LSP4

****

THE SUBLANGUAGE METHOD IN STRING GRAMMARS

by

Naomi Sager

The University of Texas at El Paso

****

NAOMI SAGER

has a degree in Electrical Engineering
from Columbia University and
a doctoral degree in Linguistics
from the University of Pennsylvania.
She is a Senior Research Scientist
and Adjunct Associate Professor
of Linguistics at New York University,
and the principal investigator of research
projects for the National Library of Medicine
and the National Science Foundation.

Grammars of natural languages are generally written in languages of the kind they describe, e.g., Jespersen's grammar of English, written in English. Even grammars which contain a formal part require supplementary statements (i.e., statements not in the formal system) which describe how the grammar applies to particular sentences.

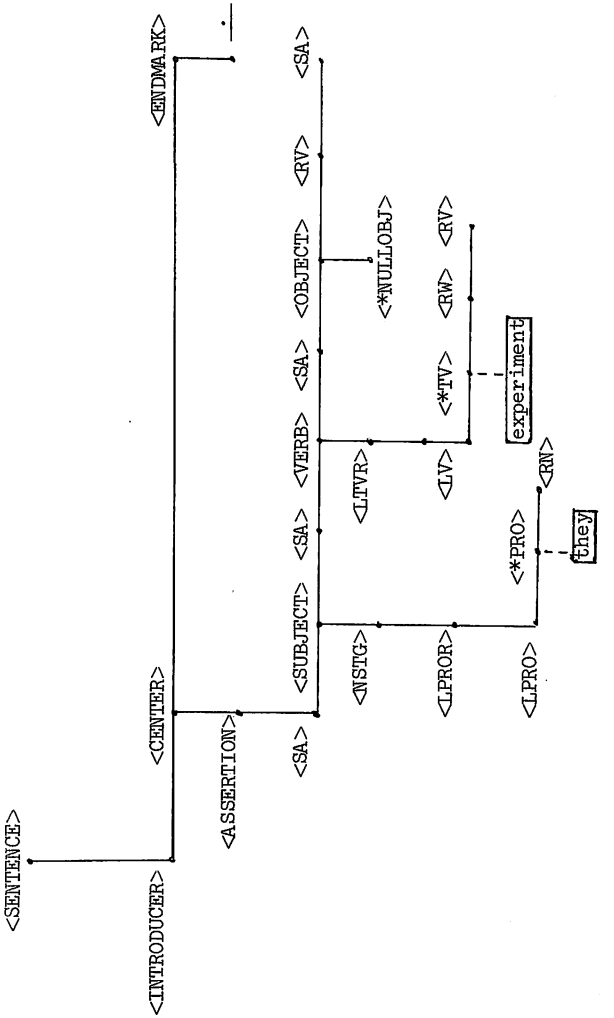
A test of the coverage and adequacy of a formal grammar is its computability: Can a computer program which incorporates the grammar produce adequate analysis of a representative corpus of sentences of the language? Consider a computer program which does just this.[1] The program includes a string grammar of English in symbolic form which is translated by a compiler into computer code. The grammar code is subsequently invoked by a parsing program to analyze input English sentences. The grammar consists of 2 parts, a set of about 200 context-free productions stating the main strings and string-sets of English[2] -- these are written as Backus Normal Form (BNF) definitions[3] -- and a set of about 350 restrictions which, in their coded form, are wellformedness tests invoked by the parsing program in the course of analyzing a given input sentence S. The tests are executed on the parse tree for S which is constructed by the parsing program from the BNF definitions.

For example the string parse tree for the sentence They experiment is shown in Figure 1. The elementary sentence, or center string, of the sentence is the word string they experiment which corresponds to the sequence of elements <SUBJECT> <VERB> <OBJECT> of the linguistic string <ASSERTION>. (The value of <OBJECT> for the case of the intransitive verb experiment is a null element.) The BNF definition of <ASSERTION> also includes mentions of adjunct sets, e.g., <SA> for sentence adjunct strings and <RV> for right adjuncts of the verb, which are string-sets whose members have optional occurrence in <ASSERTION> at the points noted (SA: Often they experiment; post-object RV: He hit the ball hard). The atoms or terminal symbols of the system stand for major word classes and null elements of the grammar, e.g., <*PRO> for pronoun, <*TV> for tensed verb.

