PARAPHRASE GRAMMARS
RICHARD M. SMABY

PARAPHRASE GRAMMARS

D. REIDEL PUBLISHING COMPANY / DORDRECHT-HOLLAND
The recent rapid development of transformational grammars has incorporated some strong claims in the areas of semantics and co-occurrence. The earlier structuralists relied on a minimum of information about the meaning of strings of a language. They asked only if strings of sounds were different in meaning—or simply were different words or phrases. Current transformational grammars, on the other hand, set as their goal the production of exactly the meaningful strings of a language. Stated slightly differently, they wish to specify exactly which strings of a language can occur together (meaningfully) in a given order.

The present book purports to show that transformational grammar is independent of the current trends in semantics. I claim that exciting and sophisticated transformational grammars are required for describing when strings of a language mean the same, that is, for describing when strings of a language are paraphrases of each other. This task can be quite naturally limited to a project of much weaker semantic claims than those which are current in transformational linguistics.

I criticize the theoretical viewpoint of Zellig Harris in an important respect—the idea that natural languages are basically "compositional", that is, can be defined in a recursive manner, like languages of logic. However, more important are the aspects of his thinking I have accepted. I follow Harris' concept of a transformation as a mapping of a set into itself. When Harris applies this concept from abstract algebra to linguistics, it becomes a mapping from the set of sentences into itself. This concept of a transformation contrasts with that of Noam Chomsky, for whom transformations are mappings on the path from abstract grammatical structures to phonetic representations. Harris' transformations are designed to have direct empirical correlates, while Chomsky's are not so designed. My own desire for such an empirical correlate is due in part to Harris. His fundamental idea that a transformation is a syntactic operation which preserves co-occurrence relations I wholeheartedly endorse; but I appropriate this idea in the strictest sense and object to the attempt to reduce lengthy and complex co-occurrence relations to atoms of co-occurrence, or to introducing binary transformations, which do not simply preserve, but rather must multiply co-occurrence relations.
The paraphrastic orientation of the present book derives from Henry Hiż, who has been my teacher for four years. I am very much indebted to him for scientific attitude as well as concrete linguistics.

The following work is an informal presentation of certain ideas for grammars of the paraphrase relation. These ideas can be formalized and I have indicated in the appendix how this might be accomplished. I have used only English examples, but I hope the generality of the techniques will be clear.

I wish to thank Henry Hiż and students in linguistics at the University of Pennsylvania for critical reading and valuable suggestions. I owe my wife, Beverly, not only for encouragement and tedious proof-reading, but also for criticism of my linguistics.
CONTENTS

PREFACE

CHAPTER I: INTRODUCTION
1. The Transformational Approach ........................................ 1
2. The Paraphrase Relation .................................................. 1
3. Paraphrase Grammars .................................................... 4
4. Compositional Grammars ............................................... 6
5. Substitution ...................................................................... 7
6. Admission Conditions ..................................................... 9
7. Equivalence ..................................................................... 11
8. Functional Notation ....................................................... 12
9. Summary ........................................................................ 14

CHAPTER II: THE PARAPHRASE RELATION
1. The Study of the Paraphrase Relation ............................... 16
2. Collection of Data ............................................................ 16
3. Selection of Data .............................................................. 18
4. Generalization of the Paraphrastic Relationships ............... 19
5. Systematization of the Paraphrase Relation ....................... 26
6. Summary ........................................................................ 31

CHAPTER III: COMPOSITIONAL GRAMMARS
1. The Compositional Approach .......................................... 32
2. A Simple Compositional Language: P .............................. 32
3. Compositional Grammars and the Co-Occurrence Problem ...... 34
4. ‘Projection Rules’ .......................................................... 40
5. Summary ........................................................................ 50

CHAPTER IV: SUBSTITUTION
1. The Substitution Concept ............................................... 52
2. The Presence of Substitution .......................................... 53
3. A Notation for Substitution ............................................. 56
4. The Generality of Substitution ....................................... 64
CHAPTER I

INTRODUCTION

1. THE TRANSFORMATIONAL APPROACH

The fundamental idea of transformational grammar is that generalizable relationships among strings lie at the base of grammar. Many pairs of English strings are related as (a) and (b):

(1) (a) an elephant escaped from the zoo
    (b) which escaped from the zoo

or as (c) and (d):

(c) an elephant escaped from the zoo
(d) it was an elephant that escaped from the zoo.

The careful generalizing of such relationships is all that is necessary to qualify a grammar as transformational. Some linguists have placed additional constraints on transformational grammars. They have required, for example, that a grammar provide a valid structural description of every string of a language. However, one can drop such a constraint and still write transformational grammars.

An important observation of transformational grammarians is that rules of string relationships such as those exemplified by (1) form a system of interrelated rules. We could not use the relationship of (a):(b) together with (c):(d), for if we did, we would obtain the anomaly

(e) ƒ it was which that escaped from the zoo.

(The ƒ indicates that the following string is non-English.) The system of all rules of a language is very intricate. It is as a system, of course, that a grammar must be evaluated.

Let us then say only that a transformational grammar is a system of rules accounting for the generalizable relationships among strings of a language.

2. THE PARAPHRASE RELATION

The paraphrase relation is a semantic relation which covers many generalizable relationships among strings of a language.
We are all familiar with the common notion of paraphrase. One utterance is a paraphrase of another if they mean the same.

(2)  
(a) *my rabbit ate the roses*
(b) *the roses were eaten by my rabbit*
(c) *I invested a lot of money in a French company*
(d) *I speculated heavily on the Paris bourse.*

In (2) we can understand (a) as a paraphrase of (b) and (c) as a paraphrase of (d). Both paraphrase relationships (a):(b) and (c):(d) are observations about English and as such form part of the basic linguistic data. However, (a):(b) is generalizable, whereas (c):(d) is not. That is, we can find a rule covering the many cases of paraphrase like (a):(b) but no general rule of which (c):(d) is an instance. Thus, out of the basic data we must select a part which we can systematically describe.

The basic data of the study of the paraphrase relation consists of sets of utterances. Each set contains utterances which all mean the same. Each set should be as complete as possible; there should be as many different ways of saying the same thing as possible in each set. We emphasize those utterances which appear to offer generalizable relationships, that is, by checking various sets we notice repeated patterns.

(3)  
(a) *the king bought the grapes*
   *the grapes were bought by the king*
   *it was the king who bought the grapes*
   *it was the king whom the grapes were bought by*
   *it was the king by whom the grapes were bought*
   *it was the grapes which the king bought*
   *it was the grapes which were bought by the king*
   *the king was the one who bought the grapes*
   *the king was the one whom the grapes were bought by*
   *the king was the one by whom the grapes were bought*
   *the grapes were what the king bought*
   *the grapes were what were bought by the king*
   *the king did some buying and the buying was of the grapes*
   *the grapes were the fruit that the king bought*
   *the king was the man who bought the grapes*
   *the ruler bought the grapes*
   *the king bought the small round fruit which the French use for wine*

(b) *the scrawny cat lapped up the milk*
   *the milk was lapped up by the scrawny cat*
INTRODUCTION

it was the scrawny cat which lapped up the milk
it was the scrawny cat which the milk was lapped up by
it was the scrawny cat by which the milk was lapped up
it was the milk which the scrawny cat lapped up
it was the milk which was lapped up by the scrawny cat
the scrawny cat was the one that lapped up the milk
the scrawny cat was the one which the milk was lapped up by
the scrawny cat was the one by which the milk was lapped up
the milk was what the scrawny cat lapped up
the milk was what was lapped up by the scrawny cat
the scrawny cat did some lapping up and the lapping up was of the milk
the milk was the beverage which the cat lapped up
the scrawny cat was the animal that lapped up the milk
the scrawny feline lapped up the milk
the scrawny cat lapped up the white beverage produced by cows.

Paraphrastic sets (a) and (b) of example (3) are not nearly complete, but they give an idea of the kinds of patterns one would find even in paraphrases of very simple utterances. The end of the lists have examples with no obvious formal patterns — that is, patterns involving permutations or deletions of elements of the utterances, or addition or substitution of certain constants. I mentioned earlier that we will not be able to account for all members of any paraphrastic set.

Another important aspect of paraphrastic sets is that many will overlap. That is, a single utterance may belong to many different paraphrastic sets.

(4) (a) the documents were found by the late king
the late king found the documents
the king who came late found the documents
(b) the documents were found by the late king
the late king found the documents
the lately departed king found the documents
(c) the documents were found by the late king
the documents were found near the late king
the documents were found near the king who came late
(d) the documents were found by the late king
the documents were found near the late king
the documents were found near the lately departed king

For example the documents were found by the late king belongs to the four distinct paraphrastic sets of (4). Colloquially we say that such an utterance
is ambiguous. Ambiguity is an important limitation on the writing of transformational rules. This limitation is caused by the fact that the transformed strings are portions of larger texts. For example, in the context

(5) The British king and the Parliament were seated as the Danish king rushed in a full half hour late but as fate would have it the documents to be signed had been mislaid and the treaty was delayed even longer as everyone in the hall including both kings looked under every chair and in every drawer. Fortunately the documents were found by the late king and he was restored to good favor among all present.

the sentence the documents were found by the late king is strongly constrained to be paraphrased as in (a) of (4). This claim can be tested by replacing the sentence in context (5) successively by the final members of (a), (b), (c), and (d). These replacements amount to possible transformational alterations of (5). But only one of the alterations had the desired result of producing a paraphrase of (5). The problem of how to write transformations which will operate only in the proper contexts will require careful selection of members of paraphrastic sets so that transformations will not resolve ambiguities. Each complete paraphrastic set has some member or members which together specify what transformations can be performed on any members of the set. The final members of (a), (b), (c), and (d) of (4) approximate this feature for the present purposes. For example, if (5') were like (5) but had the king who came late found the documents instead of the documents were found by the late king, we would be safe in altering (5') according to the other two members of (a) in (4).

3. PARAPHRASE GRAMMARS

The present work is concerned with transformational grammars of the paraphrase relation. A transformational grammar of the paraphrase relation of a language is a system of rules accounting for the generalizable relationships among strings of a language which preserve the paraphrase relation. Since we are dealing only with transformational grammars, let us abbreviate 'transformational grammar of the paraphrase relation' as 'paraphrase grammar.' A rule in a paraphrase grammar is essentially a set of directions for altering a string in its domain of application so that the resultant string is a paraphrase of the original. Of course, the more general we can write the rule, that is, the more strings which we can allow to enter the transformation without also allowing strings which result in anomalies, the more successful our grammar is.
The passive transformation provides an example

\[(6)\]

\[
\begin{array}{c}
S_1 \\
N_1 \quad V \\
N_2
\end{array} \quad \longrightarrow \quad \begin{array}{c}
S_2 \\
N_2 \quad is \quad Ved \\
by \quad N_1
\end{array}
\]

A string which functions as a sentence in a context and which is composed of a noun \(N_1\) followed by a verb \(V\) (in the present tense) followed by a second noun \(N_2\) can be altered to form \(S_2\) so that \(N_2\) is followed by \(is\), then \(Ved\) and then \(by\) \(N_1\). The result is that \(S_2\) functions as a sentence in the context and the text resulting from replacing \(S_1\) by \(S_2\) is a paraphrase of the original text. An example is

\[(7)\]

\[
\text{we will invite John even though John dislikes our dog} \quad \longrightarrow \quad \text{we will invite John even though our dog is disliked by John.}
\]

We are using very simple common notation for the informal discussion. There are much more suitable notations but they are more involved and must wait for later in the book.

Rule (6) is a very general rule covering many cases. In fact, it is too general, because strings which meet the input requirements will result in anomalies.

\[(8)\]

\[
\begin{array}{c}
John \quad resembles \quad a \quad famous \quad general
\end{array} \quad \longrightarrow \quad \begin{array}{c}
\exists \quad a \quad famous \quad general \quad is \quad resembled \quad by \quad John
\end{array}
\]

Example (8) makes it clear that not all verbs will fit the passive transformation. The precise restriction must be indicated in the transformational rule. Rules of a paraphrase grammar can be incorrect either by changing correct English into non-English or by changing a text into a non-paraphrase of itself, as indicated in the discussion of examples (4) and (5). These failings of a rule are the primary tests we will employ in evaluating a paraphrase grammar.

In all cases in a paraphrase grammar a rule will alter one string into another string. A rule of paraphrase will not combine two or more strings into a resultant string. That is, a paraphrase grammar makes no attempt to construct a sentence or text out of elements of a language; it only tries to reform existing utterances or texts. For this reason we will take as our object of interest in paraphrase grammars whole texts. There is no motivation to take partial strings of a language. A noun, a verb, or a sentence generally occurs in a context; hence, a paraphrase grammar will describe them as elements in a context.
4. COMPOSITIONAL GRAMMARS

In contrast to paraphrase grammars, some current transformational grammars do claim to compose complicated sentences out of simpler elements of a language. They seek general rules which prescribe for example that the elements

(9) \( \text{the boy, sincerity, frightens} \)

can be combined in the fashion

(10) \( \text{sincerity frightens the boy} \)

but not in the manner

(11) \( \not\exists \text{the boy frightens sincerity}. \)

Or they seek rules which construct

(12) \( \text{the boy who was tall left the door open} \)

out of

(13) \( \text{the boy was tall} \)

and

(14) \( \text{the boy left the door open} \)

but which will not at the same time yield

(15) \( \not\exists \text{the spring which is broken in the mattress dried up years ago} \)

from

(16) \( \text{the spring is broken in the mattress} \)

and

(17) \( \text{the spring dried up years ago.} \)

They wish to derive all compound sentences but avoid such senseless strings as

(18) \( \not\exists \text{John opened a door and organic chemistry was required of all students, because if seventeen were not a prime, then the dairy association would have to back down. Thus, we can conclude that television will have a beneficial effect on the nation's young.} \)

We will survey the attempts of transformational grammarians to solve such difficulties in the sequel. I wish now simply to emphasize the compositional aspect of some current transformational grammars and point out some problems.
Such grammars can hardly be grammars of the paraphrase relation in any obvious way. Combining two strings does not generally yield a paraphrase of one of the original strings. We can safely distinguish those transformational grammars from paraphrase grammars by calling them "compositional" grammars.

Compositional grammars are modeled after the very elementary grammars of formalized languages of mathematics and logic. Formalized languages are defined in such a way that any conjunction of elementary sentences is a meaningful sentence and is either true or false. As we have already indicated, the parallel with a natural language such as English breaks down in the very respect of arbitrariness of conjunctions. I shall challenge compositional grammarians that if their grammars combine perfectly sensible strings into nonsense, then they have failed to persuade me that language is compositional in any sense which linguistics can describe.

5. SUBSTITUTION

The global operation of a rule in a paraphrase grammar is the replacement of a part of a text by an alteration of that part. This replacement may take place at any location in the text. For example, any one of a number of sentences in the text may undergo an alteration from the active to passive voice. In paraphrase grammars such an alteration occurs in the text while intact; whereas in a compositional grammar the component string is altered before it is combined with the remainder of the strings which compose the text. It is particularly obvious in the case of paraphrase grammars that the location of the string to be altered must be specified, otherwise the directions for the transformation of the text would be incomplete.

Let us develop the rudiments of a notation for substitution. We reserve curved parentheses '(' and ')' exclusively for substitution. The global operation of replacing one string by another in a text can be represented as

\[ S_1(X) \rightarrow S_1(X') \]

That is, given a context \( S_1 \) which together with the string \( X \) forms the text \( S_1(X) \), replacing \( X \) by the string \( X' \) in the context \( S_1 \) yields a text \( S_1(X') \). The \( S \) indicates that the whole string \( S_1(X) \) is a sentence. For example, if \( S_1 \) is

\[ \text{John likes apples and so} \]

then the relationship of (21) and (22)

(21) \( \text{John likes apples and so John buys apples} \)

(22) \( \text{John likes apples and so apples are bought by John} \)
can be described as in

(23) \( S_1(John \text{ buys apples}) \rightarrow S_1(\text{apples are bought by John}) \).

We are actually using a degenerate case of the general substitution relation. Generally substitution replaces each occurrence of a string by a second string. Above we were concerned with replacing only one occurrence of a string. We indicate this degenerate case by a subscript \( '0' \) on the parentheses. Thus, (23) should read

(24) \( S_1_0(John \text{ buys apples}) \rightarrow S_1_0(\text{apples are bought by John}) \).

A single replacement of one string by another is hardly justification for introducing the general concept of substitution into the foundations of grammar. In fact, the motivation is much stronger. Using the above notation we can compactly describe co-ordinated operations. The formation of the relative clause in English is an example of an operation requiring many co-ordinated operations. In particular, the formation of compound clauses often requires a number of replacements.

(25) A complaint was registered but the complaint was minor
(26) which was registered but which was minor

The directions for changing (25) to (26) are simply “replace all occurrences of a/the complaint by which”.

Let \( S_2 \) be the context in which a/the complaint occurs twice. That is, \( S_2 \) can be pictured as

(27) ---- was registered but ---- was minor.

We write the change from (25) to (26) as

(28) \( S_2_2(a/the \text{ complaint}) \rightarrow A_2(\text{which}) \).

The \( A \) indicates that the whole string \( A_2(\text{which}) \) is an adjective, just as the \( S \) on the left of the arrow indicates that \( S_2(a/the \text{ complaint}) \) is a sentence. The subscript ‘2’ identifies \( S_2 \) and \( A_2 \) and in fact, both can be pictured as (27); the string that fills the slots distinguishes the grammatical category of the overall string. Of course, it is the function which the string plays in the larger text which really determines its grammatical category.

I wish to stress that the substitution concept and notation bring to the fore the co-ordinated nature of the formation of the relative clause in English. A compositional grammar would not require an explicit substitution concept, and, in fact, existing compositional grammars, to my knowledge, make no use of a general concept of substitution, although they could do
so to some advantage. They need some devices to co-ordinate the formation of the two relative clauses

(29) which was registered

(30) which was minor

later to be conjoined into the larger clause (26). In particular, they need to be explicit in requiring that the appropriate noun in each of the two source sentences (31) and (32) be replaced by a relative pronoun.

(31) a complaint was registered

(32) the complaint was minor

Of course, a relative clause is not limited to only two source sentences. The advantage of the substitution concept is particularly clear when one wants to show not only that, but how (25) and (26), for example are related.

The substitution notation yields the relationship directly, whereas non-substitutional alternatives would likely be metarules telling when and how often to apply a given relative clause rule, which could be applied to components such as (29) and (30) but not to (25) as a whole.

The basic claim of this and subsequent discussion of substitution is that the substitution concept makes the writing and understanding of a grammar easier, no matter whether it is a compositional or paraphrase type.

6. ADMISSION CONDITIONS

Any except the most trivial rules of a grammar apply properly to only certain strings of a language. A rule fails to apply to a given string because the string does not satisfy the conditions of the rule. A string may simply have the wrong 'shape' for the rule. For example,

(33) Mrs. Wilson writes to her Congressman daily

cannot undergo the transformation

(34) $N_1 V N_2 D \rightarrow N_2 is Ven by N_1 D$

simply because it has the shape $N_1 V P N_2 D$ instead of the required $N_1 V N_2 D$. The basic notation of a grammar automatically excludes certain strings from being input to a given rule.

However, more often than not the basic notation is not sufficiently discriminating. For example, (34) will apply to both

(35) Mrs. Wilson writes Congressman Walsh daily
and

(36) *Mrs. Wilson writes herself daily.*

From (35) we get

(37) *Congressman Walsh is written by Mrs. Wilson daily.*

But from (36) we obtain

(38) *herself is written by Mrs. Wilson daily.*

Often, indicating sub-categories of word classes in detail suffices to solve such a problem. In the above case, \( N_2 \) of (34) could be subscripted to indicate a class of nouns exclusive of reflexive pronouns. Then (36) would not qualify as an input to (34) and the anomaly (38) would not result. Sub-categorization could perhaps still be claimed as part of the fundamental notation.

In general, however, the situation is not so simple, as witnessed by the following pairs:

(39) (a) *Mrs. Wilson writes her own Congressman daily*
    (b) *her own Congressman is written by Mrs. Wilson daily*

(40) (a) *Mrs. Wilson visits the office of her own Congressman daily*
    (b) *the office of her own Congressman is visited by Mrs. Wilson daily.*

The source of the difficulty for rule (34) apparently can be imbedded well into the structure of \( N_2 \).

One could attempt to solve the difficulty by imposing on the rules an order in which they operate. In the above examples the introduction of reflexive pronouns would be allowed only after the introduction of the passive voice and other order changing rules. A simple global ordering of rules will prove too crude. For example, the introduction of reflexives for one noun before the introduction of a passive may be necessary and harmless. More sophisticated orderings of rules with respect to specific operands may be feasible. We will discuss this point at length later in the text.

Any procedure used to solve the difficulty of rule (34) will be complex in the details. However, some procedures are more obvious than others. In particular, the most obvious solution to the above difficulty is to forbid a rule such as (34) to operate unless certain detailed conditions are satisfied by the possible input. These conditions extend way beyond any basic notation suitable for grammarians. A string may have the proper shape to enter a transformation but unless it also satisfies a list of other conditions, we say that the string is not "admitted" to the transformation. Hence, the condi-
tions are called ‘admission conditions’. In the case of rule (34) we may simply append the condition:

\[(41) \quad N_2 \text{ contains no occurrence of a reflexive for } N_1.\]

We could then write (34) as

\[(42) \quad 41: N_1 V N_2 D \quad \rightarrow \quad N_2 \text{ is } Ven \text{ by } N_1 D.\]

That is, subject to condition (41), a sentence of the form \(N_1 V N_2 D\) may be transformed to a sentence of the form \(N_2 \text{ is } Ven \text{ by } N_1 D\).

It is easy to express informally a condition such as (41). It is more difficult to define it precisely. The careful definition of admission conditions is often a challenging and insight providing task. Furthermore, the use of admission conditions seems to me a much more direct and, hence, natural way to attack the details of grammatical rules.

7. EQUIVALENCE

In a text many physically different strings may fulfill the same semantic function. For example, in

\[(43) \quad (a) \text{ the boy from Brazil} \]

\[(b) \text{ the Brazilian boy} \]

\[(c) \text{ the boy who is from Brazil} \]

\[(d) \text{ that boy} \]

\[(e) \text{ he} \]

(a)–(e) may at different times in a text all refer to the same boy. This fact would be very important to many transformations operating on the text. If an instruction included forming a relative clause with respect to \(\text{the boy from Brazil}\) out of the string

\[(44) \quad \text{the boy who was from Brazil welcomed the Professor and then the boy from Brazil immediately left,} \]

then, without taking equivalence into account, either we would get the anomaly

\[(45) \quad \exists \text{ the boy who was from Brazil welcomed the Professor and then who immediately left} \]

or the relative clause rule would be blocked in its application to (44) or the rules could possibly have been so ordered that the relative clause rule would not be in order at the point of derivation that is (44). Again, it seems most obvious to simply treat \(\text{the boy from Brazil}\) and \(\text{the boy who is from} \)
Brazil as equivalent in the sense that both are subject to any syntactic substitution involving either one of them. Thus, under the equivalence concept our rule would transform (44) into the perfectly correct relative clause

(46) who welcomed the Professor and then who immediately left.

As examples (43)–(46) hint, equivalence of strings will be a very important relation in a grammar using the substitution concept.

8. FUNCTIONAL NOTATION

The notation I have employed so far and which I will continue to use in the informal portions of the present work is a very simple sequence of word classes. The primary exception is the use of parentheses ‘(, )’ to signify the substitution relation. I will also make use of simple tree diagrams where clarification of structure is required. The word-class notation has the advantage of being fairly readable, at least for informal discussions.

The word-class notation does, however, have serious drawbacks for any precise and explicit grammatical studies. Strings of a language cannot in general be characterized as simple sequences of word classes.

One difficulty is that plurals of nouns, for example, may not be simply a matter of an added particle such as -s. When doing a grammar of English we may write Ns to characterize the plural of a noun as the noun N followed by the plural particle s. Linguists will usually accept Ns as shorthand for a more complex situation. But perhaps the shorthand is somewhat misleading. If all English nouns formed plurals as

(47)   whip        whips
       blouse      blouses
       wind        winds
       fish        fish

then the objection would be very minor indeed; -s could be taken as a class sign over: /s/, /æz/, /z/, and /ð/ together with instructions associating proper members of -s with each noun. However, in English there are many nouns which form plurals as

(48)   goose       geese
       knife       knives
       mouse       mice.

At this point -s becomes entirely a set of instructions for changing various nouns to their plurals, which is merely to say that -s is a function from the set of nouns to the set of plurals of nouns. -s may be analyzed and related to
similar functions in interesting phonological ways, but to represent it as a simple additive function \(-s\) would be misleading. A very general functional representation should be used. For example, we could write \(f[\text{plural}] [x]\) to indicate the plural function. A slightly more sophisticated notation could indicate the grammatical category of the value of the function.

\[
N[\text{plural}] \ [\text{mouse}] = \text{mice}
\]

(49) indicates the phrase \textit{mice} plays a nominal role in the situation being described. If we wanted to indicate the plural of just any noun, we could write

\[
N[\text{plural}] \ [N].
\]

A functional notation is also useful for the notorious non-segmental features of a language, such as intonation patterns. We could distinguish, for example, the question

\[
\text{Wilson was a violent man?}
\]

from the statement

\[
\text{Wilson was a violent man.}
\]

through the distinction between the general characterizations

\[
S[?] \ [N_1, V_2]
\]

and

\[
S[.] \ [N_1, V_2].
\]

(53) describes a sentential function of the question type which has two arguments: the first argument being a noun, the second a verb. (54) describes a sentential function which differs from (53) in being of the statement type. ‘Sentential function’ means ‘functions whose values are sentences’. The parameter on the S identifies the function while the S merely indicates the grammatical category of its values.

A very interesting use of a functional notation is to indicate dependencies within a string. One example is the subject-verb dependence. In the present tense, if the subject of a sentence is \textit{mouse}, we need the verb form \textit{scampers}:

\[
\text{a mouse scampers.}
\]

With the subject \textit{mice} we need the form \textit{scamper}:

\[
\text{mice scamper.}
\]

In English there are only a couple of features of the subject that are im-
important in determining the verb form, and there are number and person in the case of pronouns. The individuality of the word is not necessary, but the noun does carry the information required, so we may as well use it, even at the price of redundancy in structural descriptions of strings. We can write a verb function which depends on a noun:

(57) \[ V[\text{finite}] [N] [\text{tense}] [X] \]

The single symbol function sign ‘f’ is replaced by a complex symbol \( V[\text{finite}] [N] [\text{tense}] \) indicating that the precise function involved depends on the choice of a specific noun \( N \) and tense for parameters. A parametric function is merely one which requires additional information, parameters, beyond that provided by its arguments, in order to give a value. Thus, the plural form *scamper* in sentence (56) would be described as

(58) \[ V[\text{finite}] [\text{mice}] [\text{present}] [\text{scamper}], \]

which is the verb root *scamper* when made into a present tense verb with the subject *mice*. Replacing *mice* by the functional description \( N[\text{plural}] [\text{mouse}] \) we have

(59) \[ V[\text{finite}] [N[\text{plural}] [\text{mouse}]] [\text{present}] [\text{scamper}]. \]

Sentence (56) might then be written as

(60) \[ S[. ] [N[\text{plural}] [\text{mouse}], V[\text{finite}] [N[\text{plural}] [\text{mouse}]] [\text{present}] [\text{scamper}]] \]

(60) provides an example of at least a partial functional description in one notational style.

9. Summary

I have attempted in this introduction to distinguish and survey the central ideas of transformational grammar which are to be discussed in the present work. I gave a deliberately broad characterization of a transformational grammar as a system of rules describing the generalizable relationships among strings of a language. The paraphrase relation among texts of a language is to be brought to the forefront since the rules of a paraphrase grammar transform texts into their paraphrases. Paraphrase grammars contrast with compositional grammars in that the paraphrase type makes no attempt to ‘produce’ exactly the meaningful strings of a language or to compose strings out of their elements. The importance of a general concept of substitution is most clear in paraphrase grammars. In whatever type grammar it is employed the central function of substitution is to co-ordinate conceptually many smaller operations, in particular, multiple replacements.
I have argued and will argue that conditions for admission of strings to transformations and equivalences among strings in a text are direct and suitable means for adjusting the various grammatical rules to each other. I recommend a functional notation because it exhibits features of a language which are glossed over or hidden by other notations; the most unusual aspect of the notation is that dependence between strings, such as subject-verb governance, is exhibited right at the surface. Indeed, my goal is to make the grammar as visible as possible – in order that when a reader looks at a rule he knows exactly when the rule applies to a string and what alterations it makes on the string.
CHAPTER II

THE PARAPHRASE RELATION

1. THE STUDY OF THE PARAPHRASE RELATION

The paraphrase relation is the acknowledged subject of the present work. Subsequent chapters can be read as merely the development of the tools necessary for the characterization of the paraphrase relation in a given language. The linguist's involvement with the paraphrase relation of a language consists of three activities. He must first gather data about paraphrase, that is, he must gather paraphrases from native speakers of the language. He must then sort out the paraphrases which promise to yield patterns from those which seem highly individual. For example, a paraphrase set such as

(1) \( \begin{align*} & \text{my rabbit ate the roses} \\ & \text{the roses were eaten by my rabbit} \end{align*} \)

is suggestive of a relationship which can be generalized beyond that one pair of sentences, while

(2) \( \begin{align*} (a) & \text{i bought the newest Oldsmobile on the lot} \\ (b) & \text{the Oldsmobile dealer sold me his newest car} \end{align*} \)

offers no such relationship. (2) would then be set aside and the linguist would concentrate his attention on data such as (1). The third activity is the generalizing and systematizing of the observed relationships.

2. COLLECTION OF DATA

Unfortunately, I cannot offer any sophisticated experimental techniques for eliciting from the native speaker responses concerning paraphrase. My own technique is simply to ask the native speaker if a number of sentences all mean the same. Some speakers are more rigid and some more casual. Some speakers will judge strings as non-paraphrases on what, for lack of clear understanding, we may call stylistic grounds. The strings say generally the same, but they have a different slant or emphasis. For example,
(3) (a) an elephant escaped from the zoo
(b) it was an elephant that escaped from the zoo

might not be paraphrases for such persons because (b) emphasizes elephant while (a) places no such emphasis.

Other speakers are quite lenient in judging paraphrase. They might judge the strings in (2) as paraphrases, even though the information content of (a) and (b) is quite different. I have not experimented with tests for the purpose of sorting out the precisely proper sense of paraphrase which is needed. I must simply rely on my own informal judgments.

I am prepared, however, to discuss some interesting theoretical aspects of paraphrase which have a direct bearing on the collection of data.

Perhaps the easiest way to collect paraphrastic data is to inquire of a native speaker whether one string means the same as another. That is, the judgment we elicit concerns pairs of strings of a language. For example, we find from a native speaker of English that both strings of (4)

(4) we got to the mill by the little stream
    we got to the mill using the little stream

mean the same, are paraphrases of each other. That is, both strings can be read as asserting

(5) we got to the mill by means of the little stream.

The same native speaker will tell us that the strings of (6) are paraphrases of each other.

(6) we got to the mill using the little stream
    we got to the little stream's mill

They are paraphrases in the sense that both can be interpreted as saying

(7) we got to the mill which was using the little stream (for power).

And again our native speaker would affirm that the pair of strings of (8) form a paraphrastic set.

(8) we got to the little stream's mill
    we got to the mill by the little stream

The reading they share is

(9) we got to the mill which was located near a little stream.

Thus, we have taken all possible combinations in pairs of the elements of (10)

(10) we got to the mill by the little stream
    we got to the mill using the little stream
    we got to the little stream's mill
and have found that in pairs they are paraphrases. However, the same native speaker tells us that the strings of (10) do not all mean the same.

I can illustrate the same point schematically

(11) Strings | Readings
   A     a  b
   B     b  c
   C     a  c

Thus, the first row shows that string A has readings a and b. From (11) we can see that strings A and B are paraphrases by virtue of sharing the reading b. B and C share c, and A and C share a. But A, B, and C together share no reading. Unfortunately objects such as a, b, and c, that is, readings, are very elusive. A language need not have strings which fill the role of readings. In (11) I have merely reified some paraphrase relationships for the purpose of visualization.

