subculture as to what words and symbols mean and how they are put together.

Natural language is the use of words and symbols to communicate between humans. It consists minimally of vocabulary and sentence structure and rules about how those two objects work together. Let's consider an example of natural language sentence. The word 'bird' means a living

object with wings and feathers that flies through the air only because there is a general agreement among this culture that when we say 'bird', that is what we mean. If you and I agreed that when we the word 'bird' we mean a young girl,probably attractive, we may find ourselves better understood among younger community than among oldies. If, instead, we decide that 'bird'means "a long,leisurely walk in the roads", we will know precisely what we mean by "let's go for a bird" but others will find us a bit strange. Similarly, we have agreement about how we put words together to form sentences.

A declarative sentence(one that states a fact) generally has the pattern of a subject followed by a verb optionally followed by an object.Ex:"I program computers " is such a sentence with 'I' the subject,'program, the verb and 'computers' the object. If I say,'program I computers' or 'computers program I', the degree to which I communicate is lessened.

CHAPTER 2

NATURAL LANGUAGE UNDERSTANDING

For a computer to be capable of understanding a natural language it should be able to understand both written language and spoken language.Understanding written language needs the use of lexical,syntactic and semantic knowledge of the language and the real world information whereas spoken language requires all the abilities for understanding written language along with the additional knowledge on phonology and on the ambiguities that arisein speech.

Hence the first step towards making computers understanding natural language is to make them understand the written language.The understanding of language is made difficult by the following major factors:

1.The complexity of the target representation into which the matching is being done.

2.The level of interaction of the components of the source representation.In a natural language sentence,changing a single word can alter the entire structure that is used to represent it.

3.The type of mapping between source and target representation:one-to-one,one-to-many,many-to-one,many-to-many.This is due to the richness of structure and vocabulary of the language.

Understanding natural language inputs can be broadly viewed to consist of the following;

1.Understanding single sentences.

2.Understanding multiple sentences.

3.Dialogue understanding.

2.1 UNDERSTANDING SINGLE SENTENCES.

Understanding single sentences form the basis for the understanding of multiple sentences and dialogues.Understanding each word in the sentence and putting these words together to form a structure that represents the meaning of th entire sentence.The next step of understanding a sentence that is combining the words of the sentence to form a structure that represents the meaning of the sentence, requires the knowledge of the language being used,knowledge of the conventions adopted in the language and knowledge of the domain.

Understanding of single sentences is divided into three parts:

2.1.1. Syntactic analysis

2.1.2. Semantic analysis

2.1.3. Pragmatic analysis

2.1.1 SYNTACTIC ANALYSIS

This consists of creating a structure from the sequence of words in the sentence that depicts the relationship between the words.Thus word sequences that do not confirm to the language rules for combining the words will be rejected.

This process is called parsing.To parse a sentence, the parser uses the grammer that describes the structure of the strings in the language and assigns to a structure to the grammatical sentences given to it.

2.1.2. SEMANTIC ANALYSIS

This consists of assigning meanings to the structure created by the syntactic analyser. As the correct interpretation of a sentence usually involves some semantic information to produce a single interpretation can be adopted. One such powerfull method is the ATN. The other approaches can be divided into three classes. They are Semantic grammer, Case grammer and semantic filtering of syntactically generated parses.

2.1.3. PRAGMATIC ANALYSIS

It concerns how sentences are used in different contexts and how contexts affect the interpretation of the sentence.

2.2 UNDERSTANDING SINGLE SENTENCES.

Understanding multiple sentences involves not only understanding the individual sentences,but also discovering the relationship among the sentences.There are a large number of such relationships that may be important in a particular text,including:

** Identical objects.

consider the text.

Bill had a red ball.John wanted it.

The word "it" should be identified as referring to the red ball.

** Parts of objects.

consider the text.

Janice opened the book she has bought.The title page was torn.

The phrase "the title page" should be recognized as being part of the book that was just bought.

** Parts of actions & objects involved in actions.

Jay went on a business trip to New York.He left on a early morning flight.

Taking a flight should be recognized as part of going on a trip.

Jessy decided to drive to the store.She went outside.But her car wouldn't start.

Jessy's car should be recognised as an object(the instrument) involved in her driving to the store.

** Planning sequences.

Sally wanted a new car.She decided to get a job.

Sally's sudden interest in a job should be recognised as arising out of her desire to get a new car and thus for money to buy one.

2.3 DIALOGUE UNDERSTANDING

For a user to communicate with the computer in natural language in order to make them carry out tasks required by him,dialogue in a very convenient way to express such information.A very useful application of natural language understanding by computers is to understand the dialogue between the user and the computer systems.Dialogue understanding is made difficult for the following reasons.

1.Mixed initiative: The control over the dialogue passes back and forth between the computers and the users.

2.Indirect speech acts:The response from the user may not be direct answers to the questions that they are asked.

3.Implicit presuppositions:The questions asked by the user can presuppose the facts that may not be true.

4.Anaphoric references.

5.Sentence fragments.

CHAPTER 3

NATURAL LANGUAGE STRUCTURES

TRANSITION NETWORKS

One approach to representing natural language structures uses the model known as Transition Networks. Transition Networks are based on the application of mathematical notations of graph theory and finite state machines to the study of grammers.

The FINITE STATE MACHINE is a theoritical (or actual) device which begins in a particular state and changes state when specific conditions occur. It is finite in that, at any point in its operation, the next state can be determined by knowing the current state and the conditions which can cause a transition. The most basic network is the FINITE STATE TRANSITION GRAPH, which consists of a group of notes, connected by directed arcs. Each note represents a state in a finite state machine and the arcs show the transitions from one state to another. Each arc is labelled with the conditions which causes the transition along that are from its tail to its head.

3.1 FINITE STATE TRANSITION NETWORKS

As the most basic level of transition network, the finite transition graph can represent the sequence in which words can appear in a sentence by following the path through the graph. For e.g., the grammer which allows the sequence

ARTICLE NOUN AUXILLARY VERB

such as in the sentence

THE BOY MAY SWIM

can be represented by the finite state diagram in Fig.1.

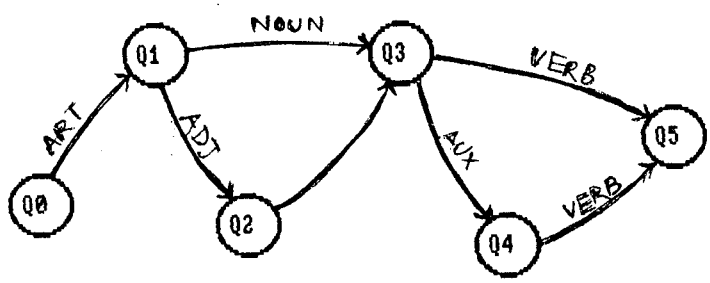


Fig.1

In the diagram the circles are the nodes and represent the particular state of the sentence recognizer or generator at that point. Each line between the nodes in an arc and indicates the terminal symbol which will cause a transition from the current state to the subsequent state. State Q0 is referred to as the initial state and Q5 as the final state. A transition network has only one initial state but have more than one final state.

3.2 RECRUSIVE TRANSITION NETWORKS

A RECRUSIVE TRANSITION NETWORK is like a finite state network in that it has one start state and one or more final states. All the states and arcs connecting them are named. In addition to these characteristics the RTN can have arc names which are state names-names of RTN's thus the term RECURSIVE.

If an arc is a terminal symbol, control moves to the node at the head of the arc and the process begins again. If the arc is a state name it represents a complete RTN. Therefore, control must pass to the initial state of the RTN named on the arc. In the lower level RTN control will be passed from arc to arc as before, until the final state is reached. If the final state is reached without error then control returns successfully to the higher level graph and continues as before.If the final state is not reached successfully then either an error has occured or more porbably, the arc being

tested does not apply. Essentially the processing proceeds form the start state by determining whether the first part of the sentence is a noun phrase or an auxiliary verb. The test for a noun phrase would be made by the saving the position of the nodes at the tail of the arc in order to be able to restore it later. If in fact the noun phrase is detected then processing proceeds by finding the state saved, following the successful arc and moving to the next state. If a noun phrase was not found, then the test for auxiliary verb would be made and that arc followed if successful. If neither noun phrase nor an auxiliary verb were found, then the string of symbols being tested would not be accepted as a sentence by this RTN. In the RTN's in Fig.2 the arc's labelled noun phrase and proposition phrase have their owngraphs.

SENTENCE ANALYSIS WITH AN RTN

Using an RTN to analyse a sentence can determine whether the sentence is grammatical according to the grammar represented therin. The usual way to determine the word type is look each word up in the lexicon where the information is stored.

for e.g. If the sentence input is

CAN THE YELLOW BIRD FLY?

