REPORT R-494 OCTOBER, 1970 UILU-ENG 70-239

COORDINATED SCIENCE LABORATORY

AN ALGORITHM FOR THE SYNTACTIC ANALYSIS IN THE R2 INFORMATION SYSTEM

JEFFREY A. SCHULTZ
WILLIAM T. BIELBY

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

AN ALGORITHM FOR THE SYNTACTIC
ANALYSIS IN THE R2 INFORMATION SYSTEM

BY
Jeffrey A. Schultz and
William T. Bielby

This work was supported by the Office of Education under contract OE C-1-7-071213-4557 and by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under contract DAAB-07-67-C-0199.

AN ALGORITHM FOR THE SYNTACTIC ANALYSIS IN THE R2 INFORMATION SYSTEM

Jeffrey A. Schultz and William T. Bielby

AN ALGORITHM FOR THE SYNTACTIC ANALYSIS IN THE R2 INFORMATION SYSTEM

Abstract

This paper presents a pure syntax parsing algorithm for the R2 question answering system. The parsing is a necessary part in the processing of information for the retrieval of answers by giving the syntactic structure of both questions and statements of fact. The syntactic analysis is performed on individual statements by using grammatical rules and various grammatical transformation. The parser has been implemented in LISP 1.5 on an IBM 360/75 computer. Experimental results from the system include examples of the analysis of some compound and complex sentences, as well as simple sentences.

ACKNOWLEDGEMENT

We wish to express our gratitude to Professor R. T. Chien for his invaluable advice and guidance throughout this research effort. Thanks also to Kenneth Biss and Fred Stahl for their many helpful and valuable criticisms of the manuscript. Special thanks to Mrs. Sherry Kallembach for the typing of this report.

INTRODUCTION

During the last decade there has been significant activity in the area of syntactic analysis [2,4,5]. Of particular interest have been the development of parsers for automatic translation [7,8,9,12,13,14] and question answering systems [1,3,11,18,19,20]. A number of automated parsers have also been described [2,6,10,15,16,17].

Unlike Simmons and Cocke[†], the algorithm described here is capable of handling complex sentences including relative phrases. In addition, the approach taken here does not have as many rules as found in either Kuno or Cocke, and the rules are less complex than those of Robinson and Friedman and thus can be executed in less time.

The grammar used for the syntactic analysis in this system is a modified context-free immediate constituent phrase-structure grammar. A phrase-structure grammar is a grammar G consisting of four parts: a terminal vocabulary V_t (words such as boy, eats, etc.), a non-terminal vocabulary V_n (parts of speech such as noun, preposition, conjunction, etc.; syntactic categories such as noun phrase, verb phrase, etc.; and types of sentences such as compound, complex, and simple), a set of grammatical rules P, and a starting symbol σ to clarify where the input begins.

An immediate constituent grammar is a grammar that builds or decomposes syntactic categories out of adjacent syntactic categories and syntactic classes. A context-free grammar is a phrase-structure grammar where the rules are of the form $\alpha \rightarrow \beta$ (α is an element of the non-terminal vocabulary

---

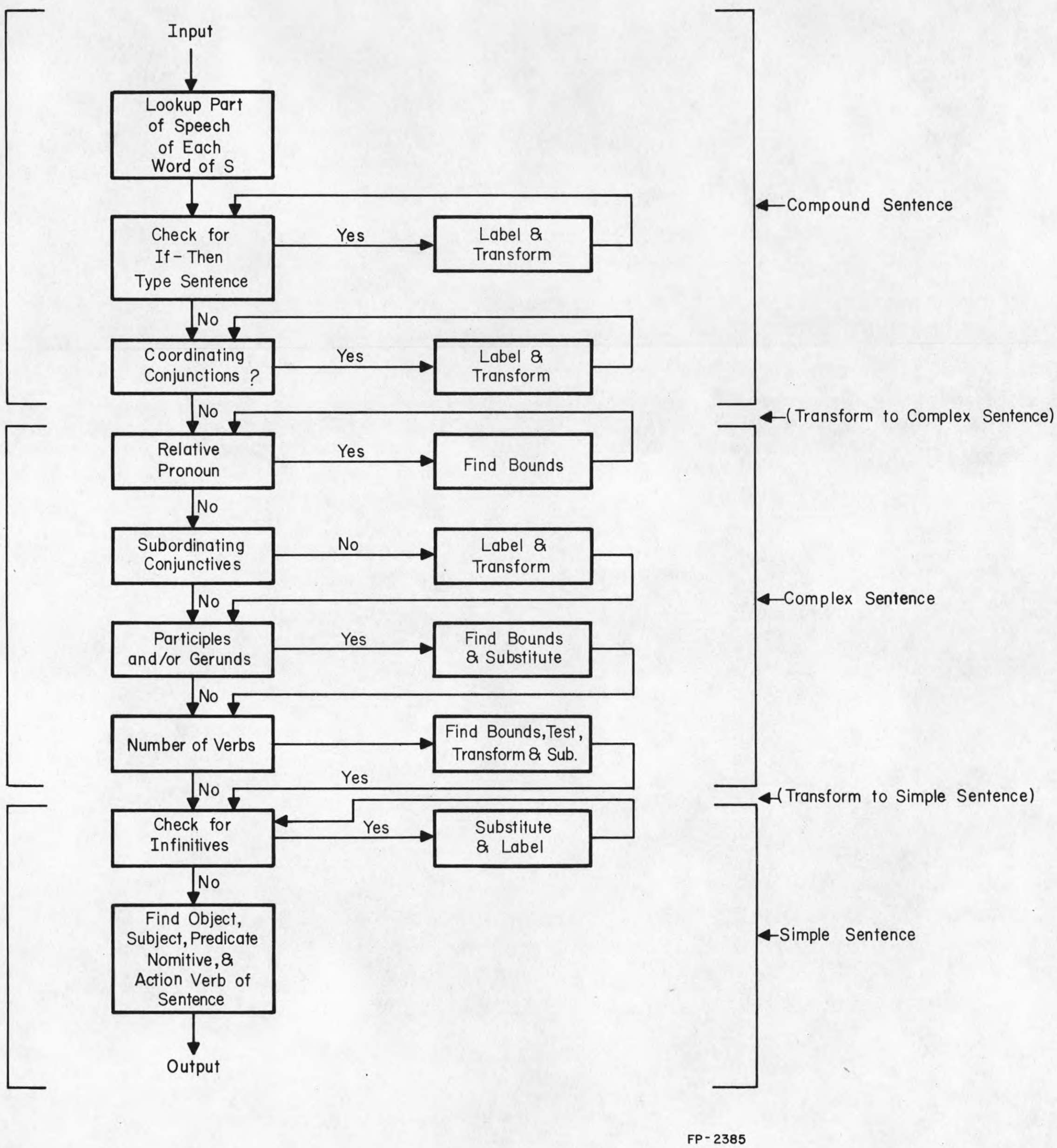
[†] As described in Hayes [7].

and β is any non-empty string), and thus the grammar is independent of the context of the sentence. A grammar is bottom-up when it combines elements of $V_n \cup V_t$ into an element of V_n.

The input to the parser is a sentence. The words are read in one by one and associated with their parts of speech by a dictionary look-up. Grammatical rules and various transformations are then applied to the sentence, and the output is in the following form:

PREDICATE(modifiers)(SUBJECT(modifiers),OBJECT(modifiers))

Thus the subject and the object of the sentence are a function of the predicate. Figure 1 shows the basic parts that the parsing algorithm has.



Input
Lookup Part of Speech of Each Word of S
Check for If- Then Type Sentence
Yes
Label & Transform
No
Coordinating Conjunctions ?
Yes
Label & Transform
No
Relative Pronoun
Yes
Find Bounds
No
Subordinating Conjunctives
No
Label & Transform
No
Participles and/or Gerunds
Yes
Find Bounds & Substitute
No
Number of Verbs
Find Bounds, Test, Transform & Sub.
Yes
No
Check for Infinitives
Yes
Substitute & Label
No
Find Object, Subject, Predicate Nomitive, & Action Verb of Sentence
Output
Compound Sentence
(Transform to Complex Sentence)
Complex Sentence
(Transform to Simple Sentence)
Simple Sentence
FP-2385

Figure 1.

THE ALGORITHM

Associated with the parser is a dictionary containing the words of the sentences to be parsed along with their parts of speech. There is also a set of hierarchial immediate-constituent rules, a set of transformational rules, and a set of phrase-markers defined in another dictionary.

The input sentence must be grammatically correct. The object of the parsing is to transform the input sentence, which is assumed to be compound, to a kernel sentence with its modifiers or to a set of kernel sentences and its modifiers. A compound sentence contains two or more independent sentences. Each statement in a compound sentence is a main clause and is coordinate with the other statements. Clauses in a compound sentence are joined in one of three ways: (1) by coordinating conjunctions - and, but, or, for, either...or, neither...nor, (2) by semicolons, or (3) by conjunctive adverbs - accordingly, however, also, consequently, therefore, etc. Kernel sentences are the sets of elementary sentences and combiners, such that all sentences of the English language are obtained from one or more kernel sentences by means of one or more transformations. Each kernel sentence is a construction of classes. The kernel constructions of English are the following:

Noun Phrase Verb ....

Noun Phrase-Verb Phrase-Noun Phrase

Noun Phrase-Verb (to be)-Noun Phrase

Noun Phrase-Verb (to be)-Adjective Phrase

Noun Phrase-Verb (to be)-Adverb.

The intermediate step between a compound and a simple sentence is a complex sentence. Thus, if a sentence is not a compound sentence, we assume it to be a complex sentence. A complex sentence consists of one main clause and one or more subordinating clauses. The main clause expresses the principle statement, the subordinate clause the secondary statement.

Complex sentences offer more variety than simple sentences and are generally more exact than compound sentences because the subordinating conjunctions are more numerous and more precise in meaning than coordinating conjunctions. Most parsing programs can only handle simple sentences; however, in the design of this algorithm compound and complex sentences are considered for the simple reason that it is desirable to accept the full range of sentences.