Thus, there is no global reduction of the concept “all mean the same” to the concept “both mean the same.” For a given concrete language there may be such a reduction procedure. For example, a language may have a syntactic form of strings which is unambiguous, at least for the purpose of performing syntactic transformations. Such strings could then be the basis for comparison for all others. But later we will discuss this point further. For the purposes of collection of data we would be advised to ask a native speaker whether all the strings of a set mean the same.

3. SELECTION OF DATA

The data one obtains concerning paraphrase is rather unmanageable. There is a whole spectrum of kinds of paraphrases, from the very mechanical such as (1) above to the very imaginative such as (2).

I have no difficulty deciding to try to formalize (1) while forgetting (2). I have no hopes of fitting (2) into a generalized relationship. However, the in-between cases are more difficult. How generalizable is the relationship exemplified by

(12) the boy from Harvard
    the Harvard boy?

It doesn’t extend to

(13) the girl from home
    AsStream the home girl
But I am still tempted to formalize it. I would like to formalize operations with classifiers, but I am not very confident that more than a solution of degrees is possible.

(14) when the dog barked the cat jumped and we never saw that cat again
    when the dog barked the cat jumped and we never saw that animal again

Concerning general synonymy, as in

(15) John was a bachelor
    John was an unmarried man

I am very skeptical.

I have simply used this section to give the reader an idea of my attitudes. Until general agreement is reached in linguistics as to what data must be accounted for, the problems one chooses to tackle depend greatly on the individual. I will select the data and problems which involve the least amount of semantical analysis of individual words.

4. Generalization of the Paraphrastic Relationships

Of course, the procedures of collecting, sorting and generalizing data are not independent activities. In particular, the scientist always has his eye on tentative generalizations, and so his collection of data and sorting of the data will be highly determined by what he expects to find. Thus, one must take care to make his generalizations carefully. It is the generalized relationships that one must account for with a system of hypotheses. To leap to large hypotheses is the most fun, and this book has its quota of them. However, I will try later in the book to write one transformation very precisely.

The first thing we notice in many paraphrastic sets is that certain strings recur in all members of a set. For example in the set

(16) the man whom we met left early
    the man we met left early
    the man who was met by us left early
    the man whom we met took leave early
    early leave was taken by the man whom we met
    it was the man whom we met that took early leave
    we met a man and the man left early
we note that the strings

\[(17) \quad \textit{man} \quad \textit{met} \quad \textit{early}\]

are repeated in all the sentences. We then wonder whether other strings can fill those same slots or take the same "role" in a paraphrastic set. We find that

\[(18) \quad \textit{girl} \quad \textit{invited} \quad \textit{late}\]

respectively can be used in place of the strings of (17). Thus, we are encouraged to generalize (16) to

\[(19) \quad \textit{the} \ 1 \ \textit{whom} \ \textit{we} \ 2 \ \textit{left} \ 3 \]
\[
\textit{the} \ 1 \ \textit{we} \ 2 \ \textit{left} \ 3
\]
\[
\textit{the} \ 1 \ \textit{who} \ \textit{was} \ 2 \ \textit{by} \ \textit{us} \ \textit{left} \ 3
\]
\[
\textit{the} \ 1 \ \textit{whom} \ \textit{we} \ 2 \ \textit{took} \ \textit{leave} \ 3
\]
\[
3 \ \textit{leave} \ \textit{was} \ \textit{taken} \ \textit{by} \ \textit{the} \ 1 \ \textit{whom} \ \textit{we} \ 2
\]
\[
\textit{it} \ \textit{was} \ \textit{the} \ 1 \ \textit{whom} \ \textit{we} \ 2 \ \textit{that} \ \textit{took} \ 3 \ \textit{leave}
\]
\[
\textit{we} \ 2 \ \textit{a} \ 1 \ \textit{and} \ \textit{the} \ 1 \ \textit{left} \ 3
\]

Let us attempt to read (19) as claiming that for all strings, 1, 2, and 3, the sentences of (19) all mean the same. That is, no matter what strings we pick for the roles 1, 2 and 3 the result will be seven sentences which are all paraphrases of each other. This claim is, of course, patently false. For example, if we set

\[(20) \quad 1 = \textit{appreciate} \]
\[
2 = \textit{taxi cab} \]
\[
3 = \textit{brown} \]

we obtain utter nonsense.

We could simply read (19) as saying that for significantly many and significantly varied strings 1, 2, 3 the sentences of (19) all mean the same. But let us rather start with the strongest claim and weaken it progressively. However, each modification will be stated in universal form. That is, it will begin with the words "for all strings...".

The most obvious step is to subclassify the words of English. Let us do so in a fairly crude and conventional way at first, increasing the complexity of the classifications only as formal manipulations dictate. We require that
I be a noun: \( N_1 \), that 2 be a verb: \( V_2 \) and that 3 be an adverb: \( D_3 \). (19) is now written

\[
\begin{align*}
& (21) \quad \text{the } N_1 \text{ whom we } V_2 \text{ left } D_3 \\
& \quad \text{the } N_1 \text{ we } V_2 \text{ left } D_3 \\
& \quad \text{the } N_1 \text{ who was } V_2 \text{ by us left } D_3 \\
& \quad \text{the } N_1 \text{ whom we } V_2 \text{ took leave } D_3 \\
& \quad D_3 \text{ leave was taken by the } N_1 \text{ whom we } V_2 \\
& \quad \text{it was the } N_1 \text{ whom we } V_2 \text{ that took } D_3 \text{ leave} \\
& \quad \text{we } V_2 \text{ a } N_1 \text{ and the } N_1 \text{ left } D_3
\end{align*}
\]

The triple

\[
(22) \quad N_1 = \text{car} \\
V_2 = \text{reserved} \\
D_3 = \text{fast}
\]

causes an anomaly. The phrase \textit{car whom} is not acceptable, though we know that \textit{car which} is correct. In general there are two subclasses of nouns: those which occur with \textit{who} or \textit{whom} and those which occur with \textit{which}. We could subclassify the nouns of English further and have two generalizations: the one would have \( N_1 \), say, where the other had \( N_2 \). However, it is not very pleasing to leave two such similar patterns unrelated. A more satisfactory solution is found by regarding the choice of \textit{who} or \textit{which} as a function depending on \( N_1 \). Thus, we write

\[
(23) \quad N[\text{wh}][\text{man}] = \text{who} \\
N[\text{wh}][\text{girl}] = \text{who} \\
N[\text{wh}][\text{car}] = \text{which}.
\]

Of course, the full definition of the function \( N[\text{wh}] \) is a recursive definition to take account of more complex arguments such as

\[
(24) \quad N[\text{wh}][\text{Russian student}] = \text{who} \\
N[\text{wh}][\text{airplane carrying the President's advisors}] = \text{which}.
\]

Our generalization now has the form

\[
(25) \quad \text{the } N_1 \text{ } N[\text{obl}][N[\text{wh}][N_1]] \text{ we } V_2 \text{ left } D_3 \\
\quad \text{the } N_1 \text{ we } V_2 \text{ left } D_3 \\
\quad \text{the } N_1 \text{ } N[\text{wh}][N_1] \text{ was } V_2 \text{ by } N[\text{obl}][\text{we}] \text{ left } D_3 \\
\quad \text{the } N_1 \text{ } N[\text{obl}][N[\text{wh}][N_1]] \text{ we } V_2 \text{ took leave } D_3 \\
\quad D_3 \text{ leave was taken by the } N_1 \text{ } N[\text{obl}][N[\text{wh}][N_1]] \text{ we } V_2 \\
\quad \text{it was the } N_1 \text{ } N[\text{obl}][N[\text{wh}][N_1]] \text{ we } V_2 \text{ that took } D_3 \text{ leave} \\
\quad \text{we } V_2 \text{ a } N_1 \text{ and the } N_1 \text{ left } D_3
\]
I have also added the function \( N[obl] \) to take care of the “oblique” cases in English, excluding genitive.

\[
\begin{align*}
\text{N[obl] [who]} & = \text{whom} \\
\text{N[obl] [which]} & = \text{which} \\
\text{N[obl] [he]} & = \text{him} \\
\text{N[obl] [she]} & = \text{her} \\
\text{N[obl] [it]} & = \text{it} \\
\text{N[obl] [they]} & = \text{them} \\
\text{N[obl] [we]} & = \text{us} \\
\text{N[obl] [you]} & = \text{you} \\
\text{N[obl] [N]} & = \text{N for all other N.}
\end{align*}
\]

Thus,

\[
(27) \quad \text{N[obl] [N[wh] [man]]} = \text{N[obl] [who]} = \text{whom}
\]

I have chosen to use a functional notation in making the generalizations about the paraphrase relation. The notation is not the central question here. The reader may use any notation he pleases in order to make careful generalizations.

In surveying the various paraphrastic sets of English we notice many which are in the pattern of (25) except for having \( A_3 \) in some places where we have \( D_3 \) and having a different form of \( V_2 \) in some of the slots. \( A_3 \) is a symbol for any adjective. For example,

\[
(28) \quad \begin{align*}
\text{the person whom we like leaves quickly} \\
\text{the person we like leaves quickly} \\
\text{the person who is liked by us leaves quickly} \\
\text{the person whom we like takes leave quickly} \\
\text{quick leave is taken by the person whom we like} \\
\text{it is the person whom we like that takes quick leave} \\
\text{we like a person and the person leaves quickly.}
\end{align*}
\]

The change of tense in (28) has resulted in the alteration of \( left \) to \( leaves \) and \( was \) to \( is \). Let us provisionally write

\[
(29) \quad \begin{align*}
\text{V[pres] [leave]} & = \text{leaves} \\
\text{V[past] [leave]} & = \text{left} \\
\text{V[pres] [be]} & = \text{is} \\
\text{V[past] [be]} & = \text{was.}
\end{align*}
\]

That is, \( leave \) in the present tense has the form \( leaves \) and in the past tense the form \( left \). Similarly for \( be \). Of course, we have not taken account of the plural and singular alternations of the verb in English.
The *quick/quickly* alternation can also be taken care of in functional notation.

\[(30)\] \[D[ly]\] [quick] = *quickly*  
\[D[ly]\] [early] = *early*

Because of observing patterns like

\[(31)\] prefer the magazine/the magazine which is preferred/the preferred magazine  
dislike the boy/the boy who is disliked/the disliked boy  
lost the war/the war which was lost/the lost war  
buy the candy/the candy which is bought/the bought candy

and comparing them to

\[(32)\] the rug which is green/the green rug

we shall attempt to treat the past participle of the verb as adjectival.

\[(33)\] \[A[ed]\] [meet] = *met*  
\[A[ed]\] [like] = *liked*  
\[A[ed]\] [reserve] = *reserved*  
\[A[ed]\] [take] = *taken*

The generalization now becomes

\[(34)\] the \[N_{1}\] \[N[obl]\] [\[N[wh]\] [\[N_{1}\]]] we \[V[t_{4}]\] [\[V_{2}\]] \[V[t_{4}]\] [leave]  
\[D[ly]\] [\[A_{3}\]]  
the \[N_{1}\] we \[V[t_{4}]\] [\[V_{2}\]] \[V[t_{4}]\] [leave] \[D[ly]\] [\[A_{3}\]]  
the \[N_{1}\] \[N[wh]\] [\[N_{1}\]] \[V[t_{4}]\] [be] \[A[ed]\] [\[V_{2}\]] by \[N[obl]\] [\[we\]] \[V[t_{4}]\] [leave] \[D[ly]\] [\[A_{3}\]]  
the \[N_{1}\] \[N[obl]\] [\[N[wh]\] [\[N_{1}\]]] we \[V[t_{4}]\] [\[V_{2}\]] \[V[t_{4}]\] [take] \[N[Vn]\] [leave] \[D[ly]\] [\[A_{3}\]]  
\[A_{3}\] \[N[Vn]\] [leave] \[V[t_{4}]\] [be] \[A[ed]\] [take] by the \[N_{1}\] \[N[obl]\]  
\[N[wh]\] [\[N_{1}\]] we \[V[t_{4}]\] [\[V_{2}\]]  
it \[V[t_{4}]\] [be] the \[N_{1}\] \[N[obl]\] \[N[wh]\] [\[N_{1}\]] we \[V[t_{4}]\] [\[V_{2}\]] that \[V[t_{4}]\] [take] \[A_{3}\] \[N[Vn]\] [leave]  
we \[V[t_{4}]\] [\[V_{2}\]] a \[N_{1}\] and the \[N_{1}\] \[V[t_{4}]\] [leave] \[D[ly]\] [\[A_{3}\]]

In the above generalization I have also added a verbal nominalization function

\[(35)\] \[N[Vn]\] [leave] = *leave*  
\[N[Vn]\] [speak] = *speech*  
\[N[Vn]\] [dictate] = *dictation* and so forth.
I note that the replacement of *us* by $N[obl][we]$ depends on generalizing *we* to any noun $N_s$.

One very important syntactic feature of English we have ignored so far. In fact, (34) is not really a generalization of (28), because the verb *like* is in the plural form due to the plural subject *we* and we have not provided for the differences caused by singular and plural. For example, consider the paraphrastic set

(36)  
- *the people whom we like leave early*
- *the people we like leave early*
- *the people who are liked by us leave early*
- *the people whom we like take leave early*
- *early leave is taken by the people whom we like*
- *it is the people whom we like that take early leave*
- *we like some people and the people leave early.*

As we have developed our notation so far, (34) cannot generalize both (28) and (36). $V[\text{pres}][\text{be}]$ would have to do double duty in entry number three of both sets. In (28) $V[\text{pres}][\text{be}]=\text{is}$ and in (36) $V[\text{pres}][\text{be}]=\text{are}$. It is clear to us that the same alternation occurs for all verbs. The subject of the verb together with the tense determine the form the verb takes in a sentence. Thus, we should rewrite (29) as

(37)  
- $V[\text{person}][\text{pres}][\text{leave}]=\text{leaves}$
- $V[\text{people}][\text{pres}][\text{leave}]=\text{leave}$
- $V[\text{person}][\text{past}][\text{leave}]=\text{left}$
- $V[\text{people}][\text{past}][\text{leave}]=\text{left}$
- $V[\text{person}][\text{pres}][\text{be}]=\text{is}$
- $V[\text{people}][\text{pres}][\text{be}]=\text{are}$
- $V[\text{person}][\text{past}][\text{be}]=\text{was}$
- $V[\text{people}][\text{past}][\text{be}]=\text{were}$ and so forth.

The other difficulty is the indefinite article or determiner in entry seven of the two sets: *a* in (28) and *some* in (36). Let us tentatively say that the indefinite article has two realizations: *a* for singular nouns and *some* for plural nouns. We can express this alternation by a function

(38)  
- $N[a][\text{person}]=\text{a person}$
- $N[a][\text{people}]=\text{some people}$

Keeping the structure of determiner constructions parallel we can also write

(39)  
- $N[\text{the}][\text{person}]=\text{the person}$
- $N[\text{the}][\text{people}]=\text{the people}$ and so forth.

Our generalization is nearing final form.
Let me emphasize that (40) is not the result of induction on one or two paraphrastic sets. The limited space may give such an impression. However, (40) is the result of comparing many similar paraphrastic sets and is a generalization in the proper sense of the word. (40) claims that for any nouns 1 and 5, any verb 2, any adjective 3 and any tense 4 the sentences related as described in (40) all mean the same.

What about the verb leave – can we generalize it? Entries four through six make that difficult, for they depend on the fact that leave and take leave are paraphrastic. There are other similar constructions in English: walk/take a walk, talk/give a talk, speak/make a speech, joke/make a joke; but they are rather individual and often of questionable paraphrastic nature. We won’t try to generalize over such cases here. However, if we leave out entries four through six we can generalize leave to any verb: V₆. In fact, it is an important part of the methodology to observe that some portions of the paraphrastic sets exhibit very general patterns, while others are bound closely to idiosyncratic usage of some word in a sentence.

Now we come to a very important theoretical question concerning the kind of generalization we are expressing by a configuration of symbols such as (40). If we follow the claim that (40) holds for any N₁, N₅, V₃, A₂ and t₄, we quickly come up with the following kind of counter-examples.

(41)  
the neutron which the bear entered left sadly
the neutron the bear entered left sadly
the neutron which was entered by the bear left sadly
the neutron which the bear entered took leave sadly
sad leave was taken by the neutron which the bear entered
it was the neutron which the bear entered that took sad leave
the bear entered a neutron and the neutron left sadly
All elements of (41) sound very nonsensical. Only by extrapolating from more conventional examples like those of (40) can we say that the strings of (41) all mean the same, since they all seem to mean nothing. Thus, it is apparent that we have done nothing toward solving the co-occurrence problem in any absolute frame. Our techniques have not, nor will they, indicate the nonsensical nature of the elements of (41).

An example such as (41) does not, however, invalidate the generalization (40); the paraphrase relation will preserve nonsense, just as it preserves the sense of understandable English. Indeed, it is only proper to leave the domain of the paraphrase relation quite open to new locutions and fairy tales. In fact, (41) might correspond to part of some outlandish fairy tale about a sensitive neutron and a rather small bear.

Perhaps we should, however, reformulate the claim of (40) and similar generalizations as follows.

(42) If in an example of a generalization of paraphrastic sets one entry is acceptable English, then all entries are acceptable and mean the same as the first.

The formulation (42) is merely an explicit acknowledgment that no attempt is made to solve the co-occurrence problem. One could, of course, attempt to include a solution to that problem in the statement of generalizations about paraphrastic sets. I have not done so, for reasons already indicated and to be discussed in detail in the next chapter.

5. SYSTEMATIZATION OF THE PARAPHRASE RELATION

In the preceding section we gave an example of a procedure for generalizing paraphrastic sets. The generalizations were understood according to (42) as claiming to preserve acceptability and paraphrase as long as any one member of a set of strings satisfied the requirements of one entry of the generalization and was an acceptable string.

This claim is still too strong. We give a counter-example using the generalization (40)

(43) the man whom the door saw left silently
    the man the door saw left silently
    the man who was seen by the door left silently
    the man whom the door saw took leave silently
    silent leave was taken by the man whom the door saw
    it was the man whom the door saw that took silent leave
    the door saw a man and the man left silently.
Entry number three of (43) is perfectly good and sensible English, for it can be read as claiming that the man who was seen while he was near the door left silently. Yet the other entries are unlikely English.

The source of the difficulty is the ambiguity of the preposition by. It is clear that just any entry will not do. One entry may be acceptable by virtue of reading or "sense" which is not preserved by the generalization. A syntactic ambiguity may be accompanied by a "semantic" ambiguity. For example

(44) the man who was seen by the girl

can be an entry in either (45) or (46)

(45) the man who was seen by the girl left silently
    the man whom the girl saw left silently
    the girl saw a man and the man left silently

(46) the man who was seen by the girl left silently
    the man who was by the girl when he was seen left silently
    a man was by the girl when he was seen and the man left silently.

(45) and (46) can not be combined into a single paraphrastic set.

(47) Thus, that a string is "semantically" ambiguous means more precisely that it is a member of two paraphrastic sets which cannot be combined to yield a single paraphrastic set.

Thus, more is required to validate generalization (40) than the acceptability of an arbitrary entry in a set of strings. It is probably the case that a syntactic ambiguity will usually, especially in context, fail to be accompanied by a semantic ambiguity.

(48) That a string is syntactically ambiguous means that it fits a generalized form which is an entry in two generalizations which cannot be combined into one generalization.

In example (43) above the third entry satisfies the form of the third entry of generalization (40), hence has that syntactic ambiguity, but fails to possess the semantic ambiguity required in order for (40) to have a paraphrastic set for a realization. Instead, the result was (43).

Usually strings of the length of those in the examples above are found in a larger context. So the above paraphrastic sets really are the result of zeroing in on a portion of text and judging in the given context what strings are paraphrases. Indeed, in order to make the material manageable we seek patterns within patterns. Furthermore, it is of systematic interest to discover
the elementary patterns of which the more complex patterns are composed. In the case of (40) the passive relationship:

\[
(49) \quad N_1 \ V[N_1] [t_4] [V_2] \ N_3 \\
N_3 \ V[N_3] [t_4] \ [be] \ [ed] \ [V_2] \ by \ N[obl] \ [N_1]
\]
is involved in the change from entry one to entry three. Actually, the alteration is

\[
(50) \quad N[obl] \ [N[wh] [N_1]] \ N_5 \ V[N_5] [t_4] [V_2] \\
N[wh] [N_1] \ V[N_1] [t_4] \ [be] \ [ed] \ [V_2] \ by \ N[obl] \ [N_5]
\]
The important observation is that only the relative clause of entry one is altered in the change to entry three.

In general alterations will occur only in a part of a text and not on the whole text. This fact introduces additional difficulties, which are due to the same dichotomy of syntactic ambiguity versus semantic ambiguity. Consider the following paraphrastic set in the pattern of (40).

\[
(51) \quad the \ man \ whom \ the \ camp \ saw \ left \ stealthily \\
the \ man \ the \ camp \ saw \ left \ stealthily \\
the \ man \ who \ was \ seen \ by \ the \ camp \ left \ stealthily \\
the \ man \ whom \ the \ camp \ saw \ took \ leave \ stealthily \\
stealthy \ leave \ was \ taken \ by \ the \ man \ whom \ the \ camp \ saw \\
it \ was \ the \ man \ whom \ the \ camp \ saw \ that \ took \ stealthy \ leave \\
the \ camp \ saw \ a \ man \ and \ the \ man \ left \ stealthily
\]
Entry three is clearly paraphrasable by all the other entries. However, the strings of (51) are being taken quite in the abstract, that is with no given context. The story is quite different when a context is present. As we should expect, a context selects and de-selects possible ways of paraphrasing a string. We have put the above entry three in a context in example (52).

\[
(52) \quad Last \ Saturday \ night \ while \ some \ campers \ were \ sleeping \ on \ the \ beach, \ a \ man \ approached \ their \ campsite \ with \ apparent \ ill \ intent. \\
Fortunately \ the \ beach \ was \ being \ patrolled \ by \ alert \ Forest \ Service \ personnel \ who \ noticed \ the \ dark \ figure \ moving \ around \ their \ huddled \ sleeping \ bags. \ Their \ sleep \ was \ never \ disturbed, \ however, \ because \ the \ first \ they \ were \ aware \ of \ this \ potential \ danger \ was \ the \ next \ morning \ when \ the \ Forest \ Service \ clerk \ told \ them \ that \ the \ man \ who \ was \ seen \ by \ the \ camp \ left \ stealthily \ and \ they \ were \ unable \ to \ find \ him \ anywhere \ in \ the \ woods.
\]
If we replace entry three as it occurs in (52) by any other entries of (51), the resulting text is not a paraphrase of (52). Furthermore, it borders on
nonsense because of the lack of connection between the man whom the camp saw and any other mention of a person in the text. Try reading (53)

(53) Last Saturday night while some campers were sleeping on the beach, a man approached their campsite with apparent ill intent. Fortunately the beach was being patrolled by alert Forest Service personnel who noticed the dark figure moving around their huddled sleeping bags. Their sleep was never disturbed, however, because the first they were aware of this potential danger was the next morning when the Forest Service clerk told them that the man whom the camp saw left stealthily and they were unable to find him anywhere in the woods.

Thus, given a paraphrastic set and the occurrence of one entry of the set in a context, as in the case of (51) and (52), we cannot claim anything about the relationship of the other entries to the context. In particular, we cannot claim that if we alter the given text by replacing the given entry by other entries, the resulting string is a paraphrase of the original text. The sentence the man who was seen by the camp left stealthily is syntactically and semantically ambiguous. However, the whole text (52), while syntactically ambiguous in the string who was seen by the camp, is not semantically ambiguous in the same string. The context has ruled out the reading whom the camp saw. We are unable to claim that anywhere entry three of (51) occurs the other entries also occur paraphrastically.

The conclusion I wish to draw from the above discussion is that particular attention must be paid to the situation of syntactic ambiguity not accompanied by the corresponding semantic ambiguity. If we are to make our generalizations valid, we must come to terms with that situation. It seems that we should eliminate the syntactic ambiguity. The question is how.

While some entries in a paraphrastic set are sources of syntactic ambiguity, other entries are able to disambiguate syntactic constructions. For example, while in (51)

(54) the man who was seen by the camp left stealthily

is syntactically ambiguous in the interpretation of the preposition by, the sentence

(55) the camp saw a man and the man left stealthily

resolves this ambiguity. Furthermore the general form of (55) in (40) does not appear to introduce any syntactic ambiguities (with the exception of the conjunction and versus and then). The latter statement is not true of
other entries of (51). The first entry

(56) the man whom the camp saw left stealthily

resolves the ambiguity of by in entry three, but its general form in (40) is a source of another syntactic ambiguity: the antecedent of the relative pronoun

(57) the brother of the man whom the class met left suddenly.

Following the pattern of (40) we could paraphrase (57) as

(58) the class met a brother of the man and the brother of the man left suddenly.

However, another distinct paraphrase exists.

(59) the class met a man and the brother of the man left suddenly.

Thus, (55) seems singled out as a particularly good choice for resolving syntactic ambiguities.

Let us now reformulate generalization (40) as follows.

(60) If a number of strings of English words are related by differing in a local part according to the description of (40), and if the string which corresponds to entry seven of (40) is an acceptable English text, then the other strings are also acceptable texts of English and paraphrases of the first string.

The broad statement becomes

(61) If a number of strings of a language are related by differing in a local part according to a general description and if the strings which correspond to entries of the general description which have specific general forms are acceptable texts of the language and are paraphrases of each other, then the other strings are also acceptable strings of the language and all the strings are paraphrases of each other.

We note that the possibility has been left open that more than one entry of a general description may be necessary to resolve all syntactic ambiguities. There may be no single form of text which is syntactically entirely unambiguous. However, the rules of a grammar are shorter if the number of such forms can be reduced to one.

In this section on systematization of the paraphrase relation, we have begun to cross over into the systematic organization of the grammar, which we will discuss in Chapter VII. But this is of no concern, for there is no
clear dividing line between generalization and theorem. We could proceed in the manner of the present chapter stepwise to a systematic theory; however, many of the points yet to be made can be made more clearly if the theory is organized into elementary binary relationships according to presentation of Chapter seven.

6. SUMMARY

In this chapter I have tried to indicate some of the procedures involved in obtaining information about the paraphrase relation of a language. First of all care must be taken to collect true paraphrastic sets, that is, sets of strings all of which mean the same. From the basic data of paraphrastic sets, generalizations are made. The generalizing process involves selecting promising patterns and developing notation allowing maximum generality in describing the relationships between members of a paraphrastic set. I used a functional notation which went beyond the customary labeled tree or labeled grouping in order to include more varied paraphrastic sets in the generalizations. I emphasized that the generalizations about paraphrastic sets are not ways of distinguishing nonsensical co-occurrences from acceptable ones, but are simply rules for preserving the sense or nonsense of strings. We noted finally that certain strings, because of their lack of syntactic ambiguity, could play the special role of the source or "base" strings in a paraphrastic generalization, that is, the strings whose sense is preserved throughout the set.
CHAPTER III

COMPOSITIONAL GRAMMARS

1. THE COMPOSITIONAL APPROACH

It has generally been regarded as good science to analyze a complex event into simpler components, and then to give the rules for combining simple events into complex ones. Indeed, this attitude has assisted us in increasing our understanding of much of the world. It has been called the ‘Cartesian method’ or the ‘atomistic approach’. In the present chapter we will discuss the attempt to apply this method to the description of strings of a language.

Current transformationalists have attempted to formalize this kind of description into a grammar. They specify a finite number of elementary strings of the language and attempt to give the rules for combining the elementary strings into all the possible strings of the language. We shall call such grammars ‘compositional’. To an extent, the project was successful. However, a very large stumbling block occurred. The rules produced a lot of nonsense along with the ‘grammatical’ or ‘acceptable’ strings. Attempts have been made to solve this problem. We will criticize those attempts in this chapter.

2. A SIMPLE COMPOSITIONAL LANGUAGE: P

The abstract definition of compositional grammar given in the last section was hardly very graphic. Let me illustrate the concept by a simple formal language. Indeed, it is the formal languages of logic and mathematics whose grammars are the archetypes of compositional grammars.

The language P is merely a part of a propositional calculus. The propositional constants are derived from the single symbol p by adding any number of strokes 11...1. Thus, p1, p11, p111, ... stand for propositions. Any two propositions can be combined using the conjunction &, for example &p11p1, &p111p11. The conjunction precedes the two propositions which are conjoined. Clearly there is no limit to the number and variety of combinations of propositions. However, it is quite easy to specify the elementary strings and the rules which compose the complex strings out of the elementary ones.
We need only one elementary string, namely

(1) \( pI \)

All other propositional constants (PC) can be derived by the repeated application of a rule which adds a single stroke.

(2) \( \text{PC} \rightarrow \text{PCI} \)

We then need to state that a propositional constant is also a proposition (P). Then all other propositions are obtained by conjoining any two propositions with \&.

(3) \( P_1, P_2 \rightarrow \& P_1 P_2 \)

For example, let us derive \&&pII & pII & pIII pIII pIIII.

(4) \( pII, pIII, pIII \)

are the result of one, two, and three applications, respectively, of (2) to \( pI \).

By applying (3) to \( pIII \) and \( pIII \) we have

(5) \( \& pIII pIII \)

Applying (3) to \( pII \) and (5) yields

(6) \( \& pII & pIII pIII \).

Repeating for \( pII \) and (6) we get

(7) \( \& pII & pII & pIII pIII \).

And finally applying (3) to (7) and \( pIII \) gives us

(8) \( \& & pII & pII & pIII pIII pIII \).

We can, of course, express the grammar using the wellknown technique of rewrite-rules. A proposition could be merely a propositional constant

(9) \( P \rightarrow \text{PC} \) (rewrite the symbol P as PC)

or it could be a conjunction of propositions

(10) \( P \rightarrow \& PP \) (rewrite the symbol P as \& PP)
A propositional constant can be simply \( p_I \)

(11) \[ \text{PC} \rightarrow p_I \] (rewrite the symbol PC as \( p_I \))

or it may have more than one stroke

(12) \[ \text{PC} \rightarrow \text{PCI} \] (rewrite the symbol PC as PCI)

The rewrite-rules lead to the familiar tree-representation of the strings of the language P. Consider the tree-representation of (8).

(13)

\[
\begin{array}{c}
P \\
& \& \\
& P \\
& \& \\
& PC \\
& p_I \\
\end{array}
\]

Whichever way we wish to express the grammar, the result is that the rules produce exactly the 'grammatical' or 'acceptable' strings of the language P. Of course, the situation of such formal languages is quite easy, for they are defined to be precisely the strings generated by the grammar. Natural languages, on the other hand, were not created by definition. And we might even question whether they possess the above illustrated compositional property.

3. COMPOSITIONAL GRAMMARS AND THE CO-OCCURRENCE PROBLEM

It is a relatively straightforward task to group the words of a language into grammatical classes and to state which classes occur together and in what sequence. Thus, in English we have nouns (N), verbs (V), adjectives (A) and so forth, which can occur in patterns such as

(14) \[
\begin{align*}
\text{NV} \\
\text{NVN} \\
\text{ANVN} \\
\text{NVA}.
\end{align*}
\]
Or we can give tree graphs of sentence forms

(15) $S \quad /.. N \quad V$

$S$

$N \quad V$

$V \quad N$

$S$

$N \quad A \quad N \quad V \quad N$

$S$

$N \quad V$

$V \quad A$

The task that remains after the forms such as in (14) or (15) are properly described is called the 'co-occurrence problem'. Which nouns, verbs, adjectives and so forth can co-occur in the designated positions so as to produce acceptable English strings?

