FORMAL SPECIFICATION OF NATURAL LANGUAGE SYNTAX
USING TWO-LEVEL GRAMMAR

Barrett R. Bryant
Dale Johnson
BhalanJanitha Edupuganty
Department of Computer and Information Sciences
The University of Alabama at Birmingham
Birmingham, Alabama, U. S. A. 35294

ABSTRACT

The two-level grammar is investigated as a notation for giving formal specification of the context-free and context-sensitive aspects of natural language syntax. In this paper, a large class of English declarative sentences, including post-noun-modification by relative clauses, is formalized using a two-level grammar. The principal advantages of two-level grammar are: 1) it is very easy to understand and may be used to give a formal description using a structured form of natural language; 2) it is formal with many mathematical properties; and 3) it is directly implementable by interpretation. The significance of the latter fact is that once we have written a two-level grammar for natural language syntax, we can derive a parser automatically without writing any additional specialized computer programs. Because of the ease with which two-level grammars may express logic and their Turing completeness, we can derive a parser automatically without writing any additional specialized computer programs. Because of the ease with which two-level grammars may express logic and their Turing completeness, we can derive a parser automatically without writing any additional specialized computer programs.

1. INTRODUCTION

Formal specifications of natural language syntax should serve as a standard definition for the syntax of the subject language. The specification must be complete, concise, consistent, precise, unambiguous, standard definition for the syntax of the subject language. The specification must be complete, concise, consistent, precise, unambiguous, understandable, and useful to language scholars, users, and implementors. The mctalanguage we propose is the metalanguage for specifying the syntax and semantics of programming languages [2] but semantics and knowledge representation. Furthermore the specification should be mathematically rigorous to the degree that an implementation of the language can be automatically derived from the specification [10]. Unfortunately, many of these aims are difficult to accomplish primarily because of the dynamic and informal nature of natural language. Formal specification is still a worthy goal to the degree allowed by present knowl" edge about natural language and in this paper we propose a metalanguage for specifying both syntax and semantics of natural language that has potential for satisfying these goals. The metalanguage we propose is the two-level grammar [10] (also called W-grammars and 2lg). Two-level grammars have been used extensively for specifying the syntax and semantics of programming languages [2] but their use in specifying natural language was first introduced by the authors [7, 8, 9].

Existing formal specification methods for natural language syntax take many forms. Of these, some of the more common are augmented transition network grammars [18], transformational grammars [1], and generalized phrase-structure grammars [5]. These methods and others are also surveyed in [17]. The degree to which any formal specification method satisfies the above stated goals is sometimes difficult to evaluate and relies on subjectivity. The authors do not intend to evaluate these existing methods with respect to the requirements of formal specification languages but will instead concentrate on why two-level grammars satisfy the necessary goals in a mathematically rigorous but readable and easy to understand way. In this paper, the two-level grammar metalanguage will be used to define a large classification of English declarative sentences, extending the work described in [8] and [9]. We will emphasize the method of using two-level grammars for this purpose and the advantages gained rather than any particular characteristics of the given grammar.

2. TWO-LEVEL GRAMMARS

A two-level grammar consists of two separate grammars, the *metaproduction rules* (metarules) and the *hyperrules*. The metarules are generally context-free rules which take the form:

**METANOTION :: hyperrule-1; hyperrule-2; ...; hyperrule-n.**

where METANOTION is the left-hand side "nonterminal" symbol of the production and hyperrule-1, hyperrule-2, ..., hyperrule-n are the n alternatives of the production. Each hyperrule consists of *protonotions* (terminal symbols) and other metanotions. In the case of English, the terminal symbols of the meta-grammar are English words. The meta-grammar itself is used to define the context-free aspects of English. Example metarules are:

**SENTENCE :: DETERMINER NOUN VERB.**

**DETERMINER :: a; an; they; the; those; these; this; that.**

The hyperrules are of the form

**hypertonotion :: hypersonion-1 hypersonion-2 ... hypersonion-n.**

The hypersonions separated by semicolons are distinct production alternatives. Each of these hypersonions may be divided into a sequence of protonotions separated by commas. In a two-level grammar derivation tree, there will be one branch for each element in the sequence. A two-level grammar with either hyperrules having more than one hypersonion or two distinct hypersonions having the same hypertonotion on the production left-hand side is nondeterministic. If each hyperrule has only one hypersonion and all hypertonotions in production left-hand sides are distinct from one another then the type is deterministic.

