A NUMERICAL SYMBOLISM FOR TRANSFORMATIONAL SYNTAX

以數字表示的變換律造句法

By

CHAO-HUI TUNG

董 昭 輝

PREFACE

This monograph was first written in the summer of 1964 at the University of Texas. Thus many of its contentions have probably been rendered pointless by subsequent development as reflected in Chomsky's new exposition of the transformational theory (Noam Chomsky: *Aspects of the Theory ofe Syntax*. Cambridge, Massachusetts: MIT Press, 1965).

A transformational syntax, according to the current view, consists of a *base component* producing the *deep structure*, on which the semantic rules operate, and a *transformational component* producing the *surface structure*, on which the phonological rules ope rate. In other words, meaning is no longer regarded as something to be totally cut away from grammar.

I have nevertheless decided to publish the monograph, with some deletions and minor changes, in the belief that, despite the many irrelevancies to the current view, it will still contribute to the general dedate through the previously unattempted approach it introduces.

Taipei, Taiwan

February, 1966

CHAPTER I

INTRODUCTION

The system of grammatical description presented in this monograph is an accidental outgrowth from an unfulfilled study to have been entitled "The Syntactic Subclasses of English Verbs." The purpose was to investigate the syntactic distribution of English "verbs" thoroughly so that the resulting classification might be really useful to teachers as well as students of English as a second language.

I wanted to exclude from the study words like *will, can, might* and so forth, but was not sure whether the exclusion would be justifiable. So I searched for a definition of the term "verb" which, I hoped, would tell me definitely whether or not I should exclude those words. Nowhere could I find a satisfactory answer. Although the structuralists are supposedly agreed on how to define parts of speech (not by meaning but by the criteria of morphological and syntactic characteristics alone), I found that some of them would expect me to call words like *can* and *may* verbs while others would not. The result was equally disappointng when I turned to transformational grammar. For here, as we know, an item labeled one way by one transformational grammarian is almost certain to be labeled another way by a second, and the difference often seems just a matter of taste.

I was disturbed. Why cannot one linguist's verb or adiective be completely equivalent to that of another, when they are talking about one and the same language? On the other hand, why is there no difficulty with terms like voiceless dental stop or voiced velar fricative? The question led to the following conclusions.

Syntactic classes are incommensurable.

While it is possible to have universal elements in phonetics (as the development of the past half century has fully demonstrated) and presumably also in semantics (if it is true that all human beings live in the same physical world and if the operation called translation really makes sense), in syntax there can be no universal sets that may in any sense be said to be shared by any two languages and may thus justifiably be represented by any common names comparable to names like velar stop, implosive, etc. A *syntactic* class, as a set with peculiar *syntactic* properties, can be defined only in terms of the syntax of a given language—that is in terms of the relations of position, of co-occ urrence, of set membership, etc., with other syntactic classes in the same language. Thus there are no universal syntactic classes any more than there can be universal phonemes.

Names like noun and verb are semantic names.

Since syntactic classes of any two languages cannot be identical in any of the senses in which phones are said to be identical in phonetics, it follows that the set of terms commonly used today as their names in the description of all languages, as if they could be identical, must be based on nothing more than *similarities* alone. What kind of similarities? First, they could possibly be syntactic similarities. For relations of position, co-occurrence, set membership, etc., may be used to compare and liken the syntactic classes of two different languages in a very general way, resulting in very broad classifications as reflected in terms such as stem, prefix, suffix, and so forth. To the extent that they reflect the above-mentioned relations, these terms are truly syntactic terms. But the basis used could not conceivably be similarities in these relations alone when two kinds of stems, for example, are spoken of as noun stem or verb stem. For it is now a well-known fact that what are called by an identical name such as *noun* or *verb* in two different languages usually show radically different behaviors in terms of position, co-occurrence, set membership, etc., if the two languages are really different. It follows, then, that something other than purely syntactic similarities is used as the basis, and the only other kind of similarities conceivable are *semantic* similarities. That is to say: when we give the same name *verb* to two syntactic classes of two different languages, say English and Chinese, on the basis of some parallelism, the parallelism is not so much parallelism in syntactic characteristics as parallelism in semantic characteristics. Only if we accept this argument, indeed, is it possible to explain why one linguist's adjectival is sometimes another's nominal or adverbial. [1]

Names like noun and verb are just convenient loose labels.

Why are semantic names used in syntactic description? The reason is this.

[1]This footnote is to be found on p.4.

Every syntactic class definitely exhibits semantic characteristics peculiar to itself, just as every phoneme has peculiar phonetic characteristics. While we have this situation on the one hand, we need names for all syntactic classes and syntactic units on the other, so that we may be able to talk about them. Consequently, just as we refer, loosely, to the English phoneme /k/, and the Mandarin Chinese phoneme represented by the same symbol, both by the phonetic name "voiceless velar stop," so, in the absence of better systems of designation, we simply assign to every syntactic class of any language a name descriptive of its semantic characteristics, such as ***verb, abverbial of time,*** and so forth.

Names like noun and verb are inadequate for grammar.

Because of their inherent lack of precision, terms like ***noun*** and ***verb,*** which we have inherited from our predecessors, are not fully suited for scientific purposes. Because they are essentially indicative of semantic rather than syntactic characteristics,[2] they also tend to blur the boundary between meaning and structure, which we want to separate completely in order to have accurate pictures of each. For a generative grammar, in particular, they are useless in that, while conveying information irrelevant to sentence generation, they are no more effective in simplifying the grammar than meaningless and arbitrary digits or alphabetical symbols, which at least do not lead us to unnecessary confusion by giving irrelevant information.

On the basis of these conclusions, I cast about for ways of describing syntax, not by names like ***noun*** and ***verb,*** or NP, VP, etc., but by designations which would carry the kind of information we really need in syntax. The

---

[1]This does not contradict the doctrine of modern structuralism that parts of speech are definable without reference to meaning, as demonstrated by Gleason's "The iggle squiggs trazed wombly in the harlish goop." For, as ***syntactic*** (formal, structural) classes, they are certainly so definable, and can only be so defined, ***within the framework of a given language.*** Otherwise they would not constitute ***syntactic*** classes at all.

[2] Equally indicative of meaning, of course, are the names of "slots" and "filler classes," like ***time slot, possessor,*** etc., used in tagmemics.

result is the numerical symbolism described in this paper, which replaces symbols like NP, VP, Nom, Vs, etc., used in transformation syntax, with numbers. These numbers are not arbitrarily given, as was the case with the numbers in a similar system used in Fries, 1952,[3] but are systematically organized so that each designation carries information about relations of set membership, and to a limited extent, about relations of position.

CHAPTER II

WORKING PRINCIPLES

The proposed system, which concerns syntax only, leaves intact the fundamental principle of transformational theory: that a transformational level is indispensable for any adequate grammar. In fact, it is an attempt to fur ther develop that principle because it aims at simpler and more explicit trans-formational statements.

The system differs primarily from the conventional system in the following four points:

1) Instead of *semantic labels*, it uses *numbers*, each number being essentially a discrete sequence of systematically combined digits. A string is thus no longer a string of units represented by semantic labels, but a string of numbers. Because of this basic difference, we shall refer through-out this monograph to the conventional system as *system S* and to the proposed system as *system N*.
2) It introduces a new device, the slash (/), for expressing duplication in the phrase structure (PS), and for stating the exact "points of attachment" in transformational (T) rules.
3) Each of its PS rewrite rules has more "depth" than a rule by system S in that, while the latter represents immediate constituent analysis on one, and only one, level, the former can represent IC analyses on more than one level.

---

[3]References throughout this paper are to the works listed by author and year in the Bibliography, beginning on p. 34.

