17

1965 International Conference on Computational Linguistics

A SYSTEM FOR TRANSFORMATIONAL ANALYSIS

Susumu Kuno

The Computation Laboratory
Harvard University
Cambridge, Massachusetts 02138

ABSTRACT

A system is proposed here for assigning a derived P-marker to a given transformed sentence and obtaining the corresponding base P-marker at the same time. Rules of analytical phrase-structure grammar for such a system have associated with them information pertaining to the transformational histories of their own derivation. When a phrase-structure analysis of the sentence is obtained, the set of grammar rules used for the analysis contains all the information necessary for the direct mapping of the derived P-marker into the corresponding P-marker. The system can also be used for decomposing a given complex sentence into "kernel" sentences for the purpose of structure matching between a query sentence and stored document sentences in information retrieval. An experimental program for the proposed system has been written and is currently tested with a small sample grammar. Study is underway to see if there is any mechanical procedure for obtaining an analytical phrase structure grammar of the proposed type for a given transformational grammar.

A SYSTEM FOR TRANSFORMATIONAL ANALYSIS *

Susumu Kuno

1. Introduction

Numerous systems for the automatic recognition procedures of context-free languages have been proposed:[1] among them, two systems are in operation with comparatively large English grammars. One is J. Robinson's English parser[2] based on J. Cocke's algorithm,[3] and the other is the Kuno-Oettinger predictive analyzer of English.[4,5,6]

The proponents of neither of the two systems have been satisfied with simply assigning phrase-structure descriptions to each given sentence. A paraphrasing routine has been added to Robinson's English parser[7] so that a set of kernel sentences can be obtained in addition to the phrase-structure description of the sentence. For example, the analysis outputs of "X commands the third fleet.", "The third fleet is commanded by X.", and "X is commander of the third fleet." would all contain the information that the kernel is "S -- X, V -- commands, O -- third fleet". In connection with the Kuno-Oettinger predictive analyzer, three kernelizing routines have been proposed by J. Olney,[8] B. Carmody and P. Jones,[9] and D. Foster,[10] which accept as input the output of the predictive analyzer and produce either kernel sentences or pairs of words which are in certain defined syntactic relationships. The SMART information retrieval system,[11,12,13,14,15] <u>S</u>alton's <u>M</u>agic <u>A</u>utomatic <u>R</u>etriever of <u>T</u>exts, has a routine which compares the structure diagram (part of the analysis output of the predictive

*This work has been supported in part by the National Science Foundation under Grant GN-329.

analyzer) of a request sentence with the structure diagrams of sentences to be retrieved, so that paraphrases of the same kernel sentence can be identified.

The aim of the present paper is to investigate the role of the predictive analyzer in a transformational grammar recognition system, and to propose a system for analysis of a language of a given transformational grammar. Before going into details of the proposed system, it is worthwhile to discuss briefly two other systems so far proposed as transformational grammar recognizers.

