LSP7

THE STRING PARSER FOR SCIENTIFIC LITERATURE

Naomi Sager
New York University

This paper describes a working parsing system which analyzes scientific articles syntactically. The grammar contains a BNF component and restrictions which operate on the BNF trees. The heart of the system is a small set of locating-relations written in a special Restriction Language (RL) which suffices for locating the arguments of a restriction test and for characterizing the transformational decomposition of the sentence.

INTRODUCTION

This paper describes a parsing program which has been operative since 1965 for analyzing scientific literature [1]. Our current field of application is pharmacology. There have been three implementations: in IPLV [2], in FAP for the IBM 7094 [3], and in Fortran for the CDC 6600. The Fortran program will be described by R. Grishman [9]. at this Symposium. This paper will deal primarily with the organization of the grammar and the type of analysis obtained.

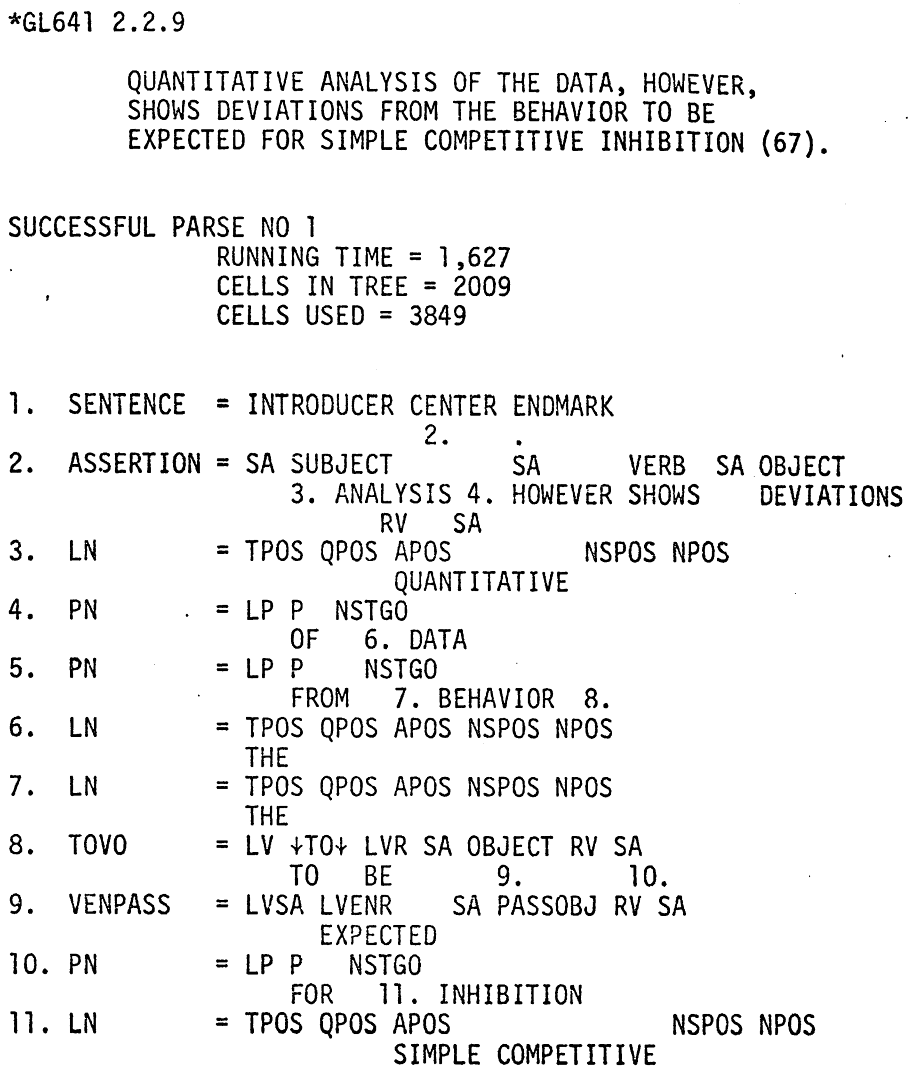
LSP 7

I. PARSING PERFORMANCE

The program produces a segmentation of the sentence into elementary word-strings of fixed grammatical structure. The output shows the point at which each elementary word-string enters into another elementary word-string of the sentence. Figure 1 shows the first output-parse obtained for a sentence from a pharmacology article [8].

One distinguished word-string, which enters into no other word-string, is called the center string of the sentence, and is itself an elementary sentence. In Figure 1, the center string is an ASSERTION (*analysis shows deviations*) appearing in line 2. In addition to a center string, a sentence may contain optional adjunct strings, i.e., modifiers, which occur at interstitial points of their host string. One type, called sentence adjunct SA, adjoins an entire string (e.g., *however* in line 2). Other types of adjunct strings are more closely associated with particular word-class elements of the host string, and usually occur to the left or right of the word class they modify. E.g., *analysis* in line 2 has a left adjunct written on line 3, the adjective *quantitative* (i.e., the analysis is quantitative), and a right adjunct on line 4, the prepositional phrase or "PN string" *of data* (i.e., the analysis is of data). *Data* has its own left adjunct *the* on line 6. The special status of the definite article is treated in restrictions.