If we choose strings of each grammatical category at random, we obtain unacceptable strings such as

(16) $\notin John may frighten sincerity$

and

(17) $\notin Euclid's fifth postulate has always been an awful shade of blue.$

Noam Chomsky has tried to solve this problem by subclassifying the strings of each category. For example, sincerity is subclassified as an abstract noun and frighten as a verb which cannot take abstract nouns, but only animate nouns. Thus (16) could not be produced. Likewise, postulate is an abstract noun which cannot have an adjective predicated of it which applies only to concrete objects. Chomsky, Fodor, and Katz have devoted considerable time to formalizing statements such as those above. Certainly the task is completable for the elementary sentences, for there is only a finite number of them. It is another question whether the sub-classifications required for acceptability correspond in the completed task to the intuitive concepts of 'abstract', 'concrete', 'animate' and so forth.
Zellig Harris on the other hand sees no intrinsic value in accounting for co-occurrence restrictions in the finite set of elementary sentences. Instead, he hypothesizes that the acceptable elementary sentences form a basis or 'kernel' for accounting for the acceptability of other more complex sentences. The plan is that if the rules for combining the elementary sentences into complex ones are formulated properly, the acceptability of the complex sentences will be a consequence of the acceptability of the elementary ones. Indeed, it is at the point of producing sentences of unbounded size that the co-occurrence problem is the most forbidding.

For example, if we allow any two sentences to be conjoined by any conjunction, we obtain results such as

(18) \( \neg \) John opened a door and organic chemistry was required of all students, because if seventeen were not a prime, then the dairy association would have to back down. Thus, we can conclude that television will have a beneficial effect on the nation's young.

The above string is unacceptable in the same manner and at least the same degree as \( \neg \) John may frighten sincerity or \( \neg \) Euclid's fifth postulate has always been an awful shade of blue.

However, the device of sub-categorizing words seems impractical when it must be extended beyond simple examples such as (16) and (17) to indefinitely long and complex strings such as (18). Both Harris and Chomsky depend on a similar device to account for (18). They require that word-sharing take place. Harris (1968, pp. 133f) requires that each main word occur in more than one kernel sentence.

We find that the acceptability of these reduced-acceptability \( S_1CS_2 \) [sentence \( S_1 \) conjoined with \( S_2 \)] can be raised to equal \( \min \) acc(\( S_1, S_2 \) [the degree of acceptability of the least acceptable of the two sentences \( S_1 \) and \( S_2 \)], and this by adding certain CS...CS [string of conjoined sentences]. Furthermore, for each such \( S_1CS_2 \), many of the CS...CS which raise the acceptability have the property of repeating the main words of \( S_1 \) and \( S_2 \). This suggests that the C imposes a restriction requiring word repetition.... If we start from the longer (word-repeating) form, which is always acceptable, we can say that the conjoining of sentences has assured acceptability only if each main word occurs in at least two of the sentences.

In the phrase structure part of his grammar of English Chomsky uses the introduction of adjectival relative clauses to obtain sentences with more than one clause. The relative pronouns of an adjectival relative clause must refer to a noun in the main clause. Thus, each clause shares a noun, at least in the form of a relative pronoun, with some other clause. This formulation is slightly weaker than Harris'.

The informal idea is that if we either account for or assume the acceptability of the elementary sentences we eliminate examples such as John
frightens sincerity and the postulate is blue from the input to the formation of complex sentences. If we then further require an overlapping or sharing of topics in the component sentences, by stipulating that each main word occur in at least two of the input sentences, then we will at least have a resultant string in which all the ideas associate well together.

Following Harris' suggestion I have tried to see how close we come to the proper degree of acceptability. The following is an actual portion of text.

(19) Linguistic structure poses no problems for conditioning theory so long as it simply requires that responses be learned to complex sign processes which are different from the responses learned to the component signs. These effects have been demonstrated.... The fact is, however, that linguistic syntax is more than simple contextual variation or stimulus patterning. Syntax possesses certain regularities; it is systematic.

I kernelized the above sentence and obtained extensive wordsharing. I then replaced each word in the kernels by other words, like by like and unlike by unlike, in such a manner that each kernel sentence was perfectly acceptable:

(20) express is efficient
    express demands
    one forms questions
    questions are about nations
    nations are from continents
    nations are new
    continents are rich
    express carries books
    books are for a class
    class is of geometry
    mathematics is efficient
    mathematics is a science
    science is worthless
    science is interesting
    mathematics is difficult
    mathematics is a learning
    learning is of numbers
    learning is interesting
    mathematics bothers students
    mathematics is important

Recomposing according to the decomposition of the above text, I
obtained:

(21) \[ \text{Efficient express carries no books for geometry class so long as it simply demands that questions be formed about new continent nations which are different from the questions formed about the rich continents. These effects have been known. The fact is, however, that efficient mathematics is more than interesting worthless science or number learning. Mathematics bothers students; it is important.} \]

If we glance at a small portion of the result we find a string that seems relatively acceptable, though the entire string is not a text of English. The result is reminiscent of the early probabilistic linguistics which produced sequences of letters resembling English words within a certain scope, while the overall output was utterly incorrect English.

Perhaps we could limit selection of words to certain essential words in each given science or topic of discussion. The results are not encouraging:

(22) \[ \text{Knowing that seventeen is a prime number we can conclude that two divides two hundred sixteen evenly, since three plus two is smaller than seven and eleven is an odd number.} \]

Even compounding the two requirements: single topic and wordsharing, we obtain (using the same text and decomposition as above):

(23) \[ \text{Prime numbers have no divisors of number pairs so long as they simply demand that ratios be multiplied with small decimal fractions which are different from the ratios multiplied with the small decimals. These effects have been considered. The fact is, however, that prime integers are more than even undivisible objects or number curiosities. Integers have divisors; they are essential.} \]

It is easy to multiply examples. Of course, if we were to work very hard we could produce a complex system of co-occurrence relations of perhaps paragraph length that would be almost satisfactory, but then the problem occurs all over again when we extend the length of sentence we consider.

Though I have talked in terms of 'acceptability' of strings of English, I could as well have used the concept of 'grammaticality'. The difficulties are not solvable by drawing a distinction between 'acceptable' and 'grammatical'. They are not problems of memory or style; (21) and (23) present precisely the same kind of problem as in \[ \text{John frightens sincerity: a co-occurrence problem, but a co-occurrence problem that cannot be solved by finite means. I would like to emphasize the importance of the co-occurrence relation to the present discussion, for it is precisely on the ability to charac-} \]
terize the co-occurrence relation that the disagreement lies. The examples of senseless strings which I gave served to indicate that there are no bounds on the length of text which must satisfy the co-occurrence relation in a language. Furthermore – and this is where the disagreement comes – I do not believe that the co-occurrence relation can be completely characterized by a finite number of tuples of co-occurring strings together with a finite number of rules.

The unboundedness of the co-occurrence problem also affects judgments of acceptability or grammaticality.

(24) John frightened sincerity

seems quite unacceptable when read in isolation. We might be tempted to draw conclusions about the co-occurrence restriction between frighten and sincerity. However, let us read the following paragraph.

(25) A tragic event in one's life can pervert normal human relationships. The case of John Henke illustrates this point. John's profession was negotiating for large firms. He was quite successful and provided a comfortable home for himself and his wife. After the sudden death of his wife he lost almost all his clients. John had a special need for sincerity on the part of his clients in order to successfully negotiate for them. Upon talking to his former clients we discovered that they all complained that he began without reason to pry into their personal lives. Of course, they felt threatened and began to fabricate fictitious motives for their business dealings. The source of the problem was very simple. John frightened sincerity. A client who was tempted to establish a sincere relationship with John was upset by his immediately demanding to know the innermost feelings of every new acquaintance.

If we now refer back to (24), it no longer seems unacceptable, but rather appears as very natural English.

An important point here is that judgments of acceptability are not reliable when applied to strings out of context. What is being evaluated in such cases is not so much the language as the imagination of the native speaker in finding a context for the cited string. The latter, I have argued, is not a task for grammar. The lesson to be taken from examples such as (24) and (25) is that linguistic judgments should be made in substantial, if not complete, linguistic contexts.

To avoid being regarded as playing with words, such as 'sentence' or 'grammatical', let me state my criticism in direct terms. The above examples of senseless strings of English words indicate that linguistics cannot account
for many aspects of the actual production of texts by native speakers. One important element in the production of texts is the native speaker's knowledge of the world. (Hiż, 1968, p. 248)

In order to construct an acceptable text, one has to know that some sentences are true. In the text My brother and his wife went to the theater. The man bought the tickets, the phrase the man refers to my brother and the phrase the tickets is short for the tickets for it where it is a referential for going to the theater. In order to construct this text one has to know some facts, namely that my brother and not his wife, nor a theater, is a man, and that going to a theater requires tickets. No syntax can give these facts, though a syntactic analysis of texts may discover many of the facts which are explicitly or tacitly assumed by the speakers.

In fact, so much remains unexplained that linguistics can hardly claim to characterize the concatenation of sentences to make larger sentences and texts. The essential problems of organizing and stringing together kernel sentences to make longer sentences are solved by the native speaker every time he speaks or writes, but apparently not on the basis of grammar.

4. "PROJECTION RULES"

Recently, Chomsky, Katz, and Fodor have proposed a formalization of what could constitute a solution of the co-occurrence problem. The idea is that the grammar would "overgenerate" strings of the language. That is, the grammar would yield strings such as \( \sim \) John may frighten sincerity. Then the semantic component of the grammar would have the job of "interpreting" these strings. For example, it would interpret \( \sim \) John may frighten sincerity as an anomaly due to the property of the verb frighten to require an animate object. As we noted in the last section this task of interpretation or exclusion is certainly performable for the finite number of elementary sentences. The question is how to extend the solution to the general case of long and complex sentences. This extension, according to Chomsky, Katz, and Fodor, is accomplished by "projection rules". Since no even partly complete set of projection rules has been given for any language, people are often unclear about what kind of thing they are. They are basically a refinement of the exclusion approach. That is, rather than have the grammar produce only the grammatical strings to the exclusion of ungrammatical strings, it would produce a great variety of strings, but each string would be labeled as to its grammaticality and reason for ungrammaticality or semantic anomaly. Rather than continuing in the abstract let us look at a pair of formal languages as an illustration of projection rules and the assumptions that underlie them.

The formal languages, northern and southern Bese, will share a number of syntactic features with English. However, our main concern in the present
section is with the semantics of the two languages; for, indeed, they differ from each other fundamentally in their semantics, which in turn affects the set of transformations of each language.

Briefly, the world which Bese has to describe is a succession of graphs. The graph of a given moment in time is quite simple; it contains a selection from among eight nodes, a, b, c, d, e, f, g, and h, and the selected nodes may or may not be bound together by lines. To write Bab is to assert that a and b are bound by a line. To write \( \sim \)Bab is to assert that a and b are not bound by a line. To write \&BabBcd is to say that a is bound to b and c is bound to d. To write Babcd is to say that a, which is bound to c, is bound to d, which expresses the same as writing \&BacBad. Thus, \( aBc \) is a noun a modified by a kind of relative clause Bac which is derived from the sentence Bac.

But let us begin with the simple phrase structure grammar for the elementary sentences. It is the same for both Bese languages. To avoid any confusion with 'grammatical sentence' let us use the term 'possible sentence' to mean a string not necessarily grammatical, but resembling a grammatical sentence in form. Thus, we take S to mean 'possible sentence' and N to mean 'possible noun'.

An elementary sentence is a simple statement about two nodes being bound together.

\[(26)\quad S \rightarrow B \ N \ N\]

That is, we rewrite the symbol S as B, N, N in succession.

There are only eight elementary nouns.

\[(27)\]

\begin{align*}
N & \rightarrow a \\
N & \rightarrow b \\
N & \rightarrow c \\
N & \rightarrow d \\
N & \rightarrow e \\
N & \rightarrow f \\
N & \rightarrow g \\
N & \rightarrow h
\end{align*}

An example is:

\[(28)\]

\[\text{S \ respecting N \ respecting N}
\quad \text{B \ respecting a \ respecting b}
\]

'a is bound to b'
The more complex sentences and nouns will be given by the transformations.

The two speech communities, the northern and southern Bians, derived from a single ancient folk. They have been separated for thousands of years as the result of a tear in the surface which they inhabit, for these people live in a two-dimensional space – an incidental, but anthropologically interesting fact.

The physical world the Bians observe obeys the same physical laws on both sides of the tear: at any given moment in time the physical world is a set of nodes: a, b, c, d, e, f, g, and h, connected by unbroken, loop-free lines; a pair of nodes may be connected by more than one line or by no line at all. A node is never connected to itself. Northern Bese is able to express how many lines directly connect any two nodes at a given moment, while southern Bese can only express whether or not two nodes are directly connected by some line, or lines. This difference between northern and southern Bese makes a profound difference in the kind of grammar appropriate to each language. Southern Bese has by far the simplest kind of grammar, so let us discuss it first.

The southern Bians do not have any means of expressing time in their language, but they do have quite good memories. When they have once seen a graph they never forget it. Of course, a statement is true or false only at a given moment. If node x is connected at a given moment by at least one line to node y, then the string $B_{xy}$ is true at that moment. If node x is not connected at a given moment to node y, then $B_{xy}$ may be false or it may simply be an ungrammatical southern Bese text. If the southern Bians are aware that x was ever connected to y, then $B_{xy}$ is grammatical, but false. If they have had no encounter with that state of affairs, then $B_{xy}$ is ungrammatical.

In southern Bese we have a very simple case of the applicability of projection rules. It should be a very simple task for a linguist to ascertain which of the elementary strings are grammatical and which are not:

<table>
<thead>
<tr>
<th></th>
<th>Bab</th>
<th>Bac</th>
<th>Bad</th>
<th>Bae</th>
<th>Baf</th>
<th>Bag</th>
<th>Bah</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bba</td>
<td>Bbc</td>
<td>Bbd</td>
<td>Bbe</td>
<td>Bbf</td>
<td>Bbg</td>
<td>Bbh</td>
<td></td>
</tr>
<tr>
<td>Bca</td>
<td>Bcb</td>
<td>Bcd</td>
<td>Bce</td>
<td>Bcf</td>
<td>Bcg</td>
<td>Bch</td>
<td></td>
</tr>
<tr>
<td>Bda</td>
<td>Bdb</td>
<td>Bdc</td>
<td>Bde</td>
<td>Bdf</td>
<td>Bdg</td>
<td>Bdh</td>
<td></td>
</tr>
<tr>
<td>Bea</td>
<td>Beb</td>
<td>Bec</td>
<td>Bed</td>
<td>Bef</td>
<td>Beg</td>
<td>Beh</td>
<td></td>
</tr>
<tr>
<td>Bfa</td>
<td>Bfb</td>
<td>Bfc</td>
<td>Bfd</td>
<td>Bfe</td>
<td>Bfg</td>
<td>Bfh</td>
<td></td>
</tr>
<tr>
<td>Bga</td>
<td>Bgb</td>
<td>Bgc</td>
<td>Bgd</td>
<td>Bge</td>
<td>Bgf</td>
<td>Bgh</td>
<td></td>
</tr>
<tr>
<td>Bha</td>
<td>Bhb</td>
<td>Bhc</td>
<td>Bhd</td>
<td>Bhe</td>
<td>Bhf</td>
<td>Bhg</td>
<td></td>
</tr>
</tbody>
</table>

So we first have a finite list of grammatical elementary sentences (GRS) and ungrammatical elementary possible sentences (UNGRS).
The truth or falsehood of complex strings and their grammaticality can be computed very easily with the aid of the transformations. The elementary transformations of southern Bese are, to use a usual notation:

(30) \[ \text{(I)} \quad B \ x \ y \rightarrow B \ y \ x \quad \text{(change of order)} \]

(II) \[ x \rightarrow \sim x \quad \text{(negation)} \]

(III) \[ x, y \rightarrow \& x \ y \quad \text{(conjunction)} \]

(IV) \[ B \ x \ y, \ B \ x \ z \rightarrow B \ x \ B \ y \ z \quad \text{(relative clause)} \]

We can formulate the projection rules of southern Bese as follows:

(31) (a) If \( x \) is GRS and if \( y \) is the transform of \( x \) by (I) or (II), then \( y \) is GRS.
(b) If \( x \) is UNGRS and if \( y \) is the transform of \( x \) by (I) or (II), then \( y \) is UNGRS.
(c) If \( x \) and \( y \) are GRS and if \( z \) is the transform of \( x, y \) or \( y, x \) according to (III) or (IV), then \( z \) is GRS.
(d) If \( x \) is UNGRS and \( y \) is S and if \( z \) is the transform of \( x, y \) or \( y, x \) according to (III) or (IV), then \( z \) is UNGRS.

(a)–(d) are merely a reformulation of the exclusion method discussed earlier. Presumably, with more knowledge of southern Bese we might be able to write a more discriminating set of projection rules, so that more than two different values would be assignable to the strings. Nevertheless, what we have presented is a crude set of projection rules.

Southern Bese is a language which is amenable to the type of grammar which Chomsky and Harris are constructing for English. The grammar of
southern Bese given above is a simple parallel to the grammar they hope to provide for English. We will now turn to the grammar of an essentially different language: northern Bese, which I regard as sharing some important features with English in contrast to southern Bese; in particular, we shall see that northern Bese does not have a compositional character.

The northern Bians have evolved the ability to count in their language. The set of possible sentences for northern Bese is identical with the set of possible sentences for southern Bese. However, the semantics of northern Bese differs from that of southern Bese in an essential way. The result of this difference is that the set of grammatical sentences of northern Bese is quite different from the set of such sentences of southern Bese and a grammar of northern Bese must be of an essentially different kind from that of southern Bese given above.

We recall that for the southern Bians it would be redundant to have two occurrences of Bab, for example, in the same text, because to say twice that a is connected to b is simply to say that a is connected to b. However, in northern Bese exactly two occurrences of Bab in a text would express that a is connected to b by exactly two lines; in general, in northern Bese exactly n occurrences of an elementary sentence Bxy in a text with no relative clauses claims that x is connected to y by exactly n lines. Now, the northern Bians have as fine, if not finer memories than the southern Bians; for they not only remember which nodes have been connected in the past but also by how many lines the nodes were connected. From talking to the northern Bians one learns that for each node the total number of lines which attach to that node is an important concept. In fact, certain totals are literally unthinkable for certain nodes. We can only conjecture that for each particular node there are some total numbers of lines which have never been observed attached to that node; and hence, to assert a possible sentence which claims that many lines attached to the node may be to assert nothing, but merely to utter nonsense, that is, such a string is an ungrammatical possible sentence. With each node, then, there is associated a set of numbers such that each number in the set is a number of lines which can be attached to the node at any given moment. Clearly, these associated sets are based on past observations on the part of the northern Bians. But also they seem to represent an element of conjecture, because in many cases a northern Bian will claim that a certain number of lines can be attached to a particular node even though he cannot recall ever having seen that number attached to it at any time in the past. Careful examination of the associated sets reveals that, though there are distinct patterns in portions of each set, over-all each set is quite irregular, that is, they do not appear to be completely describable in any systematic way. A further difficulty is that the associated sets change as a
result of the scientific investigations of the northern Bians. Apparently, their expectations as to what numbers of lines could be attached to some particular node change to follow the current scientific theories.

A formal semantic characterization of the kind given for southern Bese is not possible for northern Bese. Suppose that sentence A is true at a given moment and sentence B is also true at that moment and that A and B share no elementary nouns. Then $\&AB$ will also be true at that moment. However, if A claims there are five lines attached to node a and B claims the same, then $\&AB$ would claim that there are ten lines attached to node a. Thus, if ten is in the set associated with a, then $\&AB$ is false. However, if ten is not in the set, then $\&AB$ is simply an ungrammatical possible sentence. Hence, two true sentences when conjoined by $\&$ could yield a true sentence, a false sentence, or an ungrammatical string. Similarly, it is possible for two false sentences to be conjoined to yield a true sentence.

With the above informal semantic background let us proceed to the task of presenting a grammar for northern Bese.

If we try to write a grammar in the way we did for southern Bese we run into difficulties. If we have two grammatical sentences A and B, their conjunction $\&AB$ need not be a grammatical sentence. Suppose A claims that there are five lines attached to node b and B claims that there are three and suppose both five and three are in the set associated with b but eight is not. Then the possible sentence, $\&AB$, would be ungrammatical. The difficulty is that the semantics of northern Bese is not 'compositional'.

There is the possibility that northern Bese could fit the compositional conception of language. This could happen if Bian science were able to produce a unified picture of all graphs which could occur, and if the Bians all were able to acquire this knowledge. For example, suppose the total science said that all graphs which had an odd number of lines attached to each node could occur and no other graphs could occur. Then we could state the projection rules for northern Bese. Each possible sentence generated would be accompanied by an eight-tuple: $\langle x_a, x_b, x_c, x_d, x_e, x_f, x_g, x_h \rangle$, each place filled with either a 1 or a 0, corresponding to an odd or an even number, respectively, of lines attached to each node. The elementary sentences would be assigned their eight-tuples:

- $Baa: \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle 0$
- $Bab: \langle 1, 1, 0, 0, 0, 0, 0, 0 \rangle$
- $Bac: \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle$
- $Bgh: \langle 0, 0, 0, 0, 0, 0, 1, 1 \rangle$
- $Bhh: \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle 0$
Thus, we have our dictionary of ‘lexical entries’. Then transformations (I)–(IV) for southern Bese provide the eight-tuples for all other grammatical and ungrammatical sentences.

Every string produced by the grammar will have associated with it via the projection rules a statement concerning its grammaticality. If the string is perfectly grammatical, it will be followed by the eight-tuple \( \langle 1, 1, 1, 1, 1, 1, 1, 1 \rangle \), which simply states that all nodes have an odd number of lines attached. If the string is ungrammatical, then the statement that follows it will indicate the extent and source of the ungrammaticality.

There are four sets of projection rules corresponding to the four transformations above. Let \( \alpha_1 \) stand for any eight-tuple and \( \beta_j \) for any string of eight-tuples, including a string of no eight-tuples.

\[
(32) \quad B\, N_1\, N_2: \beta_1 \alpha_1 \rightarrow B\, N_2\, N_1: \beta_1 \alpha_1 \\
B\, N_1\, N_2: \beta_1 \alpha_1 \circ \rightarrow B\, N_2\, N_1: \beta_1 \alpha_1 \circ
\]

Thus, transformation (I) results in no change in the semantic characterization of a string. For example,

\[
(33) \quad B\text{ac}:(I, 0, 1, 0, 0, 0, 0, 0) \rightarrow B\text{ca}:(I, 0, 1, 0, 0, 0, 0, 0).
\]

The \( \circ \) following the eight-tuple \( \alpha_1 \) in the second part of (30) marks a string which will make any string of which it is a part ungrammatical. For example,

\[
(34) \quad B\text{aa}:(0, 0, 0, 0, 0, 0, 0, 0) \circ \rightarrow B\text{aa}:(0, 0, 0, 0, 0, 0, 0, 0) \circ.
\]

The reason for the ungrammaticality of \( B\text{aa} \) is simply that no node is ever connected directly to itself.

\[
(35) \quad S_1: \beta_1 \langle 1, 1, 1, 1, 1, 1, 1, 1 \rangle \rightarrow \sim S_1: \beta_1 \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle \\
S_1: \beta_1 \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle \rightarrow \sim S_1: \beta_1 \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle \\
S_1: \beta_1 \alpha_1 \left( \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle \neq \langle 1, 1, 1, 1, 1, 1, 1, 1 \rangle \right) \rightarrow \sim S_1: \beta_1 \alpha_1 \circ \\
S_1: \beta_1 \alpha_1 \circ \rightarrow \sim S_1: \beta_1 \alpha_1 \circ
\]

The negation transformation closes off a sentence. That is, the description under the negation sign in no way implies that the configuration it describes be compounded with that of any other description with which it may be conjoined. This closing-off property holds of double and multiple negation as well. Thus, if the string \( S_1 \) is grammatical, it will have no effect as a component of a larger string on the grammaticality of that string, except in the case of simple iterations of the negation operator. The function of \( \circ \) is as a zero. It does not indicate an even or odd number of lines; it simply
indicates that counting must begin all over again. The following is an example of the third rule of (33).

(36) \[ \text{Bac:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle \rightarrow \sim \text{Bac:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle 0 \]

An example of the fourth rule:

(37) \[ \sim \text{Bac:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle 0 \rightarrow \sim \sim \text{Bac:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle 0. \]

The final 0 indicates that what precedes it is an explanation for the ungrammaticality of the string and no eight-tuple preceding it should be added to any other eight-tuple. Adding eight-tuples is necessary for transformations (III) and (IV).

(38) \[
\begin{align*}
S_1: & \beta_1 \alpha_1; \ S_2: \beta_2 \alpha_2 & \& S_1 \ S_2: \beta_1 \beta_2 (\alpha_1 + \alpha_2) \\
S_1: & \beta_1 0; \ S_2: \beta_2 \alpha_2 & \& S_1 \ S_2: \beta_1 \beta_2 \alpha_2 \\
S_1: & \beta_1 \alpha_1; \ S_2: \beta_2 0 & \& S_1 \ S_2: \beta_1 \beta_2 \alpha_1 \\
S_1: & \beta_1 0; \ S_2: \beta_2 0 & \& S_1 \ S_2: \beta_1 \beta_2 0
\end{align*}
\]

In rule one of (38) we add the last eight-tuples of the two conjuncts. The addition follows a two number or modulo-two number system for obvious reasons:

(39) \[
\begin{align*}
0 + 0 &= 0 & \text{even} + \text{even} &= \text{even} \\
0 + 1 &= 1 & \text{even} + \text{odd} &= \text{odd} \\
1 + 0 &= 1 & \text{odd} + \text{even} &= \text{odd} \\
1 + 1 &= 0 & \text{odd} + \text{odd} &= \text{even}.
\end{align*}
\]

The symbol 0 does not signify odd or even, but it adds just like 0, that is, just like a zero. As an example of rule one of (38) we have

(40) \[
\begin{align*}
\text{Bac:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle; \ \text{Bbd:} \langle 0, 1, 0, 1, 0, 0, 0, 0 \rangle \rightarrow & \ \& \ \text{Bac} \\
& \ \& \ \text{Bbd:} \langle 1, 1, 1, 1, 0, 0, 0, 0 \rangle
\end{align*}
\]

That is, since a is connected once to c and b once to d, there is an odd number of lines attached to a, b, c, and d and an even number or zero lines attached to all other nodes. Rule two of (38) can be exemplified by

(41) \[
\sim \text{Bac:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle 0; \ \& \ \text{Bac Bbd:} \langle 1, 1, 1, 1, 0, 0, 0, 0 \rangle \rightarrow & \ \& \sim \ \text{Bac} \ \& \ \text{Bac Bbd:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle \langle 1, 1, 1, 1, 0, 0, 0, 0 \rangle
\]

The last eight-tuple of the transform of (41) can still be added to other final eight-tuples, as in another example of rule one of (38).

(42) \[
\begin{align*}
& \& \sim \text{Bac} \ \& \text{Bac Bbd:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle \langle 1, 1, 1, 1, 0, 0, 0, 0 \rangle; \\
\text{Bac:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle \rightarrow & \ \& \ \& \sim \text{Bac} \ \& \text{Bac Bbd Bac:} \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle \langle 0, 1, 0, 1, 0, 0, 0, 0 \rangle
\end{align*}
\]
Notice that the next to last eight-tuple underwent no change and is carried along to indicate one reason for the ungrammaticality of the transforms in (41) and (42). In this case the reason is that a closed off sentential component refers to an even number of lines being attached to nodes b, d, e, f, g, and h, something out of the realm of conception for northern Bians. An example of rule four of (38) is

\[(43) \quad \text{Baa}: \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle 0; \quad \sim \text{Bac}: \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle 0 \]
\[\rightarrow \& \ \text{Baa} \sim \text{Bac}: \langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle 0\]

Thus, there are two sources of ungrammaticality for the transform in (43). Transformation (IV) has the same projection rules as (III)

\[(44) \quad BN_1N_2: \beta_1x_1; \quad BN_1N_3: \beta_2x_2 \rightarrow BN_1BN_2N_3: \beta_1\beta_2(x_1 + x_2)\]
\[BN_1N_2: \beta_10; \quad BN_1N_3: \beta_2x_2 \rightarrow BN_1BN_2N_3: \beta_1\beta_2x_2\]
\[BN_1N_2: \beta_1x_1; \quad BN_1N_3: \beta_20 \rightarrow BN_1BN_2N_3: \beta_1\beta_2x_1\]
\[BN_1N_2: \beta_10; \quad BN_1N_3: \beta_20 \rightarrow BN_1BN_2N_3: \beta_1\beta_20\]

We take the following as examples of rule one of (44)

\[(45) \quad \text{Beg}: \langle 0, 0, 0, 0, 1, 0, 1, 0 \rangle; \quad \text{Beh}: \langle 0, 0, 0, 0, 1, 0, 0, 1 \rangle \rightarrow \]
\[\rightarrow \text{BeBgh}: \langle 0, 0, 0, 0, 0, 0, 1, 1 \rangle\]
\[(46) \quad \text{Beg}: \langle 0, 0, 0, 0, 1, 0, 1, 0 \rangle; \quad \text{Bea}: \langle 1, 0, 0, 0, 1, 0, 0, 0 \rangle \rightarrow \]
\[\rightarrow \text{BeBga}: \langle 1, 0, 0, 0, 0, 0, 1, 0 \rangle\]
\[(47) \quad \text{BeBgh}: \langle 0, 0, 0, 0, 0, 0, 1, 1 \rangle; \quad \text{BeBga}: \langle 1, 0, 0, 0, 0, 0, 1, 0 \rangle \rightarrow \]
\[\rightarrow \text{BeBgBha}: \langle 1, 0, 0, 0, 0, 0, 0, 0 \rangle\]

Examples (45)-(47) illustrate an important point about northern Bese. In a relative clause, the B can stand for more than one bond. In the transform \text{BeBgBha} of (47), for example, \text{Bg} stands for two bonds of g to e, as the transformational history shows. In fact, the general rule is that the number of bonds that a B stands for is doubled when it occurs as part of the noun-phrase which is the shared noun in the relative clause transformation. This mathematics simply makes up for the fact that one occurrence of the shared noun is deleted in the relative clause transformation.

The other rules of (44) are too similar to (38) to warrant any examples. (32), (35), (38) and (44) constitute the projection rules for northern Bese. That is not to say that we cannot add to or refine these rules. In fact, we have not even inquired whether self-contradictory sentences are ungrammatical as well as being false. If they are ungrammatical, then we would need an elaborate structure of projection rules indeed. But that is a matter for another day.
Let us present a negative and positive example of how the projection rules work. A negative example is the following string:

\[(48) \quad \& \text{Be} \land \& \land \sim \text{Bac} \land \text{BacBbdBacBeBbBha} : \langle 1, 0, 1, 0, 0, 0, 0, 0 \rangle \]
\[\langle 1, 1, 1, 1, 0, 0, 1 \rangle \]

which is simply a conjunction of the results of (42) and (47) and the string \text{Bce}. (48) is ungrammatical. Our formalized semantics, that is, the lexically entered eight-tuples plus the projection rules, “explain” the ungrammaticality as follows.

\[(49) \quad \begin{align*}
(a) & \quad \text{The larger string describes an even number of lines connected to nodes \text{f} and \text{g}.} \\
(b) & \quad \text{It also contains a closed-off component string which describes an odd number of lines only on nodes \text{a} and \text{c}.}
\end{align*} \]

We derive our positive example beginning with the result of (47).

\[(50) \quad \text{BeBbBha} : \langle 1, 0, 0, 0, 0, 0, 0, 0 \rangle ; \quad \text{Bbc} : \langle 0, 1, 1, 0, 0, 0, 0, 0 \rangle \]
\[\& \text{BeBbBhaBbc} : \langle 1, 1, 1, 0, 0, 0, 0, 0 \rangle \]

\[(51) \quad \text{Bbd} : \langle 0, 1, 0, 1, 0, 0, 0, 0 \rangle ; \quad \text{Bhf} : \langle 0, 1, 0, 0, 0, 1, 0, 0 \rangle \]
\[\& \text{BbdBdf} : \langle 0, 0, 0, 0, 1, 0, 0, 0 \rangle \]

\[(52) \quad \text{BbBdf} : \langle 0, 0, 0, 1, 0, 0, 0, 0 \rangle ; \quad \text{Beg} : \langle 0, 0, 0, 0, 1, 0, 1, 0 \rangle \]
\[\& \text{BbBdfBeg} : \langle 0, 0, 0, 1, 1, 1, 0, 1 \rangle \]

\[(53) \quad \& \text{BeBgBhaBbc} : \langle 1, 1, 0, 0, 0, 0, 0, 0 \rangle ; \quad \& \text{BbBdf} \text{Beg} : \langle 0, 0, 0, 1, 1, 1, 1, 1, 1 \rangle \]

The output of (53) describes a situation in which an odd number of lines is connected to each node, giving that sentence the maximum grammaticality.