We call the first node which represents either a word class or a linguistic string and which lies below a given string element E on the parse tree (disregarding adjunct strings in the subtree of E) the core of E ; e.g., <*PRO> is the core of <SUBJECT> in Figure 1. The word class core is the case of an endocentric construction, and the string core the case of an exocentric construction (What he said is clear). The points of occurrence of the set of left adjuncts <LX> of a word class <*X>, and of the set of right adjuncts <RX> of <*X>, are indicated by the respective positions of the elements <LX> and <RX> in definitions of the form: <LXR> :: = <LX> <*X> <RX>, which occur as the value of elements whose core can be <*X>.[4]

Figure 1

Parse tree for the sentence They experiment.



The function of the restriction part of the grammar is to rule out instances of wellformed word class sequences where the particular words corresponding to the word classes in the sequence have incompatible sub-classifications and hence do not constitute wellformed sentences of the language. For example, they experiment is a wellformed sentence, but He experiment is not, although both conform to the word-class sequence <*PRO> <*TV>. An example of a restriction, then, is one which checks that the words corresponding to the elements <SUBJECT> and <VERB> of <ASSERTION> are in compatible subclasses with respect to number, i.e., have compatible SINGULAR vs. PLURAL attributes. In terms of the computer representation of the analysis, we might express this by requiring in any instance of <ASSERTION> that if the core of the <VERB> is plural (e.g., experiment in Figure 1), then the core of the <SUBJECT> is not singular (e.g., they in Figure 1), and similarly for the case of singular verb, plural subject.[5]

The computable form of thefirst half of this restriction is a sequence of routines executed starting at the ASSERTION node of a putative parse tree: (1) Go to the element VERB; (2)go to the CORE of the VERB; (3) does it have the ATTRIBUTE PLURAL? If yes, result is true; if no, result is false. (4) (Again, starting at ASSERTION), go to the element SUBJECT; (5) go to the CORE of the SUBJECT; (6) is it the case that the CORE of the SUBJECT does NOT have the ATTRIBUTE SINGULAR? If yes, result is true; if no, result is false. (7) Overall test, Logical Implication: Result is true unless (3) is true and (6) is false. This routine sequence is represented in the computer as: IN ASSERTION: IMPLY (A,B). A = STARTAT (VERB), CORE ROUTINE, ATTRIBUTE (PLURAL). B = STARTAT (SUBJECT), CORE ROUTINE, NOT (ATTRIBUTE (SINGULAR)).[6]

This example shows how a computable restriction of the English grammar is composed of a sequence of routines. There is a vocabulary of about thirty basic routines from which all the restrictions of the grammar are composed. Clearly, not all sequences of routines (and not all choice of arguments for particular routines) will constitute wellformed (i.e., executable) restrictions. Equally clear from the limited nature of the elements is the fact that the particular sequences of routines (with their appropriate arguments) which constitute wellformed restrictions can be specified in a fixed format, for example as a set of BNF definitions, as is common in some syntax-driven compilers for symbolic programming languages.[7] This means that every English grammar restriction in its coded form is a text written in a highly restricted, in fact context-free, language which has its own (BNF) grammar.

Now let us consider translations of these call sequences (or routine-sequences) into other forms. We can assign an arbitrary symbol or word or phrase to each routine so long as the assignment is unique, and we can change the order in

which the routine-names appear within a given wellformed routine-sequence so long as the proper order can be recovered before the routines are executed. We do this now in such a way that every restriction has the form of an English sentence. The parts of the restriction sentence are in 1:1 correspondence with the routines which comprise the computable form of the restriction. An example of a wellformed restriction sentence is the one which corresponds to the sequence of routines displayed above: IN THE ASSERTION, IF THE CORE OF THE VERB IS PLURAL, THEN THE CORE OF THE SUBJECT IS NOT SINGULAR.