To complete the description of the parse shown in Figure 1: In line 2 under "SA" following *deviations* there is a reference to line 5, which contains the PN string *from behavior*, here taken as a sentence adjunct. The ambiguity represented by this assignment is discussed below. *Behavior* has the left adjunct *the* in line 7 and the right adjunct *to be expected* in lines 8-9 (i.e., a certain behavior is to be expected), where *expected* is analyzed as occurring in the passive with no part of its active object remaining as a residual passive object. The sentence adjunct (line 10) referred to in line 8 adds the condition that the behavior to be



```
*GL641 2.2.9

     QUANTITATIVE ANALYSIS OF THE DATA, HOWEVER,
     SHOWS DEVIATIONS FROM THE BEHAVIOR TO BE
     EXPECTED FOR SIMPLE COMPETITIVE INHIBITION (67).


SUCCESSFUL PARSE NO 1
               RUNNING TIME = 1,627
               CELLS IN TREE = 2009
               CELLS USED = 3849


1.  SENTENCE  = INTRODUCER CENTER ENDMARK
                           2.     .
2.  ASSERTION = SA SUBJECT      SA      VERB  SA OBJECT
                   3. ANALYSIS 4. HOWEVER SHOWS   DEVIATIONS
                         RV   SA
3.  LN        = TPOS QPOS APOS          NSPOS NPOS
                          QUANTITATIVE
4.  PN        = LP P  NSTGO
                   OF   6. DATA
5.  PN        = LP P    NSTGO
                   FROM   7. BEHAVIOR  8.
6.  LN        = TPOS QPOS APOS NSPOS NPOS
                THE
7.  LN        = TPOS QPOS APOS NSPOS NPOS
                THE
8.  TOVO      = LV ↓TO↓ LVR SA OBJECT RV SA
                   TO   BE       9.       10.
9.  VENPASS   = LVSA LVENR    SA PASSOBJ RV SA
                     EXPECTED
10. PN        = LP P   NSTGO
                   FOR   11. INHIBITION
11. LN        = TPOS QPOS APOS              NSPOS NPOS
                          SIMPLE COMPETITIVE
```

Figure 1

expected is *for (the case of) inhibition* (line 10), where the *inhibition* is *simple* and *competitive* (line 11).

The program syntactically distinguishes two types of ambiguity. One, called *permanent predictable ambiguity* (p.p.a.), is of the type where a group of parses differ only in assigning the same adjunct string to be a modifier of a different element of the sentence. For example, the prepositional phrase *for simple competitive inhibition* in lines 10 and 11 of Figure 1 could have adjoined the center string *analysis shows deviations* in line 2, instead of *to be expected* in line 8, with a corresponding shift in meaning, (i.e., The analysis shows deviations for the case of simple competitive inhibition). Similarly the string *to be expected* in line 8 could have been an adjunct of *deviations* (line 2) instead of *behavior* (line 5), with the sense that the analysis shows the deviations which are to be expected for simple competitive inhibition. Also *from behavior* in line 5 has a preferred adjoining as a right adjunct of *deviations*, rather than as a sentence adjunct, though the latter assignment might be the preferred assignment for a different choice of words following the preposition *from*, e.g., *Quantitative analysis of the data, however, shows deviations from time t and on.*

An option of the program allows one to have printed only one parse for each set of permanent predictable ambiguities in which a given adjunct string is shifted from one adjunct slot to another. The other readings of the set can be read off this output or, if desired, a routine could print a statement of all the alternative positions of adjunction for the given string. If the option for suppressing p.p.a.'s is not taken, all syntactic analyses will be printed.

The other type of syntactic ambiguity recognized by the program is one in which the sentence can be segmented into string components in different ways, or if the sentence has the same string components in two parses, then at least one component has a different syntactic function in the second parse. Alternative parses due to this type of ambiguity are always printed out, though sometimes these parses are strange and suggest secondary syntactic features which might be used to rule them out. An example of a resegmentation of the sentence in Figure 1 is a parse which takes *deviations to be expected* as an object of *show* (as in *The hotel register shows him to be expected*); this parse was not predicted. An example of a change in syntactic function is the parse which takes *to be expected* as a sentence adjunct, where *to* is a shortened form of *in order to* (as in *Conferees must show identification to be admitted to the meeting*). Taken as a right adjunct of a noun (*behavior*), as in Figure 1, *to be expected* is a shortened form of *which is to be expected*.

With regard to performance, the following estimates are based on the operation of the FAP program,since the FORTRAN program is only now emerging from the final stages of debugging. More detailed performance figures appear in a book of published outputs prepared by Beatrice Bookchin [4]. Tested on consecutive, unedited sentences of scientific articles which are keypunched as they appear in print (except for trivial keyboard conventions and substitutions) we count the parser successful if:

- The intended analysis is obtained. (By "intended analysis" in the case where the p.p.a. option is in effect we mean the parse from which the intended adjunct assignment can be read, cf. above.)

- When all parses are obtained (again, with the p.p.a. option in effect) they should be small in number and represent only real syntactic potentialities of the sentence, not imprecisions of the grammatical system.

- The parses should be obtained in reasonable time (seconds).

On this basis the program succeeds in 60-80 per cent of the sentences, depending on the complexity of the text.

On the whole, when the program succeeds, as it does in an average of about 75 per cent of the sentences, it succeeds very well, producing the correct parse, usually as the first one, quite quickly, and rarely producing a syntactically invalid parse. When it does not perform this well it usually fails completely. There is very little in-between ground. The sentences on which it fails are of several main types: (1) They may involve science-special usages (like mathematical formulas, or special conjunctions such as a colon followed by a list) which have not yet been incorporated into the grammar; or(2) they involve several interacting coordinate conjunctions, commas, or comparatives; or (3) they are long and complex for other reasons (but usually involving commas and at least one conjunction).