Thus, we could have an “explanatory semantic component with projection rules” for northern Bese. I wish to emphasize, however, that the possibility of such an account of northern Bese depended in a necessary way on the regularity of the total knowledge of graphs. At the present time the northern Bians do not possess such a regular understanding of their world, even supposing that their world possessed such regularity.

I am even less convinced that we speakers of English have or will ever attain such a regular conception of our world, even supposing our world to be in fact so regular. I do not share the optimism implicit in Chomsky’s *Aspects of the Theory of Syntax* (1965, p. 160):

Concerning dictionary definitions, two major problems are open to investigation. First, it is important to determine the universal, language-independent constraints on semantic features – in traditional terms, the system of possible concepts. The very notion “lexical
entry” presupposes some sort of fixed, universal vocabulary in terms of which these objects are characterized, just as the notion “phonetic representation” presupposes some sort of universal phonetic theory. It is surely our ignorance of the relevant psychological and physiological facts that makes possible the widely held belief that there is little or no a priori structure to the system of “attainable concepts”.

A dictionary with this kind of definition would be like the original set of lexical entries which we supposed for northern Bese.

Using such a dictionary, the grammar envisioned by Chomsky, Katz, and Fodor generates strings of symbols and notes which strings are grammatical sentences and perhaps some additional information, for example, directions as to how to interpret the sentence as to anomalies that may occur in it. The burden of this description of generated strings lies with their projection rules. They argue correctly that since a native speaker can understand an infinite number of sentences though he has encountered only a finite number of sentences, there must be some way the native speaker’s finite experience is converted into an ability to understand an infinite number of sentences. The conclusion that this conversion is achieved by a set of rules of the speaker’s language does not follow. It does not even follow that the conversion is achieved by any ‘rules’ as that term is usually understood. The point I am criticizing is clearly stated by Katz and Fodor (1964, p. 482):

Since the set of sentences is infinite and each sentence is a different concatenation of morphemes, the fact that a speaker can understand any sentence must mean that the way he understands sentences he has never previously encountered is compositional: on the basis of his knowledge of the grammatical properties and the meanings of the morphemes of the language, the rules the speaker knows enable him to determine the meaning of a novel sentence in terms of the manner in which the parts of the sentence are composed to form the whole. [My emphasis] Correspondingly, then, we can expect that a system of rules which solves the projection problem must reflect the compositional character of the speaker’s linguistic skill.

Certainly I cannot prove that projection rules are not the key to the linguistics of the future. A person who believes that they are will not be dissuaded by what I have argued; nor do I wish anyone to abandon a project he believes in. The above detailed example was merely my attempt to highlight the assumptions of a project such as writing projection rules and formulating compositional semantics in general. I cannot operate on the assumption that there is an “a priori structure to the system of attainable concepts” that is refined enough to solve the co-occurrence problem.

5. SUMMARY

In this chapter I have briefly characterized what I termed ‘compositional grammars’. The goal of a compositional grammar is to produce the whole infinite array of grammatical or acceptable strings of a language through
a finite list of atomic strings and the repeated application of a finite number of rules which combine strings into more complex strings. Such a grammar is judged unsatisfactory, if along with the good strings it produces a whole infinite array of nonsense strings, that is, if it does not solve the co-occurrence problem. The co-occurrence problem is simply to show what strings of a language can occur together acceptably and in what configurations. I indicated why I felt that the proposed solutions to this problem are only partial solutions covering the more elementary strings with no obvious extension to the generally complex strings. We examined in detail what such an extension would be like for a very simple formal language. Through that formal language I tried to indicate that the solution to the co-occurrence problem depended on an unlikely regularity in the state of human belief and knowledge.

Are we then to abandon the co-occurrence problem? My belief is that it is unsolvable in the form in which it is currently expressed. I prefer to work on a much weaker study of co-occurrence — call it 'the relative co-occurrence question': given a string of acceptably co-occurring strings of a language, what other acceptable configurations of those strings can be obtained from the given string by means of rules?
CHAPTER IV

SUBSTITUTION

1. THE SUBSTITUTION CONCEPT

The idea of substitution is very simple. Given a string which has a number of occurrences of a particular string, we may replace that string by another in all places. For example, the sentence

(1) \( \text{John went to the store and John bought some apples} \)

has two occurrences of the word \( \text{John} \). It would be very natural to replace both occurrences by \( \text{he} \) to obtain

(2) \( \text{he went to the store and he bought some apples} \)

in the context

(3) \( \text{John loves apples, so he went to the store and he bought some apples.} \)

The substitution operation is nearly as simple to conceive of as the concatenation operation which has been so fundamental to grammar. Simplicity of conception is, of course, important to us in making substitution an elementary and central operation in our grammars. In order to introduce substitution, we need some notational conventions. We will rely upon the general functional notation of mathematics: \( f(x) \) stands for a function, for example

(4) \( x^2 + 2x + 3 \)

and \( f(2) \) for the value of the function for the number 2:

(5) \( 2^2 + 2 \cdot 2 + 3 = 11. \)

Likewise, \( S(x) \) will stand for the function or context, for example

(6) \( x \text{ went to the store and } x \text{ bought some apples} \)

and \( S(\text{John}) \) for the value or string

(7) \( \text{John went to the store and John bought some apples.} \)

Of course, the analogy is only approximate. In particular, \( S(x) \) will have an
acceptable value only for a portion of specifications for $x$. For example, $S(\text{after})$ yields

(8) $\exists \text{ after went to the store and after bought some apples.}$

However, the similarities are suggestive enough to make the notation heuristically valuable. In a transformational analysis, examples such as (8) will be seen to present no problem.

The substitution operation will assist us in writing transformations in greater generality than heretofore. We will not have to write separate transformations for operations that differ only in the number of occurrences of a string to be replaced. Thus, the change from

(9) $\text{John went to the store}$

to

(10) $\text{he went to the store}$

has the same form

(11) $S(\text{John}) \rightarrow S(\text{he})$

as the change from

(12) $\text{John went to the store and John bought some apples}$

to

(13) $\text{he went to the store and he bought some apples.}$

In both cases the operation involved is the substitution of $\text{he}$ everywhere for $\text{John}$, as (11) indicates. We will see in this chapter where the added generality can be very valuable, and nearly a necessity for expressing certain transformational patterns.

2. THE PRESENCE OF SUBSTITUTION

The substitution operation has many uses in English. Let us survey some of these uses.

In the preceding section the introduction of a personal pronoun was used as an example of substitution. One could claim that, except for certain constructions which we will examine shortly, pronominalization is reducible to single item replacements and therefore the general notion of substitution as co-ordinated multiple replacement is not required. For example, one could claim that (3) is derived in the following manner. Beginning with

(14) $\text{John loves apples, so John went to the store and John bought some apples}$
we could pronominalize the first John of the second conjunct, that is, after so. We obtain

(15)  *John loves apples, so he went to the store and John bought some apples*

and then pronominalizing the last John would give us

(16)  *John loves apples, so he went to the store and he bought some apples*

which is the same as (3).

To my ear, however, there seems to be something strange about (15). It is not a paraphrase of (14) and (16). Whereas (14) and (16) assert that John’s buying of some apples is a consequence of his loving apples, (15), at least if bought is not specially stressed, indicates that John’s buying apples was co-incidental to his loving apples. Instead of (15) we should have used

(17)  *John loves apples, so John went to the store and he bought some apples.*

(17) is easy to read as a paraphrase of (14) and (16). Apparently, to preserve paraphrase and even sense

(18)  pronominalization must apply to all occurrences of the noun in the second conjunct.

(15) violated this procedure, but (17) satisfied it, because

(19)  *John bought some apples*

is the second conjunct, not of (14), but of

(20)  *John went to the store and John bought some apples.*

Thus, the change from (14) to (16) could be accomplished by a sequence of single replacements. We emphasize that co-ordination of operations was necessary to insure the proper derivation and that in the formulation of (18) the use of the general concept of substitution is quite natural. (18) can be more precisely stated as

(21)  Given a conjunction of sentences of which $S_1(N)$ is the first conjunct and $S_2(N)$ the second, the pronominalization operation on this sentence yields a new conjunction of sentences of which the first conjunct is $S_1(N)$ and the second is $S_2(prnN)$, where $prnN$ is the proper personal pronoun for $N$.

The relationship between direct and indirect discourse is characterized very
nicely using the substitution concept. Consider, for example, the sentence

(22) Mr. Donne declared, "I am not about to agree with Mr. Wilson on this issue, for I have my principles"

which in indirect discourse becomes

(23) Mr. Donne declared that he was not about to agree with Mr. Wilson on this issue, for he had his principles.

The relationship between direct and indirect discourse involves two or more uses of general substitution. In the declaration of (22) all occurrences of *I/my* were replaced by *he/his* respectively and all uses of the present tense were replaced by the past tense. Thus, we could write (22) as

(24) Mr. Donne declared, "S(I/my) (present)"

which is changed to

(25) Mr. Donne declared that S(he/his) (past)

where the two pairs of parentheses indicate the two substitution operations. Of course, the representations of the strings of English must be formalized in order for the relationship between (22) and (23) to be entirely formal. The topic of how to represent strings will be handled in a later chapter.

The most powerful tools are needed to describe the formation of the relative clause of English. We will discuss most of them in this book in various chapters. The fundamental operation is substitution. A sentence is formed into a relative clause by replacing all occurrences of a noun by a relative pronoun or personal pronoun.

The relationship between the sentence

(26) the proposal certainly was not spontaneous, but rather the President had instructed two months ago he should put forth the proposal, after they had exhausted their own proposals and he was certain they would accept the proposal

and the relative clause

(27) which certainly was not spontaneous, but rather which the President had instructed two months ago he should put forth, after they had exhausted their own proposals and he was certain they would accept it

can be described as the substitution of *which* for every occurrence of *the proposal* excepting one which is replaced by *it*. How the choice between relative pronoun and personal pronoun is made formally will be discussed
in the chapter on functional representation. The obligatory shifting of the relative pronoun will also be characterized in that chapter. The problem of which occurrences of *proposal* to identify and then replace and which to distinguish as having different referents will be put in a framework in the chapter on the structure of paraphrase grammars.

We wish simply to point out here that substitution is frequent and natural in English.

3. A NOTATION FOR SUBSTITUTION

Formal notations have a great advantage over discursive English. They concentrate the attention of the reader on some particular point or points. They are graphic. Often on the same line one can take in source, operation, and result. Most important they enable one to obtain an overview, "to visualize" a rule or function. We have been making informal use of a substitution notation. We will define that notation more carefully now.

The curved parentheses "( " and " )" are reserved exclusively to signal the substitution operation. The fundamental notational device is one grammatical description X followed by another Y which is enclosed in curved parentheses:

\[
(28) \quad X(Y)
\]

(28) describes a string as made up of a context X which surrounds generally multiple occurrences of Y. A context X is a rule which tells how to obtain a string when given a value for Y. For example, X could say

\[
(29) \quad \text{Follow } Y \text{ by the number of } *'s \text{ corresponding to the number occurrence of } Y \text{ for four occurrences of } Y.
\]

Pictorially, (29) looks like

\[
(30) \quad Y*Y**Y***Y****.
\]

If Y is chosen to be the string 1, for example, then (29) applied to 1 is

\[
(31) \quad 1*1**1***1****.
\]

Thus, if X is as described in (29), X(I) is (31). X is a context, which is in fact a function in arbitrary strings of symbols, and X(I) is the value of X for the argument I. X(I) stands for a string of symbols, namely (31).

Similarly, if X is

\[
(32) \quad \text{Follow the first occurrence of } Y \text{ by } \text{came home because} \text{ and the second occurrence of } Y \text{ by } \text{was sick}
\]
then $X(\text{my brother})$ stands for the string

(33) $\text{my brother came home because my brother was sick}$

which is the value of the context $X$ for the argument $\text{my brother}$.

What we are primarily interested in is a relation or transformation. We want to assert that $W$ is related to $Z$ or $W$ is a transform of $Z$, which we will write in the conventional manner as

(34) $Z \rightarrow W$

or

(35) $Z \leftrightarrow W$

if the change is equally valid in both directions.

Combining the two notational devices we can write rules such as the following:

(36) $X(Y) \rightarrow Z(Y)$

where $X$ is the rule

(37) Precede one occurrence of $Y$ by $\text{they destroyed}$

and $Z$ is

(38) Follow one occurrence of $Y$ by $\text{were destroyed by them}$.

Thus, if the argument $Y$ is taken to be $\text{the weapons}$, then

(39) $X(\text{the weapons}) \rightarrow Z(\text{the weapons})$

is

(40) $\text{they destroyed the weapons} \rightarrow \text{the weapons were destroyed by them}$

(39) asserts that the string

(41) $\text{they destroyed the weapons}$

is transformed into the string

(42) $\text{the weapons were destroyed by them}$

The results are a bit disappointing so far. (36)-(38) constitute a rule that, even if correct, is too specific to be of much interest to grammarians. However, it does serve to illustrate the claim made by the notation. The claim of (36)-(38) is

(43) If the value of $X$ for any argument string $Y$ is a “good” string of English, then the value of $Z$ for the argument $Y$ is also a “good” string of English and preserves the “essence” of the value of $X$.

One may understand “good” as meaning acceptable, grammatical, sensible,
or whatever predicate seems fundamental. "Essence" can be read as meaning, information, or degree of acceptability corresponding to whatever property is being characterized. As I have indicated earlier, I prefer to think in terms of acceptability and the paraphrase relation.

I emphasize that (43) is a generalization over all strings \( Y \). There is no difficulty with such arguments as \( Y = \text{after} \). \( X(\text{after}) \) is

(44) \( \exists \text{ they destroyed after} \)

which is not an acceptable string of English. Therefore, the condition of (43) is not satisfied and it cannot be invalidated by the output

(45) \( \exists \text{ after was destroyed by them.} \)

For \( Y = \text{after} \) the proper predicates are not satisfied and no relation or property is preserved. The string \( \text{after} \) is simply irrelevant to (43).

However, let \( Y = \text{the weapons and the guards were angry} \). Then the value of \( X \) is

(46) \( \text{they destroyed the weapons and the guards were angry} \)

and the condition of (43) is satisfied. (46) is "good", that is, an acceptable string of English. But the value of \( Z \) for this \( Y \) is

(47) \( \exists \text{ the weapons and the guards were angry were destroyed by them.} \)

In order to achieve generality and accuracy, additional notational devices are needed to augment the fundamental symbols for substitution.

The ability to distinguish the grammatical categories of strings will be of great assistance to these ends, just as it has been for all grammars, past and present. Capital letters will indicate grammatical categories. If the description of an English string begins with \( N \), for example, that string is asserted to be playing the role of a noun in the given context. Thus, in (36) instead of \( Y \), which indicates no grammatical category, we write \( N \). That is, we restrict \( Y \) to the strings which are nouns. Such an \( N \) is the most elementary case of a grammatical category symbol, for it is followed by no arguments. The symbol \( S \) beginning a grammatical description of a string asserts that the entire string is of the grammatical category of a sentence. For example, we can use \( S \) in place of \( X \) in (36) and (37), restricting the context \( X \) to those contexts which form sentences out of strings of English. If \( N \) is

(48) \( \text{the weapons} \)

and \( S \) is

(49) Form a sentence by preceding one occurrence of a string by \( \text{they destroyed} \)
then $S(N)$ is the value of $S$ for $N$, which is

\[(50)\quad \text{they destroyed the weapons.}\]

Here (50) is $S(N)$. $S(N)$ as a whole is the name or description of (50). The symbol $S$ stands for a context or function. $S$ by itself is not a string of the grammatical category of a sentence, but is a context which forms a sentence out of its argument. $S(N)$ is the sentence.

Some device is necessary for identifying and distinguishing strings and functions in a grammatical description. For this purpose we will use Arabic numerals as subscripts on the grammatical category symbols. Let us re-formulate (36)–(38) in this way.

\[(51)\quad S_1(N_3) \rightarrow S_2(N_3)\]

where $S_1$ is

\[(52)\quad \text{Precede one occurrence of the argument by they destroyed, forming a sentence}\]

and $S_2$ is

\[(53)\quad \text{Follow one occurrence of the argument by were destroyed by them, forming a sentence.}\]

The claim of (51)–(53) is

\[(54)\quad \text{If the value of } S_1 \text{ for any argument string which is a noun, } N_3, \text{ is a "good" sentence of English, then the value of } S_2 \text{ for the argument } N_3 \text{ is also a "good" sentence of English in which the argument } N_3 \text{ plays the role of a noun and which preserves the "essence" of the value of } S_1.}\]

(51)–(54) is certainly a valid rule of English, except for adjustments due to singular versus plural $N_3$. The counterexample of (46)–(47) does not apply to (51)–(53), since the weapons and the guards were angry is certainly not a noun, which is one of the input conditions on the transformation.

(51)–(53) may state a valid rule, but it is just not a very interesting one. We do now, however, have the tools for beginning to write interesting rules.

We will continue to make informal use of simple tree descriptions until a precise functional representation is developed in the chapter on that topic.

Consider now the change from

\[(55)\quad \text{the proposal was merely the visible sign of an already existing but unexpressed concurrence, because the proposal certainly was not spontaneous, but rather the President had instructed two months}\]
ago he should put forth the proposal after they had exhausted their own proposals and he was certain they would accept the proposal to

(56) the proposal, which certainly was not spontaneous, but rather which the President had instructed two months ago he should put forth after they had exhausted their own proposals and he was certain they would accept it, was merely the visible sign of an already existing but unexpressed concurrence.

We can describe the change provisionally in the following fashion

(57) 

Thus, the proposal, which is \( N_1 \), is replaced by which everywhere in \( S_3(N_1) \), which is everything after because in (55). (57) is neither the most general form of the transformation we are seeking to describe, nor is it very accurate. In particular, problems of word order and selection of it as a required variant on which must wait for a more sophisticated representation of English sentences.

We wish for the moment to examine the potential of the substitution notation.

There is one detail of (57) which should be noticed. \( S_3(N_1) \) was changed to \( A_3(which) \). The only indication of the relationship of the two strings is the subscript on the grammatical category symbols. \( S_3(N_1) \) plays the role of a sentence in (55), while \( A_3(which) \) functions as an adjective in (56). Therefore, the initial symbol S of the grammatical description \( S_3(N_1) \) must be changed to the A of \( A_3(which) \). The subscript identifies the context or function and the capital letter gives the grammatical category of the value of the context for the given argument and given transformation.

Clearly, the description \( N_1 V_2 \) is rather incidental to the transformation we are interested in. The idea of the transformation is that \( S_3(N_1) \) is changed to \( A_3(which) \) and adjoined to one occurrence of \( N_1 \) in the first conjunct of the because construction. In other words, one occurrence of \( N_1 \) in that first conjunct is replaced by

(58) 

If we let a 0 subscript on the parentheses indicate the degenerate case of
substitution which is single replacement, then we can generalize (57) as

\[ S_4 \{N_1\} \rightarrow S_3(N_1) \rightarrow S_4 \{N_1 \text{ (which)} \} \]

In (55) \( S_4_0(N_1_0) \) denotes

\[ \text{the proposal was merely the visible sign of an already existing but unexpressed concurrence} \]

and \( S_4 \) can be indicated graphically as

\[ Y \text{ was merely the visible sign of an already existing but unexpressed concurrence.} \]

The right-hand side of (59) shows a substitution within a substitution. This causes no difficulty for the understanding if we read it from the inside out.

\[ A_3(\text{which}) \text{ is an adjective which obtained from the sentence } S_3(N_1) \text{ by replacing every occurrence of } N_1 \text{ in } S_3(N_1) \text{ by } \text{which, and then the string } A_3(\text{which}) \text{ is adjoined to the right of } N_1 \text{ to form a noun (58), which now yields} \]

\[ S_4 \{N_1 \text{ (which)} \} \]

by replacing the selected occurrence of \( N_1 \) in \( S_4_0(N_1_0) \).

We have to be a bit more precise in using the \( _0 \) subscript. Since it indicates the replacement of only one occurrence of the specified string, we must be told which occurrence, for it makes a difference in the result. Suppose that \( S_4_0(N_1_0) \) had been

\[ \text{the proposal was brief and the proposal was merely the visible sign of an already existing but unexpressed concurrence} \]

then \( S_4 \) could have been

\[ Y \text{ was brief and the proposal was merely the visible sign of an already existing but unexpressed concurrence} \]

or it could have been

\[ \text{the proposal was brief and } Y \text{ was merely the visible sign of an already existing but unexpressed concurrence.} \]

Thus, (59) as it stands is an ambiguous transformation. In order for it
to be fully defined we must in each case specify the context $S_4$. To indicate this fact we include $S_4$ in the title of the transformation. $S_4$ is then called a ‘parameter’ of the transformation and the transformation is a ‘parametric’ transformation. For example, if we name (59) as $T_{wh}$, its full name would be $T_{wh} \langle S_4 \rangle$.

The $0$ subscript has a very important general application. It serves to indicate that a transformation takes place in a context. The change from (55) to (56) certainly took place in a larger context. For example, suppose (55) is preceded by

(66)  
*today the ambassador proposed to dismantle one military base for the withdrawal of each enemy division.*

Then we are dealing not with

(67)  
$S_4 \langle N_1 \rangle$ because $S_3(N_1)$

in an absolute sense, but as it occurs in context (66). The transformation is really from

(68)  
*today the ambassador proposed to dismantle one military base for the withdrawal of each enemy division. The proposal was merely the visible sign of an already existing but unexpressed concurrence, because the proposal certainly was not spontaneous, but rather the President had instructed two months ago he should put forth the proposal, after they had exhausted their own proposals and he was certain they would accept the proposal*

to

(69)  
*today the ambassador proposed to dismantle one military base for the withdrawal of each enemy division. The proposal, which certainly was not spontaneous but rather which the President had instructed two months ago he should put forth, after they had exhausted their own proposals and he was certain they would accept it, was merely the visible sign of an already existing but unexpressed concurrence.*

If we let context (66) be called $S_5$, then the change from (68) to (69) is described as

(70)  
$S_5 \langle S_4 \langle N_1 \rangle \rangle$ because $S_3(N_1) \rightarrow S_5 \langle S_4 \langle N_1, A_3(\text{which}) \rangle \rangle$
and the title of (70) must include the parameter $S_5$ as well as $S_4$. We note that all transformations will have at least this one parameter of the overall context.

There is also one further parameter in the above transformation (70). There may be more than one noun eligible for the relative clause transformation. So we must also specify the identity of $N_1$.

The full title of (70) is

$$(71) \quad T_{\text{wh}} \langle S_5, S_4, N_1 \rangle.$$  

In section one of the present chapter we saw another complication of the substitution notation: two sets of substitution parentheses in succession. The example of section one was the change from direct discourse to indirect discourse. Two substitutions were involved: the substitution of \textit{he}/\textit{his} for \textit{I/my} and the substitution everywhere of past tense for present tense. It didn't really matter for that example in what order the successive substitutions were performed. However, in general it does matter.

Let us illustrate this point with a simple formal language we discussed earlier. Suppose we are given the sentence

$$(72) \quad \& pI \ pII$$  

and the instruction

$$(73) \quad \text{Replace } pI \text{ everywhere by } \& pII \ pI \text{ and } pII \text{ everywhere by } pI.$$  

We could carry (73) out in a number of ways. We could first replace $pI$ everywhere by $\& pII \ pI$ to obtain

$$(74) \quad \& \& pII \ pI \ pII.$$  

and then in (73) replace $pII$ everywhere by $pI$, giving

$$(75) \quad \& \& pI \ pI \ pI.$$  

Or we could first replace $pII$ everywhere by $pI$ and get $\& pI \ pI$ in which we then replace $pI$ everywhere by $\& pII \ pI$ to obtain

$$(76) \quad \& \& pII \ pI \ & pII \ pI.$$  

Or finally we could simultaneously replace all occurrences of $pI$ and $pII$ in (72) by $\& pII \ pI$ and $pI$ respectively to get

$$(77) \quad \& \& pII \ pI \ pI.$$  

The results: (75), (76), and (77) are all quite different.

A similar situation arises with our formal notation if we replace (73) by

$$(78) \quad S_1(pI)(pII) \to S_1(\& pII \ pI)(pI).$$
We have no idea what path to take simply by looking at (78). We must establish some conventions. Let us say then that in a string of paired parentheses: \( X_n(Y)(Z)\cdots(W)(U) \) the string in the last parentheses on the right-hand side of the rule is substituted for the string in the last parentheses on the left-hand side, then the next to the last for the next to the last, and so on. Thus, in

\[
(79) \quad X_n(Y)(Z)\cdots(W)(U) \rightarrow V_n(Y')(Z')\cdots(W')(U')
\]

first we substitute \( U' \) for \( U \), then in the result so obtained we substitute \( W' \) for \( W, \ldots, Z' \) for \( Z \), and then \( Y' \) for \( Y \).

(78) thus describes the change from (72) to (76). The change from (72) to (74) would be written

\[
(80) \quad S_1(p|l|)(p|l) \rightarrow S_1(p|l) \&(p|l|p|l).
\]

What about the change from (72) to (77)? We need a slight addition to include this third possibility. Let \( X_n(Y|Z|\cdots|W|U) \) indicate that corresponding elements are substituted for each other simultaneously.

\[
(81) \quad X_n(Y|Z|\cdots|W|U) \rightarrow V_n(Y'|Z'|\cdots|W'|U')
\]

means that \( Y, Z, \ldots, W \) and \( U \) are replaced simultaneously by \( Y', Z', \ldots, W' \) and \( U' \) respectively. So the change from (72) to (77) can be written

\[
(82) \quad S_1(p|l|p|l) \rightarrow S_1(\&p|l|p|l|p|l).
\]

We now have a fairly rich substitutional notation system. Let us see what the advantages are of making such a notation basic to our grammars.

4. THE GENERALITY OF SUBSTITUTION

I argued in Chapter III that the whole idea of a compositional grammar was on unsure ground. The present section does not depend on those arguments. In fact, the arguments here also apply within the framework of compositional grammars. Some rules of a compositional grammar could be stated with more generality and transparency by using a substitutional notation.

Generality and transparency are our central concerns in this section. We will represent a transformational relationship of English in two different ways. We will describe it using the substitutional notation and we will show how it might be handled with more strictly compositional techniques. We will argue that, while both approaches will work, the substitutional one yields an immediate generalization not available on the compositional approach.
The central idea of the following illustrations is that the number of occurrences of a string which is to be replaced everywhere is not relevant to the form of the transformation being investigated, which is a relative clause transformation.

Consider the relationship between

\[(83) \quad \text{the plate didn't break, because the plate was plastic}\]

and

\[(84) \quad \text{the plate, which was plastic, didn't break.}\]

The substitutional description of this relationship is

\[
S_1 \{N_2\} \quad \because \quad S_3 (N_2) \quad \rightarrow \quad S_1 \{N_2 \ A_3(\text{which})\}
\]

where \(N_2\) is \textit{the plate} and \(S_1\) is

\[(86) \quad Y \quad \text{didn't break}\]

and \(S_3\) is

\[(87) \quad Y \quad \text{was plastic.}\]

A compositional description would appear as follows.

\[
N_1 V_2 \quad \because \quad N_1 V_3 \quad \rightarrow \quad N_1 \quad \text{which} \quad V_3 V_2
\]

where \(N_1\) is \textit{the plate}, \(V_2\) is \textit{didn't break}, and \(V_3\) is \textit{was plastic}. A lattice diagram like (88) is to be read from left to right. \(N_1 V_2 \because N_1 V_3\) is broken down into \(N_1 V_2\) and \(N_1 V_3\). \(N_1 V_3\) is changed to \textit{which} \(V_3\) and then adjoined to \(N_1\) in \(N_1 V_2\) to obtain \(N_1 \ which \ V_3 V_2\).

We note that we are not investigating the source of (83). We may regard it as a given starting point or as derived from its component strings in a compositional grammar. It does not matter which. We are now discussing how best to proceed given such a string as (83): whether in the fashion of (85) or (88). Both partially describe the change that has taken place. As they now stand there is no obvious recommendation.

Consider now the following relationship of

\[(89) \quad \text{the plate didn't break, because the plate was plastic and the plate was in a box besides}\]
(90) *the plate, which was plastic and which was in a box besides, didn't break.*

In substitutional terms it appears as follows

(91)

```
\[
S \rightarrow S_1 \{N_2 \}_{N_2} \quad \text{because} \quad S_2(N_2) \rightarrow S_1 \{N_2 \ A_2 \{\text{which} \} \}
\]
```

where \( N_2 \) is *the plate* and \( S_1 \) is

(92) *\( Y \) didn't break*

and \( S_3 \) is

(93) *\( Y \) was plastic and \( Y \) was in a box besides.*

The important fact to note is that (91) is identical to (85). The number of occurrences of \( N_2 \) in \( S_3(N_2) \) makes no difference in the substitutional description.

The situation is quite different in a compositional description of the same change.

(94)

```
\[
N_1V_2 \quad \text{because} \quad N_1V_3 \text{ and } N_1V_4 \text{ besides } \quad N_1 \text{ which } V_3 \text{ and which } V_4 \text{ besides } \quad V_2
\]
```

The great similarity between the relationship of (83) to (84) and (89) to (90) is not easily seen by comparing (88) and (94). Rather, the overwhelming impression one gets from comparing (88) and (94) is that (94) involves a much more complex change than (88) does.

Let us return to the example of (55) and (56), which constitute a very complex example of the relative clause relation we are discussing. We recall also that the substitutional description (59) of the relationship of (55) to (56) is identical with that of (85) and (91). The substitutional notation is claiming that the relationships: (83) to (84), (89) to (90), and (55) to (56) are all essentially the same.

Let us see, on the other hand, what a lattice description of (55) to (56) would be.
In order to represent the transformation compactly, we must resort to some abbreviations.

(95) \(S_1:\) the proposal was merely the visible sign of an already existing but unexpressed concurrence

\(S_2/A_2:\) the proposal certainly was not spontaneous

\(S_3[ ]/A_3[ ]:\) the President had instructed two months ago [ ]

\(S_4/A_4:\) he should put forth the proposal

\(S_5/A_5:\) they had exhausted their own proposals

\(S_6[ ]/A_6[ ]:\) he was certain [ ]

\(S_7/A_7:\) they would accept the proposal

The alternative \(S_i/A_i\) is given because of the different roles the strings can play, depending on whether they are in a complex sentential combination or are part of a relative clause. The subscripts identify the strings.

The square brackets in \(S_i[ ]\) or \(A_i[ ]\) indicate a containing string.

The prefixes \(wh\) and \(it\) are rough indications of the kind of alteration that has taken place on a string. \(wh\) means that the proposal is replaced by which and \(it\) means the proposal is replaced by it.

\(S_i\) followed by \(whX\) means that \(whX\) is imbedded in \(S_i\) in the appropriate place.

Using the above abbreviations the relationship of (55) to (56) appears as shown in (96) on page 68.

(96) is a rough description of the rule changing (55) to (56). If we compare (88), (94) and (96) there appears little to suggest grouping them under the same general rule.

If we invested the prefix \(wh\) with more power, that is, with the capability of changing arbitrarily complex sentences into relative clauses with respect to a given noun, then we could generalize (88), (94) and (96) as

\[(97)\]

But there is little transparency to recommend this generalization. In particular, the substitution process is completely obscured.

