FORMAL SPECIFICATION OF NATURAL LANGUAGE SYNTAX USING TWO-LEVEL GRAMMAR

Barrett R. Bryant
Dale Johnson
Balanjaninath 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 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. Because of the ease with which two-level grammars may express logic and their Turing computability we expect that they will also be very suitable for future extensions to semantics and knowledge representation.

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, understandable, and useful to language scholars, users, and implementors who wish to develop a parser for the language to run on a computer. 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 knowledge 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* [16] (also called W-grammars and tlgs). 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 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 :: hypernotion-1; hypernotion-2; ... ; hypernotion-n.

where **METANOTION** is the left-hand side "nonterminal" symbol of the production and **hypernotion-1, hypernotion-2, ... hypernotion-n** are the n alternatives of the production right-hand side. Each hypernotion 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; the; these; those; this; that.

The hyperrules are of the form

hypernotion : hyperaltern-1; hyperaltern-2; ... ; hyperaltern-n.

The hyperalternatives separated by semicolons are distinct production alternatives. Each of these hyperalternatives may be divided into a sequence of hypernotions 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 hyperalternative or two distinct hyperrules having the same hypernotion on the production left-hand side is nondeterministic. If each hyperrule has only one hyperalternative and all hypernotions in production left-hand sides are distinct from one another then the tlg 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 metarules (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 metarules (and root metanotion **SENTENCE**) but need not have the same instantiations.

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 :: aardvark; abacus; ...; zucchini.
VERB :: abandon; abate; ...; zoom.
ADJECTIVE :: abdominal; abhorrent; ...; zoned.
PROPER_NAME :: Aberdeen; Abilene; ...; 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 restricter.

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; personally; politically; psychologically; publically; theoretically; visually.
5. SUBJECT_SEQUENCE :: MAIN_SUBJECT; MAIN_SUBJECT and MAIN_SUBJECT.
6. MAIN_SUBJECT :: MODIFIED_NAMED_SUBJECT; PRE_NOUN_MODIFICATION NOUN_HEAD POST_NOUN_MODIFICATION.
7. MODIFIED_NAMED_SUBJECT :: RESTRICTERS NAMED_SUBJECT.
8. NAMED_SUBJECT :: PROPER_NAME; GIVEN_NAME; SURNAME; TITLE SURNAME.
9. RESTRICTERS :: chiefly; especially; even; just; largely; mainly; mostly; primarily; not even; only; EMPTY.
10. NOUN_HEAD :: NOUN; NOUN and NOUN; NOUN_LIST COMMA_OPTION and NOUN.
11. NOUN_LIST :: NOUN_LIST COMMA NOUN; NOUN COMMA NOUN.

The verb phrase consists of a predicate sequence and an object sequence. The predicate sequence consists of an auxiliary sequence (an optional auxiliary adverb such as a focusing or maximizing adverb followed by an active or passive auxiliary verb) and the main verb of the sentence.

12. VERB_PHRASE :: PREDICATE_SEQUENCE OBJECT_SEQUENCE.
13. PREDICATE_SEQUENCE :: AUXILIARY_SEQUENCE VERB.
14. AUXILIARY_SEQUENCE :: AUXILIARY_ADVERB_OPTION; AUXILIARY_ADVERB_OPTION ACTIVE_OR_PASSIVE_AUXILIARY.
15. AUXILIARY_ADVERB_OPTION::AUXILIARY_ADVERB; EMPTY.
16. AUXILIARY_ADVERB :: FOCUSING_ADVERB; MAXIMIZING_ADVERB.
17. FOCUSING_ADVERB :: again; also; as well; at least; equally; especially; even; further; in addition; in particular; just; largely; likewise; mainly; merely; mostly; notably; only; particularly; primarily; principally; purely; purely and simply; similarly; simply; specifically.
18. MAXIMIZING_ADVERB :: absolutely; altogether; completely; entirely; fully; in all respects; perfectly; quite; thoroughly; totally; utterly; very fully; very thoroughly.
19. ACTIVE_OR_PASSIVE_AUXILIARY :: ACTIVE_AUXILIARY; PASSIVE_AUXILIARY.
20. ACTIVE_AUXILIARY :: AUXILIARY_HAVE AUXILIARY_ADVERB_OPTION.
21. PASSIVE_AUXILIARY :: AUXILIARY_BE AUXILIARY_ADVERB_OPTION; AUXILIARY_HAVE AUXILIARY_ADVERB_OPTION been.
22. AUXILIARY_BE :: am; is; were; was.
23. AUXILIARY_HAVE :: have; had; has.
24. AUXILIARY_VERB :: AUXILIARY_BE; AUXILIARY_HAVE.
25. AUXILIARY_TRAILER :: AUXILIARY_ADVERB_OPTION; AUXILIARY_ADVERB_OPTION been.