4) Lexical replacement takes place, not at the end of the PS section, but after T operations, creating a sharper boundary between syntax proper and the lexicon.

2.1 Basic Notions

The central notion of system N is that of *number*. A number is essentially a discrete sequence of digits, but may contain other material as well, notably the slash (/), which has a special function, and occasionally morphophonemic items, boundary markers, etc.

In a string, numbers are joined by the hyphen, which serves as the concatenation sign.

A number may be segmented into continuous *components* of any length, the shortest being a single digit, or anything equivalent to a single digit, and the longest being the number itself. A component is represented by the letter C. When it is equivalent to the number, it is represented as –C–. Thus one way to define a number is to say that it is any –C–.

A number has one or more initials, where *initial* means a C of any length that contains the leftmost digit. The longest initial of a number is therefore necessarily also its longest C or the number itself.

Now the principles governing the formation of numbers as names of syntactic units by system N may be stated as tollows:

Principle 1. A given number must be identical to an initial of any number which is part of that number.

Principle 2. In general, numbers will occur from left to right in an ascending order in PS rules.

In addition to these two principles, which are basic to system N, it is of course necessary to meet the following two general requirements:

General requirement 1. The numbers (names) representing different units must be different.

General repuirement 2. Numbers (names) must be as short as possible.

The two general requirements are universally recognized, and therefore require no comments. Principle 1 is intended to show the relations of set membership among units, while principle 2 aims at indicating the relations of order, to a limited extent.

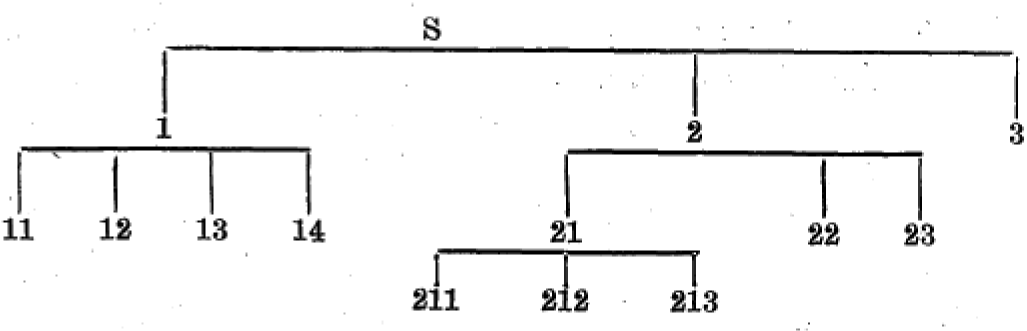
Of all the principles and requirements, principle 2 is the only one that may be ignored. Hence the qualification "in general." But principle 1 ***must always*** be observ ed. So must the two general requirements.

The relative laxity of princiiple 2 is due to the fact that its strict enforcement is not always possible. For, in the first place, there are the cases of mutually exclusive units, which clearly can be numbered only in an arbitrary order. There are also the cases of discontinuous units, which will inevitably require an inverted order for the units involved. For it will be found that, while disconinuous units are usually treated as continuous in the phrase structure by system S, with the proper order restored only by means of obligatory transformations, by system N It is possible to a certain extent to introduce discontinuous units in the phrase structure (see PS 1 of the sample syntax in Chapter III) Furthermore, there are the cases of what we shall call "complex numbers" (see p. 9 below). In spite of these facts, it would surely be unwise if we completely ignored the question of order in numbering units. Consequently we require that the relations of position be shown whenever possible. Needless to say, the general order that will be shown is the order of units in the phrase structure, not the order after transformations.

The phrase "part of" in principle 1 is used in the conventional sense—that is, every number is part of at least itself. It is because every number is part of itself that every number is identical to one of its own initials—the longest.

Only one unit is not represented by a number and consequently excepted from all the above principles and requirements: the sentence. It is simply represented as S, as in system S.

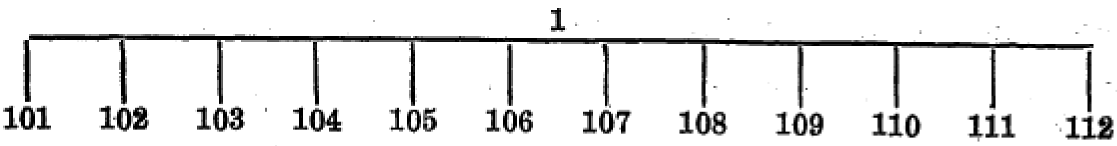
A constituent structure of a sentence, with units numbered according to the principles of system N, will then look something like the following diagram:



The three IC's of S are named 1, 2 and 3, which conform to general requirement 1 in that they are all different, to general requirement 2 in that each consists of only a single digit, and to principle 2 in that no number is smaller than any occurring to its left. Here principle 1 does not yet come into consideration. When we come to the IC's of 1, however, we must meet principle 1 as well as all the other requirements. That is, the number representing each unit here must begin with the digit 1. Thus we assign the number 1 to each of the units here. As far as principle 1 and general requirement 2 are concerned, we do not have to do anything more than this. However, we would then violate, not only principle 2, but also general requirement 1. So we assign a different digit to each unit in addition to 1 and call the four IC's of 1 here 11, 12, 13 and 14. These considerations hold with all the numbers assigned to all the other units in the diagram.

Since every number will, of necessity, appear in its entirety as a portion of every number belonging to it, the inevitable result is that every number automatically tells a complete story of its own set membership.

We see in the last diagram that no use is made of the digit 0 for any of the numbers. This is because 0 is reserved for use in cases where a unit has more than nine immediate constituents, as shown in the following diagram:



As system S speaks of terminal and nonterminal symbols, so system N speaks of *terminal* and *nonterminal numbers*. A terminal number may be defined as a number whose (full) shape is identical to no initial of any other number, while a nonterminal number may be defined as a number whose (full) shape is identical to one initial of at least one other number. In plain language, a terminal number represents an ultimate constituent of the sentence, while a nonterminal number represents a unit larger than the ultimate constituent.

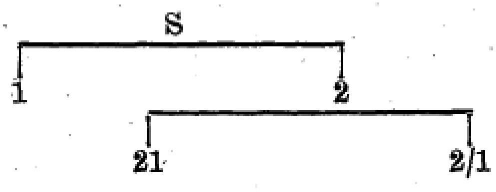
A number furthermore may have an *attached initial* and an *included number*. An attached initial of a number is a C extending from its left end to any slash(inclusive) found in the number, while an included number is a C between any slash (exclusive) and the right end. In 21/43, for example, 21/ is an attached initial, and 43 an included number. If there is more than one slash in the number, then of course some attached initials and included numbers will overlap, as do the attached initial 12/34/ and the included number 34/56 in the number 12/34/56.

A number of the kind just mentioned, which contains the slash, we may call a ***complex number***. Complex numbers are needed when units serving as part of two or more different things happen to have an identical structure.

One readily available example of this situation is the case of two *NP* 's in the first two PS rules in Chomsky, 1957 (p. 111.) The first PS rule cuts the sentence into *NP* and *VP*, then the second rule cuts *VP* into *Verb* and *NP*. The two *NP*'s are given an identical symbol apparently because of their identical internal structure. By system N, however, the two *NP*'s cannot have identical numbers because that would violate at least principle 1 and—depending on how we interpret the word "different"—also general requirement 1. In any case, the numbers to be given to the two units must differ at least in their initials.

On the other hand, it would certainly be extremely unwise to conceal the fact that the two units have identical internal structure by assigning completely different numbers. The numbers we need, in short, must indicate, clearly, both the difference and the identity.