Both notations will have to account for the many restrictions and interrelationships of the components in a relative clause. Many of them will have to do with occurrences of the noun to be replaced: where it must occur and what it must be replaced by at those various occurrences. Such requirements seem to fit well conceptually with the substitution notation. We will discuss these features in the next chapter.
S₁ because S₂ but rather S₃[S₄ after S₅ and S₆[S₇]]

S₂ but rather S₃[S₄ after S₅ and S₆[S₇]]

S₃[S₄ after S₅ and S₆[S₇]]

S₄ after S₅ and S₆[S₇]

S₅ and S₆[S₇]

S₆[S₇]

S₇
5. OPERATIONS AND TRANSFORMATIONS

Let us now illustrate some formal considerations by using our simple propositional calculus: P of Section 2 of Chapter 3. We recall that to produce all the well-formed strings of P a compositional grammar was appropriate. Its grammar requires only one elementary string and two rules. The single elementary string is \( pI \) and the two rules are

\[
\begin{align*}
(U) & \quad PC \rightarrow PCI \\
(C) & \quad \pi_1, \pi_2 \rightarrow \& \pi_1 \pi_2
\end{align*}
\]

We will now investigate how well such a compositional approach can characterize the substitution relationship. We will also draw the distinction between 'operation' and 'transformation'.

Suppose we wish to describe the relationship between

\[
(98) \quad \& \& pI I \& pI I \& pI I I pI I I I pI I I I I
\]

and

\[
(99) \quad \& \& pI \& pI \& pI I I pI I I I I.
\]

Conceptually the relationship is quite simple. The propositional constant \( pI I \) has been replaced in both occurrences by \( pI \). Of course, substitutional notation is designed exactly for such relationships, and we can express (98) to (99) as

\[
(100) \quad P_1(pI I) \rightarrow P_1(pI)
\]

This rule ignores an important detail, the concept of a proper occurrence of a string. We only replace occurrences of a string playing a definite grammatical role, not occurrences such as \( pII \) in the first part of \( pI I I I I \).

A compositional description of the relationship of (98) to (99) requires besides the rules (U) and (C) also their inverses

\[
\begin{align*}
(S) & \quad PCI \rightarrow PC \\
(D) & \quad \& \pi_1 \pi_2 \rightarrow \pi_1, \pi_2
\end{align*}
\]

in order to accomplish the decomposition and recomposition. If we use the lattice notation we can describe the relationship (98) to (99) as shown in (101) on page 70.

Reading from left to right, when two lines split off from a node it is a case of (D). When two lines merge to a node, it indicates (C). The only other transformation involved here is (S); only a single line is necessary for this transformation since no real decomposition or composition of two elements...
SUBSTITUTION

takes place. We have labeled the lattice with the names of the transformations to reinforce what was just said.

We repeat the finding of the last section that a simple substitution has been obscured by following a strict compositional approach. The operation of (101) can be abstracted as follows.

\[(102)\]

Many strings of \(P\) will fit the proper input shape to undergo this transformation. However, it is unclear that there is anything enlightening about such a generalization. It fails to be generalizable in a very important way. If there were three occurrences of \(pII\) in the input instead of two, or even if the two occurrences were in different locations, the appearance of the lattice would have been quite different. Neither of these factors would affect a substitutional description of the same relationship.

The preceding discussion has concerned a simple compositional language, that is, a language which is easily describable by a compositional grammar. In a compositional language, a lattice such as (102) is merely a description of a complex transformation of \(P\). All such lattices or transformations can be generated by a largely arbitrary selection of the elementary transformations arranged in an appropriate lattice form. The nature of the appropriate lattice forms does not concern us here. The point is that the elementary transformations: \((U)\), \((S)\), \((C)\), and \((D)\) could be taken as telling the whole story, though nothing like the concept of paraphrase could be characterized in such a way.

Let us now see how the situation changes if we replace the language \(P\) by a similar language \(P'\) which is non-compositional. A proposition of \(P'\) is always a proposition of \(P\). It is the case, however, that many propositions of \(P\) are not acceptable strings of \(P'\). There is no way of deciding completely which propositions of \(P\) are also propositions of \(P'\).

It seems that certain configurations of propositional constants are acceptable and others are not. Thus, given an acceptable proposition of \(P\) we can predict others. For example, if (98) were an acceptable proposition of \(P\),...
then (99) would also be acceptable; because they both have the same configuration of propositional constants

\[(103) \quad \& \& X \& X \& Y Z Y\]

The X, Y, and Z indicate the locations which carry identical propositional constants. Thus, the exact propositional constant is irrelevant. What matters is that the identity and distinctness of the constants in each location is preserved.

Let us limit our concern to transformations which preserve these configurations.

It will be useful for the points to be made to express the various operations co-ordinated by the lattice (102) in a linear fashion. To do this we simply need some way of naming the lines of the lattice and indicating on which line a given transformation works. When moving from left to right on a branch x, if a branching occurs, the top branch receives the name xL and the bottom branch the name xR, corresponding to the left conjunct being placed on the top branch and the right conjunct on the bottom branch. When a merger occurs, it will always be of a top line xL and a bottom line xR, or so we assume for simplicity in the present example. The resulting single line will be named simply x, the L and R being erased. Otherwise a line will have the same name as its predecessor. Thus, the lattice lines of (102) will be named as follows.

\[(104)\]

Let us indicate which line and hence which string a transformation operates on by subscripting it with the name of the line. D is subscripted with the name of the line it splits while decomposing a string into two conjuncts. S and U are subscripted with the name of the line which connects their input and output. On C let us subscript the names of the two merging lines, top before bottom.

In the above fashion one linear representation of (102) is

\[(105) \quad C_{L,R} \cdot C_{LL,LR} \cdot C_{LRL,LRR} \cdot S_{LRL,D_{LR}} \cdot S_{LL,D_{L,D}}\]
The component operations are applied from right to left. For example,

\[ (106) \quad C_{L,R} \cdot C_{LL,LR} \cdot C_{LRL,LRR} \cdot S_{RL} \cdot D_{LR} \cdot S_{LL} \cdot D_L \cdot D \]
\[ \& pII & pII & pII pII pII pII pII pII \]
\[ = C_{L,R} \cdot C_{LL,LR} \cdot C_{LRL,LRR} \cdot S_{RL} \cdot D_{LR} \cdot S_{LL} \cdot D_L \]
\[ \& [pII & pII & pII pII pII pII pII pII] \]
\[ = C_{L,R} \cdot C_{LL,LR} \cdot C_{LRL,LRR} \cdot S_{LRL} \cdot D_{LR} \cdot S_{LL} \]
\[ \& [pII, & pII & pII pII pII pII pII pII] \]
\[ = C_{L,R} \cdot C_{LL,LR} \cdot C_{LRL,LRR} \cdot S_{LRL} \]
\[ \cdot D_{LR} \cdot [pII, & pII & pII pII pII pII] \]
\[ = C_{L,R} \cdot C_{LL,LR} \cdot C_{LRL,LRR} \cdot S_{LRL} \]
\[ \cdot D_{LR} \cdot [pII, & pII & pII pII pII pII] \]
\[ = C_{L,R} \cdot C_{LL,LR} \cdot C_{LRL,LRR} \cdot S_{LRL} \]
\[ \cdot D_{LR} \cdot [pII, & pII & pII pII pII pII] \]

I referred to the various uses of C, D, and S as ‘operations’ rather than as ‘transformations’. In the language P they would all have been transformations, for they would always lead from acceptable propositions to acceptable propositions. Indeed, the very concept of a transformation is that it does not give values outside the set being studied. On the other hand, in the language P' the operations C, D, S, and U are not necessarily transformations. Even though \& pI pII and \& pI pII pII pII are acceptable propositions of P, their conjunction \& pI pII & pI pII pII pII may not be. Thus, in the language P' the transformation of (98) to (99) can be described as in (105), using C, D, and S; but we must be clear that they are merely operations which make up the transformation and are not themselves transformations.

The result of this distinction between transformation and operation for the language P is found in the description of the set of transformations of P. The set of transformations of P could be roughly characterized as all possible combinations of U, S, C, and D. Such a characterization is not possible for P. A transformation such as (105) can be described using the operations U, S, C, and D, but it cannot be generated from them as elementary transformations. If we did try to describe the transformations of P using the operations U, S, C, and D, every substitution which involves a different number of replacements or a different configuration of replacements must be described separately. One becomes involved in a task of characterizing an infinite number of transformations.

The general substitutional description, however, still stands as a unified description of the change on the whole set of propositions of P. The unification is easy and natural to achieve, for all the compositionally described transformations are defined on mutually exclusive subsets of the set of
propositions of \( P \). A transformation whose description includes three uses of the operation \( S \) will apply to different propositions than one with two uses of \( S \).

The central advantage of a description such as

\[
(107) \quad P_1(PC_2) \rightarrow P_1(PC_3)
\]

is clearly the unification of many conceptually similar transformations into a single transformation which emphasizes that conceptual similarity.

### 6. SUMMARY

The English language clearly shows the presence of general substitutional relationships. Pronominalization, direct to indirect discourse, and relative clauses are examples. We have developed an underlying substitutional notation and indicated extensions on it. The curved parentheses in \( X(Y) \) generally act as a universal quantifier, saying 'for all occurrences of \( Y \), replace them by...'. However, we also agreed to use them for a single occurrence of \( Y \) by subscripting them with '0' as in \( X_0(Y) \).

The substitutional idea and notation has a definite conceptual advantage over a strictly compositional or concatenational approach for expressing some transformations. It also has a mathematical value in unifying the domains of infinite numbers of transformations. I indicated that if we are dealing with a language not amenable to a compositional grammatical description, the argument for taking such a notation as fundamental is even stronger; for in such a case substitutional relationships could not be regarded as mere combinatorial consequences of a finite number of weaker compositional transformations.
CHAPTER V

ADMISSION AND EQUIVALENCE

The substitution notation is very powerful and we must be careful to place adequate conditions on its employment for any specific transformation. Strings which have the same referent should be regarded as equivalent for the purpose of substitution. There must be a procedure for forbidding certain syntactic configurations from entering certain transformations. The concepts of equivalence and admission apply to transformations in general—not merely substitutionally based transformations.

We treat the two concepts in the same chapter because the equivalence relation completes the description of admission conditions in an important way. Admission conditions often refer to all occurrences of a string.

1. ADMISSION CONDITIONS

A given grammatical notation automatically restricts the set of acceptable inputs to any significant transformation. Generally, however, such automatic restrictions are not sufficient, and additional conditions must be set on the admission of strings as inputs to a transformation.

For example, if we use a simple word-class and tree notation, we can give a description of the passive transformation.

(1) covers many examples.

(2)    the House and Senate ratify such agreements
       such agreements are ratified by the House and Senate

       many children chew pencils
       pencils are chewed by many children
few people like war
war is liked by few people

However, it does not bear up under the following examples.

(3) my dog is a patient creature
∧ a patient creature is been by my dog

the major seems a good leader
∧ a good leader is seemed by the major

John married Susan
- Susan was married by John

The ‘∧’ indicates that the paraphrase relation does not hold. One approach used to avoid (3) is the device of sub-classifying the word classes. In example (3) above we would distinguish two kinds of verbs: the class of verbs which participate in the passive transformation satisfactorily and the remaining verbs. Hopefully, this sub-classification will also have independent motivation, so it will not be too ad hoc. What we would have done in this case is to add a condition for admission to the passive transformation: that of requiring the verb to belong to a certain sub-class. The conditional is additional because it is not an automatic result of the basic tree notation. An example of a built-in or automatic condition is that the left-most node be a noun. The distinction between automatic and additional is highly relative. Indeed, in the above example the basic notation can be extended to incorporate sub-classification. Subscripts, superscripts, and additional nodes with more detailed description of restrictions on the word-classes have all been employed by linguists for this purpose. Whether such devices are integral or extra to the basic notation is not always clear. We will shortly give some examples in the substitution notation which are more decisive. We have mentioned the sub-classification approach, which can be seen as a way of adding conditions for admission to the transformation.

A second remedy for (3) is based on leaving formal traces of the manner of derivation of a string, that is of the transformational history of the string. Thus,

(4) John married Susan

is derived from

(5) the priest married John to Susan

via some permutations and deletions.
From (5) we first obtain, say

(6) **John through the priest married Susan**

and the adverbial phrase is deleted.

(7) **John married Susan**

Thus, the tree representation of (4) might be

(8)

\[
\begin{array}{c}
S \\
N \quad D \quad V \quad N \\
\text{John} \quad \emptyset \text{logical subject} \quad \text{married} \quad \text{Susan} \\
\text{through the priest}
\end{array}
\]

(8) would thus not have the correct form for the input to (1). In this way we could avoid the third example of (3). It is clear that the manner in which the strings of the language are represented has an effect on how transformations are formulated, but let us discuss the inclusion of transformational history in representations of strings in the chapter on functional representation.

In the present chapter we wish to explore the use of explicit conditions for admitting a string to a transformation. Let us turn our attention again to the relative clause. We recall the formulation of the previous chapter

(9)

\[
\begin{array}{c}
S \\
S_1 \{N_2\} \quad \text{because} \quad S_3 \{N_2\} \quad \rightarrow \quad S_1 \{N_2\} \quad A \quad \{wh[N_2]\}
\end{array}
\]

Remember that **wh**[**N_2**] is simply the relative pronoun for **N_2**, in this case, **which**. We note that the only thing clear about the string **S_1**(**N_2**) is that it is a sentence and has some occurrences of a noun **N_2**. The following example indicates that we need to be more restrictive about the strings that are admitted to (9)

(10) **the proposal was merely the visible sign of an already existing but unexpressed concurrence, because the proposal certainly was not spontaneous, but rather everyone expected a break-through**

(10) satisfies the input description of (9) with **N_2** being **the proposal**. However, the result

(11) **\not\exists the proposal, which certainly was not spontaneous but rather everyone expected a break-through, was merely the visible sign of an already existing but unexpressed concurrence**
is clearly incorrect. Roughly stated, the rule we have broken is

(12) Every sentential conjunct of a main co-ordinating construction in $S_3(N_z)$ must have at least one main occurrence of $N_z$.

In the case of (10) this condition requires that both of the conjuncts

(13) the proposal certainly was not spontaneous

and

(14) everyone expected a break-through

have exactly one occurrence of the proposal as a subject or object. The conjunct (14), of course, violates this condition. If instead of (14) we had

(15) everyone expected the proposal

then (13) and (15) would meet the condition and we could allow

(16) the proposal was merely the visible sign of an already existing but unexpressed concurrence, because the proposal certainly was not spontaneous, but rather everyone expected the proposal

to be input to (9) with the satisfactory result

(17) the proposal, which certainly was not spontaneous, but rather which everyone expected, was merely the visible sign of an already existing but unexpressed concurrence.

Condition (12) as formulated is extremely vague and contains a number of undefined terms. In order to state (12) in a more careful manner we must define some auxiliary concepts. The definitions of such concepts are interesting in themselves, for they often illustrate best the 'recursive' elements of a language.

2. ORDERING OF TRANSFORMATIONS

One of the technical difficulties of transformational grammars is that transformations often interfere with one another. For example, the introduction of the reflexive pronoun can disturb the result of the passive.

(18) Mrs. Wilson writes herself daily

(19) $\exists$ herself is written by Mrs. Wilson daily

One solution to this specific problem of interference is to order the transformations so that the reflexive pronoun is introduced after the passive and other order changing transformations. Indeed, it might be advantageous to have all pronominalization operations be delayed until the last.
However, ordering restrictions can be replaced by other kinds of conditions, namely admission conditions. But ordering restrictions are not themselves powerful enough to cover all admission conditions. These two facts can be seen quite easily.

**Fact 1:** Any ordering of the transformations can be replaced by admission conditions (with some notational adjustment). We need only be able to place at specific locations the names of the transformations which have already operated on a string. In general, we speak of a transformation which has operated as leaving a “trace”. The traces that are left depend on notational conventions, of course. Then, each transformation can be accompanied by an admission condition which specifies the names that cannot occur at those locations. Those names, of course, are the names of the transformations which occur in the ordering before the given transformation.

**Fact 2:** Ordering the transformations cannot account for all admission conditions. A simple example can illustrate this fact. Suppose we take the formal language \( \mathcal{P} \) of the last chapter. We recall that it was the form or configuration of interreference that was crucial, and that two strings which had the same configuration could be treated as paraphrases of each other. The rule for obtaining paraphrases according to this principle was

\[
(20) \quad S_1(\text{PC}_2) \rightarrow S_1(\text{PC}_3)
\]

For example,

\[
(21) \quad \& \& pI \ pII \ & \ pII \ pI
\]

is changed by (20) to

\[
(22) \quad \& \& pIII \ pII \ & \ pII \ pIII
\]

However, suppose we wish to replace \( pII \) in (21) by \( pI \). Using (20) we obtain

\[
(23) \quad \& \& pI \ pI \ & \ pI \ pI
\]

which has a quite different configuration from (21) or (22). Hence, we cannot regard (23) as a paraphrase of (21), and the transformation (20) is put in question. The difficulty is caused by the fact that \( pI \), which is to replace \( pII \), already occurs in (21). Thus, we can allow as inputs to (20) only those sentences which have no occurrence of \( \text{PC}_3 \). This kind of condition clearly cannot be obtained by ordering the transformations of \( \mathcal{P} \), for the presence of a particular \( \text{PC}_3 \) is either not due to any transformation or is due to (20) itself. Let us abbreviate this condition as ‘-occ \( \text{PC}_3 \)’ that is, ‘no occurrence of \( \text{PC}_3 \)’. We must also have a way of indicating which string must satisfy
the condition. We will place the condition under the first letter of the description of the string. Thus, the above transformation becomes

\[(24) \quad S_1(\text{PC}_2) \rightarrow S_1(\text{PC}_3)\]

Admission conditions are thus seen to be a more powerful device for the interrelation of transformations. We can hardly claim to have proven this point, primarily because no formal constraints have been put on the concept of admission condition. However, when the concept of admission is defined precisely, the above arguments can be converted into careful proofs.

Let us for the present simply say that an admission condition is a condition which qualifies a transformation according to the presence or absence of specified strings and configurations of strings in the input text. We will give a precise definition in Chapter seven.

3. A RECURSIVE DEFINITION AND AN ADMISSION CONDITION

The idea of a recursive definition is borrowed from mathematics. It is often a neat way of defining a concept and is useful for certain grammatical concepts. Roughly characterized, a recursive definition has three parts:

(0) a list of the elementary objects which fall under the concept

(1) rules which show how to find what other objects fall under the concept, if some are already given

(2) a statement that only those objects which are obtained from (0) and (1) fall under the concept.

For example, suppose you want to define recursively the concept of a string of alternating a's and b's; that is, \(ab, ba, aba, bab, \ldots\)

(0) \(ab\) and \(ba\) are \(a\)-\(b\) alternating strings

(1) if a string ending in \(a\), respectively \(b\), is an \(a\)-\(b\) alternating string, then the string obtained by adding \(b\), respectively \(a\), at the end of the original string is also an \(a\)-\(b\) alternating string

(2) only those strings given by (0) or (1) are \(a\)-\(b\) alternating strings.

Usually (2) is left tacitly understood and not written out. Thus, from (0) we know that \(ab\) and \(ba\) are \(a\)-\(b\) alternating strings; from what we have just said and by using (1) we know that \(aba\) and \(bab\) are \(a\)-\(b\) alternating strings, and from that knowledge and (1), we get \(abab\) and \(baba\), and so forth. We will discuss the concept of recursive definitions in detail in the appendix.
We will give a recursive formulation of (12) above. But before we do we need some auxiliary concepts.

In any text of English the component strings are grouped together in particular ways. Tree diagrams indicate the groupings

\[
\begin{array}{c}
\text{S} \\
\text{N} \\
\text{A} \\
\text{N} \\
\end{array} \\
\begin{array}{c}
\text{V} \\
\text{N} \\
\text{A} \\
\text{N} \\
\end{array}
\]

\[
\text{people from Philadelphia \hspace{1cm} like green curtains and red tassels}
\]

We will discuss these groupings as cases of 'joining', that is, the grouping of two strings of given grammatical categories to form a third string of some grammatical category. The most general case of joining is junction:

\[
\begin{array}{c}
\text{Z} \\
\text{X} \\
\text{Y}
\end{array}
\]

Two strings of different categories X and Y are joined, possibly with additional material, to form a string of a yet different category Z. One example of junction is

\[
\begin{array}{c}
\text{S} \\
\text{N} \\
\text{A} \\
\text{N} \\
\end{array} \\
\begin{array}{c}
\text{V} \\
\text{N} \\
\text{A} \\
\text{N} \\
\end{array}
\]

\[
\text{people from Philadelphia \hspace{1cm} like green curtains and red tassels.}
\]

Another example is

\[
\begin{array}{c}
\text{V} \\
\text{N}
\end{array}
\]

\[
\text{like \hspace{1cm} green curtains and red tassels.}
\]
A second kind of joining is con-junction. Here two strings of the same grammatical category are joined, possibly with additional material, to form a string again of the same category. Thus, conjunction can be represented schematically as

\[(29)\]

\[\begin{array}{c}
X \\
X_1 \\
X_2
\end{array}\]

In the above example we have a case of con-junction

\[(30)\]

\[\begin{array}{c}
N \\
N \\
\text{green curtains and red tassels}
\end{array}\]

A third kind of joining is ad-junction. Two strings, usually of different categories, are joined so that the resultant string has the category of one of the original strings, schematically

\[(31)\]

\[\begin{array}{c}
X \\
Y \\
X
\end{array}\] or

\[\begin{array}{c}
X \\
X \\
Y
\end{array}\]

For example,

\[(32)\]

\[\begin{array}{c}
N \\
A \\
N \\
\text{green curtains}
\end{array}\]

If both strings are of the same category, ad-junction resembles con-junction. For example,

\[(33)\]

\[\begin{array}{c}
N \\
N \\
\text{Philadelphia people}
\end{array}\]
The distinction between adjunction and conjunction rests ultimately on the concept of paraphrase. The conjunction (30) in (25) derives from

(34)

That is, *green curtains* and *red tassels* occupy parallel slots in the source sentences. But in

(35)

the strings *Philadelphia* and *people* have a non-parallel relationship in the source sentences

(36)

In fact, the whole discussion of joining should be founded on definitions based on paraphrase. However, our goal here is not to define joining itself, but rather merely to illustrate the use of recursion in admission conditions. So we are giving only a most general and tentative sketch of joining.

Another example of ad-junction is in order.
We note that the above use of *I know* is valid only for non-con-juncted sentences. The D must find one of the positions marked by arrows. In more complex sentences there are more than three positions, of course. For con-juncted sentences and sentences in general *I know* has a different analysis, which we will illustrate directly below.

The three types of joining above complete the system, except for a class of preparatory operations, which we call pre-junction. Pre-junction modifies a string to prepare it for one of the other cases of junction. For example,

(38)  
```
A  
/  
N  
```

for ad-junction to *people*.

(39)  
```
N  
/  
S  
```

for junction to *know* in *I know that people from Philadelphia like green curtains and red tassels*.

The distinction between the two uses of *I know* will be very important in the correct formulation of condition (12) above. The importance will be illustrated after we complete the precise formulation of (12).

Further examples of pre-junction are the following

(40)  
```
D  
/  
S  
```

because *people from Philadelphia like green curtains and red tassels*
One of the most complex forms of pre-junction is the formation of the relative of a sentence:

(41)  
\[
\begin{align*}
\text{wh-} & \quad \text{A/N} \\
& \quad \text{S} \\
& \quad \text{N} \\
& \quad \text{[people from Philadelphia]} \\
& \quad \text{V} \\
& \quad \text{like} \quad \text{green curtains and red tassels} \\
& \quad \text{V} \\
& \quad \text{N} \\
\end{align*}
\]

which we might have in

(42)

\[
\begin{align*}
\text{N} & \\
& \quad \text{A} \\
& \quad \text{people from Philadelphia} \\
& \quad \text{who} \quad \text{like green curtains and red tassels} \\
\end{align*}
\]

or

(43)

\[
\begin{align*}
\text{S} & \\
& \quad \text{N} \\
& \quad \text{V} \\
& \quad \text{I} \\
& \quad \text{know} \quad \text{who} \quad \text{like green curtains and red tassels} \\
\end{align*}
\]

Another case of the relative is

(44)

\[
\begin{align*}
\text{wh-} & \quad \text{A} \\
& \quad \text{S} \\
& \quad \text{N} \\
& \quad \text{[people from Philadelphia]} \\
& \quad \text{like} \quad \text{green curtains and red tassels} \\
& \quad \text{I} \\
& \quad \text{[or]} \\
& \quad \text{N} \\
& \quad \text{V} \\
& \quad \text{N} \\
\end{align*}
\]
For example,

\[ (45) \]

\[
\begin{array}{c}
N \\
\downarrow \\
N \quad A
\end{array}
\]

\textit{green curtains and red tassels people from Philadelphia like}

We have given a partial definition of the various kinds of joining in order to show how a careful definition would proceed. Now let us use the concepts of joining as though they were carefully defined.

The central concept in the precise formulation of condition (12) above is the level of an occurrence of one string in another. We wish to distinguish a level 0 occurrence of a noun in a sentence from higher level occurrences. Intuitively, level 0 occurrences of a noun are those that occur right at the surface, not in any imbedded or subordinate clause.

\[ (46) \]

(0) a noun \(N_1\) occurs in itself at level 0

(1a) if \(N_1\) occurs in \(X_2\) at level \(n\), then that occurrence in a con-/ad-/junction of \(X_2\) with \(Y_3\) is also at level \(n\)

(1b) if \(N_1\) occurs in \(X_2\) at level \(n\), then that occurrence of \(N_1\) in a pre-junction \(Y_3\) of \(X_2\) is at

(i) level \(n+1\), if \(X_2\) is a sentence

(ii) level \(n\), for any other category

Let us illustrate this definition with an earlier example.

\[ (47) \] \textit{the proposal}

occurs in itself at level 0. It occurs in

\[ (48) \] \textit{would accept the proposal}

as part of junction, hence still at level 0. Through junction again it occurs in

\[ (49) \] \textit{they would accept the proposal}

at level 0. The ad-junction of \textit{he was certain} to form

\[ (50) \] \textit{he was certain they would accept the proposal}

keeps the occurrence of \textit{the proposal} at level 0. It then participates in a con-junction to form

\[ (51) \] \textit{they had exhausted their own proposals and he was certain they would accept the proposal}

staying at level 0. (51) then undergoes pre-junction by \textit{after} to put the occur-
rence of the proposal at level 1 in

\[(52) \quad \text{after they had exhausted their own proposals and he was certain they would accept the proposal.}\]

The occurrence of the proposal participates next in an adjunction to form

\[(53) \quad \text{he should put forth the proposal after they had exhausted their own proposals and he was certain they would accept the proposal}\]

and remain at level 1. Then by ad-junction of the President had instructed two months ago it remains at level 1 in

\[(54) \quad \text{the President had instructed two months ago he should put forth the proposal after they had exhausted their own proposals and he was certain they would accept the proposal.}\]

The final con-junction leaves it at level 1 in

\[(55) \quad \text{the proposal certainly was not spontaneous, but rather the President had instructed two months ago he should put forth the proposal after they had exhausted their own proposals and he was certain they would accept the proposal.}\]

We note that the other two occurrences of the proposal are both at level 0, by similar arguments to the above.

**An Admission Condition**

The most difficult concepts underlying condition (12) are behind us. We now need to refer to the level of a sentence.

\[(56) \quad \text{An occurrence of } S_1 \text{ is at level } n \text{ in } S_2 \text{ if and only if it contains nouns at level } n, \text{ but no nouns at any lower level in } S_2.\]

For example,

\[(57) \quad \text{the proposal certainly was not spontaneous}\]

occurs at level 0 (S⁰) in (55), while (49) occurs at level 1 (S¹).

Finally, we need a definition of a minimal sentence.

\[(58) \quad S_1 \text{ is a minimal sentence (min } S_1) \text{ if and only if it is not a conjunction of two sentences.}\]

Thus, (49) and (50) are minimal sentences (min S), but (51) is not. However, (53) is again a minimal sentence, since the subordinating ‘conjunction’ after constitutes an adverbial ad-junction and not a con-junction as defined above.
Condition (12) can now be restated precisely as follows.

(59) With respect to and in $S_3(N_2)$, every minimal level 0 sentence ($\text{min } S^0$) must have some level 0 occurrences of $N_2(N^0_2)$: $N_2^0/\text{min } S^0$.

If we examine (55) we find that it has two minimal level 0 sentences ($\text{min } S^0$), namely (54) and (57). There are two occurrences of the proposal in (54); the first is at level 0 in (55), but the second is at level 1, as we showed earlier. Thus, (54) has one level 0 occurrence of $N_2(N^0_2)$, with respect to (55). (57) obviously has $N_2^0$ with respect to (55). So (55) satisfies $N_2^0/\text{min } S^0$, where $N_2$ is the proposal. Therefore we can admit (55) to (9), which we now write as follows.

(60)

\[ S \rightarrow S_1 \{N_2\} \quad \text{because} \quad S_3 \{N_2\} \quad \rightarrow \quad S_1 \{N_2\} \quad A_3 \{\text{wh [N}_2]\} \]

$N_2^0/\text{min } S^0$

We know that $N_2^0/\text{min } S^0$ applies to $S_3(N_2)$, because its statement begins directly below the beginning of $S_3(N_2)$.

We have seen in this section how auxiliary concepts and recursive definitions can lead to the succinct formulation of a rather difficult admission condition.

4. EQUIVALENCE

The concept of equivalence of strings in a text forms a topic worthy of detailed and independent exposition. However, we shall treat it as a means to an end. We treat several occurrences of a string as equivalent. It is an essential concept for the substitution operation and for admission conditions, both of which generally refer to 'all occurrences' of a given string, or to the number of occurrences of the string, as in the condition defined in the last section.

Whatever definition of equivalence we give must satisfy the three basic properties of any equivalence relation:

(61)

\[ \begin{align*}
(0) & \quad \text{any object } x \text{ is equivalent to itself: } x \equiv x, \\
(1) & \quad \text{if } x \text{ is equivalent to } y, \text{ then } y \text{ is equivalent to } x: x \equiv y \rightarrow y \equiv x, \\
(2) & \quad \text{if } x \text{ is equivalent to } y \text{ and } y \text{ to } z, \text{ then } x \text{ is equivalent to } z: (x \equiv y \land y \equiv z) \rightarrow x \equiv z.
\end{align*} \]

The objects we are considering are occurrences of linguistic strings, in the spelling of some systematic notation.
At the very outset we assume that

(62) occurrences composed of the same symbols in the same manner are equivalent.

For example, the left-hand occurrence of the Philadelphia boy is equivalent to the right-hand one in

(63) the Philadelphia boy's father likes the Philadelphia boy

But for the purposes of our analysis we cannot stop here. Consider, for example, the following sentence.

(64) we cannot invite the boy who is from Philadelphia, because the boy from Philadelphia has a friend who is coming and the Philadelphia boy's friend doesn't like the boy from Philadelphia anymore

A very superficial analysis would say that the because clause doesn't have any occurrences of the boy who is from Philadelphia, so the relative clause would not yield

(65) we cannot invite the boy who is from Philadelphia, who has a friend who is coming and whose friend doesn't like him anymore.

It would, however, apply to

(66) we cannot invite the boy who is from Philadelphia, because the boy who is from Philadelphia has a friend who is coming and the boy who is from Philadelphia's friend doesn't like the boy from Philadelphia anymore

but the result would be

(67) \( \neg \) we cannot invite the boy who is from Philadelphia, who has a friend who is coming and whose friend doesn't like the boy from Philadelphia anymore.

The transformation failed to affect one variation on the boy who is from Philadelphia, because it wasn't identical to the others.

For certain transformations, for example, the restrictive relative clause, we wish to ensure that we have identified the first occurrence of the noun-string by specifying that it not occur to the left of a given point. It is important in such cases to recognize non-identical variants as being subject to the condition. Thus, in a purely graphic sense the first occurrence in (64) of the boy from Philadelphia is immediately after because. However, the semantic pattern of interreference clearly includes an earlier string, namely the boy who is from Philadelphia. In this case we wish to refer to this last string also
as an occurrence of *the boy from Philadelphia*, in the sense that it is an occurrence of an equivalent string.