II. THE GRAMMAR

The grammar consists of three parts:

1. A word dictionary which for each word entry lists all of the word's major parts of speech and subclasses in terms of:

   a. about 25 major classes: N noun, V verb, TV tensed verb, VING verb with -ing suffix, VEN past participle, etc.
   b. about 120 subclasses, of which
      - about 60 are closed grammatical classes
      - about 60 are used in classifying new words.

There are about 8,000-9,000 words now in the dictionary.

2. A BNF specification of the major constructions of English, in which the terminal symbols of the definitions are the major word classes, N, V, etc.

3. Restrictions, which are wellformedness tests performed on the BNF tree for a sentence, and which mainly check that the words corresponding to the terminal symbols have compatible attributes, e.g., number agreement of subject and verb, proper case of pronoun.

The BNF grammar contains four types of definitions, as illustrated in Figure 2. Exclusive of the fourth type — conjunctional strings — there are about 200 definitions in the grammar. About 100 definitions are of the type "linguistic string." These are the definitions which were assigned separate lines in the output. An example is ASSERTION, which contains the required elements SUBJECT, VERB, OBJECT, and the optional elements: SA at points where sentence adjuncts occur, RV for post-object verb adjuncts, and CA_i for conjunctional strings which occur following the i^{th} element (not counting other CA's). CA's are not shown in the output unless a CA-string occurs in the sentence.

About 20 definitions are of the type "adjunct set." These define sets of some of the above strings, which occur at particular points of adjunction. For example, the definition of SA includes *INT (interjections, e.g., *however*) and PN (*for the case of simple inhibition*). The symbol *R means that the adjunction is repeatable: *however for the case of simple inhibition.*

About 80 definitions are collections of "positional variants," covering the different ways element positions may be filled in different sentences. For example, the positional variants of SUBJECT are:

NSTG: concrete N with its adjuncts (*external potassium*), or nominalized verb or adjective that is morphologically a noun (*quantitative analysis of*

```
.  Linguistic String

   <ASSERTION>  ::= <SA><SUBJECT><CA2><SA>
                    <VERB><CA4><SA><OBJECT><CA6><RV><SA>.


?.  Adjunct Set

   <SA>*R       ::= <*INT>/<DSTG><RD>/<PN>/<PA>/<NSTGT>/
                    <RSUBJ>/<RNSUB>/<LCS><CSSTG>/-<OBJBESA>/
                     -<SOBJBESA>/<VINGO>/<VENPASS>/<SAWH>/
                     <TOVO>/<NVSA>.

3.  Positional Variants

   <SUBJECT>    ::= <NSTG>/<VINGSTG>/<SN>/<*NULLWH>/THERE.

   <NSTG>       ::= <LNR>/<NWHSTG>/<NAMESTG>/<LPROR>.

   <LNR>        ::= <LN><CN1><NVAR><CN2><RN><CN3>.
                     a general type <LXR>=<LX><*X><RX>
                       for X atomic

   <NVAR>       ::= <*N>/<*VING>/<*NULLN>.

4.  Conjunction Strings

   <CA2>        ::= <*CONJ><QA2>/NULL.
   <QA2>        ::= <SUBJECT>.

   <CA4>        ::= <*CONJ><QA4>/NULL.
   <QA4>        ::= <VERB>/<SUBJECT><VERB>.

   <CA6>        ::= <*CONJ><QA6>/NULL.
   <QA6>        ::= <OBJECT>/<VERB><OBJECT>/<SUBJECT><VERB>
                    <OBJECT>.
```

Figure 2

Types of BNF Definitions in Grammar of English

the data, related to *someone analyzes the data quantitatively*).

VINGSTG: less N-like nominalization based on Ving (*the analyzing of the data, analyzing the data*).

SN: sentence-like nominalization (*that the data requires analysis, whether the data can be analyzed*).

*NULLWH: the null element in WH-strings, indicated by ( ) in the output.

there as in *there is another analysis* (as opposed to *over there is another analysis*).

The positional variants of NSTG are:

LNR: N or N-variant with its left and right adjuncts. There are about 15 definitions of the type <LXR> ::= <LX><X><RX> where LX and RX are, respectively, the sets of left and right adjuncts of X.

NWHSTG: noun replacement *wh*-string (*whichever ion is present*). This contrasts with the sentence-nominalization-*wh*-string based on *whether: Whichever ion is present enters the cell, Whether an ion is present is unknown.*

NAMESTG: names.

LPROR: pronoun with adjuncts.

The noun position NVAR in LNR can be filled by the atomic (word classes):

N: e.g., *potassium*

VING: e.g., *filling* in *pump filling*

NULLN: implicit N, as in *The other ( ) is ...* .

The fourth type of definition in the BNF grammar is that of conjunctional strings, illustrated by CA2, CA4, CA6, in Figure 2. The Fortran program treats conjunctions by explicit definition. The IPL and FAP programs, on the other hand, treat conjunctions by means of an algorithm which generates just those conjunctional strings which can occur at the given point p in the sentence (where the conjunction occurs as pth word) given the string analysis of words 1 through p-1. It also locates (for the ASSERTION string with "strictly parallel conjoining"[1], [5]) the words of the sentence which are repeated implicitly (i.e., zeroed) in the conjunctional string; the sentence *The cells lose potassium and gain sodium*, for example, would be filled out to *The cells lose potassium and (the cells) gain sodium*.