Compound Sentences

Assuming the sentence is a compound sentence; a check to see if it is an if-then type of sentence. This is a simple test to see if one of the four conditions exist:

if....,then....; or ....., if.....; or if....., ...; or ...., then ......

If one of these conditions occur, the program is sent into a transformation routine. The transformations are the following:

T1. if....., then..... = if-then(S_1,S_2)

T2. ...., if..... = if-then(S_2,S_1)

T3. if..........,...... = if-then(S_1,S_2)

T4. ....,then.... = if-then(S_1,S_2)

If none of the four structures occur, then the program continues and checks for coordinating conjunctions.

In this part of the program each coordinating conjunction is looked at to see if it is the connective of two independent sentences. This is done for each coordinating conjunction until it either works or we are finished in checking all the coordinating conjunctions. The procedure is the following:

1. Look at the first coordinating conjunction and see if there is a comma (phrase-marker) immediately following the coordinating conjunction. Do this for all coordinating conjunctions. If this does not produce a transformation into two independent sentences, check the environment of each coordinating conjunction. This is done by first looking at the left side and comparing its structure to the right side, since it is a coordinating conjunction and the structure at the left side must equal the structure of the right side. If the structure is a sentence, then the necessary transformations are made. If not, the structure is stored for later reference. (In the checking of the environment of the coordinating conjunction, first go to the left most boundary which is either another conjunction or a series of blanks which means the beginning of the sentence. The same is done on the right-hand side, if it is going to be transformed.) Continue this checking until the number of coordinating conjunctions has been exhausted. Some examples of if-then sentences and sentences that have coordinating conjunctions follow:

1. If a car fails to stop at a red light, it is breaking a law. This sentence would be handled the following way:

if-then(a car fails to stop at a red light, it is breaking a law)

2. Cars must obey all traffic laws, and pedestrians must do the same. This sentence would be handled the following way: A search is made for coordinating conjunctions. The word 'and' is found preceded by a comma. Left environment is a sentence, thus we have the following: and(Cars must obey all traffic laws, pedestrians must do the same). One can see the form of the transformation of the compound sentence to a coordinating conjunction and its two variables 'complex sentence one' and 'complex sentence two'.

3. Cars and buses are vehicles.

This sentence would be handled the following way: Again a search is made for the coordinating conjunction. The word 'and' is found. Check the left environment of the conjunction and find it is a Noun phrase; again, the same must be true for the right environment. Thus, both the left and the right environment are functions of the conjunction 'and', and the sentence is now considered a complex sentence with a compound subject.

Complex Sentences

If the input sentence S cannot be transformed into compound sentences S_1 and S_2, then a transformation of S into the complex sentence S' is performed and a check for subordinating conjunctions continues.

If there is a subordinating conjunction in the complex sentence, the following transformations are made:

$$\ldots\ldots\ldots\ldots, \ SC\ldots\ldots\ldots \rightarrow SC(S_1', S_2') \qquad (1)$$

$$SC \ \ldots\ldots\ldots\ldots, \ldots\ldots\ldots\ldots \rightarrow SC(S_2', S_1') \qquad (2)$$

In case (1) S_1' is the main clause and in case (2) S_2' is the main clause of the sentence.

The transformation is made by finding the subordinating conjunctions and looking to the right of the conjunction until a series of blanks is found, if the SC is not the first word of the sentence. The SC's right environment is labeled S_2 and the words to the left of the SC are labeled S_1 and the necessary transformation is performed. If the subordinating conjunction is at the beginning of the sentence, its environment is found by a complex sentence test and an immediate constituent analysis. When its environment is found, the necessary transformation is performed. After this a test for relative pronouns (adverbial and adjectivial phrases) occurs. The list of subordinating conjunctions are: although, though, that, because, since, so that, in order that, as, unless, before, where, when.

Two examples of sentences that contain a subordinating conjunction are the following:

1. We study the "Rules of the Road" booklet, because we want to pass our driver's test.

2. Unless we take the driving test, we will never get a driver's license.

The sentences are handled the following way. In sentence 1, the word 'because' is found and the sentence is transformed into a main clause - 'We study the "Rules of the Road" booklet, and a subclause - 'we want to pass the driver's test' as a function of the subordinating conjunction.

because(We study the Rules of the Road, we want to pass the driver's test)

In sentence 2, the word 'unless' is found and the sentence is transformed into a main clause - 'we will never get a driver's license' and a subclause - 'we take the driving test' as a function of the subordinating conjunction 'unless'.

unless(we will never get a driver's license, we take the driving test.)

Relative Pronouns

The next check is for relative pronouns. The pronouns in the relative pronoun dictionary are used as an adverb, adjective, or a noun, if the phrase-marker '?' does not appear in the input string.

A check of the input string to see if there is a relative pronoun is performed. If one exists its environment is found and the necessary transformations are made. The environment of a relative pronoun is almost always a simple sentence. The environment is found by looking to the right of the relate pronoun, and by immediate constituent analysis finding the simple sentence. After it is found the structure is labeled RP and it is substituted into the original sentence. Now look at what type of syntactic category RP can be. It can either be used as a noun or as a modifier. This can be handled by a simple test to see if an NP is needed to make the complex sentence have syntactic sense. It can either be subject, object, or object of a preposition if it is an NP. If there already is an object of a sentence, or a subject of a sentence, or no hanging prepositions, then the relative pronoun and its environment will be used as a modifier. It will modify either the subject or the object (or predicate nominative) or the predicate of the sentence. It will usually modify an NP (object or subject), and the one it modifies can be determined by which syntactic category it is nearest to. When it is decided what the relative pronoun and its environment's use is RP is given its equivalent value, either NP, Adj, or Adv (rare). The list of relative pronouns are: who, which, what, that, whoever, whatever, whichever, whom, whomever.

Three examples of sentences that contain relative pronouns are the following:

1. The driver, who was found drunk, will be prosecuted.

In this sentence, the relative pronoun 'who' is found. The environment of 'who' is also found easily, since it is surrounded by commas. The relative pronoun and the environment is a simple sentence with 'who' as the subject, 'was found' as the predicate, and 'drunk' as the adjective phrase. The relative phrase is used as an adjective here since it modifies the subject of the main sentence 'driver'.

2. Whoever drives the car must fill the car with gas.

In this sentence, the relative pronoun 'whoever' is found. The environment of 'whoever' is 'drives the car'. The relative phrase is a simple sentence with 'whoever' as the subject, 'drives' as the predicate, and 'the car' as the direct object. The relative phrase in this sentence is used as a noun phrase which is the subject of the sentence.

3. Who lost my hat?

In this sentence the relative pronoun 'who' is found, but the phrase marker '?' is also found. Therefore, a relative phrase does not exist.

Participles and Gerunds

The next check is for participles and gerunds. V_{ing} (a verb with an ing ending) and V_{ed} (a verb with an ed ending) can either be a participle, a verb form used as an adjective; or it can be a gerund, a verb form used as a noun; or it can be the main verb of the sentence, if it is used with another verb.

The first test is to see if it is used with another verb, in which case it must directly follow another verb (V_{ing} can only follow the verb to be). This is done first because it is the most simple form of the V_{ed} and the V_{ing} forms. If it does not follow a verb then it is assumed that the form is a gerund. This implies that it must be the object of the sentence, subject of the sentence, or object of a dangling preposition. If it fits into one of these categories there is no further analysis necessary in this part. The V_{ing} of V_{ed} is labeled G and substituted into the complex sentence. If it is neither a verb nor a gerund it is called a participle, and since participles are usually used as adjectives it will modify a noun, which will follow immediately after or two words after the participle. Participles and gerunds also have environments. These are words (or a word) immediately following the participle or gerund. The word (word and modifiers) is the object, predicate nominative, or predicate adjective of the V_{ing} or V_{ed} form. (This happens because a participle of a gerund still has properties of being a verb). Therefore, the participle and gerund will have a syntactic analysis also. Three examples of sentences that contain either a participle, gerund, or a verb of the specified type are the following:

1. Driving a bus is very hard to do.

'Driving' is a V_{ing} verb and is used as a noun; therefore, it is a gerund. 'Driving a bus' is the gerund phrase with 'a bus' being the object of the word 'driving.'

2. The speeding car received a ticket.

'speeding' is a V_{ing} verb and is used as an adjective; therefore, it is a participle.

3. He has played very hard.

'played' is a V_{ed} verb, and with the auxiliary verb 'has', is used as the predicate of this sentence.

Multiple Verbs

The next check is to see if there is more than one verb. This is done, because if there is more than one verb in the sentence (V+V construction means one verb) implies that S' is still a complex sentence. Verbs taken care of in coordinating conjunctions, if-then sentences, subordinating conjunctions, and relative pronoun phrases are not counted. Having more than one verb in the sentence usually occurs when a relative pronoun is left out. If this occurs, the appropriate relative pronoun is placed in front of the first verb, (second, third,..... if necessary) to see if it makes syntactic sense and thus eliminate the problem of extra verbs. An example of more than one verb in the sentence is the following:

1. I hope he comes.

In this sentence the relative pronoun 'that' is deleted. The two verbs are 'hope' and 'comes'. The sentence is transformed to the sentence - I hope that he comes.

Simple Sentences

What now remains are simple type sentences with a variety of possible substitutions if the sentences were processed as either compound or complex types.

Infinitives

There may be an infinitive in the simple sentence. The form of an infinitive is the following: to+verb. An infinitive may be used as a noun, adjective, or an adverb.

The testing procedure is to first check to see if it is a noun, the same way a relative pronoun phrase is checked to see if it is a noun. If it is not a noun and it follows directly after a noun or an adjective, it then modifies the noun or adjective it follows and Adj can be substituted for the infinitive. If it is not a noun and it follows directly after a verb, then it modifies the verb and is labeled an adverb. Three examples of sentences that contain infinitives are the following:

1. To be asked to the party makes any girl proud.

In this sentence 'to be asked' is the infinitive with 'to be asked to the party' as the infinitive clause. The infinitive is used as a subject, thus it is a noun phrase.

2. I have plenty of work to do.

In this sentence the words 'to do' are used as an infinitive. Since 'to do' modifies work, it is used as an adjective.

3. The students came to learn.

In this sentence the words 'to learn' are used as an infinitive. Since 'to learn' shows purpose, cause, etc., it is an adverb modifier.