A hyperrule is actually a production rule "pattern" since each hyperrule can possibly represent an infinite number of production rules in a context-free grammar. This is because each occurrence of a metanotion in the hyperrule represents all sequences of protonotions that can be derived from that metanotion. That is, a hyperrule may be viewed as a set of production rules (called *strict production rules*) in which all metanotions are replaced by the protonotions they derive. The only restriction here is that if there are more than one occurrence of a single metanotion, then each is replaced by the same protonotion sequence in deriving the strict production rules. This is called "consistent substitution". For example, in the hyperrule

**where WORD is WORD :: true.**

both occurrences of the metanotion WORD represent the same protonotion. The set of allowable protonotions in this rule is defined by the metarules for WORD. If these metarules define an infinite number of possible protonotions, then the above hyperrule also represents an infinite number of strict production rules. It is this feature of two-level grammars that allow them to define context-sensitive and recursively enumerable languages [12].

If consistent substitution is not required (or desired) for metanotions with the same root metarule (and name), then these metanotions may be distinguished by subscripts. For example,

**where SENTENCE1 and SENTENCE2 are correct :**

**where SENTENCE1 is correct, where SENTENCE2 is correct.**

In this hyperrule, SENTENCE1 and SENTENCE2 are defined by the same metanotions (and root metanotion SENTENCE) but need not have the same instanciations.

Some hyperrules called *predicates* act as conditions which must be satisfied for the derivation to be successful. A predicate begins with the word where or condition and the terminal derivation of the hyperrule is the empty string if the condition is satisfied and will derive a "blind alley" (i.e., not derive any terminal string) if the condition is not satisfied. In the two-level grammar of English presented in this paper, all hyperrules are predicates and serve to perform context checks such as subject-verb agreement, object-verb agreement, and any additional required context checks which cannot be conveniently specified by a context-free grammar (i.e., the metarules).

3. METARULES FOR ENGLISH

The metarules of the two-level grammar for English define the context-free aspects of English syntax. Some lexical items from English can not be easily defined in a formal way (i.e., using context-free rules). These include the nouns, verbs, adjectives, proper names, and titles, given names and surnames for people which are lexical categories containing a large number of elements. The formal specification of these categories would be production rules of the form:

**NOUN :: adware; abacus; ...; zucchini.**

**VERB :: abandon; abate; ...; zoom.**

**ADJECTIVE :: abdominal; abhorrent; ...; zoned.**

**PROPER_NAME :: Aberdeen; Ablay; ...; Zambia.**

**TITLE :: Admiral; Archbishop; ...; Warrant Officer.**

For simplicity we choose to omit more formal specifications of the above categories. A more complete list of words in these categories may be found in [14].
The metarules in our two-level grammar illustrate the specific subset of English grammar defined in this paper. The subset includes declarative sentences with the subject noun premodified and postmodified, including postmodification by relative clauses. The choice of this subset is rather arbitrary, since we have used two-level grammars to define a wide variety of English sentences (e.g. in [7], more extensive modification is allowed and also compound sentences). This subset will serve to illustrate the power of two-level grammars for the purposes of defining English syntax. Because the notation for metarules follows context-free grammar conventions using natural language vocabulary, our meta-grammar is fairly self-explanatory.

The rules of English syntax that have been incorporated into our grammar are based on English grammar rules given in [3], [11], [13], and [19].

We now enumerate the metarules used in our two-level grammar of English. A sentence consists of a noun phrase and a verb phrase. The noun phrase consists of an optional sentence modifier such as a "viewpoint" adverbial and a subject sequence. The subject sequence consists of two main subjects, separated by the coordinator and. The main subjects may be either a list of nouns premodified and postmodified or a proper name premodified by a restrieter.