The lexical categories would be

AUX DET ADJ N V

Starting at s, the symbol AUX would take us to Q2. Form there since the only option is noun phrase (NP), and the next word is identified as determiner (DET), the processing must

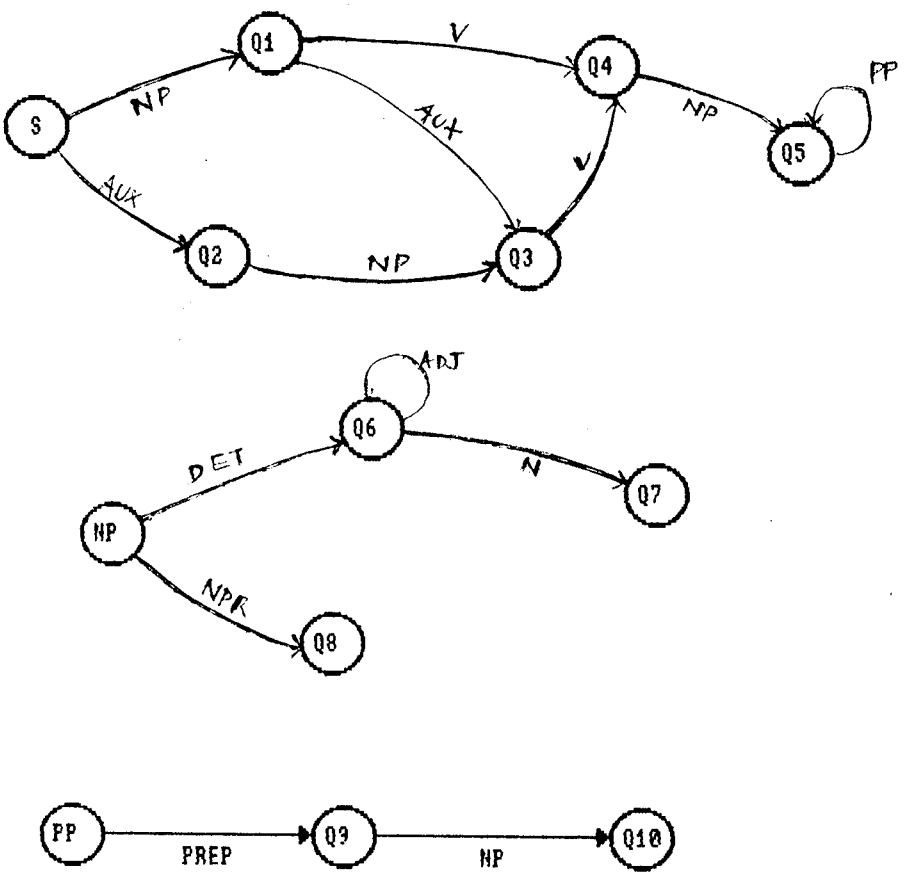

S
Q1
Q2
Q3
Q4
Q5
NP
AUX
V
AUX
NP
V
NP
PP
Q6
Q7
Q8
ART
DET
N
NP
NPR
PP
Q9
Q10
PREP
NP

Fig. 2

go beyond the initial test. Here the new graph, NP must be checked. Finding that the types DET ADJ N do indeed constitute an acceptable NP (noun phrase) we return to the higher level knowing that the arc with the state name NP is the one to take. Now at Q3 the symbol V (verb) takes up along the arc Q4 which is the terminal node. Since all the works have been used and we have reached a final stage successfully, the sentence be declared grammatically. If at any point along the path through the graph the word category did not match the possible condition along an arc or a subordinate graph tested could not be completed successfully an error would have been detected. In some instance, the error situation can be resolved by back tracking i.e., by moving back to preceding nodes and attempting a different direction.

3.3 AUGMENTED TRANSITION NETWORKS

ATN's are basically like RTN's with more conditions and actions for moving to through the network. Rather than just accepting or rejecting each word or phrase as it is encountered, the structure of the sentence is built up in the parse tree as input words match the elements of the network. The various parts of the sentence are held in registers until the structure of the sentence as a whole can be determined. For example, when a verb is encountered it is stored in the V register; similarly all the words of a noun phrase would be stored in the NP register.

Following the language specifications, the transition network is defined to be one or more arc sets, where an arc set is a state and its associated arcs. An arc set is comparable to a node and its arcs in the RTN scheme.

An important consideration in implementing ATN's concerns the flexibility allowed by the implementation. Achieving this flexibility requires creation of two phases in the program. The second phase is a program which inputs sentences for analysis and performs the analysis based on a specific grammer. In order to create the environment which allows any grammar to be used, the first stage must be a computer to generate the tables which interpret uses for second stage. The most important constraint for right now is that the grammer be deterministic, in order to avoid problems of backtracking.

Implementing an ATN primiarily involves reading an interpreter for a particular grammer. This grammer is written in a programming language like LISP. The proggram will consist of two stages: first inputs the grammar and builds the necessary tables and the second actually executes the grammar, given an input sentence, by interpreting the tables built by the compiler.

For the first stage of the program, each arc set will be read in and the state name saved with a pointer to the beginning of the list of arcs for that set. Only one of the arcs will be possible at each node. The one arc to be taken

determines the actions to be performed, and the node to process next. Once the grammar has been stored and the tables have been build, the first phase is completed. The second phase is then an interpreter, which uses the grammer to control analysis of the sentence input be accessing the tables built in the first phase.

ATN's were originally developed to replace transformations as a method for sentence analysis. As such, they are a definite improvement. One inherent drawback is that the grammars,and the programs to interpret the grammers can become quite complex and unwieldy. The complexity involved cannot be eliminated or reduced without severely restricting the sentence structure.

CHAPTER 4.

FONAL

(File Operations in NAtural Language)

FONAL is a natural language software which executes the file operations when the command is given in simple sentences.

FONAL accepts three types of simple present tense sentences which should be in active form and should not be recursive in manner.

The three types of sentences are

1. NPR VP
2. VP
3. AUX NPR VP

where NPR -> NOUN

NP -> ART ADJ NOUN

-> ART NOUN

VP -> Verb NP

VP -> Verb PP

PP -> Prep NP

The examples for the above three types for "Creating a new file" are

1. I want to create a new file.
2. Create a new file.
3. May I create a file.

These sentences are constructed by taking

May -> Auxillary

I -> Noun

Create-> Keyverb

File -> Noun

New -> Adj

Want -> Verb

FONAL first parses the sentences in the phrases and finds the keyword for processing.It checks for the keyword which should be in the knowledge base,otherwise FONAL will give you the error message that your sentence is not useable for the application by FONAL.

FONAL has a modules for processing the sentences.In the first module it accepts the sentence,parses the sentences to atomic elements and stores it in a list.The second module checks the list,word by word,and finds any syntax error in the sentence.Syntax error will appear in the comment window,if the sentence is grammatically wrong (or) not in the restricted grammar adapted by FONAL.

The third module checks the semantics of the sentence by taking the keyword and noun on which the keyword is applied.Consider the follwing sentence,

CREATE A NEW RECORD

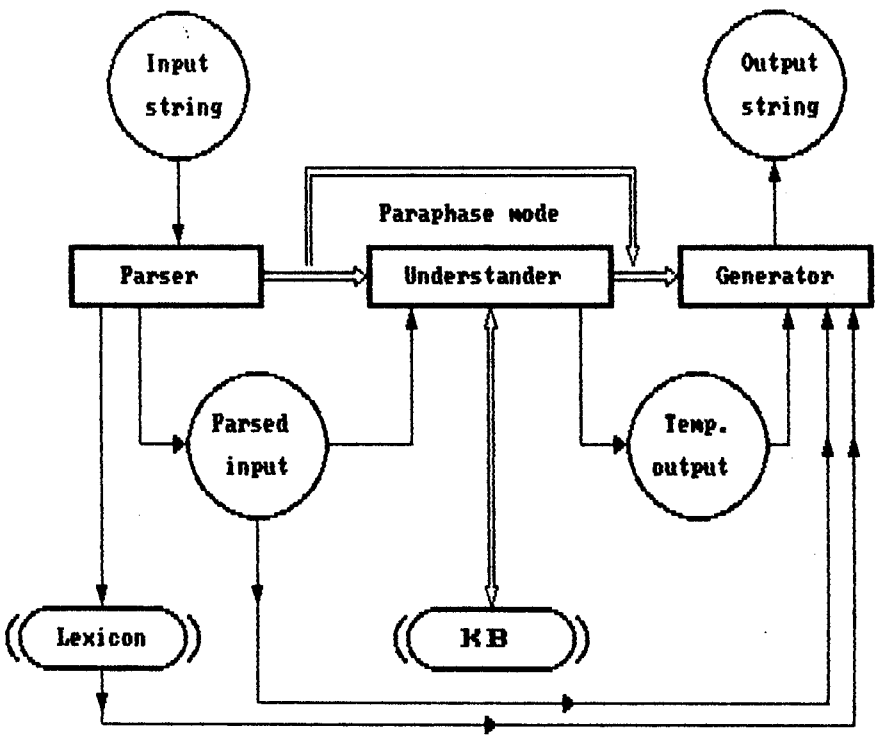
The database of the program contains both FILE & RECORD as Nouns.Here the sentence is syntactically correct but creating a new record is semantically wrong.Thus the third module detects the semantic error.

If the sentence is correct both syntactically and semantically,it executes the particular application file written in C and kept as an executable file.

FONAL also does the knowledge base operations.It adds or deletes a fact in the dynamic database used.It is a menu-driven and user friendly program in which the operation can be done easily as per the instructions.

FONAL has the introduction part to the learner of the learner of the software which explains the complete details about the usage of the software.

4.1 NATURAL LANGUAGE SYSTEM.



4.1 DESIGN OF NATURAL LANGUAGE SYSTEM

The NLS is designed to understand and manipulate language . It should be capable of accepting input, storing knowledge, drawing inferences, answering questions and generating responses. The NLS is a knowledge-based NL understanding system. The knowledge base is pre-compiled, i.e., a knowledge domain exists before the execution begins.

The major modules of any Natural language system are the

* Parser
* Generator
* Understander

PARSER

The Parser is the section of the code that reads the input sentence word by word and decides what is what.It accepts input and maps into an internal stucture compatible with the knowledge base .The Parser does the work of paraphasing the input sentence.Paraphasing is the process of splitting up the sentence word by word and separating the parts of speech.

The first step is the scanning proces , sometimes called the lexical analysis.This step breaks the input text into logical components or tokens.

GENERATOR

The Generator is the module that maps from internal structure to the output string.

The responses and the answers to the questions are generated by the generator .Consider the following dialogue

Did John attend classes today?

Yes John attended classes today.

The answer is generated by the generator after the input sentence has been passed and the meaning of the sentence recognized.The Generator or the answering part can be made more interactive and conversational so as to feel the pulse of a dialogue going on between a programmer and a computer.

An example of an interactive computer dialogue is provided by ELIZA,the most famous Natural Language Program.

Let's apply a three step process to a parser designed to process English rather than a program language.The scanner simply breaks the input stream into single units of words and punctuations.Given the text

A Boy ran fast.