Immediate Constituent Analysis

The simple sentence is easily handled by immediate constituent analysis and simple transformations. The immediate constituent rewrite rules are in hierarchial levels. Table 1 lists the symbols used in the rules.

Symbol	Meaning
N	noun
V	verb
Pro	pronoun
adv	adverb
det	determiner
adj	adjective
prep	preposition
NP	noun phrase
VP	verb phrase
PP	prepositional phrase
RP	relative phrase
AP	adjectivial phrase
S	sentence
∅	no construction possible
aux	auxiliary verb

The following seven stages of rules perform the immediate constituent analysis:

Rule	Meaning
<u>Stage 0</u>:	
Pro-------N	Pronouns such as he, they, me, etc. are rewritten as nouns.
Adverb shift	Any adverb in the sentence is shifted to the immediate right of the verb it modifies*
<u>Stage 1</u>:	
Det+N-----NP	N(Det)
Adj+N-----NP	N(Adj)
Prep+N----PP	Prep(N)
N_1+N_2-----NP	N_2(N_1)
V+adv-----VP	V(Adv)
N+∅-------N	N( )
V+∅-------V	V( )
aux+aux---aux	aux aux
etc.	

*will be discussed on page 22.

<u>Stage 2</u>:

NP+∅------NP'	N( )
NP+N------NP'	N(NP)
VP+∅------VP	VP( )
V+V-------VP	V+V ( )
V+∅-------VP	V( )
∅+N-------NP'	N( )
Adj+∅-----Adj	Adj( )
Det+NP----NP'	NP(Det)
aux+V-----VP	aux V

<u>Stage 3</u>:

$\left\{\begin{matrix}V_{ing}\\V_{ed}\end{matrix}\right\}$+NP'--NP'	NP'$\left(\left\{\begin{matrix}V_{ed}\\V_{ing}\end{matrix}\right\}\right)$
$\left\{\begin{matrix}V_{ed}\\V_{ing}\end{matrix}\right\}$+∅----NP'	$\left\{\begin{matrix}V_{ed}\\V_{ing}\end{matrix}\right\}(\quad)$

<u>Stage 4</u>:

NP'+RP----NP"	NP'(RP)
NP'+Adj---NP"	NP'(Adj)
Prep+NP'--PP	Prep(NP')
∅+RP------NP"	RP( )
RP+∅------NP"	RP( )

Adjective shift,adjective shifted immediately in front of subject**

VP+RP-----VP'	VP(RP)
VP+∅------VP'	VP( )
PP+∅------PP	PP( )

<u>Stage 5</u>:

NP"+PP----NP"	NP"(PP)
VP'+PP----VP'	VP'(PP)
PP+NP"----NP"	NP"(PP)

<u>Stage 6</u>:

NP"+VP"+∅-S	VP'(NP",∅)
NP_1"+VP'+NP_2"----S	VP'(NP_1",NP_2")
∅+VP'+NP"--------S	VP'(∅,NP")

** Will be discussed on page 21.

Several examples will now be given of how the algorithm parses sentences.

Example 1:

If a driver breaks too many driving laws, he may lose his driving license.

This sentence would be parsed:

S_1=a driver breaks too many driving laws,

S_2=he may lose his driving license.

if-then (S_1, S_2)

S_1 = breaks(too many) (driver(a),laws(driving))

S_2 = may lose (he, license(driving))

if-then(breaks(too many)(driver(a),laws(driving)),may lose(he, license(driving)))

After the words are given a parts of speech, the input would be like the following:

if Det N V Adv Adv V_{ing} N, Pro aux V adj V_{ing} N.

The sentence is found to be of the if-then type, and transformed into two complex sentences. The ing verbs are then labeled as participles (RP). The two subsentences are transformed into simple sentences. A pronoun to noun substitution is made and finally the immediate constituent rules are processed.

Example 2:

When entering a street or a highway from a driveway or a private road, drivers must yield to all other traffic, and they must not enter the roadway until it is safe to do so.

This sentence is an example of a compound-complex sentence. (A compound-complex sentence contains two or more main clauses and one or more subordinate clauses). Here 'when' and 'until' are subordinating conjunctions, 'or' and

'and' are coordinating conjunctions ('or' in both cases coordinates two noun phrases, thus it is treated later; 'and' coordinates the two main clauses of the subordinating conjunction 'when').

S_1' = entering a street or a highway from a driveway or a private road

S_2' = drivers must yield to all other traffic, and they must not enter the roadway until it is safe to do so.

S_2' is decomposed into S_1" and S_2". The final parsing is the following.

When(and(must yield(to (all))(other(traffic))(drivers,),(until(must not enter (they,roadway(the)), is (so) (it(safe)(to do),),(entering(from)(or(driveway) (road)((private) a) (,(street)(a)))))

After the words are given their parts of speech, the input would look like:

S.Conj V_{ing} Det N CC Det N Prep Det N CC Det adj N, N Aux V Prep Pro Prep N, CC Pro Aux Aux V Det N S, Conj. Pro V(to be) Adj Inf Adv.

One can see that this is an extremely difficult sentence to parse. Rule ordering plays an important role in the parsing of this sentence. If one looks at the coordinating conjunction 'and', a search to the right of 'and' finds a sentence construction as the right-hand environment. Thus, the two sentences that 'and' coordinates are the main sentences of the subordinate conjunction 'when'. This is the major problem of this sentence's parsing. By the parsing rules this is handled correctly. A second major problem of this sentence is the other subordinating conjunction 'until'. This divides the second main clause into a sub-main-clause and a dependent clause. This is also handled by the parsing rules.

Example 3:

If a policeman is directing traffic, the right-of-way laws do not apply and drivers must do as the officer tells them.

This sentence is another example of a compound-complex sentence, but it is much easier than the last one. This example will be parsed as the computer would parse it. After the words are given a part of speech, the input would look like the following:

if Det N V(to be) V_{ing} N, Det Adj N Aux Aux V CC N Aux V S Conj Det N V Pro

1. The first test is the if-then test. The word 'if' is found without then. The if-then sentence is transformed to the following:

if-then(Det N V(to be) V_{ing} N, Det Adj N Aux Aux V CC N Aux V SC Det N V Pro)

2. The next test is for coordinating conjunctions. A CC is found and the left-hand environment of the CC is:

Det Adj N Aux Aux V

This is parsed by the immediate constituent analysis rules and stored for later use.

Adj+N---NP and Aux+Aux---Aux at level 1, det+Np---NP' and Aux+V---VP at level 2. NP+ϕ---NP" and VP+ϕ---VP' at level 4, and NP"ϕVP'----S at level 5. The parsing would look like:

aux aux V (N(adj)(Det),)

3. Next, the SC is found and the right environment is parsed. This turns out to be the left environment of the coordinating conjunction 'and'.

4. Aux+V---VP at level 1 and N+ϕ----NP" at levels 2 and 4 and VP+ϕ---VP' at level 4 and NP"ϕVP'----S at level 5. The parsing would look like: Aux V (N,). This information is stored and the parsing of the subsentences begins.

5. if-then(is directing(policeman(a), Traffic)), and(do not apply(laws(right-of-way),)),(as(must do (drivers,)(tells (officer(the), them)))

This is equivalent to the part-of speech parsing that is tacked on to each individual word:

if-then(V V_{ing} (N(Det),N)), CC(aux aux V(N(Adj),)),(SC(Aux Aux (N,)(V (N(Det), Pro)

PICKING THE CORRECT PART OF SPEECH FOR THE WORDS IN THE DICTIONARY

Different parts of speech of individual words can be eliminated from the possible parsings by separating the rules that are impossible to have. For example the rule N+N---NP is legitimate, while the rule Adj+Prep----? is impossible. If a word can have two or more parts of speech the impossible rules are needed. This is done in the following way:

Example: The driver did see the speed trap.

This input can have any of the three following parts of speech associated with it:

Det N V V N N (1)

Det N V V V N (2)

Det N V V N V (3)

(1) can be analyzed the following way:

D+N---NP	∅+VP---VP'	NP"+VP'+NP'---S
V+V---VP	NP+∅----NP'	
N+N---NP	NP'+∅---NP"	

(2) cannot be analyzed because:

D+N---NP	NP+∅---NP'
V+V---VP	V+∅----VP
V+N----?	N+∅----NP

It is impossible to combine at levels past 3.

(3) cannot be analyzed because:

D+N---NP	N+∅---NP
V+V---VP	∅+V---VP
N+V----?	

Same as (2).

So, the only possible parsing is (1).

UNIQUE FEATURES OF THE PARSER

Several unique features of the parser are the adjective shift, adverb shift, and the verb-participle-gerund test.

Adjective Shift The car is a mossy green and beatleish.

$$DN, V(to\ be)\ Adj_1\ Adj_2\ C\ Adj_3$$

The first check is for conjunctions and 'and' is found. A test to the right environment of 'and' finds that the part-of-speech adjacent to it is an adjective, which implies the right side has the same part of speech and we would combine them as follows:

$DNV(to\ be)Adj_1C(Adj_2,Adj_3)$.

The next combination would be D+N=NP=N(D).

$NP\ V(to\ be)\ Adj_1\ C(Adj_2,Adj_3)$

The next combination would be $Adj_1 + Adj_2 = Adj_2(Adj_1) = AP$

$NP\ V(to\ be)\ Conj(AP,Adj_3)$

The next combination would be Conj(AP,Adj) = AP'

NP V(to be) AP'

The next combination would be the movement of AP' to left adjacent of NP and we get:

AP' NP V(to be)

The next combination is AP' + NP = NP' - NP(AP')

NP' + V(to be)

V(to be) is transformed into VP and we get

NP' + VP

The final combination is NP' + VP = VP(NP,ϕ) = S

S = is ((car(the))(and(green(mossy)),beatelish),ϕ)

Adverb Shift

He played the game well.

Pro V D N Adv

The adverb would be shifted right adjacent to the verb first, and the pronoun would be transformed to a noun. Thus:

N_1 V Adv D N_2

Next combine D + N_2 = NP = N(D)

N_1 V Adv NP

Combine V + Adv = VP and transform N_1 into NP_1.