To accomplish this, we first give each *NP* an identical number, say 1, then give one of them an attached initial containing the number that has been given to *VP*, say 2. The resulting complex number will then be 2/1, of which 1 is the included number, and 2/ the attached initial. The complex number tells us that the unit it represents is something identical to 1 in internal structure, though it serves as part of 2. The constituent structre as indicated by the two PS rules we have just discussed may then be represented as follows:



In this diagram, 1 is the first *NP*, 2 *VP*, 21 *Verb*, and 2/1 the second *NP* which has come from *VP*. Now it is to be pointed out that the number 2/1 conforms to all the principles and requirements except principle 2, which may be ignored whenever necessary. It conforms to principle 1 in that it contains an initial having the shape 2, which explicitly indicates that it is part of the number 2; to general requirement 1 in that it is unmistakably different from 21, from 2, and from 1; and to general requirement 2 in that it cannot be made any shorter than it is without undesirable consequences such as violation of either principle 1 or general requirement 1 and concealment of its structural identity with 1.

We shall see in Section 2.3 that attached initials will play a very important role in facilitating transformational statemets, especially where system S has been awkward because of its use of powerless alphabetical symbols.

Finally, an included number may be something other than digits—an item in morphophonemic spelling or some sort of boundary marker. For example we may have a number of the form *23/n't*, where *n't* is an included number. An included number of this sort usually represents a single item not defined in the phrase structure and consequently introduced by means of transformation. If necessary, however, such an included number may very well be introduced in the PS section, but must be included only in a terminal number, which cannot be further expanded. For, unless this restriction is imposed—that is, if we allow things other than digits (and the slash) to occur indiscriminately in all numbers—then we may ultimately have phrase structures where every unit is represented by "numbers" which contain no digits, and which would consequently fail to serve the purposes for which system N is intended.

2.2 Phrase Structure Rules

Except for a new convention, to be imposed because of the introduction of the slash, the PS rules under system N are to be applied in exactly the same manner as rules under system S. That is, rules are to be applied in the order given, each rule applied an arbitrary number of times before application of the next if the rule is applicable more than once. Parenthesized elemnts may or may not be selected. Braces enclose alternatives or, put differently, mutually exclusive elements. Square brackets, also enclosing mutually exclusive elements, indicate that each of the elements enclosed in one pair of brackets goes only with a positionally corresponding element in a different pair of brackets.

The new convention to be imposed involves the arrow. As in system S, it represens an instruction to rewrite something as something else, but the instructions are to be understood more broadly. Stated generally, a rule instructing us, to rewrite a certain number must be applied, not only to that number, but also to every complex number having that number as an included number. This may be stated in more explicit terms as follows:

Given a rule of the form $-Cx \longrightarrow -C_1-C_2-\cdots\cdots-Cn-$, *it will be applied in one of two ways*:

a) *A string* $W-Cx-Z$ *will become* $W-C_1-C_2-\cdots\cdots-Cn-Z$.

b) *A string* $W-Cy/Cx-Z$ *will become* $W-Cy/C_1-Cy/C_2-\cdots\cdots-Cy/Cn-Z$.

For example, if we have a string like:

#-2-31-32-3/2-#

A rule of the form 2→21-22, applied to the string, will produce:

#-21-22-31-32-3/21-3/22-#

Why a new convention like this is necessary should be clear if we remember that 2 and 3/2 represent units with an identical internal structure and must therefore be expanded in an identical manner by an identical rule.

Generally speaking, because of the greater "depth" of its PS rules, system N requires much fewer rewrite rules for the PS section than does system S. Consider the following sets of rules:

Given: #S#

G a:

PS 1a	S→A+B
PS 2a	A→a+b
PS 3a	B→c+d+e
T 1a (optional)	
	A+B→B+A
T 2a (optional)	
	B+A→B+A+B+A

G b:

PS 1b	S→1-2
PS 2b	1→11-12
PS 3b	2→21-22-23
T 1b (optional)	
	1-2→2-1
T 2b (optional)	
	2-1→2-1-2-1

G c:

PS 1c	S→11–12–21–22–23
T 1c (optional)	
	Same as T 1b
T 2c (optional)	
	Same as T 2b

G a is a set of syntactic rules by system S with three PS rules and two T rules. *G b* is identieal to *G a* in every respect except that it has replaced all the alphabetical symbols of *G a* with numbers formed according to the principles of system N. *G c* is a set of rules typical of system N, and is intended for the same hypothetial language as *G a* and *G b*. Applying the PS rules of the three grammars, we obtain the following three derivations:

S 1a	#A+B#	(by PS 1a)
S 2a	#a+b+B#	(by PS 2a)
S 3a	#a+b+c+d+e#	(by PS 3a)
S 1b	#–1–2–#	(by PS 1b)
S 2b	#–11–12–2–#	(by PS 2b)
S 3b	#–11–12–21–22–23–#	(by PS 3b)
S 1c	#–11–12–21–22–23–#	(by PS 1c)

We see that S 1c is derived by applying a single rule only once, whereas both S 3a and S 3b have been derived in three separate steps. Yet S 1c is completely the same as S 3b. In other words, we have obtained in one stroke by *G c* what it took *G b* three steps to produce. This certainly appears efficient. The question, of course, is whether this abbreviated operation is justifiable.

From the standpoint of a grammar concermed primarily with sentence generation, I think it is undeniable that PS rules exist but for two reasons: A) they convert the initial symbol into acceptable strings of terminal symbols; B) they provide, by defining the phrase structnre, the information (derivational history, structural description) that is indispensable for the application of T rules. Let us call the first operation the *expanding function* and the second, the *defining function.*

The expanding function naturally demands that there be as few interme-diate stages as possible, the most ideal case being the immediate conversion of the initial symbol into a terminal string. This, however, must inevitably affect the defining function, not to speak of its impossibility. Consequently a number of restrictions are imposed in system S as a check on the expanding function so as to insure the satisfactory execution of the defining function. However, since these restrictions are primarily intended for the protection of the defining function, there is obviously no reason why some of them should not be lifted if, by doing so, we still leave the defining function intact.

The kind of operation represented by *G c* clearly cannot be accepted by system S because it skips some important intermediate stages that are necessary for the construction of a "unique derivational tree." But we want the tree, not for its own sake, but to make T rules work. For T rules apply, not to a PS terminal string as it is, but to its underlying structure, usually represtented in the form of a tree. Thus, if *G c* can give us all the essential information about the underlying structure so that there are no difficulties whatever added to the application of T rules, then there would be no grounds on which to reject *G c*. Let us therefore see if we can elicit from *G c* all the information we want by applying the T rules in the three grammars and comparing the results.

We find that it is impossible to apply T la to S 3a unless we "look back" to earlier strings, to be sure as to whether S 3a has the specified structure A+B and, if it does, where A ends and B begins. Moreover, we see that the infom-ation would be incomplete unless the two earlier strings, S la and S 2a. are both present. In other words, the three-step derivation is the minimum required for constructing a unique derivational tree, without which there would be no way of applying any T rules to S 3a.

When we come to *G b* and *G c*, however, we find the situation rather different. In both *G b* and *G c*, with our knowledge that different numbers having a common initial are all part of the same unit, we do not have to "look back." By examining the PS terminal strings alone, we see very quickly that they have the structure 1-2 specified by T lb and T lc. So we apply the rules and produce two identical strings of the following form:

#-21-22-23-11-12-#

Again, on the basis of the information supplied by this last string alone, we find that T 2b and T 2c are applicable to it and that the new string produced by aplying the rules once will be:

#-21-22-23-11-12-21-22-23-11-12-#

We further see that T 2b and T 2c can be applied repeatedly, resulting in a string of infinite length, and that each time the rules are applied we rely solely on the information provided by the string in hand.