The change in strategy with regard to conjunctions in the Fortran program results from practical and theoretical complexities which arose in generalizing the IPL-FAP conjunction algorithm so as to treat all conjoinings as deriving from the conjunction of complete sentences, and in filling in the zeroed elements in the conjunctional strings throughout. The Fortran program resolves the problem by computing the truncated strings locally, that is, providing BNF definitions to cover the segments which may be encountered in the sentence, at the same time that it provides a behind-the-scenes facility that locates the sentence-words which should be repeated if the conjunctional string is to be filled out to a whole sentence; i.e., it locates but does not fill in the zeroed elements. Thus, in the sentence *Digoxin and other glycosides inhibit the uptake of K^{42} by red cells*, for purposes such as checking the agreement of subject and verb, the program sees the locally conjoined nouns: *Digoxin and other glycosides*. But for purposes of finding out what can be the subject of *inhibit the uptake of K^{42}*, the program sees the filled out sentences: *Digoxin inhibit(s) uptake of K^{42}; Other glycosides inhibit the uptake of K^{42}*.

We have gone into some detail concerning the content of the BNF grammar in anticipation of later remarks on transformations. It can be seen that a considerable amount of transformational information is built into the BNF grammar:

- Many BNF definitions are such as mark off a word sequence in the sentence which is the same as that resulting from the operation of a particular transformation, e.g., the sentence nominalization strings of different types.

- Named positions in such strings often indicate the transformational source.

- The string solution to handling conjunctions goes quite far toward dealing with the transformational property of zeroing in conjoined sentences.

III. RESTRICTIONS

Little has been said so far about restrictions, though the restriction mechanism is what is unique and of most interest in the grammar.

Consider the sentence: *The sodium efflux which occurs in the absence of external potassium seems to be passive*, and its output parse in Figure 3. A restriction on subject-verb number agreement applied to the words in output line 2 in Figure 3 distinguishes this sentence from the non-well-formed: ∄ *The sodium efflux which occurs in the absence of external potassium seem to be passive*. The same restriction applied at line 7 also eliminates a non-well-formed sequence similar to the original but with the verb *occur* in place of *occurs*: ∄ *The efflux occur*. To apply the restriction in this case we make use of the program's ability to supply the antecedent word (*efflux*) in place of the null subject in the ASSERTION in line 7, which is part of the relative clause WHS-N in line 4. The mechanism of restrictions can

```
GL641  2.  3.  5  A

          THE SODIUM EFFLUX WHICH OCCURS IN THE
          ABSENCE OF EXTERNAL POTASSIUM SEEMS
          TO BE PASSIVE.


   PARSE 1

.  SENTENCE  = INTRODUCER CENTER ENDMARK
                              2.     .

.  ASSERTION = SA SUBJECT        SA  VERB   SA OBJECT  RV
                   3. EFFLUX 4.   5. SEEMS       6.
               SA

.  LN        = TPOS QPOS APOS NSPOS NPOS
               THE                   SODIUM

.  WHS-N     = DDD    ASSERTION
               WHICH     7.

.  PN        = LP P  NSTGO
                  IN   8. ABSENCE  9.

.  TOVO      = LV ↓TO↓ LVR SA OBJECT  RV SA
                  TO           PASSIVE

.  ASSERTION = SA SUBJECT   SA VERB      SA OBJECT  RV SA
                  ( )          OCCURS

.  LN        = TPOS QPOS APOS NSPOS NPOS
               THE

.  PN        = LP P  NSTGO
                  OF  10. POTASSIUM

0. LN        = TPOS QPOS APOS      NSPOS NPOS
                         EXTERNAL
```