In the case of the above examples we will treat the noun strings

(68)  
\[
\text{the boy who is from Philadelphia} \\
\text{the boy from Philadelphia} \\
\text{the Philadelphia boy}
\]

as transformationally related, that is, from *the boy who is from Philadelphia* the remaining two will be transformationally derived. We will use this transformational relation as also part of the equivalence relation. In general, we will strive to account for all such equivalences transformationally. Conversely, with nearly every transformation will be associated an equivalence.

Let us consider in more detail the case of (68). Two transformations are relevant.

(69) 
\[
\text{\begin{tikzpicture}
  \node (n1) {N} edge (A);
  \node (a1) {A} edge (n1);
  \node (n2) {N} edge (a1);
  \node (v1) {V} edge (n2);
  \node (n3) {N} edge (v1);
  \node (p1) {\{is\}} edge (n3);
  \node (n4) {N} edge (p1);
  \node (n5) {N} edge (n4);
  \node (p2) {\{are\}} edge (n5);
  \node (n6) {N} edge (p2);
  \node (p3) {\{was\}} edge (n6);
  \node (n7) {N} edge (p3);
  \node (p4) {\{were\}} edge (n7);
  \node (n8) {N} edge (p4);
  \node (p5) {S_1 \{N_2 \}} edge (n8);
  \node (n9) {N} edge (p5);
  \node (p6) {S_1 \{N_2 \}} edge (n9);
\end{tikzpicture}} \]

(70) 
\[
\text{\begin{tikzpicture}
  \node (n1) {N} edge (A);
  \node (a1) {A} edge (n1);
  \node (n2) {N} edge (a1);
  \node (p1) {\{\text{\}}} edge (n2);
  \node (n3) {N} edge (p1);
  \node (p2) {\{\text{\}}} edge (n3);
  \node (n4) {N} edge (p2);
  \node (p3) {\{\text{\}}} edge (n4);
  \node (n5) {N} edge (p3);
  \node (p4) {\{\text{\}}} edge (n5);
  \node (n6) {N} edge (p4);
  \node (p5) {S_1 \{N_2 \}} edge (n6);
  \node (n7) {N} edge (p5);
  \node (p6) {S_1 \{N_2 \}} edge (n7);
\end{tikzpicture}} \]

From (69) and (70) we derive the following two equivalences, respectively:

(71) 
\[
\text{\begin{tikzpicture}
  \node (n1) {N} edge (A);
  \node (a1) {A} edge (n1);
  \node (n2) {N} edge (a1);
  \node (v1) {V} edge (n2);
  \node (n3) {N} edge (v1);
  \node (p1) {\{is\}} edge (n3);
  \node (n4) {N} edge (p1);
  \node (p2) {\{are\}} edge (n4);
  \node (n5) {N} edge (p2);
  \node (p3) {\{was\}} edge (n5);
  \node (n6) {N} edge (p3);
  \node (p4) {\{were\}} edge (n6);
  \node (n7) {N} edge (p4);
  \node (p5) {S_1 \{N_2 \}} edge (n7);
  \node (n8) {N} edge (p5);
  \node (p6) {S_1 \{N_2 \}} edge (n8);
\end{tikzpicture}} = \text{\begin{tikzpicture}
  \node (n1) {N} edge (A);
  \node (a1) {A} edge (n1);
  \node (n2) {N} edge (a1);
  \node (p1) {\{\text{\}}} edge (n2);
  \node (n3) {N} edge (p1);
  \node (p2) {\{\text{\}}} edge (n3);
  \node (n4) {N} edge (p2);
  \node (p3) {\{\text{\}}} edge (n4);
  \node (n5) {N} edge (p3);
  \node (p4) {\{\text{\}}} edge (n5);
  \node (n6) {N} edge (p4);
  \node (p5) {S_1 \{N_2 \}} edge (n6);
  \node (n7) {N} edge (p5);
  \node (p6) {S_1 \{N_2 \}} edge (n7);
\end{tikzpicture}}
\]
Thus, in general if we strip away the overall context from a transformation, that is, $S_1$, then we automatically have an equivalence. There may be other equivalences associated with a given transformation and these will have to be stated explicitly, but many words can be saved by the above convention.

There is still a very important extension of the equivalence relation remaining.

(73) Strings which are identical except for equivalent parts are equivalent, that is,

$$(x \equiv y \& z \equiv w) \rightarrow xz \equiv yw.$$  

For example

(74) *the boy from the Philadelphia team*

and

(75) *the boy from the team from Philadelphia*

are equivalent because

(76) *the Philadelphia team*

and

(77) *the team from Philadelphia*

are equivalent.

The general statement (73) may be too broad for a detailed grammar, that is, some of the equivalences it produces may cause incorrect transformational relationships. However, it does characterize well the fundamental idea of equivalent parts implying equivalent wholes.

The co-ordination of the substitution operation and the equivalence relation can be refined. Instead of replacing each of the equivalent strings by one and the same string, we can simply apply the same operation to each equivalent, yielding different but still equivalent strings. For example, if we use the relative clause for the purpose of distinguishing the referents, there is no reason to cancel all previous transformations. Thus, from

(78) *we know a boy who is from Philadelphia, but we cannot invite the boy who is from Philadelphia, because the boy from Philadelphia has a friend who is coming and the friend of the Philadelphia boy doesn't like the boy from Philadelphia anymore*
we don't want

(79)  we cannot invite the boy who is from Philadelphia whom we know, because the boy who is from Philadelphia whom we know has a friend who is coming and the friend of the boy who is from Philadelphia whom we know doesn't like the boy who is from Philadelphia whom we know anymore

but rather prefer

(80)  we cannot invite the boy who is from Philadelphia whom we know, because the boy from Philadelphia whom we know has a friend who is coming and the friend of the Philadelphia boy whom we know doesn't like the boy from Philadelphia whom we know anymore.

The single operation of adjoining whom we know to each equivalent yields equivalent but different results – results which preserve the effects of previous transformations:

(81)  \[
\text{the boy who is from Philadelphia} + \text{whom we know} \\
\text{the boy from Philadelphia} + \text{whom we know} \\
\text{the Philadelphia boy} + \text{whom we know.}
\]

Thus, the distinguishing relative clause transformation would contain the following direction

(82)  \[
\ldots S_1(N_2) \rightarrow \ldots S_1(N_2 \xrightarrow{A_3} \ldots)
\]

where \( A_3 \), for example, would be whom we know. (82) requires that all occurrences of \( N_2 \), including occurrences of equivalents be replaced by \( N_2 A_3 \). But in each place the precise form of the replacing string \( N_2 A_3 \) depends on the equivalent on \( N_2 \) occurring in that place.

5. Summary

The precise statement of the conditions that a string must satisfy to enter a transformation is an important part of the description of the transformation. While some of these conditions can be built into the notation of structural descriptions, in general they must be simply appended to that notation, and properly in a highly visible way.

Though they may appear extraneous to the basic transformational notation, the admission conditions can involve very penetrating and interesting
facts about the structure of a language. In the example of this chapter we made significant use of the notion of recursive definition for one such admission condition.

The concept of equivalence among strings in a text is necessary to the proper use of the substitution operation and admission conditions. The sketch of the definition of this concept of equivalence also had the form of a recursive definition.

Auxiliary concepts in a transformational grammar are clearly revealing about the language and demand significant mathematical tools for their description.
CHAPTER VI

FUNCTIONAL REPRESENTATION

A stretch of speech in a language can be described or represented in many different ways. Traditionally, there are phonetic, phonemic, morphophonemic, morphological, syntactic and semantic descriptions. I wish to talk about two broad kinds of description: syntactic and phonological, and to relate them to each other. I am proposing two basic representations, one syntactic and the other phonological, such that the phonological representations are the arguments and values of the functions which are notated in the syntactic representation. There is an analogy with mathematics. The natural numbers are to the arithmetical functions what phonological strings are to syntactic functions. Since this is a book on syntax, I am primarily concerned with the syntactic functions. Thus, in this chapter I will discuss the syntactic functions from two aspects, the first their use for defining syntactic transformations and the second their phonological definition. The latter aspect will involve us in certain elements of phonological representation.

1. FUNCTIONAL NOTATION

A general functional notation is employed by mathematicians when they do not wish to specify the precise operation involved in finding the value of the function, given the value of the arguments. For example, the function of adding two numbers \(x\) and \(y\) is normally written

\[(1) \quad x + y.\]

However, to emphasize its character of being a function of two arguments, rather than of one or three, we can write it as

\[(2) \quad +[x, y]\]

thereby reflecting the general form of two-argument functions

\[(3) \quad f[x, y]\]

as different from one argument functions

\[(4) \quad f[x]\]
or from three argument functions

(5) \( f[x, y, z] \).

In linguistics we also make use of a simple two argument function. When two strings \( x \) and \( y \) are concatenated to form a single string \( xy \), they are exhibiting a function: the concatenation function. However, we make no indication of the function involved. It is a two argument function and, thus, can profitably be written in the form of (3):

(6) \( C[x, y] \)

where \( C \) stands for concatenation.

In mathematics the various arguments and the value are not usually distinct in kind. They are all from a single domain: the natural numbers, the integers, the rational numbers, the real numbers and so forth. In linguistics it is useful to distinguish classes of strings: nouns (N), verbs (V), adjectives (A), adverbs (D), and others and to indicate which arguments draw from which classes and what class the value of the function belongs to. For example, by

(7) \( S \)

we have become accustomed to represent the concatenation function of two arguments, the first being a noun, the second a verb. The resultant string is a sentence. We could write the function as follows.

(8) \( S_C[N, V] \)

where the \( S \) indicates the value of the function and the subscript \( C \) shows that the operation is the concatenation operation. Actually, the operation of concatenation is an extreme simplification of the actual joining of strings in language and little information is added through that \( C \). So we will drop the symbol and let the \( S \) carry the concatenation information. We now write (7) as

(9) \( S[N, V] \)

Suppose we consider another arithmetical function

(10) \( x^2 \)

that is, \( x \) multiplied by itself. It is a single argument function and, hence, has the form of \( f[x] \). Perhaps it could be written

(11) \( P^2[x] \)
to emphasize its form. \( P^2 \) reads 'to the second power', keeping to the spirit of (10). By distinguishing the 2 we wish to relate (10) to \( x^3, x^4 \), and so forth. But we might as well have a systematic location for the value of the power. So we, with mathematicians, write

\[(12) \quad P[2][x].\]

The argument is \( x \), but the function symbol has become complex: \( P[2] \). \( P[2] \) is in a form that accommodates all the power functions, that is, it has the form \( P[m] \), where \( m \) can be any natural number. We can now write the form of (10) as

\[(13) \quad f[m][x].\]

In such a form we call \( x \) the argument and \( f[m] \) the function with a parameter \( m \). The choice of the parameter determines the precise function involved, when \( f \) is specific.

In linguistics we are often similarly motivated. Consider, for example,

\[(14) \quad \begin{array}{c} \text{N} \\
\text{the} \\
\text{N} \end{array}\]

We might want to call it the the-function and write

\[(15) \quad N_{\text{the}}[N]\]

emphasizing that the argument is a noun and so is the value. But we know it is similar to many other functions: the a-function, *some*-function, even the red-function, so we adopt a parametric role for the and write

\[(16) \quad N[\text{the}][N].\]

Let me summarize systematically the preceding ideas. The functional notation is fundamentally composed of function symbols and argument symbols. Brackets are used for punctuation. The last set of brackets contains the arguments proper of the function, set off by commas.

\[(17) \quad f[x, y, \ldots, z]\]

The function symbol itself might be complex, that is, might contain a number of parameters. The parameters are contained in the bracket pairs of the function symbol. Each pair of brackets might contain any number of parameters. Thus, the general form of the notation we are using is

\[(18) \quad f[a, b, \ldots, c][h, i, \ldots, j][p, q, \ldots, r][x, y, \ldots, z].\]

We also note that we have a much richer notation than that of labeled trees, for we could translate such trees without making any use of parameters.
The basic translation rule says to place the top node $X$ in

(19) $\begin{array}{c}
X \\
\downarrow \\
Y & Z \\
\end{array}$

to the left of the lower nodes, which are enclosed in brackets and separated by commas:

(20) $X[Y, Z]$

The question now is to what use this extra richness can be put.

2. FUNCTIONS AND TRANSFORMATIONS

In the last section we showed how the properties of a labeled tree notation can be incorporated into the functional notation. That is, we showed how the phrase structure of a string was represented. The relevance of phrase structure to transformations is well-known. Transformations are defined, not on strings of sounds, but on the phrase structure representations. Let us now see the value of the above functional notation for defining transformations.

We have just seen that the first letter of a complete functional phrase, that is, one that describes a string of the language, indicates the grammatical category of the string. For example, the $S$ of

(21) $S[the\ proposal,\ was\ spontaneous]$

indicates that

(22) $the\ proposal\ was\ spontaneous$

has the grammatical category of a sentence in the given context. The $A$ of

(23) $A[which,\ was\ spontaneous]$

states that

(24) $which\ was\ spontaneous$

is an adjective in the given context.

There is no doubt that grammatical category symbols are useful to transformations; for example, (23) can be altered to a prenominal adjective, which would not be correct if it began with an $S$.

Transformations will often require that category symbols be systematically changed. The relative clause transformations offer good examples. If we only
indicate the grammatical categories and groupings, but not the detailed structure in the following

(25) \[ S[S[the proposal certainly was not spontaneous], but rather S[the president had instructed two months ago S[he should put forth the proposal D[after S[they had exhausted their own proposals] and S[he was certain S[they would accept the proposal]]]]]] \]

then we can illustrate the change in category symbols that accompanies the formation of the relative clause. All occurrences of S in (25) are replaced by A.

(26) \[ A[A[which certainly was not spontaneous], but rather A[which the President had instructed two months ago A[he should put forth D[after A[they had exhausted their own proposals] and A[he was certain A[they would accept it]]]]]] \]

Thus, the description (60) of a relative clause transformation in the last chapter must be altered to account for this general substitution.

(27) \[ S \quad D \quad N \]

\[ S_1 \{N_2\} \quad S_2(S)\{N_2\} \quad S_1 \{N_2 A_3(A)\{N[wh]N_2\}\} \]

\[ N_2/min S^0 \]

S_2(S)\{N_2\} on the left side of the transformation co-ordinates with A_3(A)\{N[wh]N_2\} on the right side to show that two substitutions occur. The first is the replacement of N_2 everywhere by N[wh]N_2, the relative pronoun for N_2. The notation will be explained shortly. The second is the replacement of the grammatical category symbol S everywhere by the category symbol A.

We have already seen in the discussion of admission conditions one use for the types of joining relations strings enter into: junction, con-junction, and ad-junction. However, the role this three-fold distinction played there could have been filled by many other schemata. We will now incorporate the joining schema into the functional notation and indicate its importance.

We recall that junction involved a pair of strings, usually of different categories combined to form a string of still a different category. The two argument strings are on an equal basis, that is, neither dominates the construction. We represent junction as follows

(28) \[ X[Y, Z] \]
where X, Y, and Z generally indicate different categories. We might fruitfully characterize the basic sentence structure as junction,

\[(29)\] \(S[N, V]\)

for example,

\[(30)\] \(S[\textit{the proposal, was spontaneous}]\)

Conjunction also relates two strings neither of which dominates the construction. However, in conjunction the resultant string plays the same role as each of the components.

\[(31)\] \(X[X, X]\)

\(X\) indicates a single grammatical category. The conjunction of a pair of nouns illustrates this construction,

\[(32)\] \(N[N, N]\)

for example,

\[(33)\] \(N[\textit{the king, the pauper}]\).

Of course, the precise manner of the conjunction must be spelled out in English. So let us specify the function \(N\) a bit more by adding the parameter \(\text{and}\).

\[(34)\] \(N[\textit{and}] [\textit{the king, the pauper}]\)

has the value

\[(35)\] \(\textit{the king and the pauper}\).

The example (30) of conjunction also needed a parameter to indicate the manner of combination: whether as a statement, question, or whatever.

In the relation of ad-unction one of the strings dominates the construction and has the same category as the result. We treat the other string, the modifying string, as a parameter,

\[(36)\] \(X[Y] [X]\)

where \(X\) and \(Y\) generally indicate different categories, \(Y\) being the parameter. A typical example of ad-unction is the modification of a noun by an adjective,

\[(37)\] \(N[A] [N]\)

for example,

\[(38)\] \(N[\textit{red}] [\textit{shoes}]\).
The essence of the three-fold distinction of junction, con-junction, and ad-junction lies in the general categories of transformational relationships which they signal. Let us represent these relationships abstractly as follows.

\[(39)\quad \{xy\}z \leftarrow \{xy\}z\quad \text{junction}\]
\[(40)\quad \{xy\}z \leftarrow \{x\}z, \{y\}z\quad \text{con-junction}\]
\[(41)\quad \{xy\}z \leftarrow \{x\}z, \{x\}y\quad \text{ad-junction}\]

x and y are the joined strings, while z is the context, possibly null. The brackets indicate identity of roles or categories, that is, the string inside each pair of brackets has the same category at all places in the abstract transformation.

Junction does not indicate any transformational relationship beyond the identity transformation. That does not mean it permits no transformations, but only that those that do operate cause varied departures from the structure of \(\{xy\}z\) without indicating any preferred relationship of x and y. The joining of noun and verb to form a sentence in English may well be a case of junction. Consider, for example,

\[(42)\quad \text{the proposal was spontaneous.}\]

Nothing can be removed while at the same time preserving the general sentential role of (35). Alterations can occur.

\[(43)\quad \text{it was the proposal that was spontaneous}\]
\[(44)\quad \text{it was the spontaneity of the proposal}\]
\[(45)\quad \text{it was a spontaneous proposal}\]

But I have no clear picture that they indicate a particular preferred relationship of

\[(46)\quad \text{proposal}\]

and

\[(47)\quad \text{spontaneous}\]

in (35). Therefore, we write simply

\[(48)\quad S[\text{the proposal, was spontaneous}].\]

The discussion of the joining of strings is quite tentative. Research in the constraints of context is necessary to give a clearer idea of the functioning of the joining relations. Not all of (43)–(45) will be options in a given context. The present hypothesis is that a quite neutral way of joining two strings is that exemplified by (48), namely junction. Ad-junction is not so neutral.
(45) falls under the category of ad-junction. It can be analyzed in the abstract pattern of (41). The juxtaposition is of (45) and

(49) \textit{it was a proposal.}

That is, both

(50) \textit{proposal}

and

(51) \textit{spontaneous proposal}

have the same role in (49) and (45) respectively. According to (41) we expect a source pattern such as

(52) \begin{align*}
\textit{it was a \{proposal\}} \\
\textit{the \{proposal\} was spontaneous.}
\end{align*}

Thus, we have good indication of ad-junction and write (51) as

(53) N[\textit{spontaneous}] [\textit{proposal}].

The pattern of con-junction is the easiest of all to identify. Consider the example

(54) \{\textit{the king and the pauper}\} wanted a meal

which fits the pattern of (40) perfectly.

(55) \begin{align*}
\{\textit{the king}\} \textit{wanted a meal} \\
\{\textit{the pauper}\} \textit{wanted a meal}
\end{align*}

The expressions in brackets in (54) and (55) all play the same role in their respective sentences. We, thus, are justified in writing

(56) N[\textit{and}] [\textit{the king, the pauper}].

In certain positions in English we don’t need to have a conjunction particle and an ambiguity of joining can result.

(57) \textit{the king pauper episode}

We can regard \textit{king pauper} either as a con-junction

(58) N[\textit{king, pauper}]

derived from

(59) \textit{the king episode}

and

(60) \textit{the pauper episode}
or as an ad-junction

(61) $\text{N}[\text{king}] [\text{pauper}]
$

derived from

(62) the pauper episode

and

(63) the pauper was a king.

Additional motivation comes from the combination of transformational use and phonological evaluation of the functions. We analyze the sentence

(64) John laughs when he makes a mistake

as the ad-junction

(65) $\text{S}[\text{D}[\text{when}] [\text{he makes a mistake}]] [\text{John, laughs}]
$

rather than as the con-junction

(66) $\text{S}[\text{when}] [\text{John laughs, he makes a mistake}]
$

because the positions

(67) when he makes a mistake
takes are internal to the sentence very much as an adverb.

(68) John laughs when he makes a mistake

when he makes a mistake John laughs

John when he makes a mistake laughs

and so the S-function of

(69) John laughs

must operate on them in those positions. We will see this point even more clearly in the next section.

Though (67) takes internal positions we do not want to represent it inside, for then it would be difficult to write, say, the relative clause transformation we have discussed previously.

(70) $S_1 [\text{D}[\text{because}] [S_3 \text{(S)} \text{(N}_2)]]] \circ (\text{N}_2) \rightarrow S_1 (\text{N}[\text{A}_3 \text{(A)}] (\text{N}[\text{wh}]

[\text{N}_2])] [\text{N}_2])_o

\text{N}_2/\text{min} \text{S}_0
$

The transformational notation generally uses the functional notation to describe the functional notation. Noteworthy deviations are the use of
numerical subscripts to identify strings and functions and the hiding of any number of parameters in a category symbol. The schema $S_1[D]\left(N_2\right)$ on the left side of the transformation may be a bit puzzling. It has the basic form $S_1\left(N_2\right)$ except that the context $S_1$ is specified as being an ad-junct, where $D$ is the ad-junct.

So far we have discussed only one use for the parametric representations, namely to indicate ad-junction. There are other uses to which I wish to put the parametric representation. I indicated one other use already – to distinguish the individual kinds of con-junction, for example, (56) above. Other uses include governance and transformational history, which we will discuss shortly. The category symbol followed by the string of parameters contain a great deal of information and different kinds of information. We might choose a different style bracket for each kind of information, but it is not necessary; the parameters themselves will be distinction enough. We will, however, adopt some conventions. The category symbol itself may carry some information besides the category of the resultant string. The parameters immediately after the category symbol will carry computational information relating to transformational history and the overall structure of the text. The next parameters will contain symbols and strings which distinguish specific operations out of the general ones. The last parameters will contain ad-juncts as discussed above and morphological processes. The dividing line between each kind of parameter is not sharp, nor perhaps should it be.

Let us enter briefly into the topics of governance and transformational history.

Governance is that relationship between strings of a text in which one string determines which of the alternative forms of another string is selected. One of the primary examples is the form of the verb being governed by the subject. It is not sufficient to represent the verb *laughs* merely as

\[(71) \quad V[laughs].\]

Rather we must add a parameter $N$ as follows

\[(72) \quad V[N][laughs].\]

Actually, the form of *laugh* is determined by two parameters. Tense must be added to the above.

\[(73) \quad V[N][pres][laughs].\]

The relationship of direct to indirect discourse in English illustrates well the transformational significance of the nominal parameter of (73).
John says, "I laugh when I make a mistake because I am embarrassed"  

John says that he laughs when he makes a mistake because he is embarrassed

There are two series of changes from (74) to (75). The first is the replacement of

\[ N[1prn][John] = I \]

everywhere by

\[ N[3prn][John] = he \]

where 1prn and 3prn are the first person and third person parameters respectively. The second series of changes is in the verbs: laugh becomes laughs, make becomes makes, and am becomes is. These changes are all governed by the replacement of I by he. In fact, we can take care of both series of changes by one substitution operation, if we express the governance relation explicitly as in (73). There are also other changes that are generally part of the relation between direct and indirect discourse which we are ignoring in the present discussion; we characterized the tense change briefly in chapter four (22)-(25).

In a fairly rigorous representation, (74) and (75) become

\[
\]

\[
\]

The general form of both the above is the same.

\[
S[N, V[V, N[S[D][D][N, V]]]]
\]

We write the transformation relating them as

\[
S_1[N_2, V[V_3, N[quote][S_4(N[1prn][N_2])]]] \rightarrow S_1[N_2, V[V_3, N[that][S_4(N[3prn][N_2])]]]
\]
Of course, I should be more precise and sub-classify V₃, but that is not the main point. I wish to emphasize that the explicit representation of governance by means of parameters greatly systematizes the transformational relationship.

To anticipate a point to be discussed in Chapter seven, let me note that transformations generally introduce ambiguities. For example, the introduction of a personal pronoun or a relative pronoun will not necessarily result in a syntactically unambiguous antecedent for the pronoun. In such cases we write the transformation in one direction only, since the pronominalization may be syntactically determinate, but the inverse likely is not.

The functional notation is convenient for keeping traces of previous operations, that is, recording transformational history. For example, the representations

(82) N[3prn] [N₁]
(83) N[wh] [N₁]

retain the source noun N₁. Such representations allow us to have formal inverses at least. They also allow us to keep record of the precise order in which transformations were performed, when that is necessary for the inverses. For example, the category symbol A could be accompanied by a parameter of a string of strokes indicating how newly its relative clause was introduced in relation to other relative clauses in the text. One stroke would indicate it was the last relative clause introduced, two strokes the second to last, and so forth. The substitution could be used to increase the number of strokes uniformly with each relative clause transformation.

It has not been my purpose in this section to treat the topics of grammatical categories, joining, governance, and transformational history in detail. I merely wished to indicate the value of the functional notation, in particular the parametric forms, in expressing transformations.

3. FUNCTIONS AND PHONOLOGY: THE RELATIVE CLAUSE

In the preceding section the functional notation was discussed only as a vehicle for performing transformations. That use of the notation hardly warrants the name functional. It is the second role of the notation which is truly functional. In this section we examine the notation as representing functions from phonological representations to phonological representations.

The abstract functional representation must yield a phonological representation which is quite different in length, occurrences of strings, identity
PARAPHRASE GRAMMARS

of forms and order of strings. The phonological evaluation of the functions corresponds approximately to what Chomsky calls obligatory transformations.

The various functions have quite different scopes and different numbers of arguments. However, I assume that the domains are all from the same phonological notation and I will adjust that notation accordingly. Some functions will juxtapose strings in a specific order and add intonation and stress patterns. Others will modify strings according to what are traditionally called morphological patterns.

The functions with the largest scope and number of arguments are the sentential and adjectival functions. We will ignore the topic of intonation patterns, for it complicates the description considerably and I cannot offer any new insights. But it seems to me that they are profitably treated as part of the sentential and adjectival functions. These functions are very simple to define if we leave aside relative and interrogative pronouns. As I have tacitly been doing for some time, I will underline strings whose precise form is not relevant to the discussion. The basic pattern of a sentence is the junction of a noun and a verb. The phonological evaluation of this function is

\[
(84) \quad S[N, V] = NV.
\]

That is, given phonological strings \(N\) and \(V\), the phonological value of \(S[N, V]\) is \(NV\), for example,

\[
(85) \quad S[\text{the captain, found the father of the boy}] = \text{the captain found the father of the boy}
\]

The basic adjectival junction is the same.

\[
(86) \quad A[N, V] = NV
\]

When these functions have adverbial parameters, they yield the following results.

\[
(87) \quad S[D][N, V] = NVD
\]

\[
(88) \quad A[D][N, V] = NVD
\]

for example,

\[
(89) \quad S[\text{yesterday}][\text{the captain, found the father of the boy}] = \text{the captain found the father of the boy yesterday}.
\]

The alternative adverbial positions will be treated here after subtler techniques are introduced.

We treat the complex verb phrase also as a case of junction.

\[
(90) \quad V[V, N] = VN
\]
for example,

(91) \[ V[found, the father of the boy] = found the father of the boy. \]

Adverbial parameters also accompany the verb phrase.

(92) \[ V[D] [V, N] = VND \]

for example,

(93) \[ V[quickly] [found, the father of the boy] = found the father of the boy quickly. \]

Let us look briefly at complex noun phrases. We find two basic types: con-junction

(94) \[ N[C] [N, N] = NCN \]

as in

(95) \[ N[and] [the captain, the mate] = the captain and the mate \]

and ad-junction

(96) \[ N[A] [N] = AN \]

for example,

(97) \[ N[heroic] [captain] = heroic captain \]

or

(98) \[ N[A] [N] = NA \]

for example,

(99) \[ N[of the boy] [father] = father of the boy. \]

(100) \[ N[T] [N] = TN \]

for example,

(101) \[ N[the] [father of the boy] = the father of the boy. \]

The adjective in the form of a prepositional phrase appears as follows

(102) \[ A[P] [N] = PN \]

as in

(103) \[ A[of] [the boy] = of the boy. \]

The adverbial prepositional phrase is similar.

(104) \[ D[P] [N] = PN \]
I have just given a very superficial description of some possible functions of English. Let me deal with one topic in somewhat more depth: the phonological realization of the relative clause. Many of the above functions will be described more carefully in the process.

It was perfectly satisfactory to define the two argument A-function according to (86) above for examples like

\[(105)\] \(A[\text{the captain, found the father of the boy}] = \text{the captain found the father of the boy}\)

or even for

\[(106)\] \(A[\text{who, found the father of the boy}] = \text{who found the father of the boy}\).

The difficulties come when the relative pronoun occurs other than as the first element of the resultant string. For example,

\[(107)\] \(A[\text{the captain, found whom}] = \text{the captain found whom}\)

yields incorrect English.

The two argument A-function receives a rather large burden in the correct formation of the English relative clause. It will be seen to be composed of several steps, each the result of the preceding. We will also see our first examples of “syntacto-phones”, that is, phonological elements which assist in the proper definition of functions, but which may not correspond to traditional phonological elements. There are three main features of the relative clause that A must account for: the shift of a phrase to the front of the clause, the identification of the relevant relative pronouns, and the replacement of certain of these by personal pronouns.

To describe the shifting of phrases phonologically we must indicate the phrase boundaries with some phonological symbol. Let us use \# . The basic sentence divides into phrases according to

\[(108)\] \(# N \# V \#\)

and

\[(109)\] \(# N \# V \# N \#\).

Let us define a major segment as a phonological string bounded by \#' s and uninterrupted by a \#. Then the rule for shifting is at first approximation

\[(110)\] \(\text{the major segment with the relative pronoun is moved to the far left.}\)

Thus, if we write (107) according to (109) we have

\[(111)\] \(# \text{the captain} \# \text{found} \# \text{whom} \#\)
which by (110) becomes

(112) whom # the captain # found # #.

In general a sentence has more than three major segments. One additional source of major segments is adverbial phrases modifying sentences or verbs.

(113) # N # V # D #
(114) # D # N # V #
(115) # N # D # V #

An example is

(116) # yesterday # the captain # found # whom #

which by (110) becomes

(117) whom # yesterday # the captain # found # #.

Since there is some option in the placement of adverbial phrases, a notational provision must be made for it. Thus, we distinguish left and right adverbs by a parameter.

(118) D[←] [X] = #X
(119) D[→] [X] = X#

respectively.

Now we can reformulate (87) and (88) stepwise in the following pattern, which incorporates (110).

(120) A[#X] [N, V] = (step 1: group N and V by ‘#’s) #N # V #
    (step 2: add the adverbial parameter which is closest to the arguments, according to the location of ‘#’) # X # N # V #
    (repeat step 2 for each adverbial parameter)
    (step 3: shift the major segment with the relative pronoun, if any, to the front)
    (step 4: erase all ‘#’s)

We reformulate (92) as follows.

(121) V[X#] [V, N] = (step 1: group V and N by ‘#’s) #V # N #
    (step 2: add closest adverbial parameter, according to the location of ‘#’) # V # N # X #
    (repeat step 2 for each adverbial parameter)

Let us take example (116) as an illustration.

(122) V[found, whom] = (step 1) # found # whom
110 Paraphrase Grammars

(123) \[ D[-] [\textit{yesterday}] = \# \textit{yesterday} \]

(124) \[ A[\# \textit{yesterday}] [\textit{the captain}, \# \textit{found} \# \textit{whom} \#] = \]
\[ = \text{(step 1) } \# \textit{the captain} \# \# \textit{found} \# \textit{whom} \# \# \]
\[ \text{(step 2) } \# \textit{yesterday} \# \textit{the captain} \# \# \textit{found} \# \textit{whom} \# \# \]
\[ \text{(step 3) } \textit{whom} \# \textit{yesterday} \# \textit{the captain} \# \# \textit{found} \# \# \# \]
\[ \text{(step 4) } \textit{whom yesterday} \textit{the captain found} \]

We observe that there is the option of breaking the major segments. The following step 2 form

(125) \# \textit{the captain} \# \# \textit{found} \# \textit{the father of whom} \# \#

has three possible step 3 forms

(126) \textit{the father of whom} \# \textit{the captain} \# \# \textit{found} \# \# \#

(127) \textit{of whom} \# \textit{the captain} \# \# \textit{found} \# \textit{the father} \#

(128) \textit{whom} \# \textit{the captain} \# \# \textit{found} \# \textit{the father of} \# \#.