A parser divides the text into five tokens.A, Boy, ran,fast and the period at the end of the sentence.A word in this sense is any collection of letters and numbers terminated by space or a punctuation symbol.If the word ends with a punctuation symbol , then it is really two words : the group of characters up to the punctuation mark and the punctuation mark itself.

At then next phase of parsing the natural language text, the parser applies the rules of the grammar to the tokens passed to it by the scanner.

UNDERSTANDER

This module access the knowledge base to obtain knowledge to draw inference and to add knowledge to knowledge base.Data sets include the knowledge base, lexicon,the input and the output strings.

The knowledge represents the memory of the system.

LEXICON

The lexicon provides the vocabulary for the system.All the words that are encountered in the program are included in the lexicon .The words are recognized along with syntactic and semantic information to parse.

When a text is given as input, the lexicon provides the meaning of the words in the sentence.

The responses are in the upper case letters

I am angry

WHY ARE YOU ANGRY?

Because my girl friend made me come here.

YOUR GIRLFRIEND MADE YOU COME HERE?

I am not angry . She just thinks i am. Women are all alike.

IN WHAT WAYS?

They're always nagging at us.

CAN YOU GIVE ME A SPECIFIC EXAMPLE?

I am too nervous to think right now.

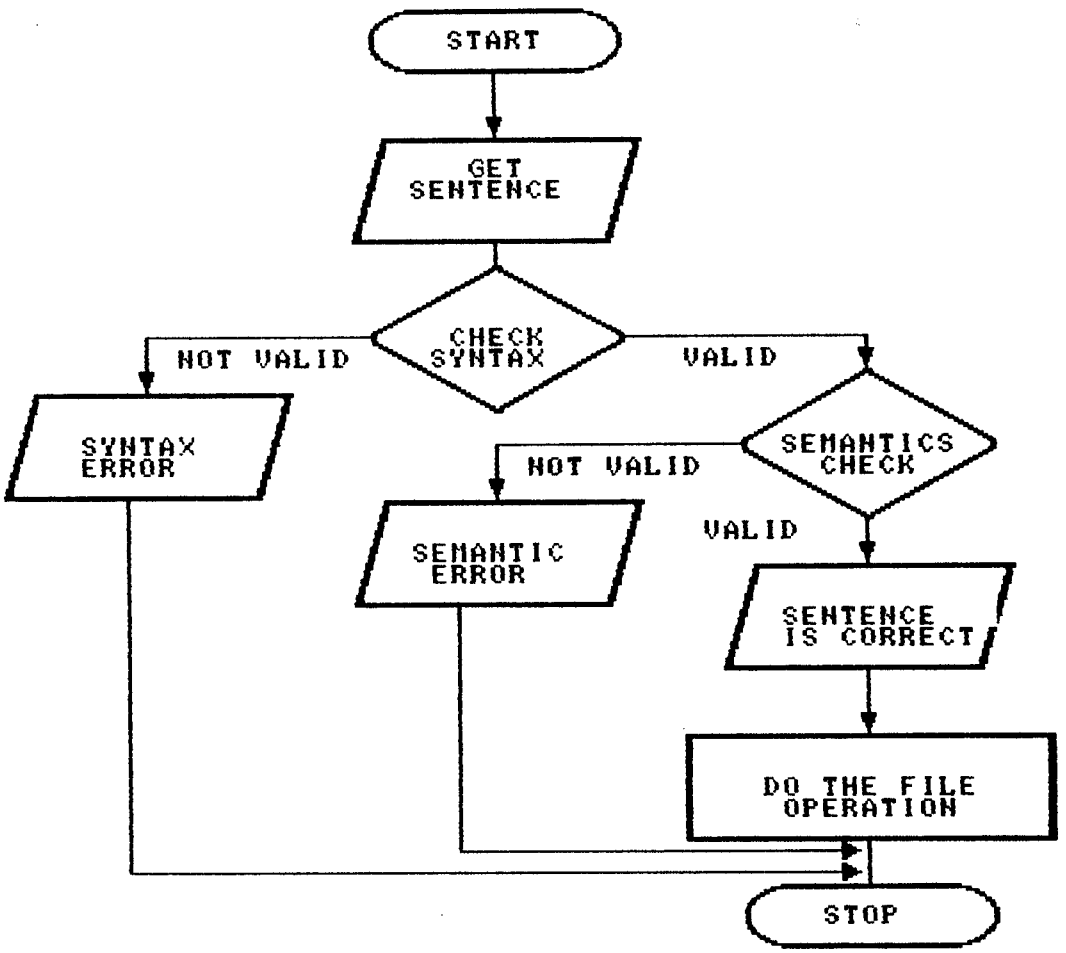
DO YOU THINK THAT COMING HERE WILL HELP YOU NOT TO BE NERVOUS?

I hope so.

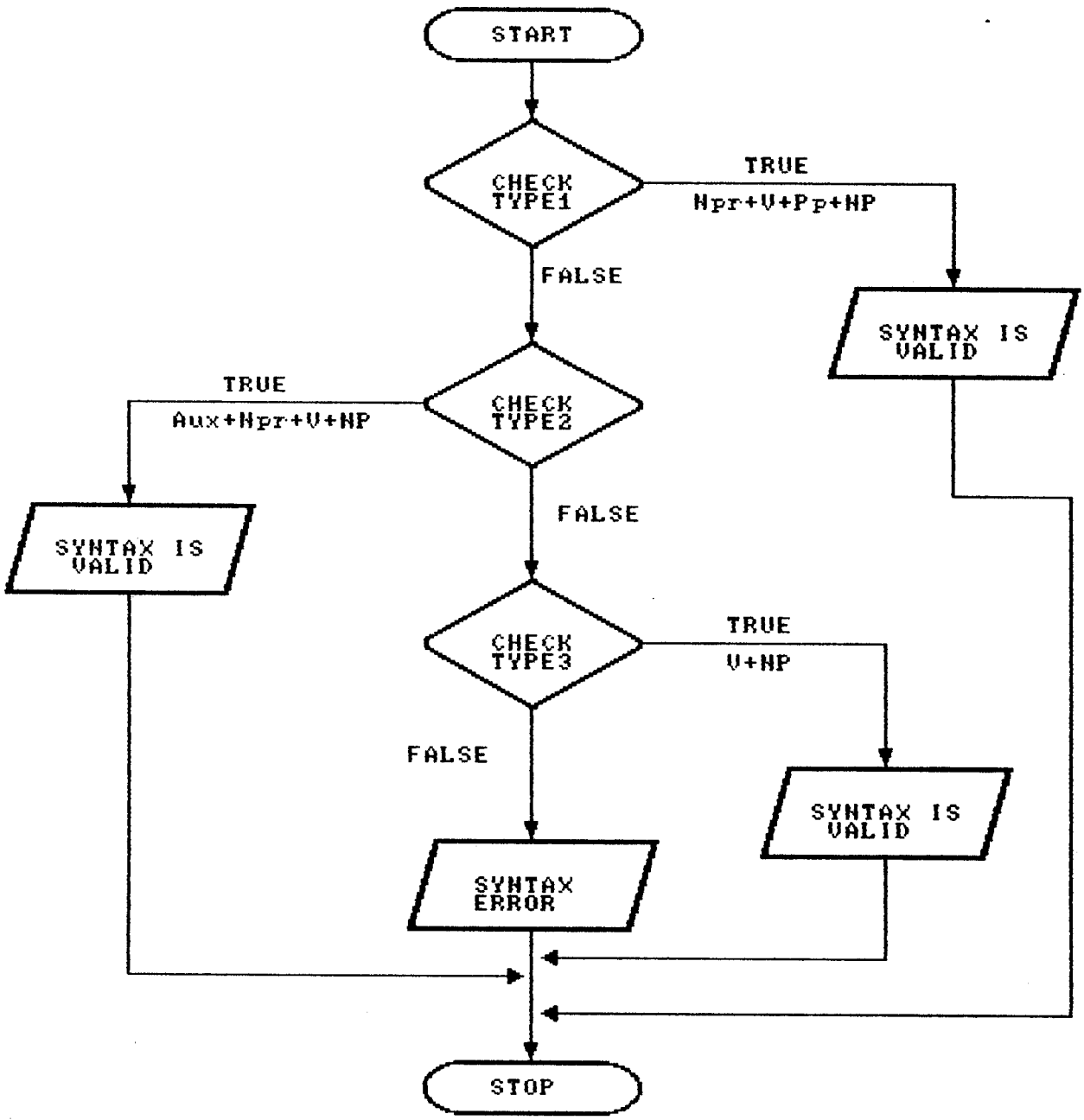
TELL ME MORE ABOUT YOUR GIRL FRIEND.

..... and the dialogue goes on.