Figure 3

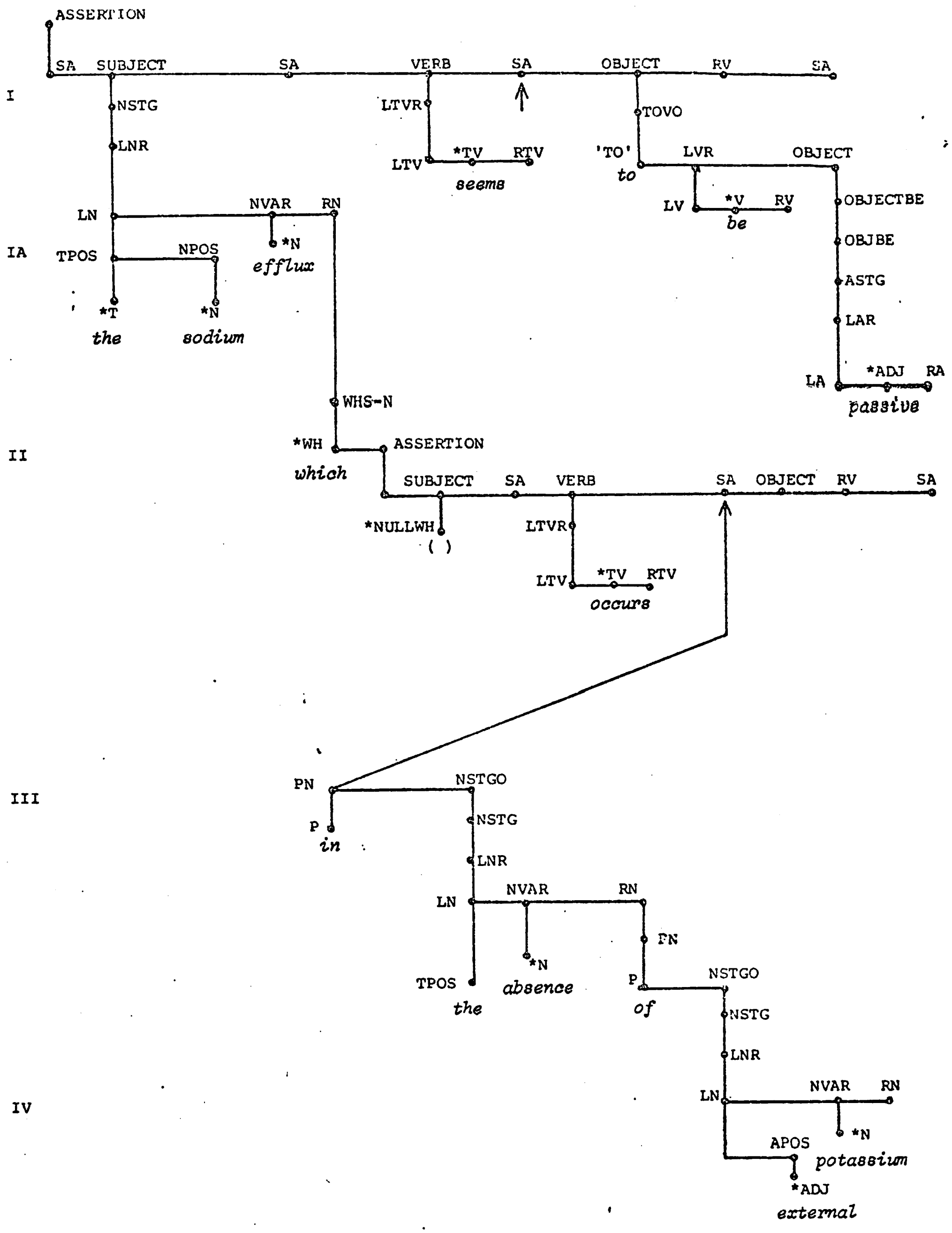


Figure 4

be introduced by noting certain features of the BNF tree (Figure 4) for the above sentence.

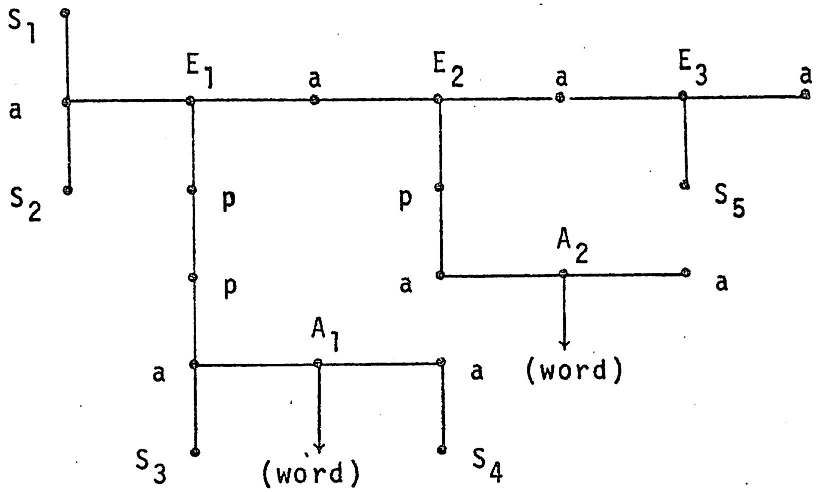
Let us define the *core* of an element E to be the first node of the type "atom" or "linguistic string" below E, provided the descent is blocked at nodes corresponding to definitions of the type "adjunct set." Thus, in the center string ASSERTION of Figure 4, the core of the SUBJECT is the atom *N (*efflux*), the core of the VERB is the atom *TV (*seems*), and the core of the OBJECT is the string TOVO. Note that every required element of the linguistic strings in this tree (ASSERTION, TOVO, WHS-N, PN) has a unique core as just defined. (We view WHS-N as having the required elements *WH, SUBJECT, VERB, OBJECT.)

Let us define "right adjunct of X" to be the string or atom lying below RX in an LXR definition. In Figure 4, the right adjunct of N (*efflux*) is WHS-N. "Left adjunct of X" is similarly defined.

A generalized representation of the BNF subtree originating from a node of the string type is shown in Figure 5. The important point is that every sentence tree is composed of just such modules. Where a module terminates in a string, a new module originates.

A small set of "locating relations," such as "core" and "right adjunct," above, have been defined on this module. Each locating relation is associated with a routine which operates in terms of the types of definitions in the BNF grammar, not individual definitions. Thus a few of them suffice to describe many sentence trees which differ in detail. Some of the main elementary locating relations are listed in Figure 5. More complex relations can quite readily be composed as products of the elementary relations.

A *restriction* is then a test (usually an attribute check) applied between two arguments which are related either by an elementary locating relation or by a product of elementary locating



Node Types

S string

a adjunction

A atom

p positional variant

Figure 5

Elementary Locating Relations

string to element
- down: S_1 HAS ELEMENT E_1
- up: E_1 IS AN ELEMENT OF S_1
S_1 IS THE IMMEDIATE STRING OF E_1

element to element E_1 HAS COELEMENT E_2

element to (core) atom
- down: E_1 HAS CORE A_1
THE CORE OF E_1 IS A_1
- up: A_1 IS THE CORE OF E_1

atom to string
- down: A_1 HAS LEFT ADJUNCT S_3
A_1 HAS RIGHT ADJUNCT S_4
- up: HOST. OF S_3 IS A_1
HOST OF S_4 IS A_1

element to (core) string
- down: E_3 HAS CORE S_5
- up: S_5 IS THE CORE OF E_3

string to string
- down: S_1 HAS SENTENCE ADJUNCT S_2
- up: S_1 IS THE HOST-STRING OF S_2

Figure 5 (continued)

relations. The locating relations, along with the syntax of BNF, have been made into a language to serve as a metalanguage for English grammar. The grammar of English is a text in this language. For the user's convenience and for linguistic interest the language has been defined as a sublanguage of English which has its own BNF grammar and compiler. This shows, incidentally, that a grammar of a natural language can be written in a language which has a simpler grammar than that of the language it describes.