In short, we have seen that, while a three-step derivation is the minimum in the case of *G a*, with *G b* it is actually redundant. For *G c*, with its abbreviated operation, can do everything just as well as *G b*, supplying all the important information needed for applying T rules that *G a* cannot fully provide with anything less than three separate PS rules. In other words, *G c* is just as effective in the defining function as either *G a* or *G b* while being far more effective in the expanding function.

In spite of what has just been demonstrated, the power of system N to condense rules is of course not limitless. When the phrase structure involves situations requiring the use of complex numbers, for example, it is clearly necessary, for avoiding wasteful duplication, to expand all the equivalent units together through a separate rule set up exclusively for that purpose, with the result that there are two or more rules. Moreover, units whose structures are conditioned by still unexpanded units must obviously be expanded, too, by some context-restricted rules after the latter have been expanded.

2.3 Transformational Rules

While in the PS section rules are applied on the basis of the full shapes of numbers in the pertinent string, ***in the T section all rules are to be applied on the basis of initials.*** The basic principle to be followed is:

While a number of the shape C_1 given in a PS rule denotes any number having C_1 as its full shape, a number of the shape C_1 given in a T rule denotes *any substring which consists entirely of numbers with an initial of the shape* C_1 *and which is not adjoined by any number with an initial of the shape* C_1. Suppose we have the following string:

#-111-112-12-211-212-213-2211-2212-222-#

A B C D E F G H I

The numbers are labeled by alphabetical letters for ease of reference. Now a number of the shape 21, if appearing in a PS rule, will not apply to this string because it contains no number whose full shape is 21. But, if the number is given in a T rule, then it will denote the substring D-E-F. Similarly A-B-C is subsumed by 1, G-H-I by 22, D-E-F-G-H-I by 2, and so forth.

There are five basic operations involved in the transformations of system

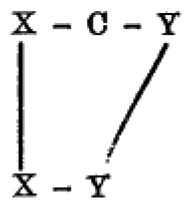
N: *deletion, addition, permutation, inclusion,* and *separation.* Each will be illustraed below.

As shown in the illustrative rules, transformational statements are made simply by placing two structural representations in vertical order, with the left ends aligning. The representation at the top shows the structure prior to transformation, and the one at the bottom shows the structure after transformation. Indications of required structural change are made by means of crisscross lines. These lines connect *all and only* identical numbers occurring in both representations, whether as full numbers or as included numbers, and whether their positions are affected or not in the transformation.

A number not connected to any line is thus either to be deletd or has been added. A number that is optionally deleted, however, will still be connected to a line, since it will appear, enclosed between parentheses, in the structural represention at the bottom.

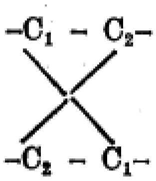
Other devices, except what are self-explanatory, are explained in the remarks following each illustrative rule.

Deletion



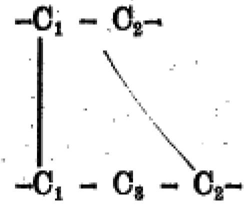
X, Y and other capitalized late alphabetical symbols are used, as in system S, to represent variable portions of a string that may be null. Wider space is given between adjoining items simply as a means of suggesting the basic difference in nature of a T rule from a PS rule. For the same trivial reason, the hyphen will be retained even when parentheses and other signs indicating concatenation are present. This rule simply says: "Delete –C–." It requires that the substring so specified be taken off in its entirety.

Permutation



The symbols for variables will be suppressed (replaced with white space) unless they are relevant to the structural description. In other words, the white space on both sides of a structural representation, unless the symbol # separates it from the string, is to be understood as representing either the absence or presence of other numbers.

Addition

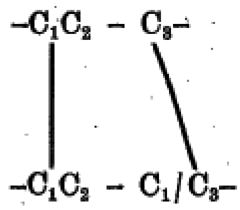


The rule says: "Insert $-C_3-$ between $-C_1-$ and $-C_2-$." It does not specify to which unit $-C_3-$ shall belong. The situation is typical of low-level rules, where it is not necessary to allow for further transformations by indicating the "points of attachment," and where an attached initial will often be found more a burden than a help. Otherwise, addition is usually accompanied by simultaneous inclusion.

The thing added may be something represented by a nonterminal number. In that case, the addition represents a generalized transformation and the added item is the embedded substring.

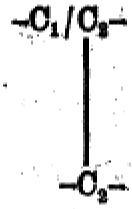
In simple addition, the thing added is a terminal number, which is frequetly made up of something other than digits. A morphophonemic item or boundary marker introduced in this fashion, even if it contains no digits, is to be treated as a "number" as long as the string is still undergoing T development.

Inclusion



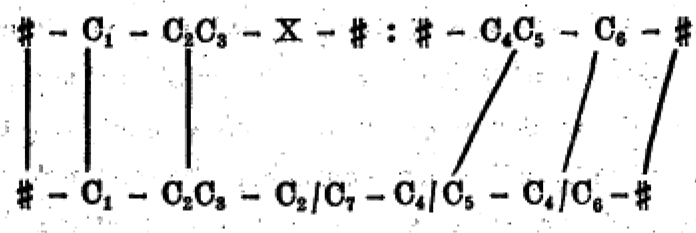
The rule says: "Make $-C_3-$ part of $-C_1-$." It requires that every number in the specified substring be given an attached initial of the shape $C_1/$. As a result of this structural change, the substring represented by $-C_1/C_3-$ can henceforward be covered by $-C_1-$.

Separation



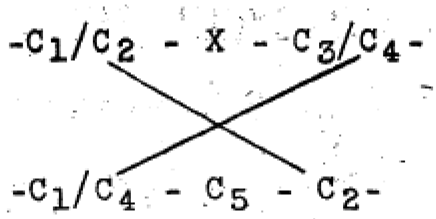
The rule says: "Separate $-C_2-$ from $-C_1-$." It requires that the attached initial $C_1/$ be taken off every number in the specified substring. The separated substring is thus no longer subject to any instruction that applies to $-C_1-$. It is to be pointed out that the word "separation" is used in a very limited sense, denoting only the loss by a complex number of an attached initial, as opposed to inclusion, which is the acquisition of an attached initjal. It does not denote all instances in which a unit is taken away from its "node," which is something that inevitably happens when a unit is deleted or, in the case of units already present in the string, included in another unit.

Most transformations naturally combine two or more of the basic operations, as in the two relatively complex cases illustrated below. The first involves de-le tion, addition, and inclusion:



The presence of the colon indicates that the rule is a case of generalized transformation involving the two strings it separates. The rule says: "Combine the two strings by deleting the # between them; delete X, add $-C_7-$, make $-C_7-$ part of $-C_2-$, and $-C_6-$ part of $-C_4-$." It is to be pointed out that the symbol #, though it need not be considered a number, is still to be considered an item making up a string and, consequently, is subject to the instructions of T rules.

The next rule involves all the five basic operations:



This rule says:" Delete X; insert $-C_5-$; shift positions of $-C_1/C_2-$ and $-C_3C_4-$; separate $-C_2-$ from $-C_1-$, and $-C_4-$ from $-C_3-$; make $-C_4-$ part of $-C_1-$."

2.4 Lexicon

The PS terminal strings derived through the application of PS rules must at least undergo proper obligatory transformations before they can be replaced by proper sequences of lexical items. The lexicon in system N is thus no longer a part of the PS section, resulting in something like a four-part grammar in contrast with the tripartite arrangement of system S.

This rearrangement is necessary in order to make the numbers useful in the T section. Without it, all the numbers which we have carefully formed so as to make the T rules work better would have to be discarded through lexical replacement before the string enters the T section.