But only the first is allowed by above procedures. So we introduce a "breaking" function transformationally.

(129) \[ S_1(N_2) \rightarrow S_1(N[\#][N_2]) \]

where

(130) \[ N[\#][N_2] = \#N_2. \]

(131) \[ S_1(P_2) \rightarrow S_1(P[\#][P_2]) \]

where

(132) \[ P[\#][P_2] = \#P_2. \]

There are important admission conditions for (129) and (131). I only mention one – that the string receiving the \# be in the V or else a right-hand D.

Now we account for (127) as follows.

(133) \[ P[\#][of] = \#of \]

(134) \[ A[#of][whom] = \#of\ whom \]

(135) \[ N[#of\ whom][the\ father] = the\ father\ #of\ whom \]

(136) \[ V[found,\ the\ father\ #of\ whom] = \#found\ #\ the\ father\ #of\ whom\ # \]

(137) \[ A[the\ captain,\ #found\ #the\ father\ #of\ whom\ #] = \]
\[ = \text{(step 1) } \# \textit{the captain} \# \# \textit{found} \# \textit{the father} \# \# \textit{of} \# \# \]
\[ \text{(step 3) } \textit{of whom} \# \textit{the captain} \# \# \textit{found} \# \textit{the father} \# \# \# \]
\[ \text{(step 4) } \textit{of whom the captain found the father} \]
We can account for (128) similarly.

We have glossed over one very crucial part in the above discussion. In (110) and (120) we referred to “the” relative pronoun. In fact, it is rather complicated to specify “the” relative pronoun, but once it is done we have finished the definition of $A[N, V]$.

The problem of determining “the” relative pronoun falls into two parts. The first is to distinguish relative pronouns referring to the head of the ultimate nominal construction from those referring to other nouns. For example, consider step 2 form

(138) # the captain who lost his ship # # found # the father of whom # #

There are occurrences of who in two different major segments. How is a phonological function supposed to distinguish the one which is free for the ad-junction from the other, the bound one? This distinction is accomplished transformationally. The transformational source for the relative clause will have clear interreference patterns, which we can keep track of through the functional notation. Let us for the moment simply suppose we have a source for this distinction and add another “syntacto-phone” to the proper who.

(139) # the captain who lost his ship # # found # the father of *whom # #

Now the instructions refer to “the” starred relative pronoun. The second part of the problem will provide us with the source of the *.

There may be many occurrences of the noun to be relativized, but only certain of those will become the starred pronouns. The primary source of the distinction between the starred and the unstarred pronouns is due to the concept of level which we dealt with in Chapter five. The basic principle can be stated as follows

(140) relative to the input sentence, level 0 occurrences of the relativized noun are starred and all higher level occurrences are unstarred.

The question is how to implement this principle. The most direct solution is to split the relative pronoun substitution of relative clause transformations into two parts. Thus, we can rewrite (70) as

(141) \[ S_1[D[because][S_3(S)(N_2 \mid N_2)]][N_2^0 \rightarrow S_1[N[A_3(A)(N[^*wh][N_2])][N_2]]] [N_2_] \]

\[ N_2^0/min S^0 \]

\[ N_2^{relS_3} \]

\[ N_2^{\geq 0relS_3} \]

The second admission condition reads that the substitution for $N_2$ applies to 0 level occurrences relative to $S_2$ and the third condition states that that
substitution for \( N_2 \) applies to greater-than-level 0 occurrences of \( N_2 \). We could write condition one as \( N_2^{relS_3} \) per \( minS_1^{relS_3} \), but we agreed earlier to understand the level as relative to the string receiving the condition, unless otherwise noted.

It remains to define \( N[wh] \) and \( N[*wh] \).

(142) \[ N[wh] [N_1] = prnN_1, \] that is, the proper personal pronoun for \( N_1 \)

(143) \[ N[*wh] [N_1] = *prnN_1 \]

for example,

(144) \[ N[wh] [the boy] = he \]
\[ N[*wh] [the boy] = *he \]

(145) \[ N[wh] [the girls] = they \]
\[ N[*wh] [the girls] = *they \]

(146) \[ N[wh] [the proposal] = it \]
\[ N[*wh] [the proposal] = *it \]

Let us return to the complex relative clause example to illustrate the progress we have made. I refer to section two of chapter five for the arguments regarding the levels of the occurrences of the proposal. The sentence to be relativized is

(147) the proposal certainly was not spontaneous, but rather the President had instructed two months ago he should put forth the proposal after they had exhausted their own proposals and he was certain they would accept the proposal.

We are dealing with two \( S[N, V] \) of level 0. That is, (147) is a con-junction of two \( S[N, V] \) using but rather. Each has the required level 0 occurrence of the proposal. The second con-junct also has a level 1 occurrence of the proposal, which is the last phrase of (147). Thus, the relative clause transformation (141) yields

(148) *it certainly was not spontaneous, but rather the President had instructed two months ago he should put forth *it after they had exhausted their own proposals and he was certain they would accept it

Now we have distinguished the pronouns of the proper level, level 0 with respect to the sentence being relativized. However, there may still be more than one starred pronoun by this procedure in each \( A[N, V] \). The A-function must select from among them. The following data provides a basis for the selection. The judgements on which the data is based are sometimes
It appears that there may be more than one possible rule and that as native speakers we do not encounter situations which distinguish them often enough to prove one wrong and the other right. I give one set of judgements. The first element in each example is the input sentence, in which the two occurrences of the boy can be understood as referring to the same boy. The second element is the result of the transformation, showing the starred pronouns. The third picks the first starred pronoun for the relative form in determining the value of A, while the fourth chooses the second. The fifth element allows both starred pronouns to become relative pronouns. The ‘−’ sign indicates that the string is acceptable, but only if the two pronouns have different referents. The ‘?’ indicates unacceptability. ‘?’ indicates indecision. I have partially systematized the writing.

(149) the boy's father liked the boy
 *hi's father liked *hi
 ab
 who's father liked hi
 - who-m hi's father liked
  who-m who's father liked
(150) the boy was liked by the boy's father
 *hi was liked by *hi's father
 ab
 who was liked by hi's father
 - by who-s father hi was liked
  by who-s father who was liked
(151) we found the boy-m at the boy-s home
 we found *hi-m at *hi-s home
 ab
 who-m we found at hi's home
 ? at who-s home we found hi
  who-m at who-s home we found
(152) at the boy-s home we found the boy-m
 at *hi-s home we found *hi-m
 a
 ? at who-s home we found hi-m
 b
 who-m at hi-s home we found
  who-m at who-s home we found
(153) near the boy-m we found the boy's father
 near *hi-m we found *hi-s father
 a
 near who-m we found hi-s father
 b
 who-s father near hi-m we found
  who-s father near who-m we found
I simply present two alternative rules, that is, two alternative definitions of $A[N, V]$. In the above the symbol 'a' marks the result of the first definition and 'b' the result of the second. Definition a is

$$A[D_1, \ldots, D_n] \cdot [D_1, \ldots, D_n] [N, V] =$$
- step 1: group N and V by #’s
- step 2: adjoin $D_i$ to the result of preceding steps according to the location of # in $D_i$; perform step 2 for each $i = 1, 2, \ldots, n$ in succession
- step 3: shift the first major segment with the starred pronouns, if any, to the front of the string replacing each *prn in it by *wh
- step 4: erase all *’s and #’s

Definition b is

$$A[D_1, \ldots, D_n] \cdot [D_1, \ldots, D_n] [N, V] =$$
- step 1: group N and V by #’s and shift the first major segment with starred pronouns, if any, to the front of the string replacing each *prn by *wh
- step 2: adjoin $D_i$ to the result of the preceding steps according to the location of # in it
- step 3: shift the major segment with *wh, if any, to the front of the string; otherwise, shift the first major segment with *prn, if any, to the front of the string, replacing each *prn in it by *wh
- repeat steps 2 and 3 for each $i = 1, 2, \ldots, n$ in succession
- step 4: erase all *’s and #’s

The generalized replacement of *prn by *wh should properly be part of the relative clause transformation and not the phonological operations of the functions. However, this requires a complex system of ranking positions for pronominalization to be built into the functional representations and
FUNCTIONAL REPRESENTATION

to be manipulated by transformations. It is unduly complex for a general presentation of a few central ideas; so I have accommodated entry five of (155) by including the generalized replacement in the definition of A[N, V].

I note that step 4 in the above definitions erases any confusion of bound and free pronouns, such as in (138).

We conclude this section with our frequent example, most recently occurring as (148). Either (156) or (157) will give the same result: however, I will use the steps of (156).

(158) \[A[^{it, \#\text{certainly\# was not spontaneous\#}] = \]

= (step 1) \[^{it\#\#\text{certainly\# was not spontaneous\#}]\]

(step 2) no change

(step 3) \[^{which\#\#\text{certainly\# was not spontaneous\#}]\]

(step 4) \[^{which\#\text{certainly was not spontaneous}]\]

(159) \[A[^{\#he\text{was certain}\} \{they, \#\text{would accept\# it\#}] = \]

= (step 1) \[^{they\#\#\text{would accept\# it\#}]\]

(step 2) \[^{he\#\text{was certain}\# they\#\#\text{would accept\# it\#}]\]

(step 3) no change

(step 4) \[^{he\text{was certain they would accept it}]\]

(160) \[A[^{they, \#\text{had exhausted\# their own proposals\#}] = \]

= (step 1) \[^{they\#\#\text{had exhausted\# their own proposals\#}]\]

(step 2) no change

(step 3) no change

(step 4) \[^{they\text{had exhausted their own proposals}]\]

(161) \[A[^{\text{and}] [160, 159] = \]

= \[^{they\text{had exhausted their own proposals and he was certain they would accept it}]\]

(162) \[D[^{\rightarrow} [after] [161] = \]

= \[^{after\text{they had exhausted their own proposals and he was certain they would accept it}]\]

(163) \[D[^{←} [the President had instructed two months ago] = \]

= \[^{the President had instructed two months ago}]\]

(164) \[A[^{163} [162] [he, \#\text{should put forth\# *it\#}] = \]

= (step 1) \[^{he\#\#\text{should put forth\# *it\#}]\]

(step 2) \[^{he\#\#\text{should put forth\# *it\# after they had exhausted their own proposals and he was certain they would accept it}]\]

(repeat for next adjunct)

\[^{\#the President had instructed two months ago \#he\#}]\]
should put forth # *it # # after they had exhausted their own proposals and he was certain they would accept it # (step 3) *which # the President had instructed two months ago # he # # should put forth # # after they had exhausted their own proposals and he was certain they would accept it # (step 4) which the President had instructed two months ago he should put forth after they had exhausted their own proposals and he was certain they would accept it

(165) A [but rather] [158, 164] =

= which certainly was not spontaneous but rather which the President had instructed two months ago he should put forth after they had exhausted their own proposals and he was certain they would accept it

To summarize the above discussion of the formation of the relative clause in English, we can say that with respect to the input sentence, each minimal level 0 sentence, that is, each conjunct, has exactly one major segment which is shifted to the front and whose pronouns referring to the head of the ultimate nominal construction take the relative form. All other pronouns referring to the head take the personal form. The choice of the major segment depends on a ranking of major segments.

4. SUMMARY

In this chapter I have argued for a functional representation of the syntax of a language. The functional representation plays a dual role. It provides the information necessary for transformations to operate on strings of the language. In this role it need not be strictly functional. Study of the relation of transformations to context may even provide a semantic argument for the functional notation. I tried to show how a functional notation was convenient for putting in relief the various joining relationships, which in turn facilitate the description of transformations.

The notation is truly functional in its use to map phonological representations to other phonological representations. The phonological notation above was highly provisional and meant only to illustrate the type of “phones” that might be used, in particular, “syntacto-phones”, which go even beyond the morphophoneme, having only the most indirect correlation with the stream of sounds in an utterance or letters in a written text. I invite the reader to be critical of the concept of “syntacto-phone”. The details of defining a function were illustrated on the adjectival relative clause function.
CHAPTER VII
THE STRUCTURE OF PARAPHRASE GRAMMARS

In the preceding chapters we have discussed various tools that could be used in constructing a grammar of the paraphrase relation. In the present chapter we will investigate the overall "shape" of such a grammar. The discussion will revolve about two central concepts: the recursive character and the algebraic character of the grammar.

A rough description of a "recursive enumeration" is a production system, that is, a system that starts with some clearly chosen objects and augments them using rules of production, whose inputs and outputs are clearly and, we might say, mechanically determined. Another locution used is that a finite set of rules "generates" from the chosen objects the remaining objects. We will give a precise definition in the appendix, but for now let us simply try to maintain a general conception. An initial set $E$ of objects, which are listed or "mechanically" chosen, are the material which a finite set $R$ of "mechanical" rules uses to produce a total set of objects.

We recall an earlier example of the set $T$ of strings of alternating $a$ and $b$: $ab$, $ba$, $aba$, $bab$, .... The set $E$, in this case, would be $\{ab, ba\}$. The rules $R$ are two: add an $a$ on the end of any string in $E$ or derived from $E$ which ends in $b$, and add a $b$ on the end of any string in or derived from $E$ which ends in $a$. Thus, applying $R$ to $E$ we successively augment $E$ as $\{ab, ba, aba, bab, abab, baba, ...\}$. By this never ending, though fully determined, process we characterize $T$. Any object or string not obtained from $E$ via successive applications of $R$ is considered as not part of the system. The first section of this chapter characterizes a paraphrase grammar in terms of a production system.

The discussion of the algebraic character of the grammar involves considerations of what is necessary for establishing a set of elementary transformations and inverses of transformations, out of which all transformations are composed. It also includes the study of the nature of the mapping from functional representations to phonological representations.

1. THE RECURSIVE ENUMERABILITY OF THE TRANSFORMATION RELATION

In Chapter III I presented arguments against using compositional grammars
for natural languages. We can place the idea of a compositional grammar under the general concept of a production system. A compositional grammar claims to be a production system that produces exactly those strings of a language which are "good", to use the terminology of Chapter III. Restated more technically, a compositional grammar claims to be a recursive enumeration of precisely the good strings of the language. Thus, I am not persuaded that the task of a grammar is to provide a recursive enumeration of the set of good strings of a language. However, I do see a grammar as giving a recursive enumeration of the generalizable paraphrastic relationships. That is, we can take $T$, in the foregoing, to be the set of transformations.

The general concept of a relation is a set of order pairs, or more generally a set of ordered, n-tuples. For example, the relation of equality between natural numbers is characterized as the set of ordered pairs: $\langle 0, 0 \rangle, \langle 1, 1 \rangle, \langle 2, 2 \rangle, \ldots$. The natural numbers: $0, 1, 2, \ldots$ constitute a recursively enumerable set. The system for producing them is very simple: starting with 0 we need only add the unit successively to obtain any natural number. Similarly the relation of equality between all the natural numbers is also easily produced, starting with $\langle 0, 0 \rangle$ and adding the unit to both places successively. Let us suppose we have a very irregular set $A$ of natural numbers, one that is not recursively enumerable. It is important to observe that the relation of equality between numbers in $A$ is given by the relation of equality on the set of all natural numbers. We call it "equality restricted to $A$". All members of $A$ are put in the proper relation. That is, if $a$ and $b$ are two different numbers in $A$, then $a$ is set equal to $a$ and $b$ to $b$, but not $a$ to $b$ or $b$ to $a$. This is a trivial fact but it contains a very important concept. It does not matter that numbers outside of $A$ also participate in the equality relation.

The transformational relation as I seek to characterize it on the set of texts, say of English, is analogous to the relation of equality on set $A$ above. Thus, I want to characterize the fact that two strings are transformationally related as an ordered pair, or possibly an ordered n-tuple, which alternative depends on the nature of the language in question. We recall that the transformational relation derives ultimately from the relation of paraphrase. It consists of those paraphrases which are universalizable through the many devices discussed in the preceding chapters. We wish, therefore, to characterize the transformation relation on the set of good strings of a language, keeping in mind that the set of good strings is not assumed to be recursively enumerable.

Let us assume for the neatness of the argument that the transformational relation is a set of ordered pairs of strings of a language, rather than the general case of an ordered n-tuple. We will illustrate the considerations that
decide the value of $n$ in the phrase ‘$n$-tuple’ later. We note that we have tacitly made this assumption in earlier chapters when we wrote sample transformations with one string $s_1$ on the left and one string $s_2$ on the right. The small $s$ is for strings.

(1) $s_1 \rightarrow s_2$

In the present framework we simply write (1) as

(2) $\langle s_1, s_2 \rangle$

which is understood to maintain the same ordered nature of left-right or input-output as the arrow notation.

Our goal now is to give a production system, that is, recursive enumeration, of the set $P$ of ordered pairs of strings of a language which characterizes the transformation relation

There are some very general properties which $P$ must satisfy to be a paraphrase grammar. The most basic is that if $\langle s_i, s_j \rangle$ is in $P$, then $s_i$ and $s_j$ are paraphrases of each other. However, we must be cautious, because we claim no production system for the set of good strings $G$. The string $s_i$ is simply a member of a general set of strings, good and bad, which we label $S$. The set of bad strings, that is, the non-good strings we call $B$. What we must be cautious about is that if $s_i$ and $s_j$ are bad strings then for $\langle s_i, s_j \rangle$ in $P$ the judgement as to whether they are paraphrases is meaningless. In other words, $P$ is a much larger set than the set of the pairs of universalizable paraphrases of the language.

The set $S$ of strings of the language includes both $G$ and $B$. We do assume that there is a production system for $S$. A very rough set of rewrite rules or a word-class sequence description could easily produce a set of strings which was larger than $G$. The difficulty arises, as we discussed in Chapter three, when we try to produce exactly $G$ and no more. Then we have to account for the co-occurrence relation.

What then are the general properties which $P$ must satisfy to be a paraphrase grammar? The nature of its elements is a fruitful starting point. The possible combinations of good and bad elements are fourfold.

(4) $\langle$good, good$\rangle$

$\langle$bad, bad$\rangle$

$\langle$bad, good$\rangle$

$\langle$good, bad$\rangle$

What combinations of good and bad do we wish to allow in $P$?
To transform good to good is precisely what we are after. For example,

(5) \(<I \text{ liked the boy because the boy smiled}, I \text{ liked the boy who smiled}>\)

is typical of the kind of ordered pair we desire to be in \(P\) for English. It is the good-to-good pairs that we require to be paraphrastic.

Since we cannot "mechanically" separate \(G\) from \(B\), we must be content that the rule that characterizes (5) also covers

(6) \(<I \text{ greeted the chemical, because the chemical was incognito,}
I \text{ greeted the chemical, which was incognito}>\)

Thus, (6) is also likely to be in \(P\) for English.

\(P\) will also contain pairs of the type \(<\text{bad, good}>\); that is, transformations will change bad strings into good strings. The linguistic reason for this is, we recall from chapter two, that a good string may be syntactically ambiguous, but semantically unambiguous. That is, one of its formally possible sources is not semantically possible, is not good. The example

(7) \(\text{the man who was seen by the door left silently}\)

has two syntactically possible sources:

(8) \(\text{the man whom someone saw by the door left silently}\)

and

(9) \(\text{the man whom the door saw left silently.}\)

However, semantically only (8) is good. Thus, since we desire a rule which relates two good strings as in

(10) \(<\text{the man whom the girl saw left silently,}
\text{the man who was seen by the girl left silently}>\)

we thereby also include bad-to-good pairs such as

(11) \(<\text{the man whom the door saw left silently,}
\text{the man who was seen by the door left silently}>\).

That a bad string can be transformed into a good string is a point relevant to inverses, but the relation can hold between such strings without interfering with the relation of good to good strings.

The last pair type causes the greatest concern. If we transform a good string into a bad string, we feel that the rule was not valid. A valid transformational rule should lead from good strings to only good strings and not to bad strings. Thus, we wish to exclude from \(P\) pairs of the type \(<\text{good, bad}>\).
The two requirements: that the good-good pairs be paraphrastic and that there be no good-bad pairs are closely related. The concrete linguistics behind the present abstract characterization involves rules which operate on large numbers of strings to produce transforms of them. We wish the rules to lead from good strings to paraphrases of them, which are also good strings. We have already admitted that we cannot prevent bad strings from being mapped to good strings, since we have no procedure for distinguishing bad strings from good ones. If in addition our rules map good strings to bad strings, we cannot avoid the possibility of having good-good pairs which are not paraphrastic, since the derivation could follow the pattern: good → bad → good, thus, going outside the set of good strings to complete the relationship. Consider the following example of a pair of possible transformational relationships.

\[
\begin{align*}
\text{(12)} & \quad \text{i. we explored} & \text{a} & \quad \text{ii. we explored} & \text{b} & \quad \text{iii. we explored} \\
& \text{the mill} & \leftrightarrow & \text{the mill} & \leftrightarrow & \text{the mill} \\
& \text{by} & \leftrightarrow & \text{by using} & \leftrightarrow & \text{using} \\
& \text{the little stream} & \leftrightarrow & \text{the little stream} & \leftrightarrow & \text{the little stream} \\
\end{align*}
\]

Rule \(a\) takes string i to string ii and rule \(b\) takes string ii to string iii, and we thereby derive that i and iii are paraphrases by the combined rule \(a \cdot b\). However, rule \(a\) leads in the left-to-right direction from good to bad strings: \(\langle\text{good, bad}\rangle\). For example,

\[
\begin{align*}
\text{(13)} & \quad \text{i. the bacteria} & \quad \text{ii. the bacteria} & \quad \text{iii. the bacteria} \\
& \text{infected the boy} & \leftrightarrow & \text{infected the boy} & \leftrightarrow & \text{infected the boy} \\
& \text{by} & \leftrightarrow & \text{by using} & \leftrightarrow & \text{using} \\
& \text{the microscope} & \leftrightarrow & \text{the microscope} & \leftrightarrow & \text{the microscope} \\
\end{align*}
\]

String \(ii\) is at best a metaphor. Let us pronounce it bad. In (13) rule \(a\) relates a good string to a bad string: \(\langle\text{good, bad}\rangle\) and \(b\) relates a bad string to a good string: \(\langle\text{bad, good}\rangle\). We agreed to allow the latter situation. But if we also allow \(\langle\text{good, bad}\rangle\), as in the above, we obtain a \(\langle\text{good, good}\rangle\) pair, such as \(\langle i, iii \rangle\) of (13) which is not paraphrastic. The reason for the lack of paraphrase is that the reading they might have shared, in analogy to (12), is semantically unacceptable, as indicated by string \(ii\) in (13) being bad. The solution we shall take is to write the rules in one direction only. Thus, if we write

\[
\begin{align*}
\text{(14)} & \quad \text{i. the bacteria} & \quad \text{ii. the bacteria} & \quad \text{iii. the bacteria} \\
& \text{infected the boy} & \leftrightarrow & \text{infected the boy} & \leftrightarrow & \text{infected the boy} \\
& \text{by} & \leftrightarrow & \text{by using} & \leftrightarrow & \text{using} \\
& \text{the microscope} & \leftrightarrow & \text{the microscope} & \leftrightarrow & \text{the microscope} \\
\end{align*}
\]

we have reduced the relationships to two \(\langle\text{bad, good}\rangle\) pairs: \(\langle ii, i \rangle\) and
The result is that \(i\) and \(iii\) have a bad source in common, which is really no source, and no compound transformation is obtained, since \(a\) and \(b\) both lead away from \(ii\).

We see by this discussion that the ordered pairs of \(P\) really have a directional character. It matters which element is first and which second. The general rule for writing transformations is that a transformation may introduce ambiguities but should never resolve any ambiguity, such as happened in \(\langle i, ii \rangle\) in (13).

Furthermore we have shown how the requirement that all \(\langle \text{good}, \text{good} \rangle\) pairs be paraphrastic entails in a transformational or rule grammar that either there be no \(\langle \text{bad}, \text{good} \rangle\) pairs or no \(\langle \text{good}, \text{bad} \rangle\) pairs. By the above conventions of representing transformations we decide to exclude \(\langle \text{good}, \text{bad} \rangle\) pairs, since a transformation which leads from a good string to a bad string is not properly a transformation on the set of good strings.

To recapitulate briefly, we see \(P\) as a recursively enumerable set, in other words, a production system, of ordered pairs of strings from \(S\). \(P\) only has pairs of the types: \(\langle \text{good}, \text{good} \rangle\), \(\langle \text{bad}, \text{bad} \rangle\) and \(\langle \text{bad}, \text{good} \rangle\). The \(\langle \text{good}, \text{good} \rangle\) pairs are all paraphrastic. Now that we have a very general set of requirements for \(P\) to satisfy, the next question is how to construct such a \(P\) for a language.

The overall procedure is to reduce the production of \(P\) to the production of a number of related concepts. The three global ones are transformation, admission condition, and elementary string. The production systems for each of these three are further analyzed into more detailed production systems for concepts of grammatical category, junction, equivalence, substitution and others. But the details are not of much assistance in obtaining an overall conception of a paraphrase grammar, so we will ignore them here, though in the appendix we will show the formal techniques for constructing the various production systems and for combining them into further production systems.

A particular transformational relation is basically a formal relationship. That is, the relationship of one element in a transformational pair to the other element can be characterized largely as a rearranging of segments, deletions, addition of constants, and substitution. In some of the preceding chapters I stressed the operation called substitution: one string \(x\) is replaced in each occurrence by another string \(y\). It is not very difficult to conceive of a production system for the substitution relationship. We start with one occurrence of the string \(x\) all by itself paired with one occurrence of the other string \(y\). Then we simultaneously and systematically add to the context and number of occurrences in a parallel fashion in each element of the pair.
The precise system for such a production must wait for the appendix. The point to be gained is that there is a production system for each formal transformation. In fact, one might even define 'transformation' or 'rule' as a production system of ordered pairs.

\[(15) \quad \langle x, y \rangle, \langle ax, ay \rangle, \langle axx, ayy \rangle, \langle axb, aby \rangle, \langle axbx, ayby \rangle, \ldots \]

The idea is that \(A_i\) simply gives the strings, good and bad, that satisfy a certain condition. Only the strings satisfying the condition appended to a transformation are allowed to enter the transformation. That is, only the strings of \(A_i\) are allowed to enter a transformation which has admission condition \(A_i\).

An admission condition is merely a set of strings which has a production system.

\[(16) \quad ft_1 = \langle s_1, s_2 \rangle, \langle s_3, s_4 \rangle, \langle s_5, s_6 \rangle, \ldots \]

We recall that besides the formal manipulations changing \(s_j\) to \(s_k\) in \(\langle s_j, s_k \rangle\) there are admission conditions on \(s_j\).

The production system of an admission condition and a formal transformation are combined to give the conditioned transformation. The technique is from the production of the rationals in mathematics. The production of the formal transformation is carried out horizontally, and that of the admission condition vertically.

Then by following the winding line every element of the admission condition is compared with every ordered pair in the transformation to see if it is identical with the first member of the pair. If it is identical, then that ordered pair is produced. If it is not identical, then it is not produced. In this fashion we obtain a sequence of produced ordered pairs. That is, we have a production system for the conditioned transformation. Call it simply \(t_i\).

There are an infinite number of such conditioned transformations. There is also a production system for the set of these transformations, which consists in finding a finite number of elementary transformations and simply producing all possible combinations of them. We will deal with this topic in the next section. Let us simply assume for now that there is a production
system for all the transformations and let us list them horizontally. Since each transformation is itself characterized as a production system of ordered pairs, let these systems unfold in the vertical direction.

We then take each pair in the sequence given by the diagonal method and we have a production system for all transformationally related pairs of strings, call this set $T$.

$T$ is not yet $P$. $T$ is merely a production system whose elements are pairs of strings whose relationship can be characterized in a "mechanical" way. $P$ has the additional requirements that the $\langle$good, good$\rangle$ pairs are all paraphrastic and that there be no $\langle$good, bad$\rangle$ pairs.

The step from $T$ to $P$ is obtained via the concept of "elementary string": $E$. We recall that the source of the $\langle$good, bad$\rangle$ pairs was syntactic am-
biguity. For example, purely formally

(20) \textit{the bacteria infected the boy by the microscope}

has both

(21) \textit{the bacteria infected the boy who is by the microscope}

and

(22) \textit{the bacteria infected the boy by using the microscope}

as transformationally related to it. A pair such as \langle 20, 22 \rangle should be eliminated from \( T \) since it is \langle \text{good, bad} \rangle. We concluded earlier that the paraphrase relation had a directional character. Thus, transformations should be written so as not to resolve any syntactic ambiguities, since syntactic ambiguities are not always accompanied by semantic ambiguities, as (20)--(22) illustrate. But perhaps a given transformation both resolves some ambiguities and adds others, as for example is the relation in

(23) \begin{align*}
& \textit{we explored} \quad \textit{we explored} \\
& \langle \textit{the mill} \quad \textit{the mill} \rangle \\
& \langle \textit{by} \quad \textit{using} \rangle \\
& \textit{the little stream} \quad \textit{the little stream}
\end{align*}

In either direction one ambiguity is resolved. From left to right the possibility of \textit{by} referring to the location of the mill is eliminated. From right to left the reading of the mill's use of the stream is eliminated. There is no syntactic procedure for determining that the shared reading is acceptable:

(24) \textit{we explored the mill by using the little stream.}

In the case of

(25) \begin{align*}
& \textit{the bacteria} \quad \textit{the bacteria} \\
& \langle \textit{infected the boy} \quad \textit{infected the boy} \rangle \\
& \langle \textit{by} \quad \textit{using} \rangle \\
& \textit{the microscope} \quad \textit{the microscope}
\end{align*}

precisely this reading is unacceptable:

(26) \textit{the bacteria infected the boy by using the microscope}

so that (25) does not constitute a paraphrastic pair. We can distinguish (23) from (25) by saying that both elements of (23) are transformationally derived from (24), which is a good string, while the only common source for both strings of (25) is (26), which is a bad string. Thus, the strings of (25) are not paraphrases because they have no good source transformationally. What is crucial to note is that we referred to a type of source string that did not
have the ambiguities found in the elements of the pairs. Thus, the transformations involved only introduced ambiguities and did not resolve any.

The above examples indicate the following solution to the discrepancy between T and P. Transformations should be defined on only strings of a certain set, call them elementary strings and call the set E. These elementary strings must have no syntactic ambiguities, and E must have a source for every string of the language. We must return to the production system for each transformation. Each transformation will operate on the left element of a pair, which must be in E, and yield the right element. (16) illustrates a production system for the formal transformations with reference to E. To the production system for each transformation we need only add the condition that the left element of the pair be in E. But if we are going to obtain a production system as a result, there must be a production system for E.

(27) \[ E = e_1, e_2, e_3, \ldots \]

We require a production system for E, otherwise the vertical axis below would not exist.

(28) \[ \langle s_1, s_2 \rangle \langle s_3, s_4 \rangle \langle s_5, s_6 \rangle \langle s_7, s_8 \rangle \ldots \]

What we are primarily interested in is, of course, the set G of good strings. We know of no production system for G, but we can say that relative to G the elementary good strings EG have a production system. This “production system” amounts to the production system for E accompanied by a check on each produced element to see if it is in G. This check is what we are lacking a production system for. In mathematical jargon we say that EG is recursively enumerable relative to G. EG might even satisfy a stronger condition, according to Henry Hizi (1968, p. 247).

Among all acceptable texts, A, of a language under study, there are some texts which will be called elementary texts, el(A). The elementary texts are characterized by having a
property, \( P \), which is recursive, i.e., there is a decision procedure which tells us which of the texts of the language are elementary texts. This does not mean that there is a procedure which tells us which of the arbitrary strings are elementary texts. The set of elementary texts is recursive relative to the set of all texts. If a string is a text, then we can find automatically whether or not it possesses property \( P \) and, therefore, whether or not it is an elementary string.

However, for present purposes we will require only the minimum, that \( \mathbb{E} \mathbb{G} \) be recursively enumerable relative to \( G \).