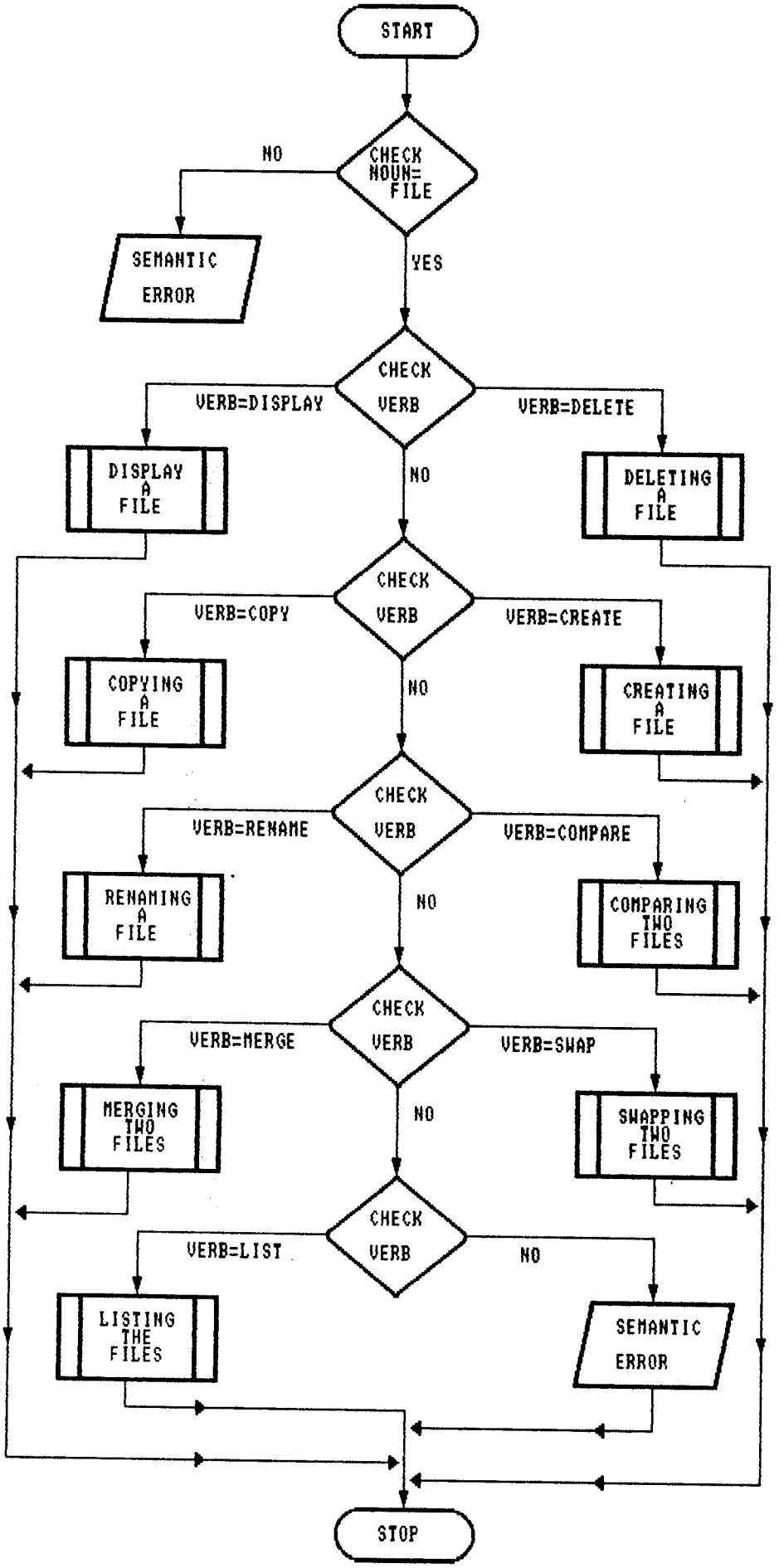
Flow of FONAL



Flow of SYNTAX



Flow of SEMANTICS



4.2 GRAMMAR ADAPTED IN FONAL

In FONAL, the sentences should be in simple present tense, active form and non-recursive.The three types are,

1.Npr VP

2.Aux Npr VP

3.VP

where NP -> Art Noun,

NP -> Art Adj Noun,

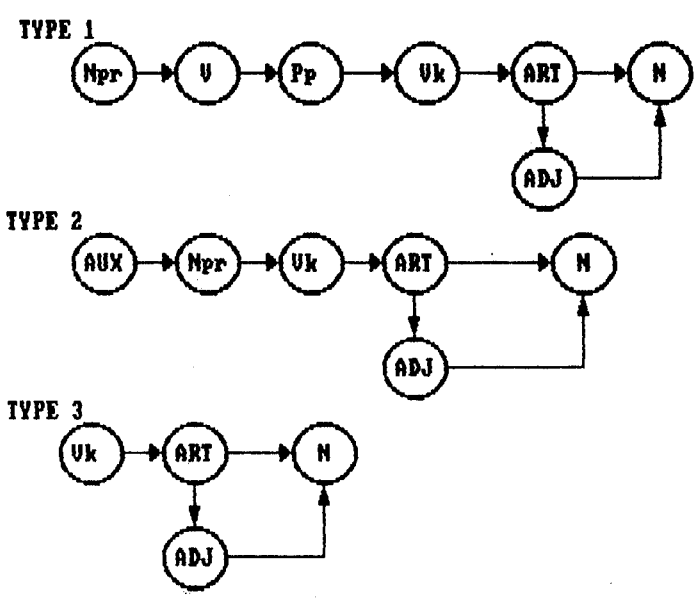
VP -> Verb(Keyverb) NP,

VP -> Verb PP,

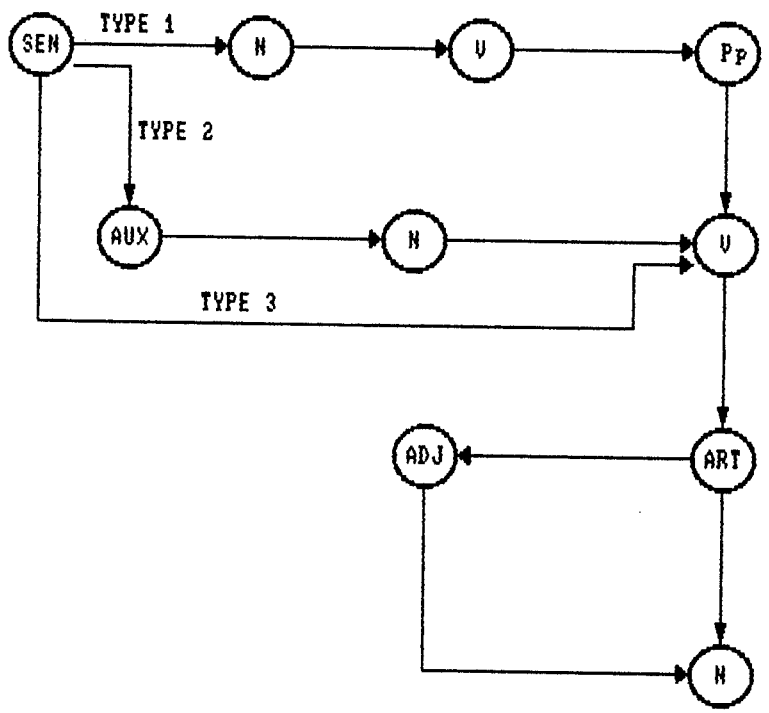
PP -> Prep VP,

Npr -> Noun.

The state diagram for the restricted grammar in FONAL is given below:



In FONAL, for all the given types of sentences,the combined diagram for the grammar in FONAL can be drawn as below:



4.3 FILE OPERATIONS

For a novice having his hands on the computer for the first time wishing to know the contents of the directory, he should know what command he should type in to list out the contents .

Natural Language comes into play at this juncture.The novice should type in "DIR" to list out the files.In case of a natural language system he can just type in "LIST OUT THE CONTENTS OF THE DIRECTORY" (or) "SHOW ME THE DIRECTORY" (or) "CAN I SEE THE FILES IN THE DIRECTORY".

The Natural Language system understands using keywords what the user aims at and what operations he wants to execute.The software used for the file operations consists of a noise disposal parser which takes into consideration only the keywords to understand the purpose of the input text.

The sentence formats are predefined and only the sentences that follow the syntax of the rigid format are accepted for parsing and further processes .

The sentences are of three types.The sentences should be in active form .The sentence should be in simple present tense.

VARIOUS FILE OPERATIONS IN FONAL

1.Creating a file.

2.Deleting a file.

3.Displaying the contents of a file.

4.Copying a file.

5.Renaming a file.

6.Listing out the directory.

7.Merging two files.

8.Swapping one file with another.

9.Comparing two files.

The above mentioned operations can be performed with the input given in the natural language which is valid in the restricted grammar both syntactically and semantically.

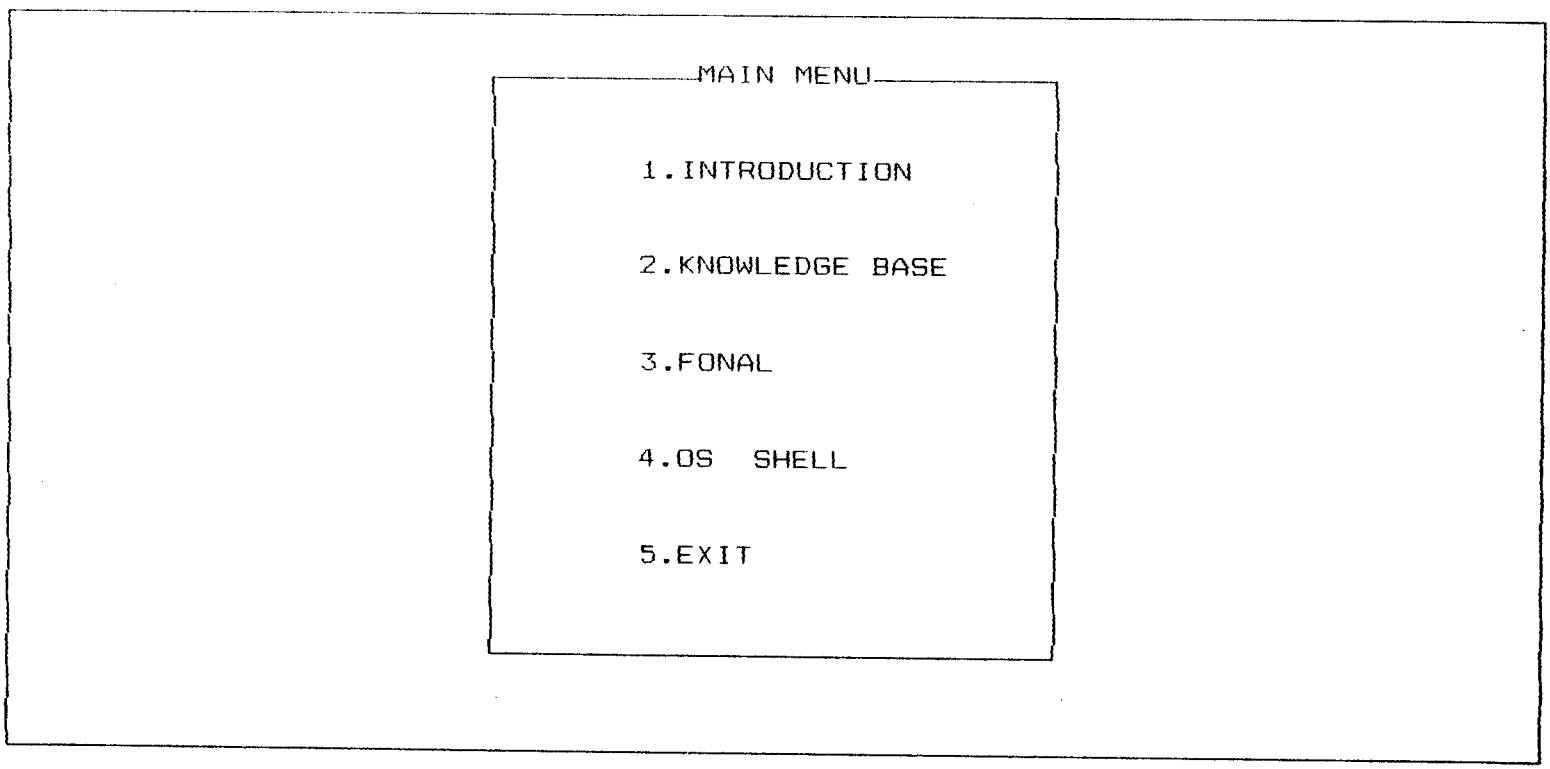
For creating a file,we can write the possible sentences by having