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 restricters given above, or a noun expression, possibly premodified and postmodified.

26. OBJECT_SEQUENCE :: INDIRECT_OBJECT DIRECT_OBJECT OBJECT_SEQUENCE_ADVERB; DIRECT_OBJECT OBJECT_SEQUENCE_ADVERB.
27. OBJECT_SEQUENCE_ADVERB :: OBJECT_SEQUENCE_ADVERBIAL; EMPTY.
28. OBJECT_SEQUENCE_ADVERBIAL :: MAXIMIZING_ADVERB; TIME_ADVERB.
29. TIME_ADVERB :: again; early; first; last; late; next; now; recently; simultaneously; since; then; today; yesterday.
30. INDIRECT_OBJECT :: OBJECT.
31. DIRECT_OBJECT :: OBJECT.
32. OBJECT :: MODIFIED_NAMED_SUBJECT; PRE_NOUN_MODIFICATION NOUN_HEAD POST_NOUN_MODIFICATION.

We now turn to the pre-noun-modifiers specified in our grammar. The modifier is a determiner optionally followed by a list of possessive nouns, an adjective, a sequence of nouns, another list of possessive nouns and a denominal noun. Examples of this type of construct include "the murderer's empty black pistol" and "a very rich man's thick wallet." For context-sensitive purposes, the determiners are divided into "universal" determiners which may precede both singular and plural nouns and determiners which may only precede singular nouns. Furthermore, a context-free restriction of the pre-noun-modifiers is that there can be at most one list of possessive nouns in a sequence. For convenience we choose to enforce this condition in the hyperrules instead of the metarules.

33. PRE_NOUN_MODIFICATION :: DETERMINER PRE_NOUN_MODIFIERS.
34. PRE_NOUN_MODIFIERS :: EMPTY; POSSESSIVE_NOUN_LIST ADJECTIVE_OPTION NOUN_SEQUENCE POSSESSIVE_NOUN_LIST DENOMINAL_NOUN.
35. DETERMINER :: UNIVERSAL_DETERMINER; SINGULAR_DETERMINER.
36. UNIVERSAL_DETERMINER :: the; some; any; my; your; his; her; its; our; their.
37. SINGULAR_DETERMINER :: either; neither; another; NOT_OPTION NEGATABLE_SINGULAR_DETERMINER.
38. NEGATABLE_SINGULAR_DETERMINER :: a; an; each; every.
39. NOT_OPTION :: not; EMPTY.
40. POSSESSIVE_NOUN_LIST :: EMPTY; POSSESSIVE_NOUN_LIST POSSESSIVE_NOUN.
41. POSSESSIVE_NOUN :: NOUN's; NOUN'.
42. ADJECTIVE_OPTION :: ADJECTIVE; EMPTY.
43. NOUN_SEQUENCE :: NOUN; NOUN and NOUN; EMPTY.

The nouns in the NOUN_SEQUENCE denote the physical composition of items (e.g. "the fisherman's rusted *iron* hook") and thus act as adjectives. Denominal nouns are adjectives which denote some quality of the noun being modified (e.g. "her *social* life" and "his *moral* 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].