To summarize, the set of elementary strings \( \mathbb{E} \) must be recursively enumerable, contain a good source for every good string of the language, and contain only strings which are unambiguous in the manner and choice of transformations which may apply to them.

One important type of ambiguity to be avoided is ambiguity of interreference. Unambiguous interreference is necessary in order to specify which occurrences of \emph{proposal} are to be relativized in

\[
(29) \quad \text{the proposal certainly was not spontaneous, but rather the President had instructed two months ago he should put forth the proposal after they had abandoned their own proposal and he was certain they would accept the proposal.}
\]

One means of keeping distinct interreferences is to require the elementary strings to include numerical descriptions on nouns, such as \emph{the first proposal} and \emph{the second proposal}, which are eliminated overtly by certain transformations but preserved as part of the transformational history of a string.

It may well happen for a given language that no one single string is elementary, that is, is syntactically unambiguous. In such a case, we would select however many strings were necessary to resolve the ambiguity. Then instead of the sources being single strings they would be elementary paraphrastic sets. Consider the string

\[
(30) \quad \text{the man entertained the boy with a train.}
\]

We might try to pick a single string for a source in English. The requirement of having a production system for such strings constrains them to have in general a specific form. For example, underlying (30) we might have

\[
(31) \quad \text{the man did some entertaining and the entertaining was of a boy and the boy had a train}
\]

or

\[
(32) \quad \text{the man did some entertaining and the entertaining was of a boy and the entertaining was by means of a train}
\]

as sources. (31) and (32) have a form that is syntactically quite unambiguous. We could choose this form as the general form for all elementary strings.
The other possibility we can also illustrate with (30). Both (30) and

(33)  the boy was entertained by the man with a train

are syntactically ambiguous. But they complement each other on one reading. If we state that (30) and (33) are paraphrases, then with a train must be an adverbial phrase. In this approach the members of E as well as the input to transformations would be two member paraphrastic sets with elements of a certain form for example,

(34)  the man entertained the boy with the train
       the boy was entertained by the man with the train

I prefer the solution of (31) and (32) myself. But the final choice depends on the language.

We will discuss in the next section just how to impose the condition that transformations operate only on strings in E. Assuming for the present that it has been imposed, we have a production system T which has no <good, bad> pairs and whose <good, good> pairs are all paraphrastic. The reason is that the original reading provided by the syntactically unambiguous source in E will not be eliminated by a transformation, because it is the very basis for the transformation. However, T is only a proper part of P, because now all pairs in T have first elements in E. To enlarge T to P, we proceed as follows. List T on both the horizontal and vertical axes of the following two-dimensional array.

(35)  \langle s_1, s_2 \rangle \langle s_3, s_4 \rangle \langle s_5, s_6 \rangle \langle s_7, s_8 \rangle \ldots

At each intersection compare the lefthand members of the two pairs: \langle s_i, s_j \rangle, \langle s_k, s_l \rangle. If s_i is the same as s_k, then \langle s_j, s_l \rangle is put as the next element in the new set T'. In this manner all strings which have the same reading, that is, the same source in E, will be paired in T'. Thus, T' will have no <good, bad> pairs, and all <good, good> pairs will be paraphrastic and will include all the paraphrastic pairs of P. In fact, T' is the production system for P.
T is also the basis for producing paraphrastic sets of any size, in particular the generalizable paraphrastic sets we discussed in chapter two as the empirical basis for our theorizing. We start the production system for T. As each pair of T, \( \langle s_i, s_j \rangle \), is produced it is compared with each preceding ordered n-tuple \( \langle s_m, s_n, \ldots, s_p \rangle \). If the first element of the pair, \( s_i \), is identical with the first element of the n-tuple, \( s_m \), then the second element of the pair, \( s_j \), is added to the n-tuple to be the \( n+1 \)st element of an \( n+1 \)-tuple. In this way all strings which have the same source in \( E \) are grouped together. Given any generalizable paraphrastic set, its members will all be found among the elements of some n-tuple of some point in the development of the above described sequence.

We now have a sketch of a formal theory accounting for the empirical data gathered about the concept of paraphrase.

### 2. Elementary Transformations

In the preceding section we assumed that there was a production system for the set of transformations. A particularly nice kind of production system results if we can find a finite list of elementary transformations: \( et_1, et_2, \ldots, et_n \) such that every transformation can be represented simply as a sequence of elementary transformations:

\[
(36) \quad t_i = et_j \cdot et_k \cdot \ldots \cdot et_s
\]

Each elementary transformation in a sequence is applied in turn starting at the right and moving to the left. For example, the transformation \( t_1 \) of

\[
(37) \quad \text{a boy came to the shop this morning and the boy bought a camera}
\]

to

\[
(38) \quad \text{a camera was bought by a boy who came to the shop this morning}
\]

might be conveniently analyzed as two transformation in sequence. A relative clause transformation, \( t_{rel} \), changes (37) to

\[
(39) \quad \text{a boy who came to the shop this morning bought a camera}
\]

And then a passive transformation, \( t_{pass} \), changes (39) to (38). We could then write

\[
(40) \quad t_1 = t_{pass} \cdot t_{rel}
\]

A production system for the set of all transformations is given by a production system for all possible combinations of the elementary transformations. The production system begins with the finite list of elementary transformations in some order. The rule is that the next list is made so that to each
element in the preceding list is added each of the elementary transformations according to the order of the elementary transformations. Each list is really a finite production system.

\begin{equation}
\text{(41)}
\end{equation}

\[
\begin{array}{l}
\text{et}_1, \text{et}_1, \text{et}_1, \text{et}_1, \text{et}_1, \ldots\\
\text{et}_n, \text{et}_1, \text{et}_n, \text{et}_1, \text{et}_1, \ldots\\
\text{et}_1, \text{et}_n, \text{et}_1, \text{et}_1, \text{et}_n, \ldots\\
\text{et}_n, \text{et}_n, \text{et}_n, \text{et}_n, \text{et}_n, \ldots
\end{array}
\]

The sequence of production follows the obvious path through the one element list, two element list, three element list, and so forth. Let us now see how we can realize this reduction to elementary transformations.

An immediate stumbling block confronts us. We decided in the preceding section that all transformations should be defined on elementary strings only. Since the outputs of the transformations will generally not be elementary strings, it is not possible to further transform the output of a transformation. Thus, we cannot represent a transformation as a sequence of transformations. It is at this point that the value of two levels of representation is apparent. They allow us both to define all transformations on elementary strings and to represent transformations as sequences of elementary transformations.

We recall that the reason for defining transformations exclusively on elementary strings was the problem of syntactic or structural ambiguity. Elementary strings had the property of being syntactically unambiguous and transformations introduced all syntactic ambiguities. We also recall from the chapter on functional representation that transformations may be defined on representations that leave no doubt about the syntactic analysis. For example,

\begin{equation}
\text{(42)} \quad \text{we explored the mill using the little stream}
\end{equation}

can be represented functionally as

\begin{equation}
\text{(43)} \quad S[\text{using the little stream #}] [\text{we, explored the mill}]
\end{equation}

or as

\begin{equation}
\text{(44)} \quad S[\text{we, V[explored, N[the] [using the little stream] [mill]]}].
\end{equation}

(43) and (44) give unequivocal functional syntactic analyses of (42). Thus, if every string could be given such analyses we could define transformations
on the functional representations and easily factor all transformations into elementary ones. But the old problem arises in a different form. How do we formalize the fact that (42) has two semantically valid functional representations and

\[(45) \textit{the bacteria infected the boy using the microscope}\]

only has one, namely

\[(46) S[\textit{the bacteria}, V[\textit{infected}, N[\textit{the}][\textit{using the microscope}][\textit{boy}]]].\]

I know of no formal procedure for deciding in general which functional representations are associated with a given phonological representation. However, the elementary strings were chosen for their being syntactically unambiguous. That is, a given elementary string should only have a single functional representation. In the case of elementary strings no decision is necessary. For example,

\[(47) \textit{a boy was using the microscope and the bacteria did some infecting and the infecting was of the boy}\]

has only one possible functional representation.

We can use the device of functional representation to remove the burden of syntactic ambiguity from the transformations. Transformations on functional representations do not introduce ambiguities. The ambiguities all derive from the mapping of the functional representation to the phonological representation. In the following diagram the top level is a sequence of transformationally related functional representations and on the bottom level are the corresponding phonological representations. The far left string is elementary.

\[(48) \begin{array}{c}
\text{efr}_1 \rightarrow \text{fr}_2 \rightarrow \text{fr}_3 \\
\downarrow \quad \downarrow \quad \downarrow \\
\text{epr}_1 \rightarrow \text{pr}_2 \rightarrow \text{pr}_3
\end{array}\]

The arrows indicate directions of mappings. The mapping from a functional representation to its corresponding phonological representation is given by the functions themselves, as we discussed in the chapter on functional representation. The inverse of this map is not unique in general, since a given phonological representation may derive from many different functional representations. The main exception to this is the elementary strings. This map is unique in both directions, as indicated by the double arrow. We can represent the transformational relation of \(\langle \text{efr}_1, \text{fr}_3 \rangle\) as composed of the elementary transformational relations of \(\langle \text{efr}_1, \text{fr}_2 \rangle, \langle \text{fr}_2, \text{fr}_3 \rangle\). But we can-
not say that the relation of \( \langle epr_1, pr_3 \rangle \) is composed of the relations of \( \langle epr_1, pr_2 \rangle, \langle pr_2, pr_3 \rangle \). The reason is that the relation of \( \langle pr_2, pr_3 \rangle \) involves a non-unique operation, a resolution of an ambiguity, as indicated by the single arrow from \( fr_2 \) to \( pr_2 \). Thus, on the phonological level \( \langle epr_1, pr_3 \rangle \) cannot be factored into elementary relations. Let us take an example

\[
(49) \quad \begin{align*}
 efr_1 : & \ S[and] \ S[and] \ S[we, V[did, some exploring]], S[the\ exploring, V[was, of the mill]], S[the\ exploring, V[was, by using the little stream]]
 
 epr_1 : & \ we\ did\ some\ exploring\ and\ the\ exploring\ was\ of\ the\ mill\ and\ the\ exploring\ was\ by\ using\ the\ little\ stream
 
 fr_2 : & \ S[and] \ S[we, V[did, N[A[which, V[was, of the mill]]] [some exploring]]], S[the\ exploring, V[was, by using the little stream]]
 
 pr_2 : & \ we\ did\ some\ exploring\ which\ was\ of\ the\ mill\ and\ the\ exploring\ was\ by\ using\ the\ little\ stream
 
 fr_3 : & \ S[and] \ S[we, V[explored, the mill]], S[the\ exploring, V[was, by using the little stream]]
 
 pr_3 : & \ we\ explored\ the\ mill\ and\ the\ exploring\ was\ by\ using\ the\ little\ stream
 
 fr_4 : & \ S[we, V[by using the little stream#] [explored, the mill]]
 
 pr_4 : & \ we\ explored\ the\ mill\ by\ using\ the\ little\ stream
 
 fr_5 : & \ S[we, V[using the little stream#] [explored, the mill]]
 
 pr_5 : & \ we\ explored\ the\ mill\ using\ the\ little\ stream
 
 fr_6 : & \ S[we, V[by the little stream#] [explored, the mill]]
 
 pr_6 : & \ we\ explored\ the\ mill\ by\ the\ little\ stream
\end{align*}
\]

We can safely treat \( \langle fr_5, fr_6 \rangle \) as exemplifying a transformational relation, since \( using\ the\ little\ stream \) is clearly indicated by the functional representation as being in an adverbial role. However, \( \langle pr_5, pr_6 \rangle \) does not have the same status. We have no method for making certain that the acceptability of \( pr_5 \) does not depend solely on an adjectival role of \( using\ the\ little\ stream \) modifying \( mill \), as for example, is the case with

\[
(50) \quad \text{the bacteria infected the boy using the microscope}
\]

the danger is again that a non-paraphrastic pair would result, especially in context. In the case of (50) we would have

\[
(51) \quad \text{the bacteria infected the boy by the microscope}
\]

and \( \langle 50, 51 \rangle \) is precisely the kind of pair we wish to avoid. Thus, on the phonological level all transformations are defined on elementary strings and \( \langle efr_1, fr_2 \rangle, \langle fr_2, fr_3 \rangle \), and so forth merely represent operations which are
not transformations. But on the functional level we can record on a string as much of its transformational history as we need for the further application of the various transformations. We are, thereby, able to factor transformations into elementary transformations, such as,

\[(52) \quad \langle efr_1, fr_6 \rangle = \langle fr_5, fr_6 \rangle \cdot \langle fr_4, fr_5 \rangle \cdot \langle fr_3, fr_4 \rangle \cdot \langle fr_2, fr_3 \rangle \cdot \langle efr_1, fr_2 \rangle\]

On the functional level, then, we have a set of elementary transformations whose combinations will produce the entire set of transformations, according to the plan outlined earlier in (41). Even on the phonological level the operations that constitute the elementary functional transformations can be used in a manner parallel to (41) to produce the entire set of transformations.

3. SUMMARY

If the present chapter has a single overarching thought, it is that the subject matter of a paraphrase grammar is not the strings of a language but relations among those strings, namely relations of paraphrase. We sought a production system, not for all the good strings of a language, but rather for the paraphrase relation among those strings. We studied not the algebra of combining strings of a language, but rather the algebra of elementary transformations.

We showed informally how the production systems for transformational relations and admission conditions can be combined to form a production system for the generalizable paraphrase relation by having recourse to a set of elementary strings, which itself has a production system. We then showed how the elegant algebraic idea of a set of elementary transformations which generates the whole set of transformations could be applied to the production system of the set of transformations.
APPENDIX

RECURSIVE ENUMERABILITY

In the last chapter we relied rather heavily on the concept of a production system without really defining it. There are a number of techniques available in the mathematics of arithmetical functions which could be used to define a production system, which is usually referred to as a recursive enumeration. The technique we shall use is developed by Raymond Smullyan under the name of 'elementary formal systems' (Smullyan, 1961). We will give Smullyan's definition of an elementary formal system and then sketch some applications which illustrate the mathematical foundations of the present book.

Elementary formal systems rely on the ideas of recognition of symbols in a finite alphabet, for example,

(1) \( a, b \)

concatenation of strings of symbols in the alphabet, for example,

(2) \( a \) concatenated with \( b \) is \( ab \)

substitution of strings of the alphabet for variables, for example,

(3) \( axbx \) becomes \( abbaa \)

when \( ba \) is substituted for the variable \( x \), and detachment of a satisfied condition, for example

(4) given that \( A \) is sufficient for \( B \), we can conclude \( B \) if we have \( A \).

The language and organization of the formal systems must be made unequivocal.

We distinguish between the language being described and the language of formal systems. The language being described is made up exclusively of strings formed out of symbols of an alphabet which is preset. A formal system does not use this language, but merely talks about it. So the strings of the language being described will be italicized to indicate that we are not using them to say anything.

The language of formal systems contains a list of descriptive predicates, whose meanings are given by their use in a formal system. The convention is that bold-face capital English letters will be used for predicate symbols: \( A, B, C, \ldots \). We will also use mnemonic combinations of capital letters, such as
SUBST for a substitution predicate. There is also a list of variables. These
will be lower case letters from the end of the English alphabet: z, y, x, w, ....

A term is an arbitrary string of variables and symbols from the language
being described. An example for the alphabet of a and b is

(5)  \( aaxbyb \).

An atomic formula is a predicate symbol followed by a number of terms
separated by commas. Thus, a comma ',' is a constant symbol of the formal
system language. The idea is that a predicate symbol might actually indicate
a relation among a number of strings. Let \( AB \) be the predicate that means
a string alternating in a and b. Then

(6)  \( AB \ ab \)

would be an atomic formula stating that \( ab \) is a string alternating in a and b.

Let \( I \) be a predicate of identity between two strings. Then

(7)  \( I \ a, b \)

is an atomic formula claiming a is identical to b.

A general formula is either an atomic formula or a string of atomic
formulas all separated by '→', the sign for detachment of a condition. For
example, a general formula is

(8)  \( I \ x, y \rightarrow I \ y, x \)

which says that if x is identical to y, then y is identical to x. '→' is a second
constant symbol of the formal system language.

The above described terms and formulas constitute the entire language
of formal systems.

An elementary formal system has two parts. The first is a set of axioms,
which is simply a finite list of general formulas, for example,

(9)  \( AB \ ab \)
\( AB \ ba \)
\( AB \ xb \rightarrow AB \ xba \)
\( AB \ xa \rightarrow AB \ xab \).

The first two axioms are absolute statements that \( ab \) and \( ba \) satisfy the pred-
icate \( AB \). The last two axioms are conditional statements, that allow
conclusions when the conditions are satisfied. The second part of a formal
system consists of two rules. Formulas derived from the axioms by the rules are called theorems.

(10) Substitution rule: in a theorem any variable may be replaced in all its occurrences by any given string of symbols of the alphabet of the language being described. All occurrences of the variable must be replaced by the same string.

Detachment rule: if the first condition of a theorem is also a theorem, then the remainder of the theorem can be detached from the condition as a theorem.

Thus, we can add to (9) by substitution of $a$, for example, in axiom three,

(11) \( AB \ ab \rightarrow AB \ aba \)

and then by detaching axiom one from (11) we have the theorem

(12) \( AB \ aba. \)

Now substituting $ab$ in axiom four we have

(13) \( AB \ aba \rightarrow AB \ abab \)

and detaching (12) we have

(14) \( AB \ abab. \)

We note that by substituting $b$ in axiom three we obtain the following theorem.

(15) \( AB \ bb \rightarrow AB \ bba \)

But we cannot detach the condition, since it is not itself a theorem. Thus, we are primarily interested in atomic formulas which are theorems. There is no 'if' involved in them. They state absolutely that, for example, \( abab \) satisfies the predicate \( AB. \)

We say that a predicate in a formal system "represents" the set of those strings or $n$-tuples of strings and only those that satisfy it. For example, \( AB \) represents the set containing $ab, ba, aba, abab, ....$

When we are trying to make an intuitive concept precise in the manner of a formal system, we must prove that the set that the predicate represents is the set we have in mind. For example, we should prove that the set represented by \( AB \) in (9) is identical to the set of strings alternating in $a$ and $b$. Such proofs are usually by induction. I refer the reader to Smullyan's *Theory of Formal Systems* for sample proofs. We will not try to verify any of the representations made here, since this appendix is merely a sketch of the method of formal systems. Let us now proceed with some examples of formal systems for natural languages.
The set of nouns of English, aside from co-occurrence problems, can be partially characterized by the following formal system.

(16) \[ \text{EN } \text{horse} \]
    \[ \text{EN } \text{man} \]
    \[ \text{EN } \text{box} \]
    \[ \text{EN } \text{ball} \]
    \[ \text{EA } \text{white} \]
    \[ \text{EA } \text{broad} \]
    \[ \text{EA } \text{old} \]
    \[ \text{C and} \]
    \[ \text{C or} \]
    \[ \text{T a} \]
    \[ \text{T the} \]
\[ \text{EN } x \rightarrow \text{EA } y \rightarrow \text{N } yx \]
\[ \text{EN } x \rightarrow \text{EN } y \rightarrow \text{N } xy \]
\[ \text{C } x \rightarrow \text{N } y \rightarrow \text{N } z \rightarrow \text{N } yxz \]
\[ \text{T } x \rightarrow \text{N } y \rightarrow \text{TN } xy \]
\[ \text{C } x \rightarrow \text{TN } y \rightarrow \text{TN } z \rightarrow \text{TN } yxz \]

The ideas are EN, elementary noun, EA, elementary adjective, C, conjunction, T, article, N, noun, without article, TN, noun with article. The formal system develops its theorems as follows.

(17) \[ \text{EN } \text{horse} \rightarrow \text{EA } \text{white} \rightarrow \text{N } \text{white horse} \]
    \[ \text{EA } \text{white} \rightarrow \text{N } \text{white horse} \]
I claim no elegance or completeness for any of the formal systems presented here. I merely wish to indicate the nature of the mathematics involved. It should be clear from (16) how categories of phrases are represented.

Joining relations are three place predicates. For example, one axiom in the ad-junction relation might be

\[(18) \quad N x \rightarrow EA y \rightarrow ADJ y, x, yx\]

where 'ADJ y, x, z' means y is adjoined to x to yield z.

The substitution relation can be roughly characterized by the formal system below. 'SUBST x, y, z, w' means that substituting x for y in z yields w. 'NOC x, y' means that x does not occur in y. We assume we already
have a formal system for ‘\textsc{NOC} x, y’. I have combined a number of axioms by using a ‘o’ subscript on variables to mean the variables are not there in a related axiom. Of course, if the variable is used once in an axiom, it must be used in all other places designated.

\begin{equation}
\text{SUBST} \ x, y, y, x
\end{equation}

\begin{align*}
\text{NOC} \ y, w_o v & \rightarrow \text{SUBST} \ x, y, z_o y w_o, u \rightarrow \text{SUBST} \ x, y, z_o y w_o, v u \\
\text{NOC} \ y, v z_o & \rightarrow \text{SUBST} \ x, y, z_o y w_o, u \rightarrow \text{SUBST} \ x, y, v z_o y w_o, v u \\
\text{NOC} \ y, w_o v & \rightarrow \text{SUBST} \ x, y, z_o y w_o, u \rightarrow \text{SUBST} \ x, y, v o y t_o, s \rightarrow \\
& \rightarrow \text{SUBST} \ x, y, z_o y w_o v o y t_o, u s
\end{align*}

The system is quite simple. First, the string by itself is replaced. Then strings can be added to either side of the substitutionally related strings as long as no occurrences of the replaced string are introduced by the added material. Finally, two substitution pairs can be concatenated into a further pair as long as the concatenation produces no additional occurrences of the replaced string. The system yields replacement of an unbounded number of occurrences, which is the essence of the general substitution relation.

The substitutional notation used for transformations in the preceding book is readily translatable into the notation of formal systems. Consider the rough pronominalization transformation

\begin{equation}
S_1(N_2) \rightarrow S_1(\text{prnN}_2)
\end{equation}

which can be represented as

\begin{equation}
\begin{align*}
N \ N_2 & \rightarrow \text{PRN} \ N_2, \text{prnN}_2 \rightarrow S \ S_1(N_2) \rightarrow S \ S_1(\text{prnN}_2) \rightarrow \text{SUBST} \\
& \text{prnN}_2, N_2, S_1(N_2), S_1(\text{prnN}_2) \rightarrow \text{TPRN} \ S_1(N_2), S_1(\text{prnN}_2)
\end{align*}
\end{equation}

That is, if $N_2$ is a noun and $N_2$ and $\text{prnN}_2$ are in the pronoun relation and $S_1(N_2)$ is a sentence and $S_1(\text{prnN}_2)$ is a sentence and $\text{prnN}_2$, $N_2$, $S_1(N_2)$, $S_1(\text{prnN}_2)$ satisfy the substitution relation in the order given, then $S_1(N_2)$, $S_1(\text{prnN}_2)$ in that order satisfy the pronominalization transformation.

Alternatively, if we wish transformations to enlarge the basic concepts as well as produce transformationally related pairs, we can represent (20) as

\begin{equation}
\begin{align*}
N \ N_2 & \rightarrow \text{PRN} \ N_2, \text{prnN}_2 \rightarrow S \ S_1(N_2) \rightarrow \text{SUBST} \ \text{prnN}_2, N_2, S_1(N_2), S_1(\text{prnN}_2) \\
& \rightarrow \text{N} \ \text{prnN}_2 & \\
& S \ S_1(\text{prnN}_2) & \\
& \text{TPRN} \ S_1(N_2), S_1(\text{prnN}_2)
\end{align*}
\end{equation}
which is an abbreviation of three rules, since the ‘&’ symbol is not really part of the notation of elementary formal systems.

The above sample translations illustrate how formal systems can be constructed for transformations by building on formal systems for other grammatical properties and relations.

The concept of a formal system is not identical with that of a production system or recursive enumeration. The latter systems are linear while a formal system is a tree or branching system. I will sketch briefly how to linearize formal systems.

We must first obtain a linear system for all possible strings in the alphabet of the language being described. This is easy. Let me illustrate with a two symbol alphabet of $a$ and $b$.

\[
\begin{array}{c}
\text{aaa} \\
\text{aab} \\
\text{aba} \\
\text{abb} \\
\text{baa} \\
\text{bab} \\
\text{bba} \\
\text{bbb}
\end{array}
\]

This linear system can now be used to construct a linear system for each n-tuple of strings of the alphabet. For ordered pairs it is simple. Let the following be the above linearization of the strings of the alphabet of $a$ and $b$.

\[\text{STR} = s_1, s_2, s_3, \ldots\]

We use (24) to construct a linearization for each n-tuple of strings of the alphabet. For ordered pairs it is simple.

\[
\begin{array}{cccc}
\text{s}_1 & \text{s}_2 & \text{s}_3 & \ldots \\
\langle \text{s}_1, \text{s}_1 \rangle & \langle \text{s}_1, \text{s}_2 \rangle & \langle \text{s}_1, \text{s}_3 \rangle & \\
\langle \text{s}_2, \text{s}_1 \rangle & \langle \text{s}_2, \text{s}_2 \rangle & \langle \text{s}_2, \text{s}_3 \rangle & \\
\langle \text{s}_3, \text{s}_1 \rangle & \langle \text{s}_3, \text{s}_2 \rangle & \langle \text{s}_3, \text{s}_3 \rangle & \\
\end{array}
\]

In general, for n-tuples we must go to an n-dimensional array.

For each n we have a linearization of n-tuples of strings in the alphabet.
of \( a \) and \( b \). These linearizations will be used in the systematization of the substitution rule.

Let us take a given formal system and list its axioms: \( a_1, a_2, a_3, \ldots, a_m \). Each axiom will have a certain number \( n \) of distinct variables. Let \( n_i \) be the number of distinct variables in \( a_i \). For each \( a_i \) we can construct a linearization of all possible substitutions in \( a_i \). We begin with the linear system of \( n_i \)-tuples. Each \( n_i \)-tuple results in a substitution instance of \( a_i \) by substituting the first element of the \( n_i \)-tuple for the first variable of \( a_i \), the second for the second, and so forth up to the \( n_i \)th for the \( n_i \)th. Going through the linear system of \( n_i \)-tuples in the above fashion yields a linear system of substitution instances of \( a_i : a_{i,1}, a_{i,2}, \ldots \). From the linear system for each \( a_i \) we obtain a linear system for all substitutions in all the axioms.

\[
\begin{align*}
\text{(26)} \\
\begin{array}{c}
a_{1,1} \\
a_{2,1} \\
a_{3,1} \\
\vdots \\
a_{m,1} \\
\end{array}
\begin{array}{c}
a_{1,2} \\
a_{2,2} \\
a_{3,2} \\
\vdots \\
a_{m,2} \\
\end{array}
\begin{array}{c}
a_{1,3} \\
a_{2,3} \\
a_{3,3} \\
\vdots \\
a_{m,3} \\
\end{array}
\end{align*}
\]

From (26) we obtain the linearization of all substitutions in all the axioms, which we write:

\[
\text{(27) } SB = sb_1, sb_2, sb_3, \ldots
\]

It is a well-known fact of formal systems that the restriction that substitutions be made only in the axioms does not change the set of theorems of a formal system.

We have only one operation left in the production of the theorems of a formal system, namely detachment. Let us define a detachment operation on a linear system as follows. An element \( k \) in the linear system is selected and we ask for each predecessor of \( k \) if it together with \( k \) allows the detachment of a condition. If it does, then we place the result of the detachment immediately after the result of the last previous such detachment, if any, otherwise immediately after \( k \). This operation is carried out in a finite number of steps, since each element of a linear system has only a finite number of predecessors. The result of the operation is a new linear system, differing only in a finite number of elements from the original.

We first apply this operation to the first element of \( SB \) and obtain the linear system \( D_1 \). Then to the second element of \( D_1 \) we again apply the
operation to obtain $D_2$, and from $D_2$ we get $D_3$ and so forth. We thus have a linear sequence of linear systems: $D_1, D_2, D_3, \ldots$. Now we form a new linear system $FS$ composed of the first element of $D_1$, the second element of $D_2$, the third element of $D_3$, and so forth. $FS$ is a linear system of all the theorems of the formal system, except perhaps some axioms, which we can add now if we wish.

The last step is to go through the linear system $FS$ searching for atomic formulas which have the predicate we are interested in. Placing these formulas in a sequence according to the order in which we found them yields a new linear system $P$ which is a production system or recursive enumeration of the concept we were seeking to characterize.

There is no need to go through the above linearization process for each case. It is sufficient to know that it can be done. The technique of elementary formal systems provides an interesting and direct way of describing recursive concepts in language.
BIBLIOGRAPHY


INDEX OF SUBJECTS

acceptability 32, 34, 38f, 58, 71f, 113
ad-junction 82f, 99f, 107
admission condition 9ff, 75ff, 123
algebra 117
ambiguity 4f, 27, 105, 122, 131
attainable concepts 50
bad string 119ff
base strings, see elementary strings
because 60
Bese 40ff
binary transformations V, 36ff
bound pronouns 111f, 116
breaking function 110
classifiers 19
collection of data 16
compositional grammars 6f, 14, 32ff, 45, 64ff
compositional languages V, 14, 32, 50
concatenation 95, 134
con-junction 82f, 99f, 107
context 39, 59f
co-occurrence V, 26, 35, 38, 39, 40, 50f
dependence 13f, 24, 103ff
direct discourse 54f, 103f
ed 23
elementary formal systems 134ff
elementary strings 29ff, 32, 35f, 126ff
elementary transformations 129ff
equivalence 11f, 75, 88ff
essence 57f, 59
finite 14
free pronouns 111f, 115
functional representation 12ff, 21ff, 94ff,
102, 116, 130ff
generalization 1f, 4, 16, 19ff, 26, 30, 53, 64ff
good string 57, 59, 118ff
governance, see dependence
grammatical category 8, 58f, 81ff, 137f
grammatical category symbols 60, 97f
grammaticality 32, 34, 38f, 40, 42f
indirect discourse 54f, 103f
intonation 13
inverse 105, 117
joining 81ff, 98ff, 106ff
junction 81, 98f, 106f
kernel 36, 49
lattice 65ff, 69f
level 86f, 111f
lexical entries 46
ly 23
major segment 108, 116
minimal sentence (min S) 87f
0 subscript 60ff, 74
0 superscript 87ff
obligatory transformations 106
oblique case (obl) 22
-ooccX 79f
operation 69ff, 72, 132f
ordering of transformations 10, 78ff
parameter 13, 62f, 96, 103, 109
paraphrase grammar 4ff, 14, 117ff
paraphrase relation VI, 1ff, 4, 16ff, 58, 117ff
paraphrastic sets 2f, 17f, 19ff, 128f
paraphrastic transformation 4f
passive transformation 5, 9, 28, 75ff, 78
past participle 23
personal pronoun 54, 55, 112f
phonological function 105f, 111, 114
phonological representation 94, 105ff, 131ff
phonology 105ff
plural 12f
possible sentence 41ff
pre-junction 84ff
production system 117f
projection rules 40ff
pronominalization 53f, 78
readings 17f
recursive definition 80ff
recursive enumerability 117ff, 126, 134ff
recursivity 117ff, 127
reflexive pronoun 10f
relation 118
relative clause 8, 11, 55f, 60ff, 65ff, 77ff, 85, 97f, 102f, 108ff, 111
relative pronoun 21, 55, 77, 108, 111ff
rewrite rules 33f
selection of data 18f
semantic ambiguity 27f
sentential function 13
subclassification 10, 20, 35f
subscripts 8, 60, 103
substitution 7ff, 11, 52ff, 64, 74f, 91, 122f, 134, 138f
substitution notation 7ff, 53, 56ff, 74
synonymy 19
syntactic ambiguity 27f, 125, 127f, 130f
syntactic function 94
syntactic representation 94ff
syntacto-phone 108, 111
tense 14
transformation V, 4f, 10, 43, 46, 57, 69ff, 97f, 123ff, 131ff, 139
transformational history 77, 103, 105
transformation relation 118ff, 131
tree notation 5, 9, 12, 35, 75, 96f
truth 40, 42ff
wh 21, 112ff
world-class notation 9, 12, 34, 75
word-sharing 36f