Every lexical rule is a simple string replacement rule specifying the replacement of a terminal number by a single lexical item, offering as many alternatives as there are items listed. The rules are again to be applied on the basis of the full shapes of numbers, not by initials, and the stipulation that a rule applicable to a number of the form C_1 is applicable also to a number of the form C/C_1 will again be observed, if there are still any complex numbers remaining in the T terminal string. However, chances are that, by the time of lexical replacement, all complex numbers will have shed all their attached initials as a result of an obligatory seprating transformation required to facilitate other lower-level transformations, for which attached initials will often be found cumbersome.

Only numbers consisting of digits need be replaced, for "numbers" consisting of other material automatically become morphophonemic entities at the same time that the string emerges from the T section.

The string, with all its numbers replaced by proper lexical items, is then referred to the morphophonemic section to be given a proper phonological shape.

CHAPTER III

SAMPLE SYNTAX

The sample syntax presented in this chapter is that of extremely oversimplified English. Since the oversimplification would probably make the name "English" unjustifiable, we shall use the name language E throughout our discussion. But, oversimplified as it is, I think the analysis at least correctly represents some of the fundamental syntactic phenomena of the English language.

The analysis is based largely on Lees, 1960, and Chomsky, 1957 and 1962, though other sources, transformational or otherwise, have been drawn upon and some portions are original.

We shall first list all the basic syntactic units (i.e., units not represented by complex numbers) and lexical items that figure in language E, much as characters are listed at the head of a play by playwrights, before proceeding to examine the syntactic rules. This full list, given primarily for ease of reference, will serve as the lexicon of language E, though in an actual grammar the lexicon will list only terminal numbers plus the lexical items they represent and will be placed after the T section, where each derived string may most conveniently be converted into a proper sequence of lexical items before enterang the morphophonemic section.

In the following full list, as in the PS rules, nonterminal numbers are italicized to minimize confusion. Parenthesized terms or remarks given after a number indicate the general characteristics, semantic and syntactic, of the unit represented by the number. In these remarks, reference will often be made to the syntactic rules to be given after the list which represent the syntactic characteristics of each unit more explicitly. Lexical items, with a few exceptions, are all given in conventional English graphemes instead of morphophonemic spelling because they are primarily intended for illustrative purposes. We shall, however, pretend that the graphemes are all morphophonemes.

1—(noun phrase)

11—the, my, his, your, our, their, her, its

12—a

13—book, table, student, pen, boy, girl, house, tree, car, dog, cat, teacher (count noun)

14—S (plural morpheme)

15—water, milk, rice, gold, silver, ink (mass noun)

16—(personal pronoun)

161—I

162—we, you, they

163—he, she

164—it (separated from 163 because of T 2)

2—what, who, which (interrogative pronoun)

3—(This unit may be called ***preverb***. The name as used here, however, is not equivalent to that usad in Lees, 1960, for unit 3 denotes part of what Lees called *Aux* and part of what he called *Prev*. The most striking syntactic peculiarity of this unit is that it can exchange positions with unit 1 to form simple interrogative sentences as shown in T 5.)

31—(One striking feature of the English sentence—striking perhaps only from the viewpoint of an Oriental — is what is traditionally called the ***finite***, as opposed to ***nonfinite***, verb. A common mistake among Chinese and Japanese students learning English is the failure to make at least one verb finite in a sentence. This unit, 31, comprises part of what were called ***anomalous finites*** in Palmer, 1924, plus three fictitious groups of morphemes which, when combined with a verb, make the verb finite, namely, 3121, 3122, and 3123.)

311—will, would, can, could, should

312—(See remarks on 31.)

3121—AM-DO, WAS-DID

3122—ARE-DO, WERE-DID

3123—IS-DOES, WAS-DID

32—not

33—E (superfix of emphasis)

4— (This whole segment constitutes the ***verb phrase***, but of course it is not identical to the VP in Chomsky, 1957, because it does not contain what he called *Aux*. The most striking characteristic of this unit is that it can combine with 31, to serve as the predicate, or either with 422 or with ***to***, to serve as a nominal, as shown in T 1 and T 2. It is separated from 3 because it cannot accompany 3 when the latter moves to the front position.)

41—(This number subsumes what are traditionally called ***verb*** minus what we have labeled 31 above. Its syntactic peculiarity is that it absorbs any 42 directly preceding it and makes it its suffix, as shown in T 15.)

411—(Covered by this number are the two words which are peculiar in that, when preceded by 312 with or without intervening 32 or 33, they move to the position before 312, make 312 their suffix, become part of 3 themselves, and thereby become shiftable to the front position like any other part of 3. This operation is shown in T 3.)

4111—have

4112—be

412—(This unit, standing for an ordinary verb, is distinguished from 411 because it reacts to 312 differently. It also absorbs 312 as its suffix, but only when directly preceded by it. After absorbing 312, moreover, it does not become part of 3 and consequently is not shiftable to the front position. The operation is shown in T 14.)

4121—hit, please, move, kill, burn, stop, catch

4122—swim, sleep, rise, work, appear, vanish, fall

42—(This is roughly equivalent to the *Af* defined, not in the PS section, but in the T secion, in Chomsky, 1957. It is only roghly equivalent, because we have subsumed under 312 part of what Chomsky defined as *Af*. The syntactic peculiarity of this group is that it becomes the suffix of any 41 directly following it, as shown in T 15 and mentioned in the remarks on 41. In PS 1, moreover, it will be noted that the two units subsumed under this number, 421 and 422, are allowed to occur discontinuously. This discontinuity is presumably what makes the redefinition in the T section necessary under system S. But, using numerical designations to represent the units and allowing them to occur discontinuously in the phrase structure, system N makes it possible to avoid makeshift redefinitions of this sort—at least in this instance.)

421—en (past participle suffix)

422—ing

43—good, interesting, bad, silly, great, important, dangerous,bettr, worse (Only adjectives capable of being used both predicatively and attributively are listed.)

44—by (The passive construction is treated as a kernel in this sample syntax, as shown in PS 1, and consequently a number is assigned to this word. In a real syntax of English, the question of whether to treat the passive as a transform or a kernel can, of course, be decided only after a more thorough analysis of the language. But, as far as the oversimplified syntax of language E is concerned, treating the passive as a kernel is simpler than treating it as a transform.)

Other items playing some part in the syntax of language E, besides the initial #S#, are $, which is the affixation marker, as well as *n't, em* (objective case marker) and *to*, all of which are introduced through transformation.

We now proceed to examine the syntactic rules of language E. The syntax of language E consists of four PS rules and 16 T rules, of which nine are optional and seven are obligatory.

In examining the PS rules, it is to be remembered that, by system N, most nonterminal numbers appear only in the T section, and consequently that only a few of the nonterminal numbers listed above will be found in the following PS rules.

In the T section, each optional T rule (marked OP, as opposed to OB) is followed by one or more typical sentences that will result from application of that rule and none of the further optional rules.