44. POST_NOUN_MODIFICATION :: RELATIVE_CLAUSE; EMPTY.
45. RELATIVE_CLAUSE :: who PREDICATE_SEQUENCE OBJECT_SEQUENCE.

Finally, the punctuation in our grammar is given below.

46. PERIOD :: . .
47. COMMA :: , .
48. COMMA_OPTION :: COMMA; EMPTY.
49. EMPTY :: .

4. HYPERRULES FOR ENGLISH

The hyperrules 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 hyperrules 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 useless symbols otherwise. The notion that the hyperrules will not generate any terminal string but instead verify context-sensitive conditions of a terminal string already generated by the context-free metarules is a unique feature of our approach to designing two-level grammars (e.g. in contrast, see [2]). This will greatly simplify parsing two-level grammars as we will see later.

We will define two types of predicates. The first of these will be preceded by the protonotion **condition** and will be given explicitly in the formal grammar. As with the meta-grammar, however, there will be some rules which can not be precisely defined in the formal system. These rules relate to qualities of the unspecified lexical classes (e.g. nouns, verbs, etc.) and will be designated by the protonotion **where**. For example, the hypernotions **where NOUN is singular**, **where VERB is past participle**, and **where NOUN and VERB agree in person and number** can not be precisely defined except by a very large number of formal rules such as those given below:

where aardvark is singular : EMPTY.
where abandoned is past participle : EMPTY.
where Adam and ate agree in person and number : EMPTY.

In the subsequent discussion of hyperrules we will use the notation Hn to denote hyperrule number n. The start hyperrule (H1) of the two-level grammar is:

1. SENTENCE : condition SENTENCE is a well-formed sentence.

This hyperrule has as its start notion an English sentence which is well-formed with respect to the context-free rules of the meta-grammar for metanotion SENTENCE. The next hyperrule (H2) expands the sentence with respect to what conditions must be satisfied. The formalization of these is self-explanatory.

2. condition SENTENCE_MODIFIER SUBJECT_SEQUENCE
AUXILIARY_SEQUENCE VERB OBJECT_SEQUENCE
PERIOD is a well-formed sentence :
condition SUBJECT_SEQUENCE shows subject-predicate
agreement with AUXILIARY_SEQUENCE VERB,
condition SUBJECT_SEQUENCE is a well-formed subject,
condition OBJECT_SEQUENCE
shows object-predicate agreement with VERB,
condition AUXILIARY_SEQUENCE VERB
is a well-formed predicate,
condition OBJECT_SEQUENCE is a well-formed object.

The first condition is that the subject sequence must agree with the predicates specified by the auxiliary sequence and verb. In our grammar, agreement means that the subject and the subject-verb must agree in person and number. There are two possibilities for subject-verbs: 1) the auxiliary sequence is empty (H3) in which case the main verb must be consistent with the subject, and 2) the auxiliary sequence is non-empty (H4) in which case it is the auxiliary verb which must be consistent with the subject. Subjects may be in one of three forms: 1) the subject is a proper name (H5), possibly modified by a restricter (e.g. "*even* Mr. Smith" or "*primarily* Mrs. Jones"), and therefore requires a singular verb; 2) the subject is a single subject (H6-H7) in which case it need only agree with the subject-verb; or 3) the subject may be a compound subject co-ordinated with *and* (H8-H9), in which case it requires a plural verb (e.g. "John *and* Bill *are* here.").

3. condition SUBJECT_SEQUENCE
shows subject-predicate agreement with VERB :
condition SUBJECT_SEQUENCE agrees in person and number
with VERB.

4. condition SUBJECT_SEQUENCE
shows subject-predicate agreement
with AUXILIARY_ADVERB_OPTION AUXILIARY_VERB
AUXILIARY_TRAILER VERB :
condition SUBJECT_SEQUENCE agrees in person and number
with AUXILIARY_VERB.

5. condition MODIFIED_NAMED_SUBJECT
agrees in person and number with VERB :
where VERB is singular.