An interesting feature of the modular tree structure and the locating relations is that the large majority of restrictions turn out to operate within one module or adjacent modules. The restrictions which reach beyond one adjacent module are those which must overcome repetitions of two types of transformational operators: verb operators (e.g., *seems to want to try to ...*, etc.), and operators on sentences (*It seems that they know that it was assumed that ...*).

As an example of a restriction which handles the effects of sentence- and verb-operators, consider W113, reproduced in Figure 6. This restriction checks that if the subject is a weakly nominalized sentence of the type SN, then the deepest verb (defined below) should be a sentence operator verb or other acceptable verb type. E.g., *That this greatly simplifies the chronologic problem seems to be obvious*, ∄ *That this greatly simplifies the chronologic problem seems to walk briskly*. In the case where the deepest verb is *be* (as in the example sentence in Figure 6) a further restriction checks that the predicate noun or adjective is appropriate.

In Figure 6, the portion of W113 immediately following the equality sign states that W113 is housed in the ASSERTION string, the YES-NO question, all the object nominalization strings, and all the permuted center strings. The restriction first checks whether the value of the subject is SN, i.e., whether SN was the option chosen for SUBJECT. This is the case in the example sentence of Figure 6. If so, two tests must be true:

```
*W113:  CORRECT VERB FOR SN SUBJECT

 W113=  IN ASSERTION, YESNOQ, SENTNOM, PERMUTLIST:  IF THE SUBJECT
        HAS THE VALUE SN, THEN BOTH $VFORSN AND $VPASS ARE TRUE.

 $VFORSN= THE CORE X2 OF THE DEEPEST VERB X4 IS VBE OR BEREP OR
          VSENT1 OR VSENT2 or VSENT3 OR VMOD OR VEXP.

 $VPASS= IF X2 IS VSENT3 THEN X4 IS AN ELEMENT OF VENPASS.


 ROUTINE DEEPEST-VERB= ITERATE $VBSTG UNTIL $OBJV FAILS.

 $VBSTG= THE PRESENT STRING HAS ELEMENT VERB OR LIVINGR OR LVENR
         OR LVR OR VERB1 (VALUE IS LTVR) OR VERB2 OR VERB3.

 $OBJV= CORE OF COELEMENT OBJECT IS VO OR VINGO OR VENO OR VENPASS
        OR TOVO.
```

Figure 6

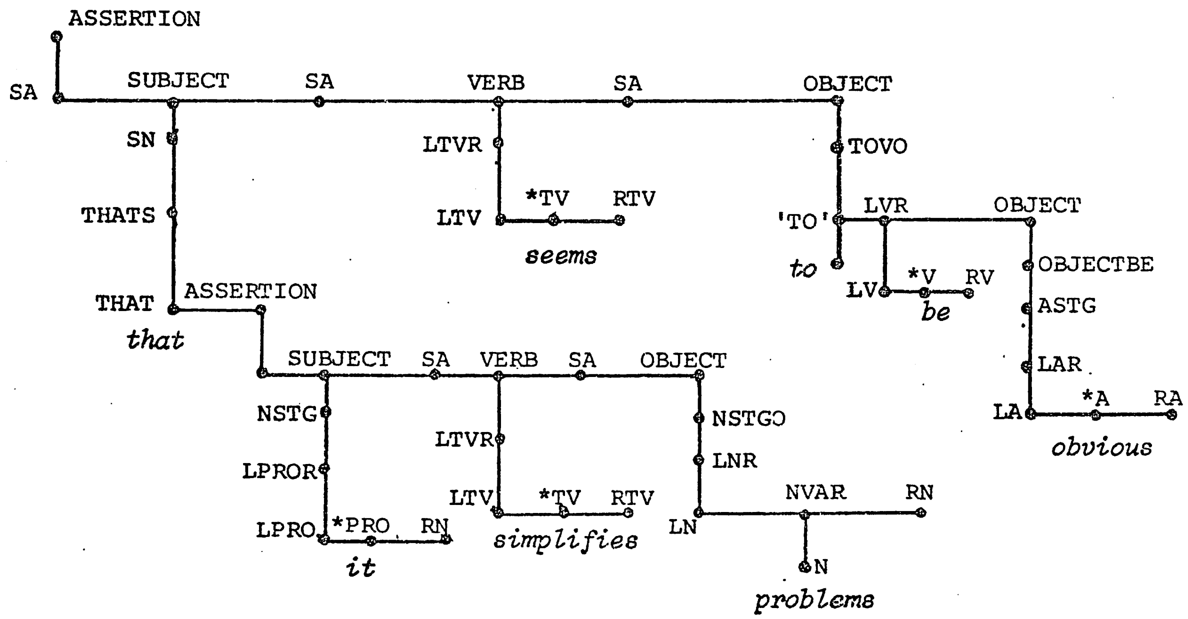


Figure 6 (continued)

$ VFORSN: The core of the deepest verb is VBE (*be*), or BEREP (*be* replacer) or one of various sentence-operator verbs, modals (*has importance*) or sentence-operator idioms (VEXP).