Consider the above-mentioned BNF specification (i.e., grammar) of the wellformed routine-sequences. When we have the ability to translate these sequences into a subset of English sentences we can construct a grammar of this subset. This grammar would admit as wellformed sentences of the subset only those sentences which correspond to wellformed sequences of routines (i.e., to sequences which constitute executable restrictions). Every such sentence corresponds to, and expresses the intent of, a possible restriction of the grammar.

We have therefore constructed two objects: (1) a subset of English sentences which has been shown by its correspondence with a working computer grammar of English to suffice for stating a grammar of English; (2) a grammar which defines a sublanguage of English[6] in which the above subset (ie., the above grammar of English) is written. The grammar of the sublanguage differs in important respects from the grammar of the language as a whole. First, and most striking, the grammar of the sublanguage is not co-extensive with the grammar of English. This sublanguage of English, in which a grammar of English is written, is itself context-free, whereas English is not. This means, perhaps not surprisingly, that the full power of English is not needed to specify English. It also shows specifically in what way two levels of context-free specification suffice to describe a natural language.

A second important point about the grammar of this sublanguage is related to the fact that the sublanguage is concerned with a specific subject matter and carries the informational burden of that area of knowledge. The categories of the grammar of the sublanguage reveal essential concepts and relations of the specific subject matter. Thus, the 30 or so relations of the restriction language and their possible combinations constitute a summary, or the materials of a theory, of computable string grammar. For example, the permitted subjects of a restriction sentence, shown in Figure 2, along with a small number of predicates which are illustrated in the accompanying examples, are the essential relations of a computable string grammar.

FIGURE 2*

Permitted Subject of Restriction Statement	Example of Restriction Statement
THE ELEMENT X [OF Y]	THE ELEMENT SUBJECT OF ASSERTION HAS THE CORE N.
THE COELEMENT X	THE COELEMENT VERB HAS THE CORE TV.
THE CORE [OF X]	THE CORE OF THE COELEMENT OBJECT IS N:SINGULAR.
THE HOST [OF X]	THE HOST OF PN IS N.
THE HOST-STRING [OF X]	THE HOST-STRING OF SUB 1 IS ASSERTION.
THE IMMEDIATE STRING [OF X]	THE IMMEDIATE STRING OF SUBJECT IS ASSERTION.
THE LEFT ADJUNCT [OF X]	THE LEFT ADJUNCT OF N IS EMPTY.
THE RIGHT ADJUNCT [OF X]	THE RIGHT ADJUNCT OF TV HAS THE CORE D.
THE (PRE/POST) X SENTENCE ADJUNCT [OF Y]	THE PRE-OBJECT SENTENCE ADJUNCT OF THE IMMEDIATE STRING IS OF THE TYPE CENTERLIKE-STRING.
THE VALUE [OF X]	THE VALUE OF THE CENTER IS ASSERTION OR QUESTION.
THE NTH ELEMENT [OF X]	THE SECOND ELEMENT OF PN HAS THE CORE PRO.
THE (ULTIMATE SUBJECT/ DEEPEST VERB)	THE ULTIMATE SUBJECT IS OCCURRING IN A CENTERLIKE-STRING.
THE PRESENT (POSITION/ ELEMENT/OPTION/STRING/ SET/ENTITY)	THE PRESENT ELEMENT IS CONJOINED BY AN AND-STRING.
THE PREVIOUS/FOLLOWING ELEMENT	THE PREVIOUS ELEMENT IS STRING INITIAL.
<*NODE>	THE OBJECT HAS THE VALUE NULLOBJ.
<*ATTRIBUTE>	ASENTI HAS THE ATTRIBUTE ATHAT.
<*REG>	X1 IS SINGULAR.

*A/B means A or B; [A] means A is optional; X,Y stand for appropriate arguments not further specified here. <*NODE> stands for names of nodes, and <*ATTRIBUTE> for names of attributes respectively. <*REG> stands for the contents of a register location. SUB1 is the name of a subordinate conjunction string in the set of sentence adjuncts. D stands for adverb.