1. SENTENCE = NOUN_PHRASE VERB_PHRASE PERIOD.
2. NOUN_PHRASE = SENTENCE_MODIFIER SUBJECT_SEQUENCE.
3. SENTENCE_MODIFIER = VIEWPOINT COMMA; EMPTY.
4. VIEWPOINT = artistically; economically; ethically; financially; geographically; linguistically; militarily; morally; personallv; politically; psychologically; theoretically; visually.
5. SUBJECT_SEQUENCE = MAIN_SUBJECT; MAIN_SUBJECT and MAIN_SUBJECT.
6. MAIN_SUBJECT = MODIFIED_NAMED_SUBJECT.
7. MODIFIED_NAMED_SUBJECT = RESTRIETERS NAMED_SUBJECT.
8. NAMED_SUBJECT = PROPER_NAME; GIVEN_NAME.
9. RESTRIETERS = initially; especially; even; just; largely; mostly; primarily; not; every; only; EMPTY.
10. NOUN_HEAD = NOUN; NOUN and NOUN.
11. NOUN_LIST = NOUN; NOUN_LIST COMMA and NOUN.
12. VERB_PHRASE = PREDICATE_SEQUENCE OBJECT_SEQUENCE.
13. PREDICATE_SEQUENCE = AUXILIARY_PHRASE VERB.
14. AUXILIARY_PHRASE = AUXILIARY_ADMEMBER OPTION.
15. AUXILIARY_ADMEMBER OPTION = ACTIVE; PASSIVE.
16. AUXILIARY = AUXILIARY_HAVE AUXILIARY_ADMEMBER OPTION.
17. PASSIVE_ADMEMBER OPTION = AUXILIARY_HAVE AUXILIARY_ADMEMBER OPTION.
18. ACTIVE_AUXILIARY = AUXILIARY_HAVE AUXILIARY_ADMEMBER OPTION.
19. AUXILIARY_HAVE = AUXILIARY_HAVE AUXILIARY_ADMEMBER OPTION.
20. AUXILIARY_BE = AUXILIARY_HAVE AUXILIARY_ADMEMBER OPTION.
21. AUXILIARY_ADMEMBER OPTION = AUXILIARY_HAVE AUXILIARY_ADMEMBER OPTION.
22. AUXILIARY_BE = am; is; are; was; were.
23. AUXILIARY_HAVE = has; have.
24. AUXILIARY_ADMEMBER OPTION = AUXILIARY_HAVE.
25. AUXILIARY_ADMEMBER OPTION = AUXILIARY_ADMEMBER OPTION.

The object sequence of a verb phrase can contain both direct and indirect objects followed by an optional adverbial such as a maximizing adverb or a time adverb. Objects can be either a proper name, possibly modified by the restrieters given above, or a noun expression, possibly premodified and postmodified by restrieters.

26. OBJECT_SEQUENCE = INDIRECT_OBJECT DIRECT_OBJECT.
27. INDIRECT_OBJECT = OBJECT; OBJECT_SEQUENCE.
28. DIRECT_OBJECT = OBJECT.

The nouns in the NOUNSEQUENCE denote the physical composition of items (e.g., "the fisherman's rusted iron responsibility"). Since there are a large number of these, we omit their formal specification here. In our grammar subset we restrict post-noun-modifiers to relative clauses involving people. Many other forms of post-noun-modification are formally specified in [7].

29. POST_NOUN_MODIFICATION = RELATIVE_CLAUSE; EMPTY.
30. RELATIVE_CLAUSE = RELATIVE_PHRASE.
31. RELATIVE_PHRASE = who PREDICATE_SEQUENCE OBJECT_SEQUENCE.
32. Finally, the punctuation in our grammar is given below.