$ VPASS: If the deepest verb is of the type VSENT3 (*think, assume*), which in the active take SN as object, then this verb should be in the passive.

The DEEPEST VERB routine iterates through certain verb-containing strings, and stops when the object of the given string no longer has such a verb-containing string as object. In the example sentence in Figure 6, the first iteration lands on VERB, the second on LVR.

As this restriction illustrates, the restrictions add to the treatment of transformational material in the grammar, since the restrictions must often overcome the effects of transformational operators, such as the distancing of the subject from the verb as a result of intervening verb operators, or the permutation of the verb-object to subject position, as in the passive.

IV. TRANSFORMATIONS

Since our interest is in processing the information carried by language in scientific articles, we hope to arrive at syntactic components of the sentence which correspond directly to informational components of the sentence, and we would like to arrive at a single form for components which carry the same information. Undoing the effects of transformations is a step in this direction. For example, *The sodium efflux is passive*, and *The sodium efflux seems to be passive* contain the same information components except for the addition of the aspectual verb *seems* in the second sentence.

For this informational goal it may be that a complete transformational analysis will not be necessary; that is, we may keep some segments in their transformed state. In this our goals are less ambitious than, say, that of Dr. Joshi's group,which aims at covering in a principled way the entire system of transformations in the language.

A survey of several hundred transformations formulated in our string computational form has been compiled by Barbara Anderson [6]. Consider the main classes of transformations:

- the binaries (conjunctions, *wh*) correspond quite directly to adjunct and conjunctional strings, in which a considerable start on the zeroing problem has been made;

- the weak sentence nominalizations correspond to subject and object nominalization strings in the BNF grammar;

- the strong nominalizations are encapsulated in LNR;

- the verb operators are those whose objects are iterated through in the DEEPEST VERB routine (cf. III above) and certain objects of *be*;

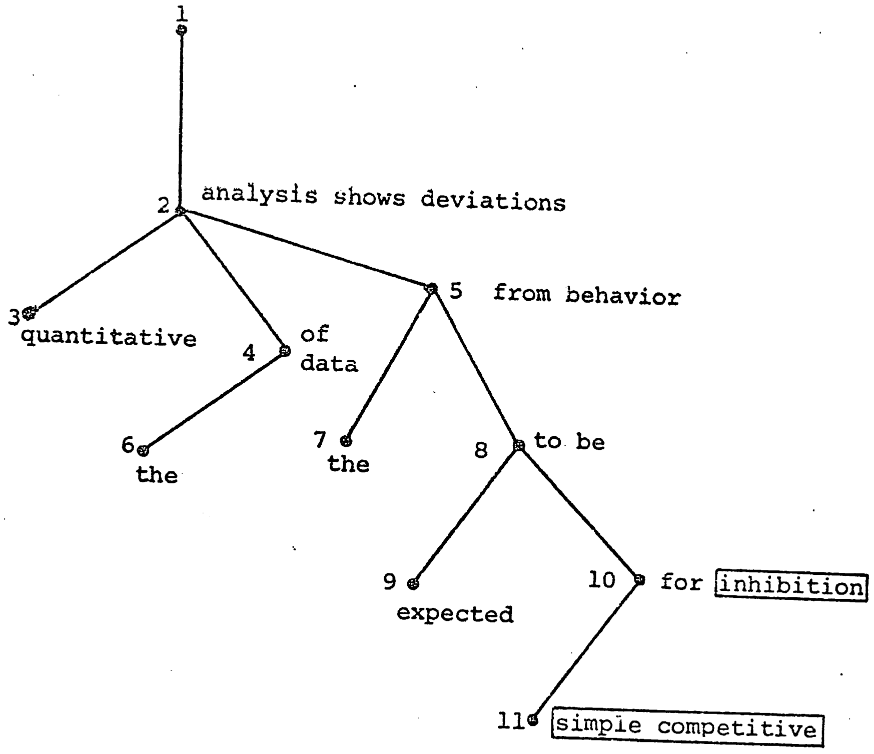
- the sentence operators are treated in a routine called the ULTIMATE OBJECT, which is similar to the DEEPEST VERB routine, but iterates (upwards) through objects of sentence-operator verbs.

It turns out that the transformations, like the grammatical constraints, operate within a module or adjacent modules. This is not surprising, if we think how transformations operated to form the sentence: each transformation could only move or add elements in its operand (the sentence to date) into positions within the operand or contiguous to the operand. Clearly the grammatical constraints also had to satisfy this condition of contiguity. It is therefore understandable that the framework which makes grammar constraints show up within a module, or immediately adjacent ones, should also be an adequate framework for recognizing the effects of transformations.

V. SEMANTIC RELEVANCE OF THE ANALYSIS

The semantic relevance of the sentence decompositions obtained by the String Program is demonstrated by a particular result which has come out of our current work on pharmacology texts. Working with texts from a particular scientific subfield (currently, the cellular pharmacodynamics of digitalis), it is possible to establish on outside grounds (e.g., by asking informants) which of the lexical items are particular to the subfield (or are English or general science words used in a particular way in the subfield). We call this the science-specific vocabulary.