Thus we have

NP_1 VP NP_2

Then combining this into VP(N_1,N_2) = S

S = played (well)(he,game(the))

Verb-Participle-Gerund Test.

Stopping at stop signs is necessary.

Stopping is an 'ing-verb' and it is not known if it is used as a verb (with auxiliary helper), a gerund or participle.

V_{ing} Prep Adj_1 N V(to be) Adj_2

Adj_1 + N = NP = N(Adj_1)

V_{ing} Prep NP V(to be) Adj_2

Prep + NP = PP

V_{ing} PP V(to be) Adj

Since there is no subject (NP_1) implies V_{ing} is this NP_1 and must be a gerund. Therefore,

NP PP V(to be) Adj

NP + PP = NP' = NP(PP)

NP' V(to be) Adj

Next the adjective shift and

NP'(Adj) V(to be)

Next, make the transformation to get

V(to be)(NP',ϕ) = S.

Thus S = is (Stopping(necessary)(at,(signs(stop))),ϕ)

PROGRAM FOR IMMEDIATE CONSTITUENT ANALYSIS OF SIMPLE SENTENCES

This LISP program is internal to the larger parsing program. The function FTEST applies the seven stages of the immediate constituent rules to a simple sentence.

Abbreviations

The abbreviations for parts of speech vary slightly from the list on page 14 . They are as in Table 2.

TABLE 2.

N	noun
PRO	pronoun
V	verb
VGD	verb ending with "ing" or "ed"
PREP	preposition
D	determiner
AJ	adjective
ADV	adverb
NP	noun phrase
PP	prepositional phrase
VP	verb phrase
RP	relative phrase

Input

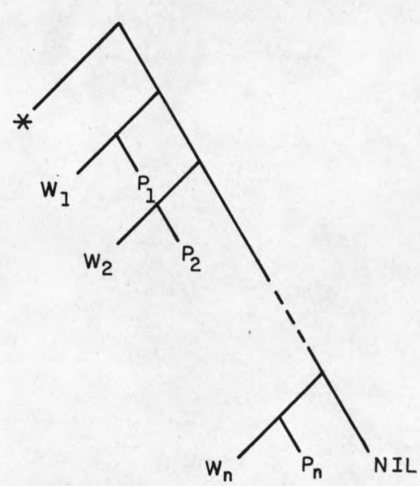
The application of FTEST to a simple sentence is best illustrated by an example. The LISP statement to apply FTEST to the simple sentence "The young boy goes to school." is as follows:

```
FTEST((* *THE•D) (YOUNG•ADJ) (BOY•N) (GOES•V)
     (TO•PREP) (SCHOOL•N)))
```

Thus the argument of FTEST for a simple sentence of n words $w_1 \ldots w_n$ with n parts of speech $p_1 \ldots p_n$ is the following LIST:

$$(* \ (w_1 \cdot p_1) \ (w_2 \cdot p_2) \ \ldots \ (w_n \cdot p_n))$$

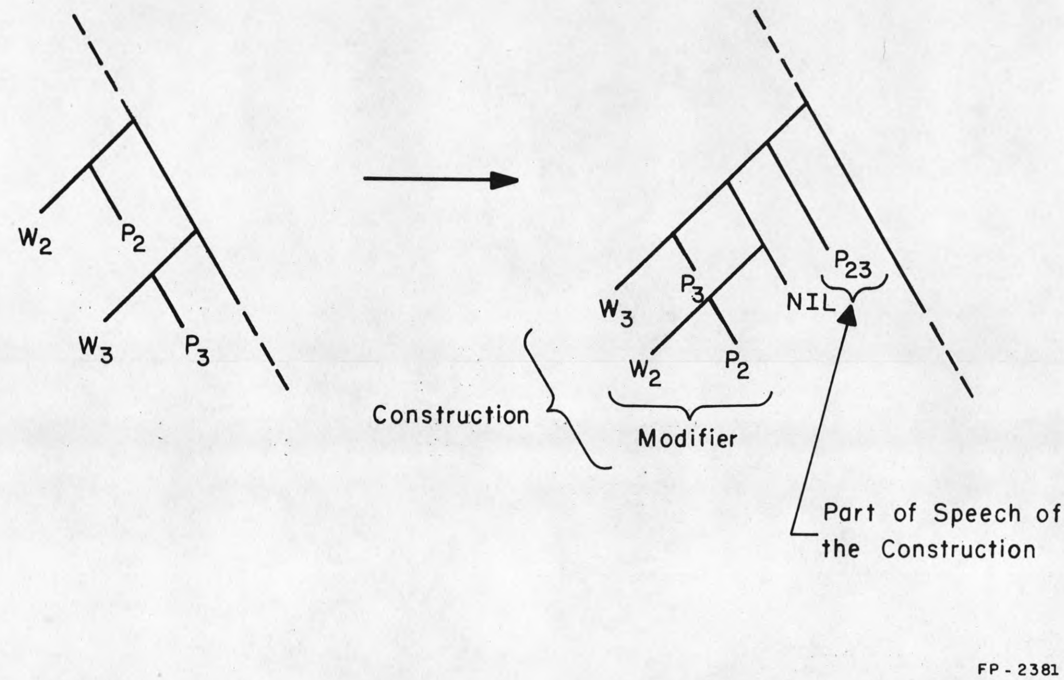
The tree structure of this list is:



FP-2383

Application of the rules to the simple sentence

The LIST is altered as the seven stages are successively applied. When the parts of speech of two adjacent list members satisfy a rule within a stage, the two members are combined into one. If w_2 modifies w_3 the construction is as follows:

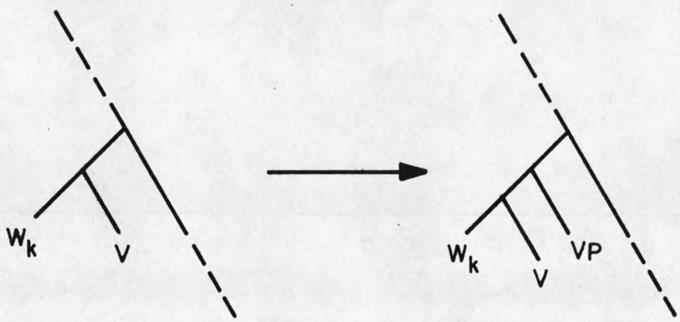


where p_{23} is the part of speech of the combination of w_2 and w_3. Had w_3 modified w_2, their positions would be switched in the transformed LIST member. At any stage, the CAR of any member of the LIST is the part of speech of the construction which is the CDR of that member. Thus at successive stages the parts of speech are compared and further constructions are made.

There are minor differences between the rules applied in the program and those listed in page 14. One is that primed parts of speech (e.g., NP′′) were not needed in the program. Another is that when a rule such as N+∅...N is satisfied where no LIST members are combined, and the part of speech is not changed, no action is taken. Thus these rules are not in the program.

There are several rules where the part of speech of a member of the LIST is changed without forming a new construction with an adjacent member.

In this case, there is no modifier. If the stage 2 rule $V+\phi \ldots VP$ is satisfied by word w_k (a verb), the following construction takes place:



FP-2384

Output

After Stage 5 has been applied by FTEST the LIST that started as a simple sentence now has (not including *) two members, a noun phrase and a verb phrase in either order, or three members, a noun phrase, a verb phrase, and a noun phrase in that order. In Stage 6, FTEST constructs the output of the analyzed simple sentence.

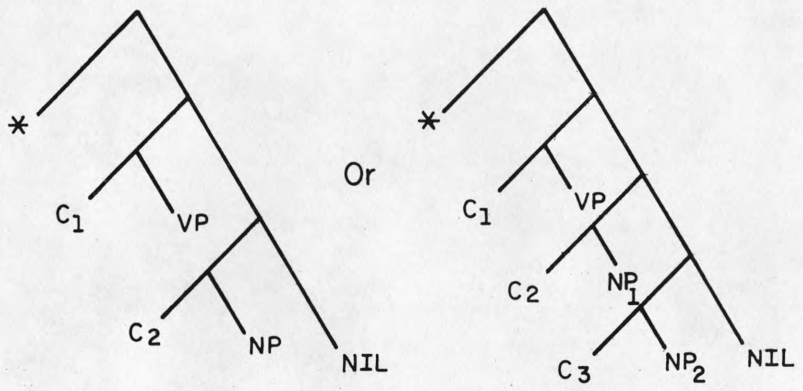
The Stage 6 rules are:

$$NP + VP + \phi \text{ ---- } S \quad VP(NP,\phi)$$

$$NP_1 + VP + NP_2 \text{ --- } S \quad VP(NP_1,NP_2)$$

$$\phi + VP + NP \text{ --- } S \quad VP(\phi,NP)$$

The final LIST returned by FTEST has the following structure:



FP-2382

where c_1, c_2, and c_3 are constructions like those noted above. In LISP notation:

$$(*\ (c_1 \cdot VP)\ (c_2 \cdot NP))$$

or

$$(*\ (c_1 \cdot VP)\ (c_2 \cdot NP_1)\ (c_3 \cdot NP_2))$$

The Program

```
//JOBLIB DD DSN=USER.P1923.LISP151,VOL=SER=UIUSR4,                  *
// UNIT=2314,DISP=OLD
// EXEC PGM=LISP151,TIME=4,REGION=348K
//LISPOUT DD SYSOUT=A
//LISPIN DD *
   CSET (N N)
   CSET (PRO PRO)
   CSET (V V)
   CSET (VGD VGD)
   CSET (PREP PREP)
   CSET (D D)
   CSET (AJ AJ)
   CSET (ADV ADV)
   CSET (NP NP)
   CSET (PP PP)
   CSET (VP VP)
   CSET (RP RP)
   CSET (PHI PHI)


  DEFINE((   (CADDDR (LAMBDA (A) (CAR (CDR (CDDR A  )))))))

  DEFINE((
        (STAGE01 (LAMBDA (X)
                    (COND ((EQ (CDR X) PRO) (CONS (CAR X) N))
                          (T X))))

        (GTEST01 (LAMBDA (X Y)
                    (COND ((NULL X) Y)
                          (T (APPEND Y (GTEST01 (CDR X)
                                      (LIST (STAGE01 (CAR X)))))))))

        (FTEST01 (LAMBDA (A)
                    (GTEST01 (CDR A) (LIST STAR))))


   (STAGE1 (LAMBDA (X Y)
     (COND ((EQ (CDR Y) N) (COND ((EQ (CDR X) D)
                                      (LIST (CONS (LIST Y X) NP)))
                                 ((EQ (CDR X) AJ)
                                      (LIST (CONS (LIST Y X) NP)))
                                 ((EQ (CDR X) PREP)
                                      (LIST (CONS (LIST X Y) PP)))
                                 ((EQ (CDR X) N)
                                      (LIST (CONS (LIST Y X) NP)))))
           ((EQ (CDR Y) ADV) (COND ((EQ (CDR X) V)
                                      (LIST (CONS (LIST X Y) VP))))))))

   (COMB1 (LAMBDA (X Y)
      (COND ((EQ Y N) (COND ((EQ X D) T)
                            ((EQ X AJ) T)
                            ((EQ X PREP) T)
                            ((EQ X N) T)
                            (T F)))





            ((EQ Y ADV) (COND ((EQ X V) T)
                              (T F)))
            (T F))))

   (GTEST1 (LAMBDA (X Y)
      (COND ((NULL X) Y)
            ((NULL (CDR X)) (APPEND Y (LIST (CAR X))))
            ((COMB1 (CDAR X) (CDADR X))
               (APPEND Y (GTEST1 (CDDR X)
                                    (STAGE1 (CAR X) (CADR X)))))
            (T (APPEND Y (GTEST1 (CDR X) (LIST (CAR X))))))))

   (FTEST1 (LAMBDA (A)
      (GTEST1 (CDR A) (LIST STAR))))
```

```
DEFINE ((
  (ADVT (LAMBDA (X)
     (COND ((NULL X) F)
           ((EQ (CDAR X) ADV) T)
           (T (ADVT (CDR X))))))

  (VFIRST (LAMBDA (X)
     (COND ((EQ (CDAR X) ADV) F)
           ((EQ (CDAR X) V) T)
           (T (VFIRST (CDR X))))))

  (SRCHA (LAMBDA (X)
     (COND ((EQ (CDAR X) ADV) (CONS (CAR X) NIL ))
           (T (CONS (CAR X) (SRCHA (CDR X)))))))

  (SRCHV (LAMBDA (X)
     (COND ((EQ (CDAR X) V) (COND ((EQ (CDADR X) V)
                                   (CONS (CAR X)
                                      (CONS (CADR X) NIL)))
                                 (T (CONS (CAR X) NIL))))
           (T (CONS  (CAR X) (SRCHV (CDR X)))))))
 ))

DEFINE ((
  (SRCHVA (LAMBDA (X)
     (COND ((EQ (CDAR X) V)
                 (COND ((EQ (CDADR X) V)
                        (SRCHA (CDR X)))
                     (T (SRCHA X))))
           (T (SRCHVA (CDR X))))))

  (SRCHAV (LAMBDA (X)
     (COND ((EQ (CDAR X) ADV) (SRCHV X))
           (T (SRCHAV (CDR X))))))
))




DEFINE ((
  (SRCHV1 (LAMBDA (X)
     (COND ((EQ (CDADR X) V)
                 (COND ((EQ (CDADR (CDR X)) V)
                        (CONS (CAR X)
                          (CONS (CADR X) NIL)))
                     (T (CONS (CAR X) NIL))))
           (T (CONS (CAR X) (SRCHV1 (CDR X)))))))

  (SRCHA1 (LAMBDA (X)
     (COND ((EQ (CDADR X) ADV)
                 (CONS (CAR X) NIL))
           (T (CONS (CAR X)
                (SRCHA1 (CDR X)))))))
))

DEFINE ((
  (SRCHA2 (LAMBDA (X)
     (COND ((EQ (CDAR X) ADV) (CDR X))
           (T (SRCHA2 (CDR X))))))

  (SRCHV2 (LAMBDA (X)
     (COND ((EQ (CDAR X) V)
                 (COND ((EQ (CDADR X) V)
                        (CDDR X))
                     (T (CDR X))))
           (T (SRCHV2 (CDR X)))))))
))
```

```
  DEFINE((
    (LAST1 (LAMBDA (X)
        (COND ((NULL (CDR X)) X) (T (LAST1 (CDR X))))))
    ))
DEFINE ((
  (OMITA (LAMBDA (X)
     (COND ((EQ (CDAR X) ADV) NIL)
           (T (CONS (CAR X) (OMITA (CDR X)))))))

  (TRANVA (LAMBDA (X)
     (COND ((EQ (CDDR X) NIL) X)
           (T (APPEND (LIST (CAR X)) (APPEND (LAST1 X)
                   (OMITA (CDR X))))))))

  (TRANAV (LAMBDA (X)
  (APPEND (CDR X) (LIST (CAR X)))))
))

DEFINE((
 (FTEST02 (LAMBDA (A)
  (COND ((ADVT (CDR A))
         (COND ((VFIRST (CDR A))
                   (APPEND (SRCHV1 A)
                    (APPEND
                     (TRANVA
                      (SRCHVA (CDR A)))
                      (SRCHA2 A))))
               (T (APPEND (SRCHA1 A)



                     (APPEND
                      (TRANAV
                       (SRCHAV (CDR A)))
                       (SRCHV2 A))))))
        (T A))))
))


 DEFINE((
 (STAGE20 (LAMBDA (X Y)
   (COND ((EQ (CDR Y) N)  (COND ((EQ (CDR X) NP)
                                 (LIST (CONS (LIST Y X) NP)))))
         ((EQ (CDR Y) NP) (COND ((EQ (CDR X) AJ)
                                 (LIST (CONS (LIST Y X) NP)))
                                ((EQ (CDR X) D)
                                 (LIST (CONS (LIST Y X) NP)))))
         ((EQ (CDR X) V)  (COND ((EQ (CDR Y) V)
                                 (LIST (CONS (CONS X Y) VP))))))))

 (COMB20 (LAMBDA (X Y)
   (COND ((EQ Y N)  (COND ((EQ X NP) T)
                          (T F)))
         ((EQ Y NP) (COND ((EQ X AJ) T)
                          ((EQ X D) T)
                          (T F)))
         ((EQ X V)  (COND ((EQ Y V) T)
                          (T F)))
         (T F))))

 (GTEST20 (LAMBDA (X Y)
   (COND ((NULL X) Y)
         ((NULL (CDR X)) (APPEND Y (LIST (CAR X))))
         ((COMB20 (CDAR X) (CDADR X))
          (APPEND Y (GTEST20 (CDDR X)
                              (STAGE20 (CAR X) (CADR X)))))
         (T (APPEND Y (GTEST20 (CDR X) (LIST (CAR X))))))))

 (FTEST20 (LAMBDA (A)
   (GTEST20 (CDR A) (LIST STAR))))
```

```
    ))
DEFINE((
(STAGE21 (LAMBDA (X)
  (COND ((EQ (CDR X) N) (LIST (CONS X NP)))
        ((EQ (CDR X) V) (LIST (CONS X VP))))))

(COMB21 (LAMBDA (X)
  (COND ((EQ X N) T)
        ((EQ X V) T)
        (T F))))

(GTEST21 (LAMBDA (X Y)
  (COND ((NULL X) Y)
        ((COMB21 (CDAR X))




          (APPEND Y (GTEST21 (CDR X) (STAGE21 (CAR X)))))
        (T (APPEND Y (GTEST21 (CDR X) (LIST (CAR X))))))))

(FTEST21 (LAMBDA (A)
  (GTEST21 (CDR A) (LIST STAR))))


    ))
DEFINE((
(STAGE30 (LAMBDA (X Y)
  (LIST (CONS (LIST Y X) NP))))

(COMB30 (LAMBDA (X Y)
  (COND ((EQ X VGD) (COND ((EQ Y NP) T)
                          (T F)))
        (T F))))

(GTEST30 (LAMBDA (X Y)
  (COND ((NULL X) Y)
        ((NULL (CDR X)) (APPEND Y (LIST (CAR X))))
        ((COMB30 (CDAR X) (CDADR X))
          (APPEND Y (GTEST30 (CDDR X)
                             (STAGE30 (CAR X) (CADR X)))))
        (T (APPEND Y (GTEST30 (CDR X) (LIST (CAR X))))))))

(FTEST30 (LAMBDA (A)
  (GTEST30 (CDR A) (LIST STAR))))


    ))
DEFINE((
(STAGE31 (LAMBDA (X)
  (LIST (CONS X NP))))

(COMB31 (LAMBDA (X)
  (COND ((EQ X VGD) T)    (T F))))

(GTEST31 (LAMBDA (X Y)
  (COND ((NULL X) Y)
        ((COMB31 (CDAR X))
          (APPEND Y (GTEST31 (CDR X) (STAGE31 (CAR X)))))
        (T (APPEND Y (GTEST31 (CDR X) (LIST (CAR X))))))))

(FTEST31 (LAMBDA (A)
  (GTEST31 (CDR A) (LIST STAR))))
      )).
```

```
DEFINE((
 (STAGE401 (LAMBDA (X Y)
      (LIST (CONS (LIST X Y) RP))))

 (COMB401 (LAMBDA (X Y)
         (COND ((EQ X NP) (COND ((EQ Y RP) T)









                                    (T F)))
                (T F))))

 (GTEST401 (LAMBDA (X Y)
    (COND ((NULL X) Y)
          ((NULL (CDR X)) (APPEND Y (LIST (CAR X))))
          ((COMB401 (CDAR X) (CDADR X))
             (APPEND Y (GTEST401 (CDDR X)
                              (STAGE401 (CAR X) (CADR X)))))
          (T (APPEND Y (GTEST401 (CDR X) (LIST (CAR X))))))))

 (FTEST401 (LAMBDA (A)
   (GTEST401 (CDR A) (LIST STAR))))
   ))


DEFINE((
 (STAGE41 (LAMBDA (X Y)
      (COND ((EQ (CDR X) PREP) (LIST (CONS (LIST X Y) PP)))
            ((EQ (CDR X) VP)   (LIST (CONS (LIST X Y) VP))))))

 (COMB41 (LAMBDA (X Y)
      (COND ((EQ X PREP) (COND ((EQ Y NP) T)
                               (T F)))
            ((EQ X VP)   (COND ((EQ Y RP) T)
                               (T F)))
            (T F))))

 (GTEST41 (LAMBDA (X Y)
    (COND ((NULL X) Y)
          ((NULL (CDR X)) (APPEND Y (LIST (CAR X))))
          ((COMB41  (CDAR X) (CDADR X))
             (APPEND Y (GTEST41  (CDDR X)
                              (STAGE41  (CAR X) (CADR X)))))
          (T (APPEND Y (GTEST41  (CDR X) (LIST (CAR X))))))))

 (FTEST41  (LAMBDA (A)
   (GTEST41  (CDR A) (LIST STAR))))
   ))


DEFINE((
 (STAGE43 (LAMBDA (X)
      (COND ((EQ (CDR X) RP) (LIST (CONS X NP)))
            (T X))))

 (GTEST43 (LAMBDA (X Y)
    (COND ((NULL X) Y)
          (T (APPEND Y (GTEST43  (CDR X) (LIST (CAR X))))))))

 (FTEST43  (LAMBDA (A)
   (GTEST43  (CDR A) (LIST STAR))))
   ))
```

```
DEFINE((
 (STAGE5 (LAMBDA (X Y)
      (COND ((EQ (CDR X) NP) (COND ((EQ (CDR Y) PP)
                                    (LIST (CONS (LIST X Y) NP)))))
            ((EQ (CDR X) VP) (COND ((EQ (CDR Y) PP)
                                    (LIST (CONS (LIST X Y) VP)))))
            ((EQ (CDR X) PP) (COND ((EQ (CDR Y) NP)
                                    (LIST (CONS (LIST Y X) NP)))))
                                                            )))

 (COMB5 (LAMBDA (X Y)
      (COND ((EQ X NP) (COND ((EQ Y PP) T)
                             (T F)))
            ((EQ X VP) (COND ((EQ Y PP) T)
                             (T F)))
            ((EQ X PP) (COND ((EQ Y NP) T)
                             (T F)))
            (T F))))

 (GTEST5 (LAMBDA (X Y)
    (COND ((NULL X) Y)
          ((NULL (CDR X)) (APPEND Y (LIST (CAR X))))
          ((COMB5   (CDAR X) (CDADR X))
             (APPEND Y (GTEST5   (CDDR X)
                         (STAGE5   (CAR X) (CADR X)))))
          (T (APPEND Y (GTEST5   (CDR X) (LIST (CAR X))))))))

 (FTEST5   (LAMBDA (A)
    (GTEST5   (CDR A) (LIST STAR))))
   ))


DEFINE((
 (FTEST6   (LAMBDA (A)
      (COND ((EQ (CDADR A) NP) (COND ((NULL (CDDDR A))
                                      (LIST STAR
                                         (CADDR A)
                                        (CADR A)))
                                    (T (LIST STAR
                                     (CADDR A)
                                       (CADR A)
                                         (CADDDR A)))))
            (T A))))

 (FTEST (LAMBDA (A)
    (FTEST6 (FTEST5 (FTEST43 (FTEST41 (FTEST401
          (FTEST31 (FTEST30 (FTEST21 (FTEST20
                    (FTEST1 (FTEST02 (FTEST01 A)))))))))))))
   ))
```

```
 ARGUMENTS FOR EVALQUOTE ...
   FTEST
   ((STAR (THE . D) (YOUNG . AJ) (BOY . N) (GOES . V) (TO . PREP) (SCHOOL . N)))

***  (ARGUMENTS OF FTEST
   ((STAR (THE . D) (YOUNG . AJ) (BOY . N) (GOES . V) (TO . PREP) (SCHOOL . N)))

***  VALUE OF FTEST
   (* ((((GOES . V) . VP) (((TO . PREP) (SCHOOL . N)) . PP)) . VP) (((((BOY . N) (YOUNG . AJ)) . NP)
    (THE . D)) . NP))
  TIME      99MS,  VALUE IS ...
   (* ((((GOES . V) . VP) (((TO . PREP) (SCHOOL . N)) . PP)) . VP) (((((BOY . N) (YOUNG . AJ)) . NP)
    (THE . D)) . NP))


 ARGUMENTS FOR EVALQUOTE ...
   FTEST
   ((STAR (THE . D) (BOY . N) (SWALLOWS . V) (QUICKLY . ADV) (AN . D) (ORANGE . AJ) (TANGERINE . N)
   ))

***  VALUE OF FTEST
   (* (((SWALLOWS . V) (QUICKLY . ADV)) . VP) (((BOY . N) (THE . D)) . NP) (((((TANGERINE . N) (ORANGE .
   AJ)) . NP) (AN . D)) . NP))
  TIME     116MS,  VALUE IS ...
   (* (((SWALLOWS . V) (QUICKLY . ADV)) . VP) (((BOY . N) (THE . D)) . NP) (((((TANGERINE . N) (ORANGE .
   AJ)) . NP) (AN . D)) . NP))


***  ARGUMENTS OF FTEST
   ((* (A . D) (CAR . N) (GOES . V) (EAST . ADV)))

***  VALUE OF FTEST
   (* (((GOES . V) (EAST . ADV)) . VP) (((CAR . N) (A . D)) . NP))


***  ARGUMENTS OF FTEST
   ((* (IT . PRO) (TURNS . V) (LEFT . ADV)))

***  VALUE OF FTEST
   (* (((TURNS . V) (LEFT . ADV)) . VP) ((IT . N) . NP))
```

```
*** ARGUMENTS OF FTEST
    ((* (HE . PRO) (MAY . ADV) (LOSE . V) (HIS . PRO) (LICENSE . N)))
*** VALUE OF FTEST
    (* (((LOSE . V) (MAY . ADV)) . VP) ((HE . N) . NP) (((LICENSE . N) (HIS . N)) . NP))


*** ARGUMENTS OF FTEST
    ((* (SSEING . VGD) (IS . V) (BELIEVING . VGD)))
*** VALUE OF FTEST
    (* ((IS . V) . VP) ((SEEING . VGD) . NP) ((BELIEVING . VGD) . NP))


*** ARGUMENTS OF FTEST
    ((* (A . AJ) (CAR . N) (GOES . V) (EAST . ADV)))
*** VALUE OF FTEST
    (* (((GOES . V) (EAST . ADV)) . VP) (((CAR . N) (A . AJ)) . NP))


*** ARGUMENTS OF FTEST
    ((* (THIS . AJ) (CAR . N) (TURNS . V) (RIGHT . ADV)))
*** VALUE OF FTEST
    (* (((TURNS . V) (RIGHT . ADV)) . VP) (((CAR . N) (THIS . AJ)) . NP))


*** ARGUMENTS OF FTEST
    ((* (WHAT . RP) (IS . V) (THE . D) (FINAL . AJ) (DIRECTION . N)))
*** VALUE OF FTEST
    (* *WHAT . RP) ((IS . V) . VP) (((((DIRECTION . N) (FINAL . AJ)) . NP) (THE . D)) . NP))
```