Phrase Structure Rules

Given: #S#

PS 1

$$S \rightarrow \left\{ \begin{array}{c} \mathit{1} \\ 2 \end{array} \right\} \left\{ \begin{array}{c} 311 \\ \mathit{312} \end{array} \right\} (32)\ (3/\$\text{–}33)\ (4111\text{–}421) \left\{ \begin{array}{l} (4112\text{–}422) \left\{ \begin{array}{l} 4121\text{–}\mathit{4/1} \\ 4122 \end{array} \right\} \\ 4112 \left\{ \begin{array}{l} 43 \\ 421\text{–}4121\ (44\text{–}\mathit{4/1}) \end{array} \right\} \end{array} \right\}$$

PS 2

$$1 \rightarrow \left\{ \begin{array}{ll} \left\{ \begin{array}{c} 11 \\ 12 \end{array} \right\} (1/43) & 13 \\ \quad (12)(1/43) & \left\{ \begin{array}{l} 13\text{–}1/\$\text{–}14 \\ 15 \end{array} \right\} \\ & 161 \\ & 162 \\ & 163 \\ & 164 \end{array} \right\}$$

PS 3

$$312 \rightarrow \left\{ \begin{array}{c} 3121 \\ 3122 \end{array} \right\} \text{ in } \left\{ \begin{array}{c} 161 \\ \left\{ \begin{array}{c} 14 \\ 162 \end{array} \right\} \end{array} \right\} \text{——}$$

PS 4

$312 \rightarrow 3123$

Transformational Rules

T 1 (OP)

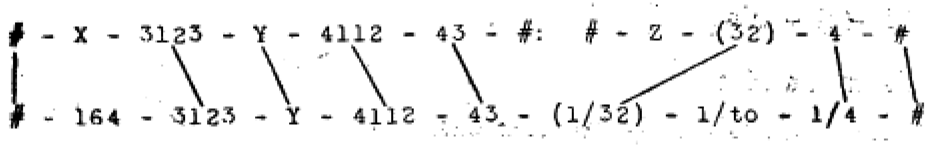
```
# - X - (32) - 4 - # : # - Y - 3123 - Z - 4112 - 43 - #

# - (1/32) - 1/422 - 1/4 - 3123 - Z - 4112 - 43 - #
```

Catching a dog is silly.

Not burning the trees was worse.

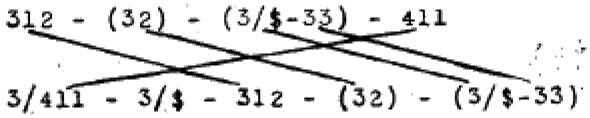
T 2 (OP)



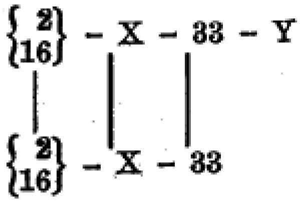
It is silly to catch a dog.

It was worse not to burn the trees.

T 3 (OB)



T 4 (OP)



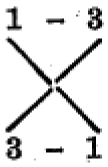
who *can?*

she *has.*

what is *not?*

I am *not.*

T 5 (OP)



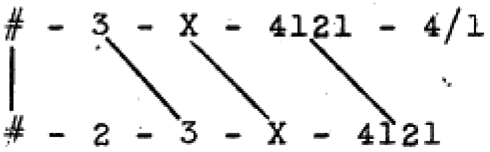
Has she?

Did it stop their cars?

Would not our teacher swim?

Is it good?

T 6 (OP)



Who could the house please?

What is she moving?

T 7 (OP) 31 – 32

31 – $ – n't

It won't rise.
The student didn't kill her cat.
Wouidn't our teacher swim?

T 8 (OP) 32 - (3/$-33) - 1

1 - 32 - (3/$-33)

Did the teacher not swim?
Should they *not?*

T 9 (OB) 32 - (3/$-33) - 16 - #

$ - n't - (3/$-33) - 16 - #

T 10 (OP) # – 1 – X – 33 – Y

– 1 – X – 33

The sthudent *could.*
Our tables *aren't.*

T 11 (OB) –C/C$_1$–

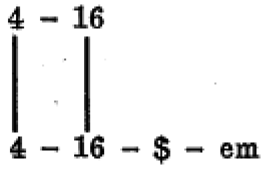
–C$_1$–

T 12 (OP) 43

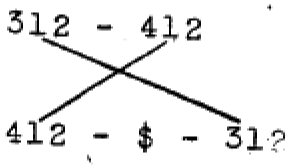
43 – 43

The girl was bad, silly, dangerous.
The good, important, interesting, great, dangerous tree vanished.

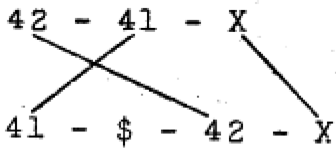
T 13 (OB)



T 14 (OB)

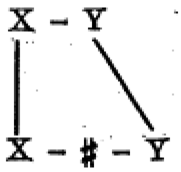


T 15 (OB)



condition: X≠$-Y

T 16 (OB)



condition: $\begin{Bmatrix} X \\ Y \end{Bmatrix} \neq Z \begin{Bmatrix} \$ \\ \# \end{Bmatrix} W$

The "condition" for T 15 is that 41 cannot be adjoined by $, while the condition for T 16 states that neither X nor Y may contain $ or #. If this is the case, then the rules are inapplicable. Both conditions are for avoiding unnecessary recursion. T 16 is the transformation for inserting word boundaries for use in the morphophonemic section.

Our next task is to try to apply the rules. First, applying the PS rules in such a way as to produce the necessary strings for applying all the optional T rules, we obtain the following five PS derivations:

Derivation A.

PS 1 #-1-312-3/$-33-4122-#

PS 2 #-163-312-3/$-33-4122-#

PS 4 #-163-3123-3/$-33-4122-# (TSA)

Derivation B.

PS 1 #-1-311-32-3/$-33-4122-#

PS 2 #-162-311-32-3/$-33-4122-# (TSB)

Derivation C.

PS 1 #-1-312-4121-4/1-#

PS 2 #-161-312-4121-4/163-#

PS 3 #-161-3121-4121-4/163-# (TSC)

Derivation D.

PS 1 #-2-312-32-3/$-33-4111-421-4112-43-#

PS 4 #-2-3123-32-3/$-33-4111-421-4112-43-# (TSC)

Derivation E.

PS 1 #-1-311-32-4111-421-4112-421-4121-44-4/1-#

PS 2 #-12-1/43-13-1/$-14-311-32-4111-421-4112-421-4121-44-4/12-4/1/43-4/13-4/1/$-4/14-# (TSE)

In the above examples of PS derivation, each string representing a stage of derivation is preceded by the label of the PS rule applied to produce it. The parenthesized labels placed at the end of all PS terminal strings will be used in the following illustrations of how these PS terminal strings may develop as they run through the T rules. By planning these developments in such a way that every optional rule will be applied once to at least one string, we will obtain a total of nine T terminal strings. In the following examples, as in the examples of PS derivation, each string representing one stage is preceded by the label of the rule applied to produce it. The PS terminal string or strings involved in each development are indicated between parentheses at the head of each example. In each example, the concatenation signs are suppressed after the application of T 16, and the T terminal string is followed by a sequence of lexical items (represented by the symbol L) which it my acquire through lexical replacement:

Development I (TSA)

T 11 #-163-3123-$-33-4122-#

T 16 #163#3123$33#4122#

L #he#IS-DOES$E#swim# (L I)

Development II (TSA)

T 10 #-163-3123-3/$-33-#

T 11 #-163-3123-$33-#

T 16 #163#3123$33#

L #she#WAS-DID$E# (L II)

Development III (TSB)

T 4 #-162-311-32-3/$-33-#
T 5 #-311-32-3/$-33-162-#
T 8 #-311-162-32-3/$-33-#
T 11 #-311-162-32-$-33-#
T 16 #311#162#32$33#
L #will#you#not$E# (L III)

Development IV (*TSB*)

T 4 #-162-311-32-3/$-33-#
T 5 #-311-32-3/$-33-162-#
T 9 #-311-$-n't-3/$-33-162-#
T 11 #-311-$-n't-$-33-162-#
T 16 #311$n't$33#162#
L #Will$n't$E#you#(L IV)

Development V (*TSC*)

T 5 #-3121-161-4121-4/163-#
T 6 #-2-3121-161-4121-#
T 16 #2#3121#161#4121#
L #who#AM-DO#I#please# (L V)

Development VI (*TSC*)

T 11 #-161-3121-4121-163-#
T 13 #-161-3121-4121-163-$-em-#
T 14 #-161-4121-$-3121-163-$-em-#
T 16 #161#4121$3121#163$em#
L #I#frighten$WAS-DID#he$em# (L VI)