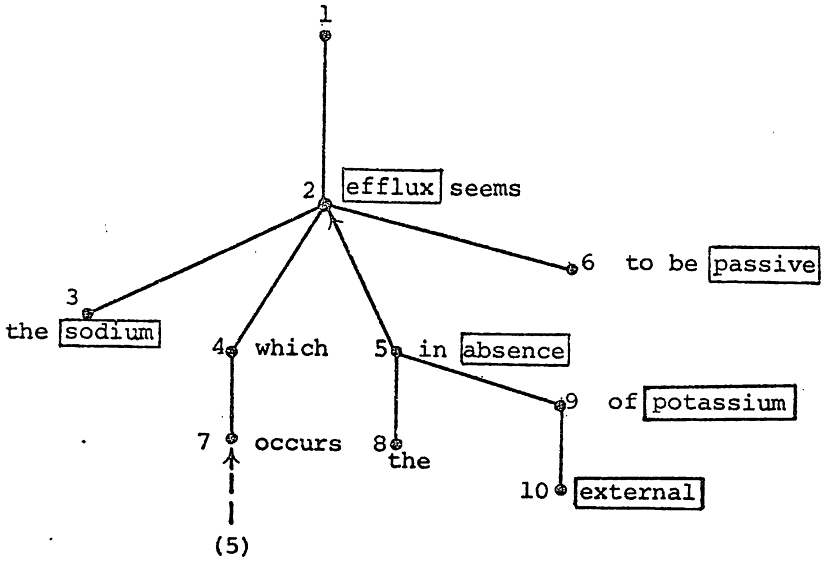
If we now take our sentence decompositions, which have been obtained using only general syntactic properties of the words (i.e., Noun, Verb, etc., as noted in II above), and mark the output lines which contain words from the science-specific vocabulary, we find that the science-specific vocabulary is clearly separated from the more general science and English vocabulary in its location in the decomposition: The output lines containing science-specific vocabulary are connected and lie at the "bottom" of the decomposition. This can be seen in Figure 7, in which the output parses of Figures 1 and 3 have been reduced to show which output lines connect to which others, and the science-specific vocabulary items have been outlined. This result is even sharper in a transformationally extended string analysis, such as that discussed in IV above, or in transformational semi-lattices [7], which show the hierarchical structure of operators. For example, in a sentence such as GL641 2.3.5A (bottom half of Figure 7), the verb *seems* is an operator on *sodium efflux is passive* and in a transformational representation would appear above its operand sentence. The same applies to *occurs*, which operates on *sodium efflux*.



GL641 2.2.9: Quantitative analysis of the data, however, shows deviations from the behavior to be expected for simple competitive inhibition.

Figure 7

String Diagrams of Output Parses



GL641 2.3.5A: The sodium efflux which occurs in the absence of external potassium seems to be passive.'

Figure 7 (continued)

Even without further transformational refinements in the string program this result shows that it is possible to isolate the object-language portions of sentences in the scientific literature by purely syntactic, and computable, means.

An even greater semantic sharpness can be obtained by syntactic methods applied within restricted subject-matter areas of science, for example, by constructing a science sublanguage grammar for a specific subfield, as we are currently doing for the digitalis subfield of pharmacology. The idea here is that the discourse within a science subfield reflects, in addition to the ordinary constraints of English grammar, the constraints imposed by the subject matter itself. For example, the sentences *Potassium enters the cell* and *The heart enters the cell* are both well-formed sentences of English, but only the former of the two would be accepted by a pharmacologist as a well-formed (i.e. possible) sentence of pharmacology. Using standard linguistic distributional analysis of the lexical items appearing in the sentence-decompositions obtained for texts in the given subfield, it has been possible to construct a grammar whose word subclasses reach a high level of semantic refinement. E.g., *potassium* and *heart* are in disjoint noun subclasses because they occur with different subclasses of verbs in elementary sentences of the sublanguage. Overall, the word classes and the hierarchy of operators (in a transformational form of the sublanguage grammar) reflect closely the classes of objects and relations which characterize the subfield.

REFERENCES

[1] Sager, N., "Syntactic Analysis of Natural Language," *Advances in Computers 8*, F. Alt, M. Rubinoff, Eds., Academic Press, New York, 1967. *String Program Reports Nos. 1-5*, Linguistic String Project, New York University, 1966-1969.

[2] ———, J. Morris, M. Salkoff, and C. Raze, "Report on the String Analysis Program," *String Program Reports (S.P.R.) 1*, New York University, Linguistic String Project, 1966.

[3] Raze, C., "The FAP Program for String Decomposition of Scientific Texts," *S.P.R. 2*, 1967.

[4] Bookchin, B., "Computer Outputs for Sentence Decomposition of Scientific Texts," *S.P.R. 3*, 1968.

[5] Sager, N., "A Computer String Grammar of English," *S.P.R. 4*, 1968.

[6] Anderson, B., "Transformationally Based English Strings and Their Word Subclasses," *S.P.R. 7*, 1970.

[7] Harris, Z.S., *Mathematical Structures of Language*, Interscience Tracts in Pure and Applied Mathematics 21, John Wiley and Sons, New York, 1968.

[8] Glynn, I.M., "The Action of Cardiac Glycosides on Ion Movements," *Pharm. Review 16*, 1964.

[9] Grishman, Ralph, "Implementation of the String Parser of English," this volume.

ACKNOWLEDGMENT

This research was supported in part by Research Grant No. LM00720-01, from the National Library of Medicine, National Institute of Health, DHEW, and in part by Research Grants GS-2462 and GS-27925, from the National Science Foundation, Division of Social Sciences, and in its earlier stages by Research Grants GN559 and GN659 from the National Science Foundation, Office of Science Information Service. I am happy to acknowledge the valuable contributions to this work by past and present members of the String Project Staff, chiefly, Morris Salkoff, James Morris, Carol Raze, Beatrice Brookchin Ralph Grishman, Richard Schoen, Goerge Chaikin, Matthew Bottkol, and Judith Clifford.