33. PERIOD = .
34. COMMA = ,
35. EMPTY = .

The hyphers for English:

The hyphers of the two-level grammar for English define the context-sensitive aspects of English syntax which can not be specified by the context-free rules of the meta-grammar. Unlike the meta-grammar, the hyphers do not generate any part of the English sentence. They serve only to verify the context-sensitive conditions of the grammar. This is done by using predicates as described earlier. Predicates will derive the empty string if they are satisfied and will derive nonterminal strings of
4. condition SUBJECT_SEQUENCE shows subject-predicate agreement with AUXILIARY,_VERB : 
   condition AUXILIARY,SEQUENCE agrees in person and number with VERB.

5. condition SUBJECT_SEQUENCE shows subject-predicate agreement with AUXILIARY
   condition SUBJECT_SEQUENCE agrees in person and number with AUXILIARY,VERB.

6. condition MODIFIED_NAMED_SUBJECT agreement in person and number with VERB:
   where VERB is singular.

7. condition PRE_NOUN_MODIFICATION NOUN,HEAD POST_NOUN_MODIFICATION
   agrees in person and number with VERB:
   condition NOUN,HEAD agrees in person and number with VERB.

8. condition NOUN_LIST COMMA_OPTION and NOUN
   agrees in person and number with VERB:
   where VERB is plural.

9. condition MAIN_SUBJECT1 and MAIN_SUBJECT2
   agrees in person and number with VERB:
   where VERB is plural.

To satisfy the second condition that the subject of a sentence must be
well-formed, the subject may fall into one of the following categories: 1) if
the subject is a noun (H10), then it is already well-formed by the
metarules; 2) if the subject is modified (H11), then the modifiers must be
correct; and 3) if the subject is a compound subject (H12), then each
component of the compound subject must be well-formed according to
rules 1 and 2.

10. condition MODIFIED_NAMED_SUBJECT is a well-formed subject:
condition DETERMINER PRE_NOUN_MODIFIERS
   NOUN,HEAD POST_NOUN_MODIFICATION
   is a well-formed subject.

conditions DETERMINER PRE_NOUN_MODIFIERS
   NOUN,HEAD is correct in premodification,
condition DETERMINER NOUN,HEAD
   POST_NOUN_MODIFICATION
   is correct in postmodification.

11. condition MAIN_SUBJECT1 and MAIN_SUBJECT2
   is a well-formed subject:
condition MAIN_SUBJECT1 is a well-formed subject,
condition MAIN_SUBJECT2 is a well-formed subject.

Correctness of modification implies that a subject must be correctly
premodified and postmodified. We first give the hyperrule which enforces
correct premodification. Premodification (H13) requires 1) correct
determiner usage (i.e. with respect to singular and plural nouns) and 2)
any premodifying nouns must be singular or "mass" nouns (i.e. nouns
which denote item composition such as aluminum, brass, etc.). A singular
determiner (e.g. a, an, that, etc.) requires a singular noun (H14) but a
"universal" determiner (e.g. the, the, etc.) may be used with singular or
plural nouns (H15). If there are no premodifying nouns, then hyperrule
H16 will apply. A single premodifying noun (H17) may be either singular
or a mass noun. Note that rule H17 is nondeterministic in that there are
two hyperalternatives. The condition is satisfied if either one of those
hyperrules is satisfied. If the premodifying nouns are co-ordinated with
and (H18), then both nouns must be mass nouns (e.g. "the wooden and
iron door" is correct but the "forest and garden path" is not).

12. condition DETERMINER POSSESSIVE NOUN LIST1
   NOUN,SEQUENCE POSSESSIVE NOUN LIST2
   DENOMINAL NOUN NOUN,HEAD
   is correct in premodification:
   condition DETERMINER correctly premodifies NOUN,HEAD,
   condition NOUN,SEQUENCE are singular or mass nouns.

13. condition SINGULAR DETERMINER correctly premodifies NOUN:
   where NOUN is singular.

14. condition UNIVERSAL DETERMINER correctly premodifies NOUN:
   where NOUN is singular.