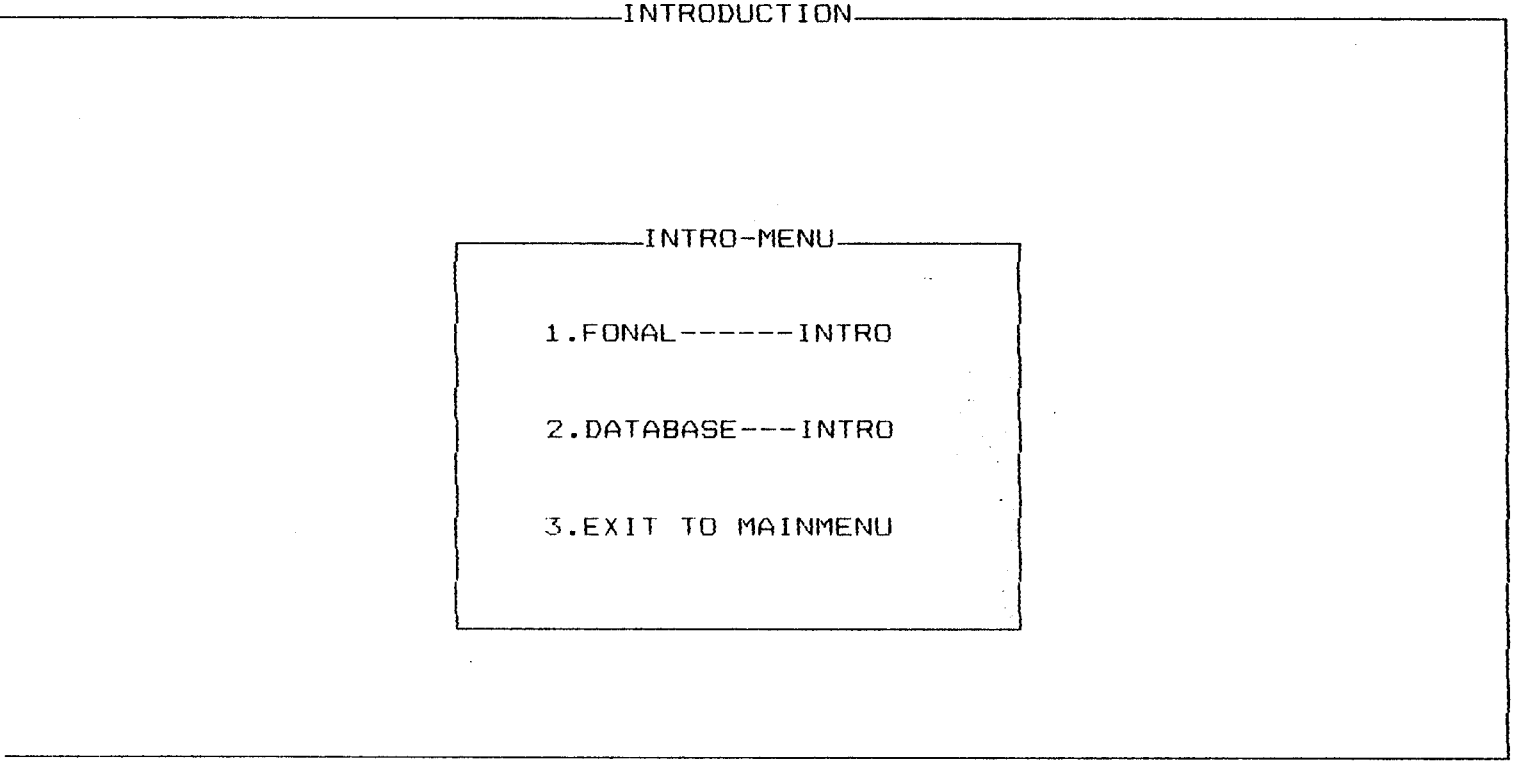
Pro noun -> I	Prep -> to
Verb -> like	Art -> a
Key verb -> create	Adj -> new
Aux -> can	Noun -> file

in the database,is as follows

1. I LIKE TO CREATE A FILE.
2. I LIKE TO CREATE A NEW FILE.
3. CAN I CREATE A FILE.
4. CAN I CREATE A NEW FILE.
5. CREATE A FILE.
6. CREATE A NEW FILE.

By having more verbs,Aux verbs and pronouns in dynamic database used we can write more combinations for a single file operation.


MAIN MENU
1.INTRODUCTION
2.KNOWLEDGE BASE
3.FONAL
4.OS SHELL
5.EXIT


INTRODUCTION
INTRO-MENU
1.FONAL------INTRO
2.DATABASE---INTRO
3.EXIT TO MAINMENU

FONAL-INTRO

Page 1

FONAL has been coined to imply File Operations in NAtural Language.It performs well both as an English sentence parser as well as an understanding system of considerable merit.The software includes an effective and extensive parsing algorithm together with a Natural Language Understanding System.The Understanding system makes optimum use of the dictionary in it's particular application.

Press Esc->Intromenu , Pgdn->Next page

FONAL-INTRO

Page 2

FONAL is a software designed to execute various file operations with the commands given in the natural language. FONAL has a restricted grammar for this particular application The grammar is restricted because of the complexity of the English Language.

Press Esc->Intromenu, Pgup->Previous Page, Pgdn->Next page

FONAL-INTRO

The input sentence would be command to carry out operations like creating a file,deleting a file,displaying the directory etc.After the sentence has been parsed,depending on the verb and the noun used,the operation would be executed.This is done by the understanding system.

INTRODUCTION

FONAL-INTRO

Page 4

In this software interfacing of prolog
with the C language has been done.C language
has been chosen becos it's the highest level
language that can be interfaced easily.

Press Esc->Intromenu , Pgup->Previous Page

INTRODUCTION

DATABASE-INTRO

The Knowledge Base represents the memory of the FONAL.In FONAL,the knowledge base is a DYNAMIC database i.e.,the facts are included during the execution of the program. The addition of the facts is done using the assertz predicate.The assertz predicate adds the fact at the end of the same type of the fact.The deletion of a fact is done by the retract predicate.The addition of the same fact is not allowed in FONAL.All the facts are stored in the file 'word.lst'.