REFERENCES

1. Biss, K., Syntactic Analysis for the R2 System, Report R-399, Coordinated Science Lab., University of Illinois, Dec. 1968.

2. Bobrow, D. G., Syntactic Analysis of English by Computer - A Survey, AFIPS Conf. Proc., Spartan, Baltimore, Vol. 24, pp. 265-387 (1963).

3. Bobrow, D. G., "Syntactic Theories in Computer Implementations," in Borko, ed., Automated Language Processing, New York, Wiley, 1967.

4. Chomsky, N., "Syntactic Structures," The Hague: Mouton and Co., 1957.

5. Fodor, Jerry A. and Katz, Jerrold, The Structure of Language: Readings in the Philosophy of Language Prentice Hall. 1964.

6. Friedman, J. A., "A Computer System for Transformational Grammar," CACM, 11, 2, Feb. 1968, pp. 77-113.

7. Hays, D. G., "Automatic Language-Data Processing," in Borko, ed., Computer Applications in the Behavioral Sciences, Englewood Cliffs, N. J.: Prentice-Hall, 1962.

8. ___________, "Grouping and Dependency Theories," RM-2646. The Rand Corp., Santa Monica, California, September, 1960.

9. ___________, "Dependency Theory: A Formalism and Some Observations," RM-4087-PR. The Rand Corp., Santa Monica, California, July, 1964.

10. Kay, M., "A Parsing Program for Computational Grammars," RM-4283-PR, Rand Corp., Santa Monica, Calif., 1964.

11. Kellogg, C. H., "On Line Translation of Natural Language Questions into Artificial Language Queries," S.D.C. Document SP-2927. Apr. 28, 1967.

12. Kuno, S., "The Predictive Analyzer and a Path Elimination Technique," Comm. of ACM, V. 8 (July, 1965), 453-461.

13. ___________, "The Augmented Predictive Analyzer for Context-Free Languages--Its Relative Efficiency," Comm. of ACM, V. 9 (November, 1966), 810-823.

14. ___________, "Computer Analysis of Natural Languages," presented at the Symposia in Applied Mathematics, 1967.

15. McConlogue, K. and Simmons, R. F., "Analyzing English Syntax with a Pattern-Learning Parser," Comm. of ACM, V. 11 (November, 1965) 687-698.

16. Robinson, J., "Preliminary Codes and Rules for Automatic Analysis of English," RM-3339-PR, Rand Corp., Santa Monica, Calif. 1962.

17. Robinson, J., "PARSE: A System for Automatic Syntactic Analysis of English Text, Part I," RM-4654-PR, Rand Corp., Santa Monica, Calif. 1965.

18. Simmons, R. F., "Automated Language Processing," in Cuadra, ed., Annual Review of Information Science and Technology, 1965, New York, Interscience Publishers, 1966.

19. Simmons, R. F., Burger, J. F., and Schwarez, R. M., "A Computational Model for Verbal Understanding," Fall Joint Computer Conference, 1968, pp. 441-456.