15. condition NOUN and NOUN are singular or mass nouns:
   condition NOUN is a mass noun, where NOUN is a mass noun.
   Hyperrule H19-H22 define the conditions for postmodification. Any
   postmodification of the subject must be in the form of a relative clause
   which begins with who. This type of relative clause requires a human noun
   and the verb of the relative clause must agree with the modified noun.
   For example, in "The men who fix computers were very helpful," the noun
   men must be a human noun since it is modified by who and the verb fix must
   be compatible with men. This type of relative clauses may be considered as
describing two separate sentences: "The men fix computers." and "The men
   were very helpful." In the hyperrules which verify these conditions, the
   subjmediate described by the relative clause is formed and then checked for
correctness using hyperrule H2 recursively.

16. condition DETERMINER NOUN,HEAD
   POST_NOUN_MODIFICATION
   is correct in postmodification:
   condition POST_NOUN_MODIFICATION
   correctly postmodified DETERMINER NOUN,HEAD.

17. condition EMPTY correctly postmodified DETERMINER NOUN,HEAD:
   condition RELATIVE_CLAUSE correctly postmodified DETERMINER NOUN,HEAD:
   condition NOUN,HEAD is a human noun, condition the verb of RELATIVE_CLAUSE
   agrees with DETERMINER NOUN,HEAD.
sentence. Context-free parsing will eliminate all sentences which do not syntax. Similarly our method of parsing a two-level grammar requires a parser for metarules and a parser for hyperrules. Since the metarules are
5. TWO-LEVEL PARSING
Our method of natural language specification has two-levels: metarules for context-free syntax and hyperrules for context-sensitive syntax. Similarly our method of parsing a two-level grammar requires a parser for metarules and a parser for hyperrules. Since the metarules are context-free, any of the well-known context-free parsing algorithms (e.g. see [17]) may be used to derive a context-free structure of some input sentence. Context-free parsing will eliminate all sentences which do not satisfy the context-free syntax of the language but is unable to eliminate structures which are correct in the context-free sense but incorrect with respect to context-sensitive syntax. The hyperrule parser will further reduce the set of sentences which are considered to be grammatically valid by analyzing the context-free parse tree for context-sensitive violations. The "parser" for the hyperrules is actually an interpreter developed by the authors in [4] which evaluates the hyperrules in much the same way as a programming language interpreter executes programs. The hyperrules are interpreted sequentially in the order that conditions are enumerated in the grammar. Interpretation proceeds by expanding the start notion and applying the hyperrules to all of the branches of the hyperrule derivation tree until all of the predicates are evaluated. As interpretation proceeds, each node of the derivation tree (corresponding to a hyperrule) is expanded by matching it with a hyperrule left-hand side. The right-hand side of the matched hyperrule is then used to create a subtree for that node. Each branch of the tree is evaluated from left to right in a pre-order traversal. The English sentence is syntactically correct if and only if the resulting terminal string derived by the hyperrule tree is the empty string.

The method of writing hyperrules to derive only the empty string greatly simplifies the parsing process. Traditionally (e.g. [9, 10]), two-level grammars use the hyperrules to generate the terminal strings of the language with the metarules being used only to instantiate hyperrules. For example, in our grammar the metanotion SENTENCE is used to generate English sentences which are then input to the hyperrules for analysis. In other two-level grammar styles, however, the components of the sentence would be generated by hyperrules. The result of hyperrules generating terminal strings is that parsing becomes considerably more difficult and is not accomplished without restrictions being placed on hyperrules (e.g. [16]). Our method of interpreting hyperrules places no restrictions, therefore allowing the tag to be more general. The differences in writing styles are explored further in [4].

The hyperrule interpretation algorithm is outlined below: Procedure Evaluate (hyperrule)
1. Find the hyperrule to apply which has the hypernotion as its left-hand side. This rule will be of the form Hyperrule: Hyperrule-l, Hyperrule-2, ..., Hyperrule-n
2. Expand the derivation tree with Hyperrule-l as the root of the current subtree and the branches being Hyperrule-2, Hyperrule-3, ..., Hyperrule-n.
3. Evaluate (hyperrule+l) for l = 1, 2, ..., n.