INTRODUCTION

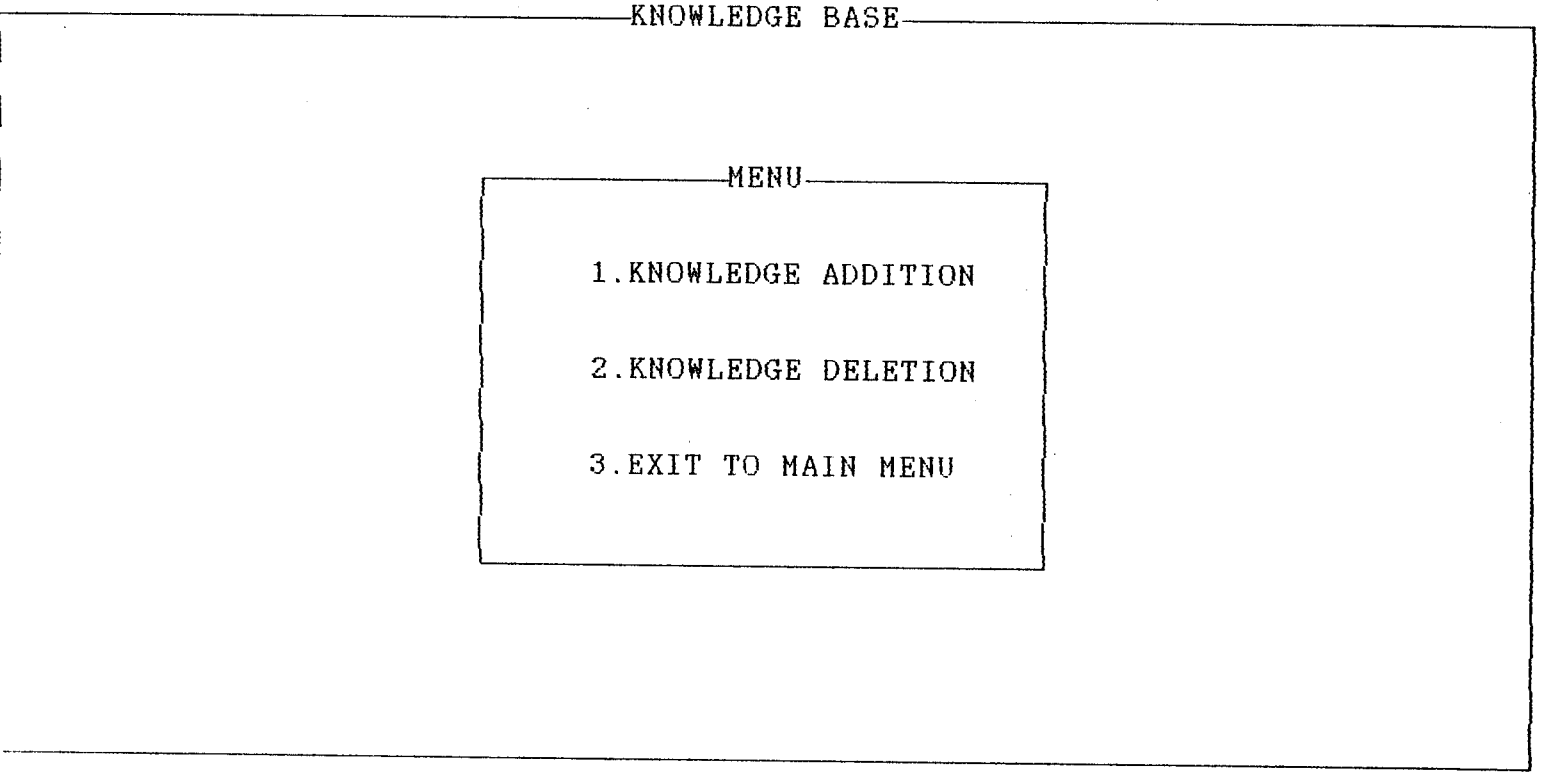
DATABASE-INTRO

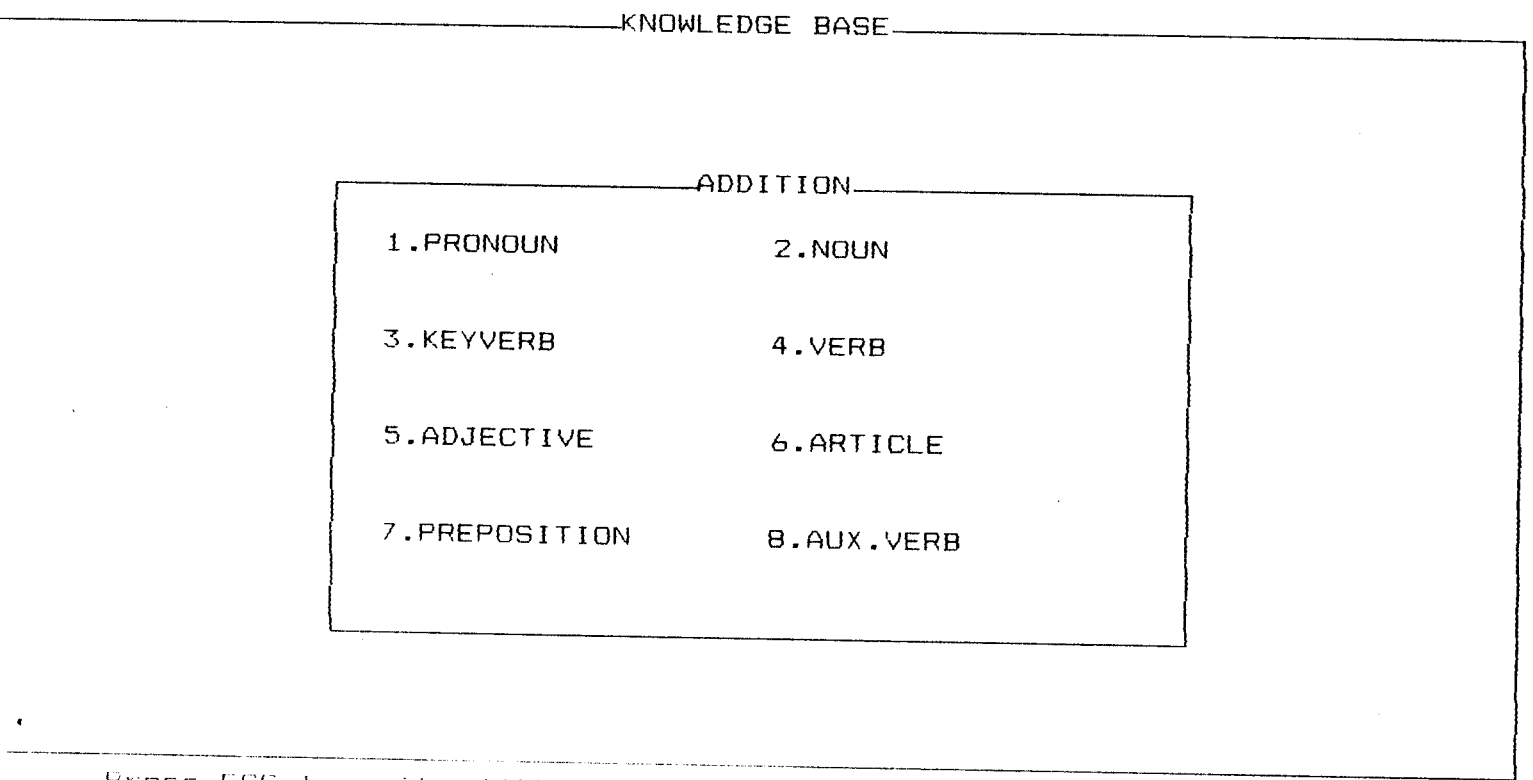
Page 2

Valid file operations in FONAL

1.CREATE	5.DELETE	9.RENAME
2.COPY	6.DISPLAY	1Ø.COMPARE
3.LISTOUT	7.LIST	
4.MERGE	8.SWAP	

Press Esc->Intromenu, Pgup->Previous page


KNOWLEDGE BASE
MENU
1.KNOWLEDGE ADDITION
2.KNOWLEDGE DELETION
3.EXIT TO MAIN MENU


KNOWLEDGE BASE
ADDITION
1.PRONOUN
2.NOUN
3.KEYVERB
4.VERB
5.ADJECTIVE
6.ARTICLE
7.PREPOSITION
8.AUX.VERB
Press ESC to quit addition

KNOWLEDGE BASE

ADDITION

Enter the keyverb to be added
rename

The fact KEYVERB(RENAME)

has been added.

Press any key to continue

KNOWLEDGE BASE

ADDITION

Enter the keyverb to be added
rename

The fact KEYVERB(RENAME)

is already present.

Press any key to continue

KNOWLEDGE BASE

DELETION

1.PRONOUN	2.NOUN
3.KEYVERB	4.VERB
5.ADJECTIVE	6.ARTICLE
7.PREPOSITION	8.AUX.VERB

Press Esc to quit deletion

KNOWLEDGE BASE

DELETION

Enter the preposition to be deleted
to

The fact PREPOSITION(TO)

has been deleted.

Press any key to continue

KNOWLEDGE BASE

DELETION

Enter the article to be deleted
an

The fact ARTICLE(AN)

is not available.

Press any key to continue

FONAL

Enter the sentence : delete the file

PARSING

Verb	DELETE
Article	THE
Noun	FILE

COMMENT

SYNTAX of your sentence is correct.
Wait for my proceedings...

Press any key to continue...

FILE DELETION

```
Enter the filename to be  deleted(with path) : b:\c\create.c
The file b:\c\create.c has been deleted .
A>
```

```
Enter the filename to be  deleted(with path) : b:\grap.c
The file b:\grap.c is not existing.
```

PARSING

Noun	I
Verb	WANT
Preposition	TO
Verb	CREATE
Article	A
Adjective	NEW
Noun	FILE

COMMENT

SYNTAX of your sentence is correct.
Wait for my proceedings...

Press any key to continue...

FILE CREATION

```
Enter the name of the file (with path) : b:rajesh

                                                The file is being created...

Do you want to enter fields (Y/N) ?

Enter the number of fields: 3
Enter the 1 field name: NAME
Enter the 2 field name: AGE
Enter the 3 field name: SEX
Do you want to enter Records (Y/N) ?

Enter the number of records: 2
Enter the data for - 1 - record's NAME : RAJ
Enter the data for - 1 - record's AGE : 21
Enter the data for - 1 - record's SEX : MALE
Enter the data for - 2 - record's NAME : MURUGI
Enter the data for - 2 - record's AGE : 45
Enter the data for - 2 - record's SEX : MALE

                                                The file has been create

Do you want to create another file (Y/N) ?
```