20. Simmons, R. F., and Burger, J. F., "A Semantic Analyzer for English Sentences, Systems Development Corp. Report, January 10, 1968.

Distribution List as of August 1, 1970

ESD (ESTI)
L. G. Hanscom Field
Bedford, Mass 01731 2 Copies

Mr I. A. Balton
Institute for Exploratory Research
Code: AMSEL-XL
U. S. Army Electronics Command
Fort Monmouth, New Jersey 07703

LTC Howard W. Jackson
Deputy Dir of Electr. & Solid St. Sciences
Air Force Office of Scientific Research
1400 Wilson Boulevard
Arlington, Virginia 22209 5 Copies

Defense Documentation Center
Attn: DDC-TCA
Cameron Station
Alexandria, Virginia 22314 50 Copies

Director, Electronics Program
Attn: Code 427
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217 3 Copies

Naval Air Systems Command
AIR 03
Washington, D. C. 20360 2 Copies

Naval Electronic Systems Command
ELEX 03, Room 2046 Munitions Building
Department of the Navy
Washington, D.C. 20360 2 Copies

Director
Naval Research Laboratory
Washington, D.C. 20390
Attn: Code 2027 6 Copies

Commander
U. S. Naval Ordnance Laboratory
Attn: Librarian
White Oak, Md. 20910 2 Copies

Commanding General
Attn: STEWS-RE-L, Technical Library
White Sands Missile Range
New Mexico 88002 2 Copies

Commander
Naval Electronics Laboratory Center
Attn: Library
San Diego, Calif 92152 2 Copies

Dr L. M. Hollingsworth
AFCRL (CRN)
L. G. Hanscom Field
Bedford, Massachusetts 01731

Division of Engineering & Applied Physics
210 Pierce Hall
Harvard University
Cambridge, Massachusetts, 02138

Director
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Miss R. Joyce Harman
Project MAC, Room 810
545 Technology Square
Cambridge, Mass 02139

Professor R. H. Rediker
Electrical Engineering Prof.
Mass. Institute of Technology
Building 13-3050
Cambridge, Mass 02139

Raytheon Company
Research Division Library
28 Seyon Street
Waltham, Massachusetts 02154

Sylvania Electronic Systems
Applied Research Laboratory
Attn: Documents Librarian
40 Sylvan Road
Waltham, Mass 02154

Commanding Officer
Army Materials & Mechanics Res. Center
Attn: Dr H. Priest
Watertown Arsenal
Watertown, Massachusetts 02172

MIT Lincoln Laboratory
Attn: Library A-082
P.O. Box 73
Lexington, Mass. 02173

Commanding Officer
Office of Naval Research Branch Office
495 Summer Street
Boston, Massachusetts 02210

Commanding Officer (Code 2064)
U. S. Naval Underwater Sound Laboratory
Fort Trumbull
New London, Connecticut 06320

Dept of Eng & Applied Science
Yale University
New Haven, Conn 06520

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-CT-A
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-CT-D
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-CT-I
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-CT-L (Dr W. S. McAfee)
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-CT-O
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-CT-R
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Fort Monmouth, New Jersey 07703
Attn: AMSEL-CT-S

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-DL
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-GG-DD
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-KL-D
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-KL-E
Fort Monmouth, New Jersey 07707

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-KL-I
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-KL-SM
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-KL-S
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-KL-T
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-NL-A
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-NL-C
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-NL-D (Dr H. Bennett)
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-NL-P
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-SC
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-VL-D
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-VL-F
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-WL-D
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-XL-DT
Fort Monmouth, New Jersey 07703

Commanding General
U. S. Army Electronics Command
Attn: AMSEL-XL-D
Fort Monmouth, New Jersey 07703

Mr Norman J. Field, AMSEL-RD-S
Chief, Office of Science & Technology
Research and Development Directorate
U. S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Project Manager
Common Positioning & Navigation Systems
Attn: Harold H. Bahr (AMCPM-NS-TM),
Bldg. 439
U. S. Army Electronics Command
Fort Monmouth, New Jersey 07703

U. S. Army Munitions Command
Attn: Science & Technology
Info Br., Bldg 59
Picatinny Arsenal, SMUPA-RT-S
Dover, New Jersey 07801

European Office of Aerospace Research
Technical Information Office
Box 14, FPO New York 09510

Director
Columbia Radiation Laboratory
Columbia University
538 West 120th St.
New York, N.Y. 10027

New York University
Engineering Library
Bronx, New York 10453

Mr Jerome Fox, Research Coordinator
Polytechnic Institute of Brooklyn
333 Jay St.
Brooklyn, N. Y. 11201

Airborne Instruments Laboratory
Deerpark, New York 11729

Dr. W. R. Lepage, Chairman
Syracuse University
Dept of Electrical Engineering
Syracuse, N. Y. 13210

Rome Air Development Center
Attn: Documents Library (EMTLD)
Griffiss Air Force Base, N. Y. 13440

Mr H. E. Webb (EMBIS)
Rome Air Development Center
Griffiss Air Force Base, N.Y. 13440

Professor James A. Cadzow
Department of Electrical Engineering
State University of New York at Buffalo
Buffalo, N. Y. 14214

Dr. A. G. Jordan
Head of Dept of Elec Engineering
Carnegie-Mellon University
Pittsburgh, Penn 15213

Hunt Library
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA. 15213

Lehigh University
Dept of Electrical Engineering
Bethelehem, Pennsylvania 18015

Commander (ADL)
Naval Air Development Center
Attn: NADC Library
Johnsville, Warminister, Pa 18974

Technical Director (SMUFA-A2000-107-1)
Frankford Arsenal
Philadelphia, Pennsylvania 19137

Mr M. Zane Thornton, Chief, Network
Engineering, Communications and
Operations Branch, Lister Hill
National Center/ Biomedical Communications
8600 Rockville Pike
Bethesda, Maryland 20014

U. S. Post Office Dept
Library- Room 6012
12th & Pennsylvania Ave. N.W.
Washington, D.C. 20260

Technical Library
DDR&E
Room 3C-122, The Pentagon
Washington, D.C. 20301

Distribution List (Cont'd.)

Director for Materials Sciences
Advanced Research Projects Agency
Department of Defense
Washington, D.C. 20301

Assistant Director, (Research)
Office of Director of Defense Research
& Engineering
Pentagon, Rm 3C128
Washington, D.C. 20301

Chief, R & D Division (340)
Defense Communications Agency
Washington, D.C. 20305

Commanding General
U. S. Army Materiel Command
Attn: AMCRD-TP
Washington, D.C. 20315

Director, U. S. Army Material
Concepts Agency
Washington, D.C. 20315

Hq USAF (AFRDD)
The Pentagon
Washington, D.C. 20330

Hq USAF (AFRDDG)
The Pentagon
Washington, D.C. 02330

Hq USAF (AFRDSD)
The Pentagon
Washington, D.C. 20330

AFSC (SCTSE)
Andrews Air Force Base, Maryland 20331

Dr I. R. Mirman
Hq AFSC (SGGP)
Andrews AFB, Maryland 20331

Naval Ship Systems Command
Ship 031
Washington, D.C. 20360

Naval Ship Systems Command
Ship 035
Washington, D.C. 20360

Commander
U. S. Naval Security Group Command
Attn: G43
3801 Nebraska Avenue
Washington, D.C. 20390

U. S. Naval Oceanographic Office
Attn: M. Rogofsky, Librarian (Code 640)
Washington, D.C. 20390

Director
Naval Research Laboratory
Washington D.C. 20390
Attn: Dr A. Brodizinsky, Sup. Elec Div

Director
Naval Research Laboratory
Washington, D.C. 20390
Attn: Maury Center Library (Code 8050)

Director
Naval Research Laboratory
Washington, D.C. 20390
Attn: Dr W. C. Hall, Code 7000

Director
Naval Research Laboratory
Attn: Library, Code 2029 (ONRL)
Washington, D.C. 20390

Dr G.M.R. Winkler
Director, Time Service Division
U. S. Naval Observatory
Washington, D.C. 20390

Colonel E. P. Gaines, Jr
ACDA/FO
1901 Pennsylvania Ave. N. W.
Washington, D.C. 20451

Commanding Officer
Harry Diamond Laboratories
Attn: Mr Berthold Altman (AMXD)-T1)
Connecticut Ave & Van Ness St., N.W.
Washington, D.C. 20438

Central Intelligence Agency
Attn: CRS/ADD Publications
Washington, D.C. 20505

Dr H. Harrison, Code RRE
Chief, Electrophysics Branch
National Aeronautics & Space Admin.
Washington, D.C. 20546

The John Hopkins University
Applied Physics Laboratory
Attn: Document Librarian
8621 Georgia Avenue
Silver Spring, Maryland 20910

Commanding Officer (AMXRD-BAT)
U.S. Army Ballistics Research Laboratory
Aberdeen Proving Ground
Aberdeen, Maryland 21005

Technical Director
U. S. Army Land Warfare Laboratory
Aberdeen Proving Ground
Aberdeen, Maryland 21005

Electromagnetic Compatibility
Analysis Center (ECAC)
Attn: (ACOAT)
North Severn
Annapolis, Maryland 21402

Commanding Officer
U. S. Army Engineer Topographic Labs
Attn: STINFLO Center
Fort Belvoir, Virginia 22060

Director (NV-D)
Night Vision Laboratory, USAECOM
Fort Belvoir, Virginia 22060

U. S. Army Mobility Equipment Research
and Development Center
Attn: Technical Document Center
Bldg 315
Fort Belvoir, Virginia 22060

Dr Alvin D. Schnitzler
Institute for Defense Analyses
Science and Technology Division
400 Army-Navy Drive
Arlington, Virginia 22202

Director, Physical & Eng. Sciences Div.
3045 Columbia Pike
Arlington, Virginia 22204

Commanding General
U. S. Army Security Agency
Attn: IARD-T
Arlington Hall Station
Arlington, Virginia 22212

Dr Joel Trimble, Code 437
Information Systems Branch
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

Commanding General
USACDC Institute of Land Combat
Attn: Technical Library, rm 636
2461 Eisenhower Avenue
Alexandria, Virginia 22314