6. 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.

7. condition NOUN agrees in person and number with VERB :
where NOUN and VERB agree in person and number.

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 name (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 :
EMPTY.

11. condition DETERMINER PRE_NOUN_MODIFIERS
NOUN_HEAD POST_NOUN_MODIFICATION
is a well-formed subject :
condition DETERMINER PRE_NOUN_MODIFIERS
NOUN_HEAD is correct in premodification,
condition DETERMINER NOUN_HEAD
POST_NOUN_MODIFICATION
is correct in postmodification.

12. 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 hyperrules which enforce 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, each,* etc.) requires a singular noun (H14) but a "universal" determiner (e.g. *some, 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 these 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).

13. 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.

14. condition SINGULAR_DETERMINER correctly premodifies NOUN:
where NOUN is singular.

15. condition UNIVERSAL_DETERMINER
correctly premodifies NOUN_HEAD : EMPTY.

16. condition EMPTY are singular or mass nouns : EMPTY.

17. condition NOUN are singular or mass nouns :
where NOUN is singular; where NOUN is a mass noun.

18. condition NOUN1 and NOUN2 are singular or mass nouns :
where NOUN1 is a mass noun, where NOUN2 is a mass noun.

Hyperrules H19-H27 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 clause 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 sub-sentence described by the relative clause is formed and then checked for correctness using hyperrule H2 recursively.

19. condition DETERMINER NOUN_HEAD
POST_NOUN_MODIFICATION
is correct in postmodification :
condition POST_NOUN_MODIFICATION
correctly postmodifies DETERMINER NOUN_HEAD.

20. condition EMPTY correctly postmodifies
DETERMINER NOUN_HEAD : EMPTY.

21. condition RELATIVE_CLAUSE correctly postmodifies
DETERMINER NOUN_HEAD :
condition NOUN_HEAD is a human noun,
condition the verb of RELATIVE_CLAUSE
agrees with DETERMINER NOUN_HEAD.

22. condition NOUN is a human noun : where NOUN is a human noun.
23. condition NOUN1 and NOUN2 is a human noun :
where NOUN1 is a human noun,
where NOUN2 is a human noun.
24. condition NOUN_LIST COMMA_OPTION and NOUN
is a human noun :
condition NOUN_LIST is a human noun,
where NOUN is a human noun.
25. condition NOUN1 COMMA NOUN2 is a human noun :
where NOUN1 is a human noun,
where NOUN2 is a human noun.
26. condition NOUN_LIST COMMA NOUN is a human noun :
condition NOUN_LIST is a human noun,
where NOUN is a human noun.
27. condition the verb of
who PREDICATE_SEQUENCE OBJECT_SEQUENCE
agrees with DETERMINER NOUN_HEAD :
condition DETERMINER NOUN_HEAD
PREDICATE_SEQUENCE OBJECT_SEQUENCE PERIOD
is a well-formed sentence.

The third condition that the English sentences defined by our grammar must satisfy is that the predicate (verb) and objects should agree. The type of verb must correspond to the number of objects in the sentence: if the verb is intransitive, then no objects are allowed except for adverbs (H28); if the verb is transitive, then a direct object is required (H29); and if the verb is ditransitive, then both a direct and an indirect object are required (H30).

28. condition OBJECT_SEQUENCE_ADVERB
shows object-predicate agreement with VERB :
where VERB is intransitive.
29. condition DIRECT_OBJECT OBJECT_SEQUENCE_ADVERB
shows object-predicate agreement with VERB :
where VERB is transitive.
30. condition INDIRECT_OBJECT DIRECT_OBJECT
OBJECT_SEQUENCE_ADVERB
shows object-predicate agreement with VERB :
where VERB is ditransitive.

The fourth condition for a well-formed sentence is that the auxiliary adverbs and main verb are in correct grammatical sequence. If there are no auxiliary verbs (H31), then the auxiliary sequence is correct according to the meta-grammar. If auxiliary verbs are present then the verb must be a past participle (H32).