Development VII (*TSD*)

T 3 #-2-3/4111-3/$-3123-32-3/$-33-421-4112-43-#
T 7 #-2-3/4111-3/$-3123-$-n't-3/$-33-421-4112-43-#
T 11 #-2-4111-$-3123-$-n't-$-33-421-4112-43-#
T 15 #-2-4111-$-3123-$-n't-$-33-4112-$-421-43-#
T 16 #2#4111$3123$n't$33#4112$421#43#
L #what#have$IS-DOES$n't$E#be$en#worse# (L VII)

Development VIII (*TSC, TSD*)

T 2 #-164-3123-32-3/$-33-4111-421-4112-43-1/to-1/4121-1/4/163-#
T 3 #-164-3/4111-3/$-3123-32-3/$-33-421-4112-43-1/to-1/4121-1/4/163-#
T 11 #-164-4111-$-3123-32-$-33-421-4112-43-to-4121-163-#
T 13 #-164-4111-$-3123-32-$-33-421-4112-43-to-4121-163-$-em-#

T 15 #-164-4111-$-3123-32-$-33-4112-$-421-43-to-4121-163-$-em-#

T 16 #164#4111$3123#32$33#4112$421#43#to#4121#163$em#

L #it#have$WAS-DID#not$E#be$en#bad#to#hit#she$em# (L VIII)

Development IX (*TSD, TSE*)

T 1 #-1/32-1/422-1/4111-1/421-1/4112-1/421-1/4121-1/44-1/4/12-1/4/1/43-1/4/13-1/4/1/$-1/4|14-3123-32-3/$-33-4111-421-4112-43-#

T 3 #-1/32-1/422-1/4111-1/421-1/4112-1/421-1/4121-1/44-1/4/12-1/4/1/43-1/4/13-1/4/1/$-1/4/14-3/4111-3/$-3123-32-3/$-33-421-4112-43-#

T 11 #-32-422-4111-421-4112-421-4121-44-12-43-13-$-14-4111-$-3123-32-$-33-421-4112-43-#

T 12 #-32-422-4111-421-4112-421-4121-44-12-43-43-43-13-$-14-4111-$-3123-32-$-33-421-4112-43-43-43-#

T 15 #-32-4111-$-422-4112-$-421-4121-$-421-44-12-43-43-43-13-$-14-4111-$-3123-32-$-33-4112-$-421-43-43-43-#

T 16 #32#4111$422#4112$421#41 21$421#44#12#43#43#43#13$14#4111$3123#32$33#4112$421#43#43#43#

L #not#have$ing#be$en#please$en#by#my#good#bad#important#table$S#have$IS-DOES#not$E#be$en#great#interesting#silly# (L IX)

Although morphophonemics is not our concern we may mention a few of the rules that will be needed to give our strings proper phonological shapes. Pretending that the graphemes are morphophonemes, we will see that the morphophonemic section will have to contain at least some sort of ordered rules like the following:

1)
$$\text{be\$}\left(\begin{matrix}\text{AM–DO}\\ \text{ARE–DO}\\ \text{IS–DOES}\\ \text{WAS–DID}\\ \text{WERE–DID}\end{matrix}\right)\longrightarrow\left(\begin{matrix}\text{am}\\ \text{are}\\ \text{is}\\ \text{was}\\ \text{were}\end{matrix}\right)$$

2)
$$\text{have\$}\left(\begin{matrix}\left\{\begin{matrix}\text{AM–DO}\\ \text{ARE–DO}\end{matrix}\right\}\\ \text{IS–DOES}\\ \left\{\begin{matrix}\text{WAS–DID}\\ \text{WERE–DID}\end{matrix}\right\}\end{matrix}\right)\longrightarrow\left(\begin{matrix}\text{have}\\ \text{has}\\ \text{had}\end{matrix}\right)$$

3)

$$\$ \begin{bmatrix} \left\{ \begin{matrix} \text{AM-DO} \\ \text{ARE-DO} \end{matrix} \right\} \\ \text{IS-DOES} \\ \left\{ \begin{matrix} \text{WAS-DID} \\ \text{WERE-DID} \end{matrix} \right\} \end{bmatrix} \longrightarrow \begin{bmatrix} \phi \\ \text{(e)s} \\ \text{(e) d} \end{bmatrix}$$

4)

$$- \begin{bmatrix} \left\{ \begin{matrix} \text{AM-DO} \\ \text{ARE-DO} \end{matrix} \right\} \\ \text{IS-DOES} \\ \left\{ \begin{matrix} \text{WAS-DID} \\ \text{WERE-DID} \end{matrix} \right\} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{do} \\ \text{does} \\ \text{did} \end{bmatrix}$$

Applying these and other rules to the sequences of lexical items we have obtained, we may then get the following sentences:

L I: He *does* swim[4].

L II: She *did*.

L III: Will you *not?*

L IV: *Won't* you?

L V: Who do I please?

L VI: I frightened him.

L VII: What *hasn't* been worse?

L VIII: It had *not* been bad to hit her.

L IX: Not having been pleased by my good, bad, important tables has *not* been great, interesting, silly.

CHAPTER IV

EVALUATION

4.1 Power of Transformational Theory

Syntax can obviously be viewed from many different angles, each revealing in one way or another. I should like to look upon the syntax of a language as

[4] The italics stand for emphasis.

the finite[5] set of designs in accordance with which the syntactic units of the language are put together to shape an infinite[6] number of forms acceptable as sentences in the language—much as machine parts are assembled into various types of machines according to the engineer's blueprints. A syntactic description of a language may thus be considered as a set of blueprints which represent all these designs in such a way that any of their users may correctly form sentences by following them.

A good engineering blueprint is not presented in the form of disorganized verbal statements like "Put this part here and that part there." This is not done undoubtedly because of its inadequacy. Not even the most unsophisticated type of machine can be built by instructions of this sort, though a layman may feel that such verbal instructions are "easier to understand." Thus, to be really useful, an engineering blueprint is normally a precise graphic representation of every detail conforming to an established form and characterized by a minimum of redundancy and a maximum of explicitness. It may strike the layman as "complicated," but it is what really enables the engineer to construct his machines or buildings.

Like the designs of machines, the designs of sentences are much too intricate to be fully describable by verbal statements alone. Their adequate representation thus requires some medium analogous to that used for engineering blueprints—some way of saying everything within limited space and with the utmost clarity. In other words, it is necessary to have some *consistent* system which makes it possible to represent sentence structures *completely, explicitly*, and most *economically*.

A model must be consistent, to begin with, because it is a truism that only if there is consistency can anything qualify as a system.

As for completeness, explicitness and simplicity, I think their indispensability is something none of us would today question.

Now, as a model of grammatical description, transformational theory is powerful, first of all, because it is truly a consistently formalized model while few other systems have received a comparable degree of consistent formalization.

---

[5] Fininite because, otherwise, it would be impossible for anybody to learn them in a matter of a few years.

[6] Infinite because some designs contain recursive mechanisms.

But what about the other requirements?

Whether or not transformational theory is complete as a model is a question which cannot be fully answered before the theory is applied to more languages and the results fully evaluated. But at the moment, at least, it is more capable of coping with problems than anything else, as demonstrated in Chomsky, 1957 and Postal, 1964.

Nor could there be much doubt that it has greater explicitness than other systems. In fact, just because it is explicit, it has raised many problems which have hitherto been taken for granted or concealed by the lack of explicitness. For example, many co-occurrence restrictions in English, such as those involving animate and inanimate nouns and verbs, were shown up in sharp relief only through attempts to represent them, sometimes without success, in the form of the mechanial rules of a transformational grammar.