FONAL

Enter the sentence : display the file

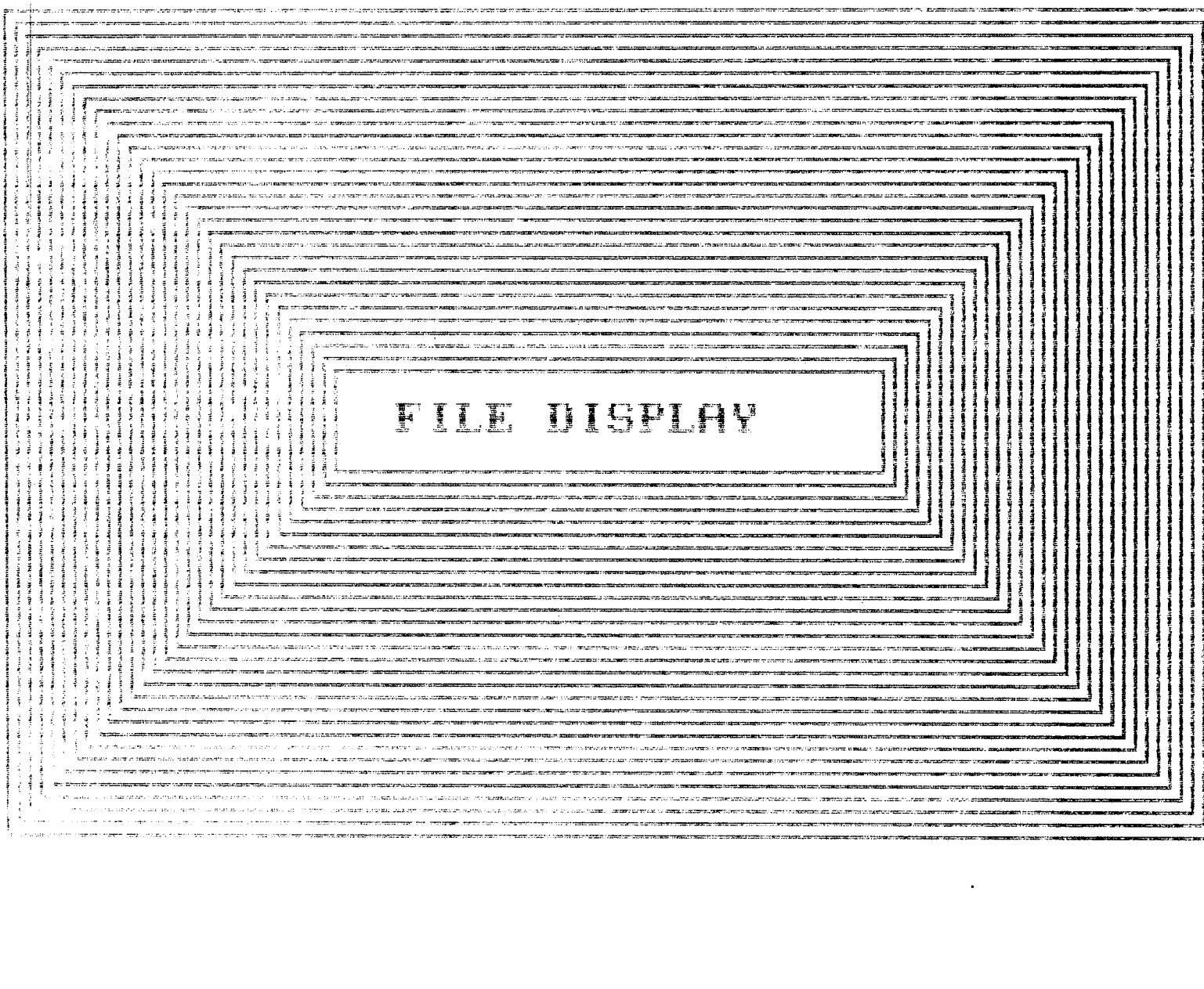
PARSING

Verb	DISPLAY
Article	THE
Noun	FILE

COMMENT

SYNTAX of your sentence is correct.
Wait for my proceedings...

Press any key to continue...


FILE DISPLAY

```
 Enter the file to be displayed : b:\c\display.c
/* program to display the file contents */
# include <stdio.h>
# include "b:\c\grap.c"
main()
{

  FILE *fp1;
  char str[80],fn[14];
  clrscr();
  g(" FILE DISPLAY ");
  clrscr();
  printf(" Enter the file to be displayed : ");
  scanf("%s",fn);
  fp1=fopen(fn,"r");
  while(!feof(fp1))
      {
        fgets(str,50,fp1);
        printf("%s",str);
      }
  fclose(fp1);
}
A
```

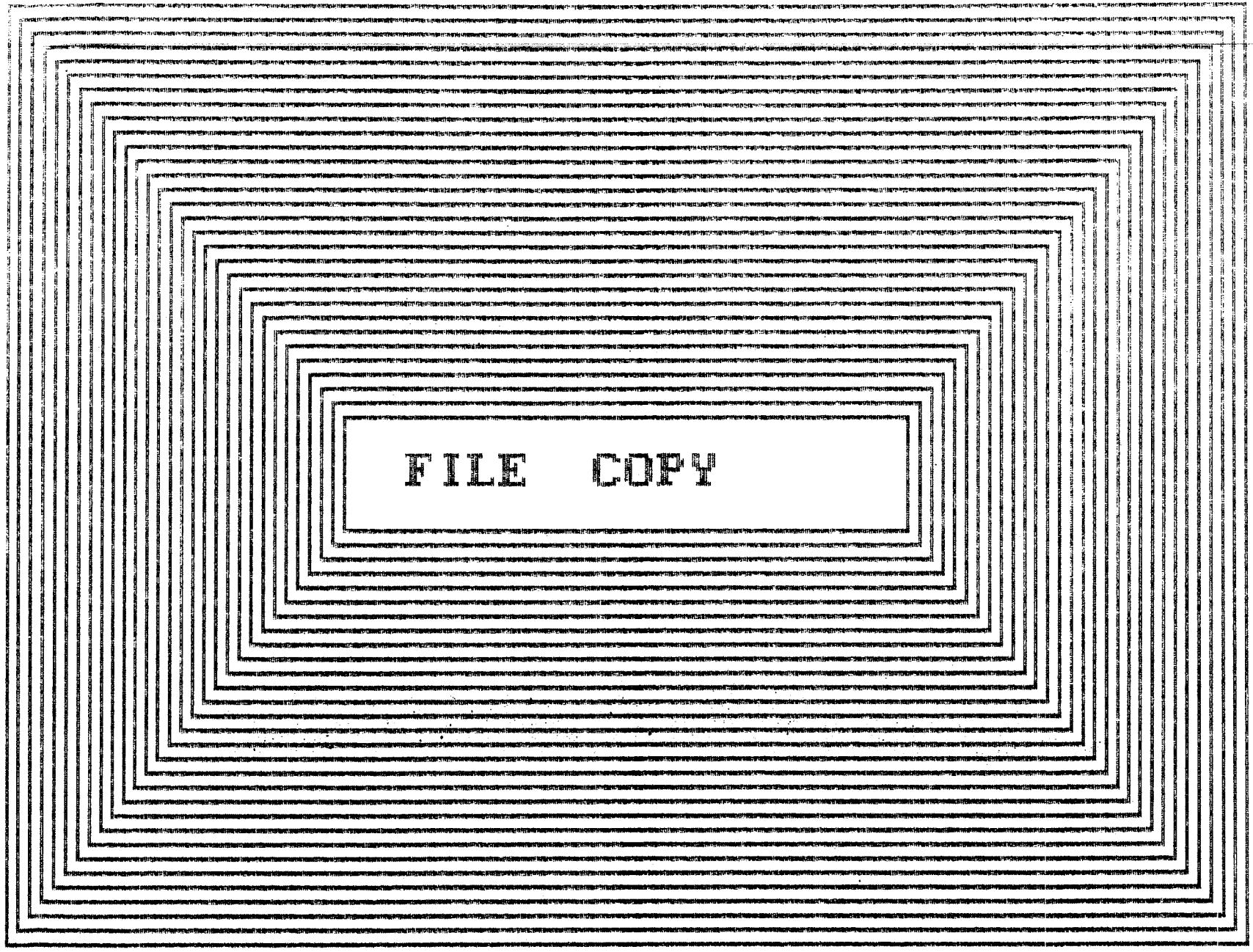
PARSING

Aux.verb	CAN
Noun	I
Verb	COPY
Article	A
Noun	FILE

COMMENT

SYNTAX of your sentence is correct.
Wait for my proceedings...

Press any key to continue...


FILE COPY

```
Enter the source file name : B:RAJESH
Enter the destination file name : B:RAJ


The file has been copied.
A>


Enter the source file name : B:GRAP.C
Enter the destination file name : B:P.P
The source file B:GRAP.C is not existing.
A>
```

FONAL

Enter the sentence : rename the file

PARSING

Verb	RENAME
Article	THE
Noun	FILE

COMMENT

SYNTAX of your sentence is correct.
Wait for my proceedings...

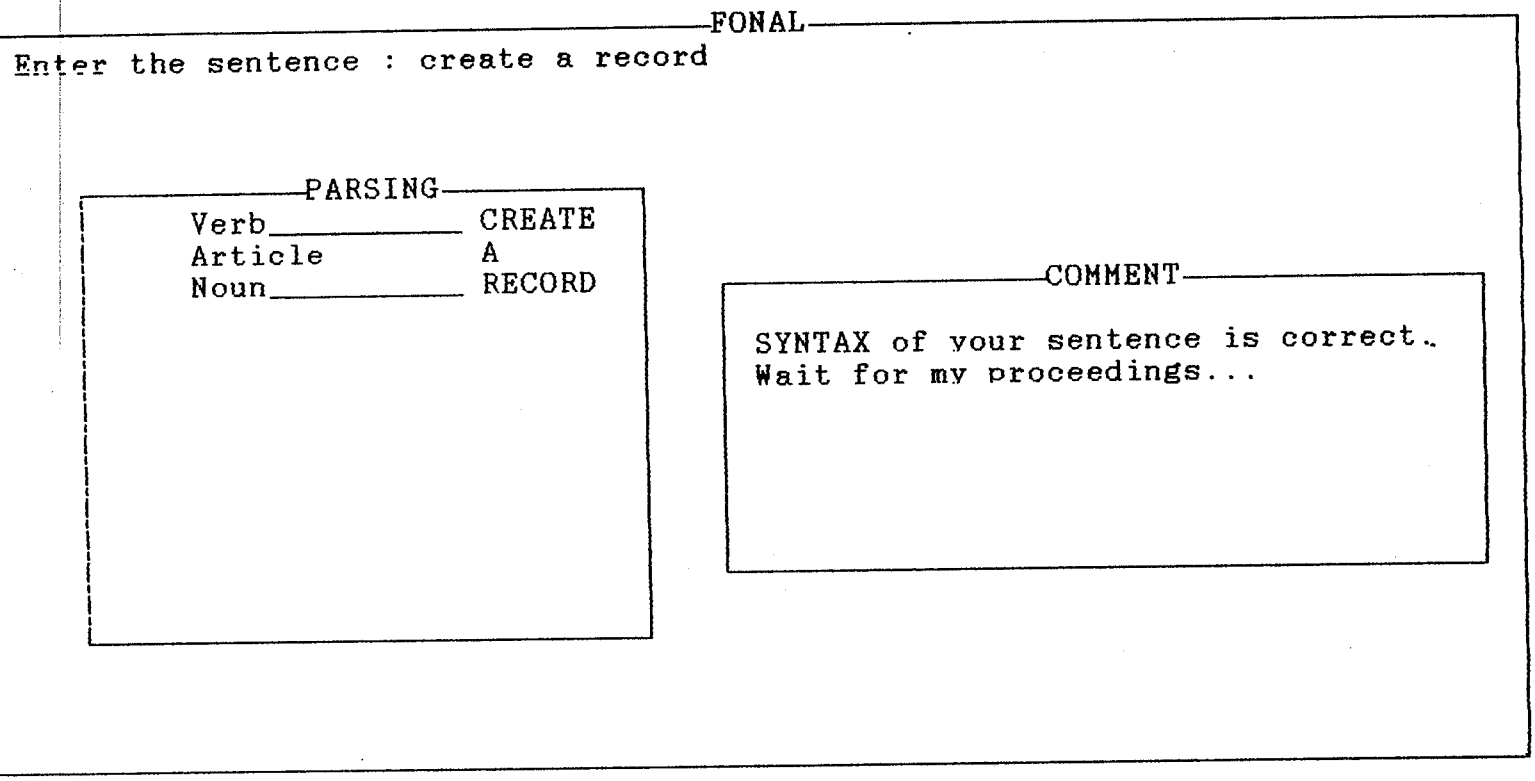
Press any key to continue...