31. condition AUXILIARY_ADVERB_OPTION VERB
is a well-formed predicate : EMPTY.
32. condition AUXILIARY_ADVERB_OPTION
ACTIVE_OR_PASSIVE_AUXILIARY VERB
is a well-formed predicate :
where VERB is a past participle.

The fifth and final condition which must be satisfied is for the object of the sentence to be well-formed. A simple object (H33) must satisfy the same conditions as a subject and hyperrules H10-H12 will apply recursively. An object sequence (H34) is well-formed if the indirect and direct objects are well-formed.

33. condition OBJECT OBJECT_SEQUENCE_ADVERB
is a well-formed object :
condition OBJECT is a well-formed subject.
34. condition INDIRECT_OBJECT DIRECT_OBJECT
OBJECT_SEQUENCE_ADVERB is a well-formed object :
condition INDIRECT_OBJECT is a well-formed object,
condition DIRECT_OBJECT is a well-formed object.

It can be seen that the above set of hyperrules is relatively concise and the conditions being described are readily understandable. We claim that the other goals of consistency, precision (for our subset of English), and unambiguity are also achieved. In the next section it will be shown how this specification may be implemented automatically.

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 hypernotion) 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. [2, 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 also 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. [15]). Our method of interpreting hyperrules places no restrictions, therefore allowing the tlg to be more general. The differences in writing styles are explored further in [4].