To explain how this interpreter works, consider the example sentence "Professor White and the students who attend the university gave Mrs. White a present today." This sentence is seen to be correct with respect to context-free syntax and its structural representation is shown in Figure 1. The specific metarules applied are numbered. We will now apply the hyperrules to this sentence to show how context-sensitive conditions are verified. For notational convenience we have italicized the pronouns which correspond to metanotions in the hyperrules. Since the tree will be traversed from left to right we will label the branches (i.e. nodes) using a number (0-8) to denote the level in the tree and a letter (a-z) to indicate left to right ordering.

The root of the hyperrule derivation tree is the sentence itself. Hyperrule H1 will be applied to initiate the verification process. This will be followed by H2 which divides the derivation tree into five separate branches, one for each condition which the sentence must satisfy.

9. Professor White and the students who attend the university gave Mrs. White a present today.

1. condition Professor White and the students who attend the university gave Mrs. White a present today, is a well-formed sentence
2. condition Professor White and the students who attend the university gave Mrs. White a present today, is object-predicate agreement with Mrs. White.
3. condition Professor White and the students who attend the university gave Mrs. White a present today, is subject-predicate agreement with Mrs. White.
4. condition Professor White and the students who attend the university gave Mrs. White a present today, is a well-formed subject.
Hyperrule H12 will be applied to expand branch 2b and decompose the compound subject into its components. Hyperrules H10 and H11 will then analyze each of the two respective sub-subjects for well-formedness.

A condition Professor White and the students who attend the university is a well-formed subject.

A condition Professor White is a well-formed subject.

A condition the students who attend the university is a well-formed subject.

A condition the students who attend the university is correct in premodification.

A condition the students who attend the university is correct in well-formedness.

Proceeding to construct the tree in a left-to-right manner, branch 6c is expanded next using hyperrule H13. Since the determiner is universal and there is no premodifying noun sequence, hyperrules H15 and H16 complete this subtree.

A condition the students is correct in premodification.

A condition the correcty premodifies students.

A condition EMPTY are singular or mass nouns.

The expansion of branch 6d is one of the more interesting aspects of the context-sensitive analysis since it involves a relative clause. The analysis is performed by hyperrules H19, H21, H22 and H27. Note that rule H27 rearranges the relative clause into a new sentence and recursively calls hyperrule H2 to analyze the new sentence.

A condition the students who attend the university is correct in well-formedness.

A condition who attend the university correctly postmodifies the students.

A condition students is a human noun.

A condition where students is a human noun.

A condition the verb of who attend the university agrees with the students.

A condition the students who attend the university is a well-formed sentence.

Instead of expanding branch 7b further, we will resume our example at branch 6c to verify the condition that the original sentence must have object-predicate agreement. Since the object sequence contains an indirect object, direct object and an adverb, hyperrule H20 will be applied next and since the verb gave is ditransitive, object-predicate agreement will be satisfied.

A condition Mrs. White a present today shows object-predicate agreement with gave.

A condition where gave is ditransitive.

A condition Returning to the top-level conditions, we next verify the well-formedness of the verb gave. Since there are no auxiliary verbs, hyperrule H31 is satisfied.

A condition given is a well-formed predicate.

The final condition that the sentence must satisfy is well-formedness of the object. Since the object is a sequence, rule H34 will be applied to branch 2d to decompose the object sequence and analyze the indirect and direct objects individually by rule H33. Rule H33 calls rule H10-H12 recursively. Since Mrs. White is a named subject, hyperrule H10 is satisfied for the indirect object. By applying hyperrules H11, H13, H14, H16, H19 and H20, the direct object a present will also be verified as a well-formed object. The analysis is now complete and the sentence has been determined to be correct through the process of our two-level grammar interpretation method.

6. CONCLUSIONS

We have shown that two-level grammars may be used very elegantly to give a formal specification of English context-free and context-sensitive syntax. In addition to the subset we have defined in this paper, many other types of English declarative sentences have been formally specified using two-level grammars [7]. There seems to be no obstacle to using tlg specifications for any type of natural language syntactic specification.