The experience of constructing a sublanguage grammar for a subset of English concerned with a particular subject matter opens the way to considering other subsets of English, similarly concentrated on a particular subject matter, such as a small subfield of science. A body of research articles in a subfield of pharmacology (digitalis) is currently under analysis with the aim of constructing a sublanguage grammar for this area. Small vocabulary classes (e.g., ions vs. drug names) are found to occupy distinctive syntactic positions relative to other classes; elementary sentence forms involving these different small classes are found to behave differently with respect to transformational operators which also divide into small, semantically significant, syntactic classes (e.g., influence, results in, vs. increase, decrease).

Results of applying the sublanguage method in this more complex area of a science sublanguage (as opposed to the simpler domain of linguistic string analysis) are still preliminary, but indicate that the sublanguage method is a promising device for overcoming the apparent -- but perhaps not real -- gap between syntactic analysis and the informational burden carried by language in specific subject matter domains.[9]

NOTES

1. The New York University String Program, is described in N. Sager, The String Parser for Scientific Literature, Natural Language Processing, R. Rustin, ed., Prentice-Hall (in press). Early documentation for the system, including the parsing program, the computer grammar of English, and outputs produced will be found in N. Sager, "Syntactic Analysis of Natural Language," Advances in Computers, ed. F. Alt and M. Rubinoff, 8 (1967), 153-188; String Program Reports, Nos. 1-5 (New York University: N.Y.U. String Project, 1965-69). The grammar in the symbolic form referred to in this paper will appear in the series Mathematics and Its Applications, published by Gordon and Breach.

2. It should be noted that while the form of the grammar rules is context-free, the content is that of a string grammar and not an immediate constituent or phrase structure grammar, the type most often associated with context-free rules for a natural language grammar. For string grammar, see the references above and Z. S. Harris, String Analysis of Sentence Structure, Papers on Formal Linguistics, No. 1 (Mouton and Co., 1962).

3. Backus Normal Form (BNF) is a particular style of syntax specification due to John Backus, used widely in the field of programming languages and compilers (cf. reference in fn. 7). An example of a BNF definition used in the English grammar is: <ASSERTION> :: = <SA> <SUBJECT> <SA> <VERB> <SA> <OBJECT> <RV> <SA>. Here each item in < > is a syntactic type which is defined in the grammar. The atomic or terminal symbols X are written <*X>.

4. The pronoun class has as its right adjuncts a restricted subset of the right adjuncts of the noun, <RN>, so that <LPROR> consists of the sequence <LPRO> <*PRO> <RN>.

5. The reason for the negative formulation (e.g., not SINGULAR) has to do with the method of assigning subclasses to the lexical entities.

6. Routines separated by a comma are executed in sequence. Arguments of routines are written in parentheses. This sequence is a copy of the printout of the routine calls for this restriction, except for the use of "A" and "B" here, and the suppression of occurrences of "EXECUTE."

7. John Cocke and J.T. Schwartz, Programming Languages and Their Compilers (New York University: Courant Institute of Mathematical Sciences, 1969, revised 1970.

8. This subset constitutes a sublanguage since it is closed under at least one operation, e.g., and. Cf. Z.S. Harris, Mathematical Structures of Language, Interscience Tracts in Pure and Applied Mathematics, No. 21 (New York: John Wiley and Sons, 1968), Section 5.9. The sublanguage includes sentences which correspond to the BNF definitions in the English grammar, e.g., the definition of <ASSERTION> can be read: THE SYNTACTIC TYPE ASSERTION CONSISTS OF THE SYNTACTIC TYPE SA FOLLOWED BY THE SYNTACTIC TYPE SUBJECT FOLLOWED BY THE SYNTACTIC TYPE SA FOLLOWED BY THE SYNTACTIC TYPE VERB FOLLOWED BY THE SYNTACTIC TYPE SA FOLLOWED BY THE SYNTACTIC TYPE OBJECT FOLLOWED BY THE SYNTACTIC TYPE RV FOLLOWED BY THE SYNTACTIC TYPE SA. This sublanguage is called the restriction language.

9. This research was supported in part by Research Grant No. LM00720-01, from the National Library of Medicine, National Institutes of Health, DHEW.