The hyperrule interpretation algorithm is outlined below:

```
Procedure Evaluate (hypernotion)
1. Find the hyperrule to apply which has the hypernotion as its left-
   hand side. This rule will be of the form:
      hypernotion : hypernotion-1, hypernotion-2, ..., hypernotion-n.
2. Expand the derivation tree with hypernotion as the root of the
   current subtree and the branches being hypernotion-1, hypernotion-2,
   ..., hypernotion-n.
3. Evaluate (hypernotion-i) for i = 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 the context-sensitive conditions are verified. For notational convenience we have italicized the protonotions 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-e) 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.

0 • *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
2a • condition *Professor White and the students who attend the university* shows subject-predicate agreement with *gave*
2b • condition *Professor White and the students who attend the university* is a well-formed subject
2c • condition *a present today* shows object-predicate agreement with *gave*
2d • condition *gave* is a well-formed predicate
2e • condition *a present today* is a well-formed object

To expand branch 2a and check the first condition, hyperrule H3 (no auxiliary verbs) is applied. Since the subject is compound, rule H9 will be applied, requiring the verb to be plural. The "library" predicate will verify the plurality of *gave*.

2a • condition *Professor White and the students who attend the university* shows subject-predicate agreement with *gave*
3a • condition *Professor White* and *the students who attend the university* agrees in person and number with *gave*
4a • where *gave* is plural
5a •

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.

2b • condition *Professor White* and *the students who attend the university* is a well-formed subject
3b • condition *Professor White* is a well-formed subject
4b •
3c • condition *the students who attend the university* is a well-formed subject
4c • condition *the students* is correct in premodification
4d • condition *the students who attend the university* is correct in postmodification

Proceeding to construct the tree in a left-to-right manner, branch 4c 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.

4c • condition *the students* is correct in premodification
5b • condition *the* correctly premodifies *students*
6a •
5c • condition EMPTY are singular or mass nouns
6b •

The expansion of branch 4d 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.

4d • condition *the students who attend the university* is correct in postmodification
5d • condition *who attend the university* correctly postmodifies *the students*
6c • condition *students* is a human noun
7a • where *students* is a human noun
8a •
6d • condition the verb of who *attend the university* agrees with *the students*
7b • condition *the students attend the university.* is a well-formed sentence

Instead of expanding branch 7b further, we will resume our example at branch 2c 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 H30 will be applied next and since the verb *gave* is ditransitive, object-predicate agreement will be satisfied.

2c • condition *Mrs. White a present today* shows object-predicate agreement with *gave*
3d • where *gave* is ditransitive
4e •

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.