The principal advantages of the two-level grammar metalanguage are:

1) it is very readable and may be used to give a formal description using a structured form of natural language;
2) it is formal with many well-known mathematical properties; and
3) it is directly implementable by interpretation. The significance of the latter fact is that once we have written a two-level grammar for natural language syntax, we can derive a parser automatically without writing any additional specialized computer programs. The combination of compactness and implementability is unique in grammar theory for natural languages.

To give a complete specification of natural language, semantics and knowledge representation must be specified in addition to syntax. Our future goals are the investigation of two-level grammar for semantic specifications. Because of the ease with which two-level grammars may express logic [9] and their Turing computability [12], we expect that tlg will also be very suitable for these goals.

REFERENCES

[1] Chomsky, N. Syntactic Structures. Mouton Publishers, The Hague, Netherlands, 1957.

[2] Clennon, J. C. and Ugalde, R. C. Grammars for Programming Languages. Elsevier North-Holland, New York, 1977.

[3] Culicover, P. W. Syntax. 2nd ed. Academic Press, New York, 1982.

[4] Edupuganti, B. and Bryant, B. R. "Two-Level Grammars for Automatic Interpretation." Proc. 1985 ACM Annual Conference, 1985, pp. 417-423.

[5] Gazdar, G. and Pullum, G. K. Generalized Phrase Structure Grammar: A Theoretical Synthesis. Indiana University Linguistics Club, Indiana University, Bloomington, Ind., 1984.

[6] Hesse, W. "A Correspondence Between W-Grammars and Formal Systems of Logic and Its Application to Formal Language Description." Comput. Linguist. Comput. Lang. 19 (1979), 18-30.

[7] Johnson, D. Using Two-Level Grammars to Describe the Syntax of English. M. S. Thesis, Department of Computer and Information Sciences, The University of Alabama at Birmingham, 1984.

[8] Johnson, D. and Bryant, B. R. "Using Two-Level Grammars to Describe the Syntax of English." Papers on Computational and Cognitive Science, ed. F. Bastida, Indiana University Linguistics Club, Bloomington, Ind., Aug. 1984, pp. 81-96.

[9] Johnson, D. and Bryant, B. R. "Formal Syntax Methods for Natural Language." Inf. Process. Lett. 19, 3 (Oct. 1984), 135-143.

[10] Kenstowicz, M. "Orthogonal Design and Description of a Formal Language." Technical Report MR 78, Mathematisch Centrum, Amsterdam, 1985.

[11] Quirk, R. et al. A Grammar of Contemporary English. Longman, White Plains, N. Y., 1972.

[12] Sintoff, M. "Existence of Wijngaarden's Syntax for Every Recursively Enumerable Set." Ann. Soc. Sci. Bruxelles 2 (1967), 115-118.

[13] Stageburg, N. C. An Introductory English Grammar. 4th ed. Holt, Rinehart and Winston, New York, 1981.

[14] Webster's Third New International Dictionary, Unabridged. The Great Book of the English Language. Merriam-Webster, Springfield, Mass., 1981.

[15] Wegner, L. M. "On Parsing Two-Level Grammars." Acta Inf. 14 (1980), 175-193.

[16] van Wijngaarden, A. "Orthogonal Design and Description of a Formal Language." Technical Report MR 78, Mathematisch Centrum, Amsterdam, 1985.

[17] Winograd, T. Natural Language as a Cognitive Process. Volume 1. Syntax. Addison-Wesley, Reading, Mass., 1983.

[18] Woods, W. A. "Transition Network Grammar for Natural Language Analysis." Commum. ACM 16 (1973), 393-402.

[19] Zandvoort, R. W. A Handbook of English Grammar. Prentice-Hall, Englewood Cliffs, N. J., 1965.
Figure 1. Meta-Grammar Derivation Tree.

[Diagram of a parse tree for a sentence, showing elements such as noun phrases, verb phrases, object modifiers, and more, with specific labels like "Professor White," "Mrs. White," "the," "today," and "university." Each node in the tree represents a grammatical element such as a subject, modifier, or auxiliary verb, and the tree structure illustrates the syntactic relationships between these elements.]