As for economy, there can be even less doubt that transformational grammar is superior to other systems. For the machinery of grammatical transformations, the central principle of the theory which has given it its name, has been introduced primarily for simplifying grammatical descriptons, which otherwise would degenerate into absurdly long and repetitious lists of sentence structures. The compact notational system it employs is another important factor leading to greater economy. Even more responsible, perhaps, is the acceptance of simplicity as the basic criterion for choosing between grammars.

Our task in the remaining part of this monograph is to see what bearing our proposed system has on this powerful model in terms of completeness, explicitness and simplicity.

4.2 Advantages of System N

We have seen that system N has at least the following advantages over system S:

1) It considerably shortens the PS section of the grammar by reducing, without any damaging effect, the number of those intermediate stages which system S cannot eliminate without the consequences of losing structural information and making T rules meaningless and inapplicable.

2) In the T section, it is able to do three things which system S cannot do;

a) It makes it possible to apply a T rule merely on the basis of the string in hand, making it unnecessary to "look back" to earlier stings.

b) By means of the device we have called attached initials, it makes explicit specification about the "points of attachment" in T rules much easier than by system S.

c) As shown in the sample syntax, by permitting discontinuous numbering of units in the phrase structure, it makes it easier to avoid such makeshift devices as "Let X=a, b, c, d," which are frequently used under system S to redefine units for particular T rules.

In view of these facts, it seems safe to conclude that system N is superior to system S at least in terms of economy (efficiency, simplicity) and, perhaps to a lesser extent, also in explicitness. But what about completeness?

There is no evidence pointing to either the superiority or inferiority of system N to system S in this respect, because completeness can be proved only by empirical tests. Our judgments at the moment are therefore based on indications alone. In general, I think there is reason to believe that system N is at least no less satisfactory than system S in terms of completeness. For, after all, system N is not radically different from system S, in that both rely on the machinery of grammatical transformation, and that it is very easy to translate one into the other. For example, we may look upon system N just as a version of system S which uses only the symbol X marked by different subscript numbers for differentiating purposes. In giving the subscript numbers, however, we take pains to organize them in such a way that, instead of being arbitrary, they may convey some sort of information. After we do this, we find the information so built in makes a number of PS rules superfluous. So we eliminate those rules. Finally, since every unit is X, with the subscript number serving as the only distinctive feature, we suppress X and deal only with the subscript numbers.

APPENDIX

BASIC TERMINOLOGY

In this appendix are listed the basic new terms used in this description of the proposed numerical symbolism, each defined briefly. Only terms really new are listed.

Number: Any discrete sequence of digits, occasionally supplemented by other material.

Component: Any segment of a number. It is symbolized as C.

Initial: Any component containing the leftmost digit.

Complex number: Any number containing at least one slash.

Attached initial: Any initial of a complex number with the slash occurring at the right end.

Included number: A complex number minus any of its attached initials.

Inclusion: The transformational operation by which a number acquires an attached initial

Separation: The transformational operation by which a complex number loses an attached initial.

BIBLIOGRAPHY

Allen, Harold B. 1964. *Readings in Applied English Linguistics.* 2nd edition. New York: Appleton-Century-Crofts.

Bach, Emmon. 1964. *An Introduction to Transformational Grammars.* New York: Holt, Rinehart and Winston.

Chomsky, Noam. 1957. *Syntactic Structures.* The Hague: Mouton and Company.

__________. 1961. "Some Methodological Remarks on Generative Grammar." *Word* 17:219-39. (Reprinted in Allen, 1964, pp. 173-92.)

__________. 1962. "A Transformational Approach to Syntax." In Hill, 1962a, pp. 124-58.

Elson, Benjamin, and Velma B. Pickett. 1962. *An Introduction to Morphology and Syntax.* Santa Ana, California: SIL.

Francis, W. Nelson, 1958. *The Structure of American English.* New Yark: Ronald Press Company.

Fries, Charles Carpenter. 1952. *The Structure of English.* New York: Harcourt, Brace and World.

Gleason, H. A., Jr. 1961. *An Introduction to Descriptive Linguistics.* Revised edition. New York: Holt, Rinehart and Winston.

Harman, Gilbert H. 1963. "Generatve Grammar Without Transformotian Rules: a Defense of Phrase Structure." *Language* 39:597-616.

Harms, Rober T. 1962. *Estonian Grammar* (identical with *Indiana Publications, Uralic and Altaic Series*, Vol. 12). Bloomington, Indiana.

Harris, Zellig S. 1957. "Co-occurrence and Transformation in Syntactic Structure." *Language* 33:283-340.

Hill, Archibald A. 1958. *Introduction to Linguistic Structures.* New York: Harcourt, Brace and World.

__________.1961. "Grammaticality." *Word.* 17:1-10 (Reprinted in Allen, 1964, pp. 163-72.)

__________. ed. 1962a. *Third Texas Conference on Problems of Linguistic Analysis in English.* Austin, Texas: University of Texas.

__________. 1962b. Review of Lees, 1960a. *Language* 38:434- 44.

Hockett, Charles F. 1954. "Two Models of Grammatical Description." *Word* 10:210-31. (Reprinted in Joos, 1958, pp. 386-99.)

———. 1958. *A Course in Modern Linguistics*. New YorK: Macmillan.

Joos, Martin, ed. 1958. *Readings in Linguistics*. 2nd edition. New York: ACLS.

Lees, Robert B. 1957. Review of Chomsky, 1957. *Language* 33:375-408.

———. 1960a. *The Grammar of English Nominalizations*. Publication 12 of Indiana U. Research Center in Anthropology, Folklore, and Linguistics (identical with *IJAL*, XXVI, No. 3, part II). Bloomington, Indiana: Indiana University.

———. 1960b. "A Multiply Ambiguous Adjectival Construction in English." *Language* 36:207-21.

———. 1960c. "Some Neglected Aspects of Parsing." *Language Learning* 11:171-81. (Reprinted in Allen, 1964, pp. 146-55).

———. 1964. "Transformation Grammars and the Fries Framework." In Allen, 1964, pp. 137-46.

Lees, Robert B., and E. S. Klima. 1963. "Rules for English Pronominalization." *Language* 39:17-28.

Longacre, Robert E. 1960. "String Constituent Analysis." *Language* 36:63-88.

Nida, Eugene A. 1960. *A Synopsis of English Syntax*. Norman, Oklahoma: SIL.

Norman, Arthur. 1958. "An Outline of the Subclasses of the Engish Nominal." *American Speech* 33:83-89. (Reprinted in Allen, 1964, pp. 156-63.)

Palmer, Harold E. 1924. *A Grammar of Spoken English on a Strictly Phonetic Basis*. Cambridge, England: W. Heffer and Sons, Ltd.

Postal, Paul. 1964. *Constituent Structure: A Study of Contemporary Models of Syntactic Description*. Publication 30 of Indiana U. Research Center in Anthropology, Folklore, and Linguistics (identictical with *IJAL*, XXX, No. 1, Part II). Bloomington, Indiana: Indiana Univeraity.

Roberts, Paul. 1962. *English Sentences*. New York: Harcourt, Brace and World.

Smith, Carlota S. 1961. "A Class of Complex Modifiers in English." *Language* 37:343-65.

———. 1964. "Determiners and Relative Clauses in a Generative Grammar of English." *Language* 40: 37-52.

Trager, George L., and Henry L. Smith, Jr. 1951. *An Outline of English Structure*. Occasional Papers 3 of *SiL*. Norman, Oklahoma: Battenburg Press. (Reprinted, Washington, D. C.: ACLS.)

Twaddell, William F. 1963. *The English Verb Auxiliaries*. Second edition

Providence, Rhode Island: Brown University.

Wells, Rulon S. 1947. "Immediate Constituents." *Language* 23:81–117. (Reprinted in Joos, 1958, pp. 186–207.)