2d • condition *gave* is a well-formed predicate
3e •

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 2e to decompose the object sequence and analyze the indirect and direct objects individually by rule H33. Rule H33 calls rules 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 readability 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 specification. Because of the ease with which two-level grammars may express logic [6] and their Turing computability [12], we expect that tlgs will also be very suitable for these goals.

REFERENCES

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

[2] Cleaveland, J. C. and Uzgalis, 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] Edupuganty, 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 Synopsis.* Indiana University Linguistics Club, Indiana University, Bloomington, Ind., 1982.

[6] Hesse, W. "A Correspondence Between W-Grammars and Formal Systems of Logic and Its Application to Formal Language Description." *Comput. Linguist. Comput. Lang. 13* (1979), 19-30.

[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. E. Battistella. Indiana University Linguistics Club, Bloomington, Ind., Aug. 1984, pp. 61-86.

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

[10] Pagan, F. G. *Formal Specification of Programming Languages: A Panoramic Primer.* Prentice-Hall, Englewood Cliffs, N. J., 1981.

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

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

[13] Stageberg, 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 Library of the English Language.* Merriam-Webster, Springfield, Mass., 1981.

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

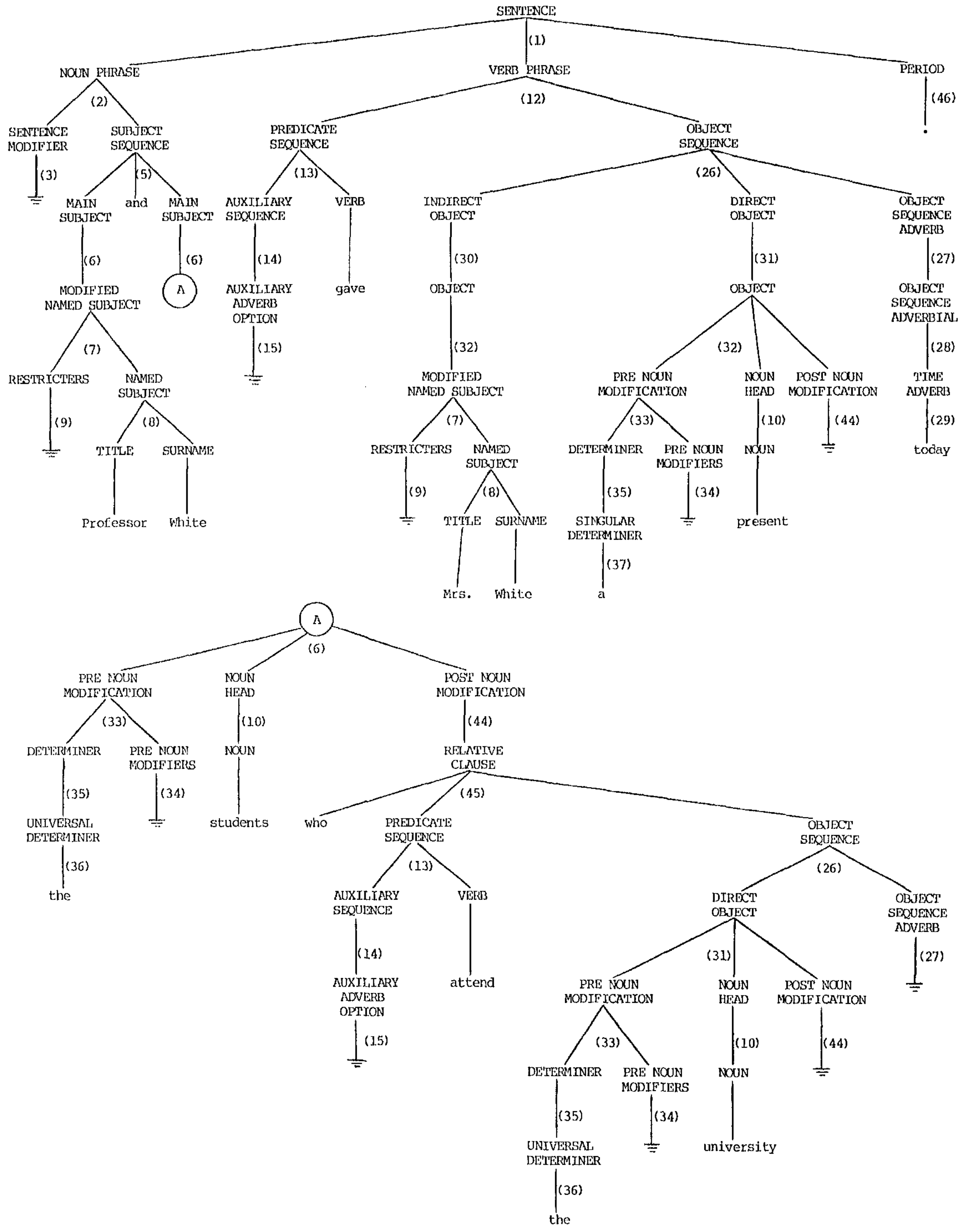
[16] van Wijngaarden, A. "Orthogonal Design and Description of a Formal Language." Technical Report MR 76, Mathematisch Centrum, Amsterdam, 1965.

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

[18] Woods, W. A. "Transition Network Grammar for Natural Language Analysis." *Commun. ACM 13* (1970), 591-602.

[19] Zandvoort, R. W. *A Handbook of English Grammar.* Prentice-Hall, Englewood Cliffs, N. J., 1965.

Figure 1. Meta-Grammar Derivation Tree.


SENTENCE
(1)
NOUN PHRASE
VERB PHRASE
PERIOD
(2)
(12)
(46)
SENTENCE MODIFIER
SUBJECT SEQUENCE
PREDICATE SEQUENCE
OBJECT SEQUENCE
.
(3)
(5)
(13)
(26)
MAIN SUBJECT
and
MAIN SUBJECT
AUXILIARY SEQUENCE
VERB
INDIRECT OBJECT
DIRECT OBJECT
OBJECT SEQUENCE ADVERB
(6)
(6)
(14)
(30)
(31)
(27)
MODIFIED NAMED SUBJECT
A
AUXILIARY ADVERB OPTION
gave
OBJECT
OBJECT
OBJECT SEQUENCE ADVERBIAL
(7)
(15)
(32)
(32)
(28)
RESTRICTERS
NAMED SUBJECT
MODIFIED NAMED SUBJECT
PRE NOUN MODIFICATION
NOUN HEAD
POST NOUN MODIFICATION
TIME ADVERB
(9)
(8)
(7)
(33)
(10)
(44)
(29)
TITLE
SURNAME
RESTRICTERS
NAMED SUBJECT
DETERMINER
PRE NOUN MODIFIERS
NOUN
today
Professor
White
(9)
(8)
(35)
(34)
TITLE
SURNAME
SINGULAR DETERMINER
present
(37)
Mrs.
White
a
A
(6)
PRE NOUN MODIFICATION
NOUN HEAD
POST NOUN MODIFICATION
(33)
(10)
(44)
DETERMINER
PRE NOUN MODIFIERS
NOUN
RELATIVE CLAUSE
(35)
(34)
(45)
UNIVERSAL DETERMINER
students
who
PREDICATE SEQUENCE
OBJECT SEQUENCE
(36)
(13)
(26)
the
AUXILIARY SEQUENCE
VERB
DIRECT OBJECT
OBJECT SEQUENCE ADVERB
(14)
(31)
(27)
AUXILIARY ADVERB OPTION
attend
PRE NOUN MODIFICATION
NOUN HEAD
POST NOUN MODIFICATION
(15)
(33)
(10)
(44)
DETERMINER
PRE NOUN MODIFIERS
NOUN
(35)
(34)
UNIVERSAL DETERMINER
university
(36)
the