FONAL

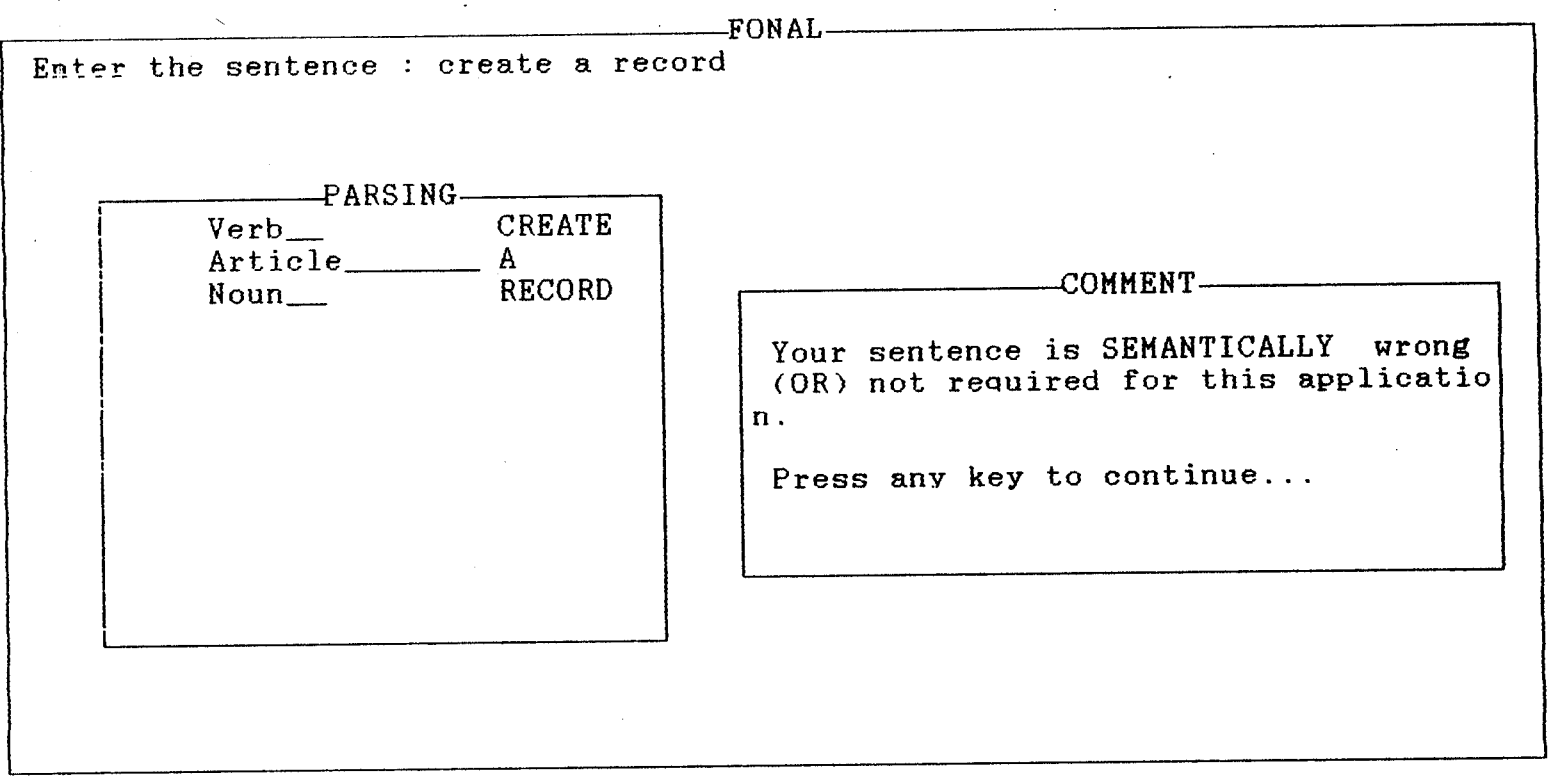
```
Enter the file name to be renamed : b:raj.cht

Enter the new file name  : b:fon.cht

The file  B:RAJ.CHT  has been renamed to B:FON.CHT .

                                        Press any key to continue...
```

FONAL

```
Enter the file name to be renamed : b:dd.d

Enter the new file name  : b:kk.l

The file B:DD.D is not existing...

                                        Press any key to continue...
```



Press any key to continue...



FONAL

Enter the sentence : create a page

PARSING

Verb___ CREATE
Article_____ A

I am not able to parse any more.

COMMENT

SYNTAX of your sentence may be wrong (OR) not in the required format.

Press any key to continue...

CONCLUSIONS

This software accepts three types of sentences that are in simple present tense and active form.It successfully executes the valid file operations .

If the input sentence is syntactically wrong , then it gives out an error message.

FONAL has a general purpose syntactic part.This software can be used for any particular appllication changing only the inference routine .

The following operations can be executed using **FONAL.**

Displaying the directory

Merging two files

Copying a file

Creating a file

Deleting a file

Renaming a file

Swapping two files

FUTURE ENHANCEMENTS

FONAL has a general purpose syntatic part.This software can be used for any Natural Language Application changing only inference part.

FONAL has been expanded to perform the other operations like sorting,searching,modifying the files etc.

CHAPTER 7.

APPENDIX-1

ABOUT PROLOG

Prolog is a object oriented language suited for formal symbolic reasoning.Prolog is different from other higher level languages as it does not use procedures to solve problems.It uses data about the object and their relationships and symbolic processing to prove the goal specifide by the user.

The process of solving problems using prolog can be viewed as an attempt to move through problem space to a specific objective.The problem space consists of nodes and links.Each node is a sub-goal or step to the final solution.Prolog programs can incorporate heuristics to aid the search for the solution of a problem.The main part of a prolog program consists of a collection of knowledge about a specific subject.This collection is expressed in the form of facts and rules.

Some important features of Prolog are:

The CUT(!):-

A special mechanism that can be used in Prolog programming is the "cut".It tells Prolog which previous choices it need not consider again,when it backtracks through the chain of satisfied goals.By using the 'cut',the program works faster as it does not waste time.The program may occupy less of the computers memory.

When a cut is encountered as a goal, the system thereupon becomes committed to all choices made since the parent goal was invoked.All other alternatives are discarded.Hence an attempt to resatisfy any goal between the parent goal and the cut goal will fail.

UNIFICATION:-

Unification is a pattern-matching process.A term is said to unify with another term if

* both terms appear in predicates that have the same number of arguments and appear in the same position in their positions.

* all terms appear as arguments of the same type.

* all subterms unify with each other.

The basic rules for unification are:-

* A variable that is free will unify with any term that satisfies the preceding conditions.After unification, the variable is bound to the value of the term.

* A constant will unify with itself or any free variable.If the constant is unified with a variable,the variable will be bound to the value of the constant.

* A free variable will unify with any other free variables.After unifying the two variables will act as one.If any of the variables become bound,the other will be bound to the same value.

LISTS:-

A list is an ordered sequence of terms.The arrangement of the terms is important.List terms can be variables,simple objects,compound objects or other lists.A list can contain unlimited number of terms.Each list is set off in square brackets,with the components of the lists separated by commas.The components of a list should be of the same domain type.

Here are several points to be remembered about lists:-

* A list is declared by using an asterix in the declaration.The name with the asterix is the same of the list element.

namelist = symbol*

* All objects in alist must be of th same type,but the types can be as complex as wished.

* Predicates can be defined using the list name as an argument.

CHAPTER 8.

REFERENCES

1.Carl Townsend,"Introduction to Turbo Prolog",BDP Publications.

2.Dan Shafer,"Advanced Turbo Prolog",Howard & Sams company.

3.Herbert Schildt,"Advanced Turbo Prolog Version 1.1".

4.Byron Gottfried,"Programming with C",McGraw Hill publications.

5.Wren & Martin,"English Grammar & Composition".

6.Mary Dee Harris,"Introduction to Natural Language Processing".

7.Elaine Rich,"Artificial Intelligence",McGraw Hill Publications.