VELA Seismological Center
300 North Washington St.
Alexandria, Virginia 22314
U. S. Naval Weapons Laboratory
Dahlgren, Virginia 22448

Research Laboratories for the Eng.
Sciences, School of Engineering &
Applied Science
University of Virginia
Charlottesville, Va. 22903

Dr Herman Robl
Deputy Chief Scientist
U. S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina 27706

Richard O. Ulsh (CRDARD-IP)
U. S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina 27706

ADTC (ADBPS-12)
Eglin AFB, Florida 32542

Commanding Officer
Naval Training Device Center
Orlando, Florida 32813

Technical Library, AFETR
(ETV,MU-135)
Patrick AFB, Florida 32925

Commanding General
U. S. Army Missile Command
Attn: AMSMI-RR
Redstone Arsenal, Alabama 35809

Redstone Scientific Information Center
Attn: Chief, Document Section
U. S. Army Missile Command
Redstone Arsenal, Alabama 35809

AUL3T-9663
Maxwell AFB, Alabama 36112

Hq AEDC (AETS)
Attn: Library/Documents
Arnold AFS, Tennessee 37389

Case Institute of Technology
Engineering Division
University Circle
Cleveland, Ohio 44106

NASA Lewis Research Center
Attn: Library
21000 Brookpark Road
Cleveland, Ohio 44135

Director
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433
AFAL (AVTA) R. D. Larson
Wright-Patterson AFB, Ohio 45433

AFAL (AVT) Dr H. V. Noble, Chief
Electronics Technology Division
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433

Dr Robert E. Fontana
Head, Dept of Electrical Engineering
Air Force Institute of Technology
Wright Patterson AFB, Ohio 45433

Dept of Electrical Engineering
Clippinger Laboratory
Ohio University
Athens, Ohio 45701

Commanding Officer
Naval Avionics Facility
Indianapolis, Indiana 46241

Dr John C. Hancock, Head
School of Electrical Engineering
Purdue University
Lafayette, Ind 47907

Professor Joseph E. Rowe
Chairman, Dept of Electrical
Engineering
The University of Michigan
Ann Arbor, Michigan 48104

Dr G. J. Murphy
The Technological Institute
Northwestern University
Evanston, Ill 60201

Commanding Officer
Office of Naval Research Branch Office
219 South Dearborn St.
Chicago, Illinois 60604

Illinois Institute of Technology
Dept of Electrical Engineering
Chicago, Illinois 60616

Deputy for Res. and Eng (AMSE-DRE)
U. S. Army Weapons Command
Rock Island Arsenal
Rock Island, Illinois 61201

Commandant
U. S. Army Command & General Staff
College
Attn: Acquisitions, Library Division
Fort Leavenworth, Kansas 66027

Dept of Electrical Engineering
Rice University
Houston, Texas 77001

HQ AMD (AMR)
Brooks AFB, Texas 78235

USAFSAM (SMKOR)
Brooks AFB, Texas 78235

Mr B. R. Locke
Technical Adviser, Requirements
USAF Security Service
Kelly Air Force Base, Texas 78241

Director Electronics Research Center
The University of Texas at Austin
Eng-Science Bldg 110
Austin, Texas 78712

Department of Electrical Engineering
Texas Technological University
Lubbock, Texas 79409

Commandant
U. S. Army Air Defense School
Attn: Missile Sciences Div., C&S Dept
P. O. Box 9390
Fort Bliss, Texas 79916

Director
Aerospace Mechanics Sciences
Frank J. Seiler Research Laboratory (OAR)
USAF Academy
Colorado Springs, Colorado 80840

Director of Faculty Research
Department of the Air Force
U. S. Air Force Academy
Colorado Springs, Colorado 80840

Major Richard J. Gowen
Tenure Associate Professor
Dept of Electrical Engineering
U. S. Air Force Academy
Colorado Springs, Colorado 80840

Distribution List (Cont'd.)

Academy Library (DFSLB)
U. S. Air Force Academy
Colorado Springs, Colorado 80840

M. A. Rothenberg (STEPD-SC(S))
Scientific Director
Desert Test Center
Bldg. 100, Soldiers Circle
Fort Douglas, Utah 84113

Utah State University
Dept of Electrical Engineering
Logan, Utah 84321

School of Engineering Sciences
Arizona State University
Tempe, Ariz 85281

Commanding General
U. S. Army Strategic Communications Command
Attn: SCC-CG-SAE
Fort Huachuca, Arizona 85613

The University of Arizona
Dept of Electrical Engineering
Tucson, Arizona 85721

Cpt C. E. Baum
AFWL (WLRE)
Kirkland AFB, New Mexico 78117

Los Alamos Scientific Laboratory
Attn: Report Library
P. O. Box 1663
Los Alamos, N.M 87544

Commanding Officer
Atmospheric Sciences Laboratory
White Sands Missile Range, N. Mex 88002

Commanding Officer
(AMSEL-BL-WS-R)
Atmospheric Sciences Laboratory
White Sands Missile Range
New Mexico 88002

Chief, Missile Electronic Warfare Tech Area
(AMSEL-WL-M)
Electronic Warfare Laboratory, USACOM
White Sands Missile Range, N.M. 88002

Director
Electronics Sciences Lab
University of Southern California
Los Angeles, Calif 90007

Engineering & Mathematical Sciences Library
University of Californaia at Los Angeles
405 Hilgred Avenue
Los Angeles, Calif. 90024

Aerospace Corporation
P. O. Box 95085
Los Angeles, California 90045
Attn: Library Acquisitions Group

Hq SAMSO (SMTTA/Lt Belate)
AF Unit Post Office
Los Angeles, Calif. 90045

Dr Sheldon J. Wells
Electronic Properties Information Center
Mail Station E-175
Hughes Aircraft Company
Culver City, California 90230

Director, USAF PROJECT RAND
Via: Air Force Liaison Office
The RAND Corporation
Attn: Library D
1700 Main Street
Santa Monica, California 90406

Deputy Director and Chief Scientist
Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91101

Aeronautics Library
Graduate Aeronautical Laboratories
California Institute of Technology
1201 E. California Blvd.
Pasadena, California 91109

Professor Nicholas George
California Inst. of Technology
Pasadena, California 91109

Commanding Officer
Naval Weapons Center
Corona Laboratories
Attn: Library
Corona, California 91720

Dr F. R. Charvat
Union Carbide Corporation
Materials Systems Div.
Crystal Products Dept.
8888 Balboa Avenue
P.O. Box 23017
San Diego, Calif 92123

Hollander Associates
P. O. Box 2276
Fullerton, California 92633

Commander
U. S. Naval Missile Center (56322)
Point Mugu, California 93041

W. A. Eberspacher, Associate Head
Systems Integration Division
Code 5340A, Box 15
U. S. Naval Missile Center
Point Mugu, California 93041

Sciences-Engineering Library
University of California
Santa Barbara, California 93106

Commander (Code 753)
Naval Weapons Center
Attn: Technical Library
China Lake, California 93555

Library (Code 2124)
Technical Report Section
Naval Postgraduate School
Monterey, California 93940

Glen A. Myers (Code 52Mv)
Assoc. Professor of Elec. Engineering
Naval Postgraduate School
Monterey, California 93940

Dr Leo Young
Stanford Research Institute
Menlo Park, Calif. 94025

Lenkurt Electric Co., Inc.
1105 County Road
San Carlos, California 94070
Attn: Mr E. K. Peterson

Director
Microwave Laboratory
Stanford University
Stanford, California 94305

Director
Stanford Electronics Laboratory
Stanford University
Stanford, California 94305

Director, Electronics Research Laboratory
University of California
Berkeley, California 94720

DELETE:

LTC H.W. Jackson

REPLACE WITH:

Dr. L. A. Wood, Director
Electronic & Solid State Sciences
Air Force Office of Scientific Res.
1400 Wilson Boulevard
Arlington, Virginia 22209

DELETE:

Mr. Norman J. Field

REPLACE WITH:

Mr. Norman J. Field, AMCPM-AA-PM
Chief, Program Management Division
Project AACOMS, USAECOM, Bldg. 2525
Fort Monmouth, New Jersey 07703

DELETE:

Mr. Robert O. Parker, AMSEL-RD-S

REPLACE WITH:

Mr. I.A. Balton, AMSEL-XL-D
Executive Secretary, TAC/JSEP
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

ADDENDUM:

Dr. H.K. Ziegler, Chief Scientist
Army Member TAC/JSEP (AMSEL-SC)
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Dr. Billy Welch
USAFSAM (SMC)
Brooks AFB, Texas 78235

Command General
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Attn: AMSEL-RD-PB
(Miss F. Morris) 2 copies

DELETE:

Director, U.S. Army Material
Concepts Agency
Washington, D.C. 20315

REPLACE WITH:

Director
USA Advanced Materiel Concepts
Agency
2461 Eisenhower Avenue
Alexandria, Va. 22314

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of Illinois Coordinated Science Laboratory Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION 2b. GROUP

3. REPORT TITLE

AN ALGORITHM FOR THE SYNTACTIC ANALYSIS IN THE R2 INFORMATION SYSTEM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Schultz, Jeffrey and William T. Bielby

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1970	38	20

8a. CONTRACT OR GRANT NO. b. PROJECT NO. QE C-1-7-071213-4557 and DAAB 07-67-C-0199 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) R - 494 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) UILU-ENG 70-239

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Joint Services Electronics Program thru U.S. Army Electronics Command Fort Monmouth, New Jersey 07703

13. ABSTRACT

This paper presents a pure syntax parsing algorithm for the R2 question answering system. The parsing is a necessary part in the processing of information for the retrieval of answers by giving the syntactic structure of both questions and statements of fact. The syntactic analysis is performed on individual statements by using grammatical rules and various grammatical transformation. The parser has been implemented in LISP 1.5 on an IBM 360/75 computer. Experimental results from the system include examples of the analysis of some compound and complex sentences, as well as simple sentences.

DD FORM 1 NOV 65 1473

Security Classification

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Parsing						
Syntactic Analysis						
Natural Language Processing						
Transformational Grammar						
Phrase Structure Grammar						
Immediate Constituant Analysis						

Security Classification