2. General Solution to Recognition Problems of Transformational Languages

(i) Analysis by Synthesis

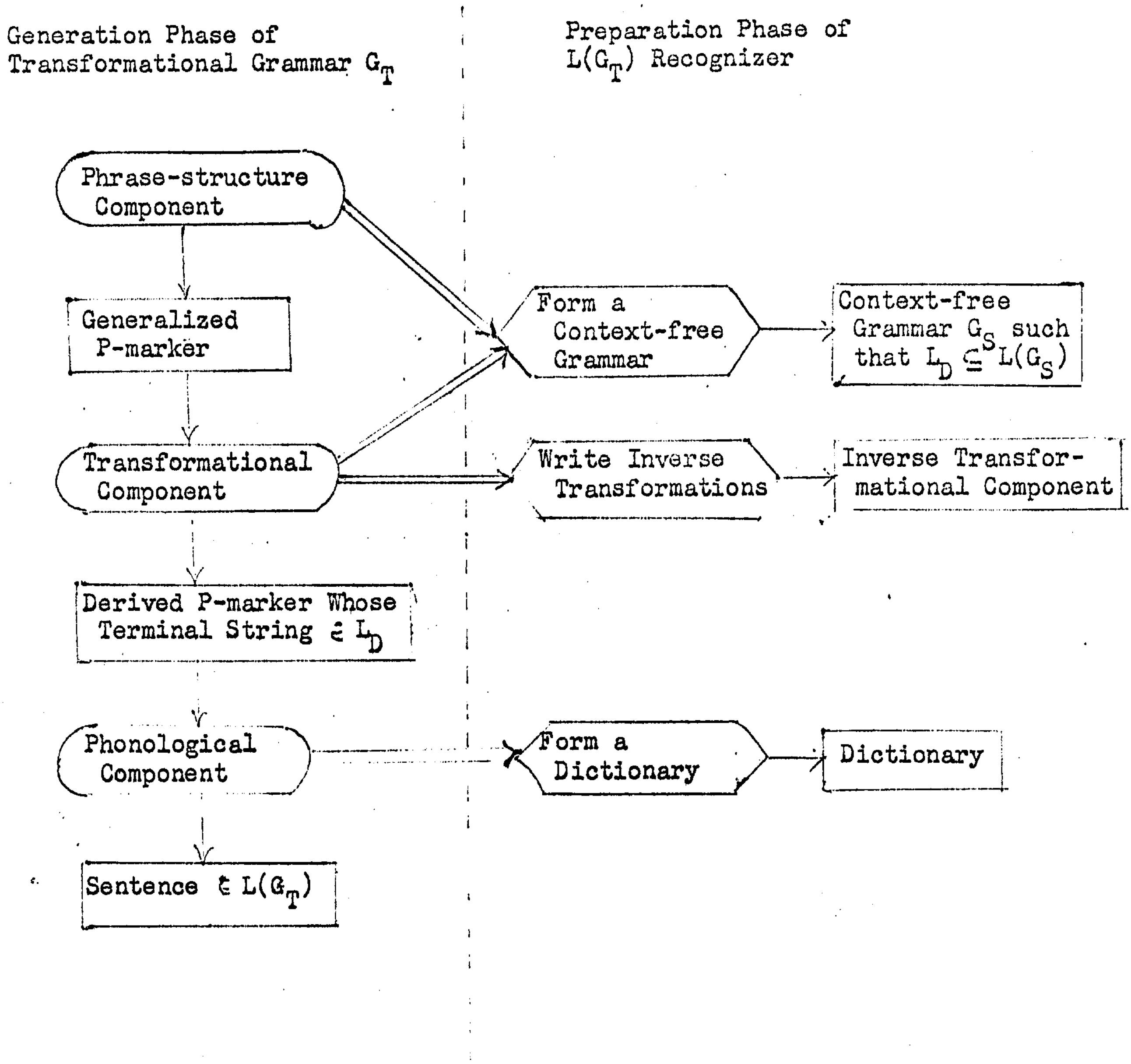
D. E. Walker and J. M. Bartlett[16] have proposed a system which parses the language of a given transformational grammar. Their system is essentially based on Matthews' proposal[17] for analysis by synthesis. Analysis of a sentence is performed by generation of all possible strings from the initial symbol "Sentence" by means of a phrase-structure component, a transformational component, and a phonological component. Each of the terminal strings thus generated is matched against the input sentence. When a match is found, the path which has led to the matched terminal string represents an analysis of the input sentence. Certain heuristics are used to distinguish transformations which could have been applied to generate the sentence under analysis from those which could not have. For example, if a sentence ends in a question mark, then it is certain that at some point the question transformation was used.

The Walker-Bartlett system, although drastically improved in efficiency compared to the proto-type proposed by Matthews, seems to be still far from being practicable because of an astronomical number of sentences that will have to be generated before the match is found.

(ii) From Derived P-markers to Base P-markers

Two similar parsing methods have been independently proposed by S. Petrick[18] and the MITRE Language Processing Techniques Subdepartment (Zwick, A. M., Hall, B. C., Fraser, J. B., Geis, M. L., Isard, S., Mintz, J., and Peters, P. S.) directed by Walker[19] as a general solution for the recognition problem of the language generated by a given transformational grammar. A transformational generative grammar G_T has three components: the phrase-structure component, the transformational component, and the phonological component (see Diagram 1). The output of the phrase-structure component are generalized P-markers which have grammatical and lexical forms emanating from the lowest nodes in the trees.* The function of the transformational rules is to map generalized P-markers into derived P-markers. If the transformational rules map the generalized P-marker M_G into the final derived P-marker M_D of the sentence X, then M_G is the deep structure (base P-marker) of X and M_D is its surface structure. The M_D is then transferred to the phonological component, whose output is the plain terminal string X.[20]

*A slightly outdated model of a transformational grammar is presented here for the purpose of avoiding delicate arguments not directly connected with the aim of the present paper.


Generation Phase of Transformational Grammar G_T
Preparation Phase of $L(G_T)$ Recognizer
Phrase-structure Component
Generalized P-marker
Transformational Component
Derived P-marker Whose Terminal String $\in L_D$
Phonological Component
Sentence $\in L(G_T)$
Form a Context-free Grammar
Context-free Grammar G_S such that $L_D \subseteq L(G_S)$
Write Inverse Transformations
Inverse Transformational Component
Form a Dictionary
Dictionary

Transformational Language Recognizer (1)

Diagram 1

Consider the (probably infinite) set of derived P-markers obtainable from a given transformational grammar G_T. Each P-marker has at the bottom a string of symbols from which no branch emanates. Regard the set of all such strings corresponding to all derived P-markers as constituting language L_D. It has been shown by Hall[*] that, given the original transformational grammar G_T, one can automatically construct a context-free grammar G_S which accepts all the strings in L_D and assigns the corresponding derived P-markers to them. It is generally the case, however, that G_S accepts nonsentences in L_D as well as sentences in L_D, and also assigns some incorrect P-markers, as well as the correct one(s), to sentences in L_D.[**]

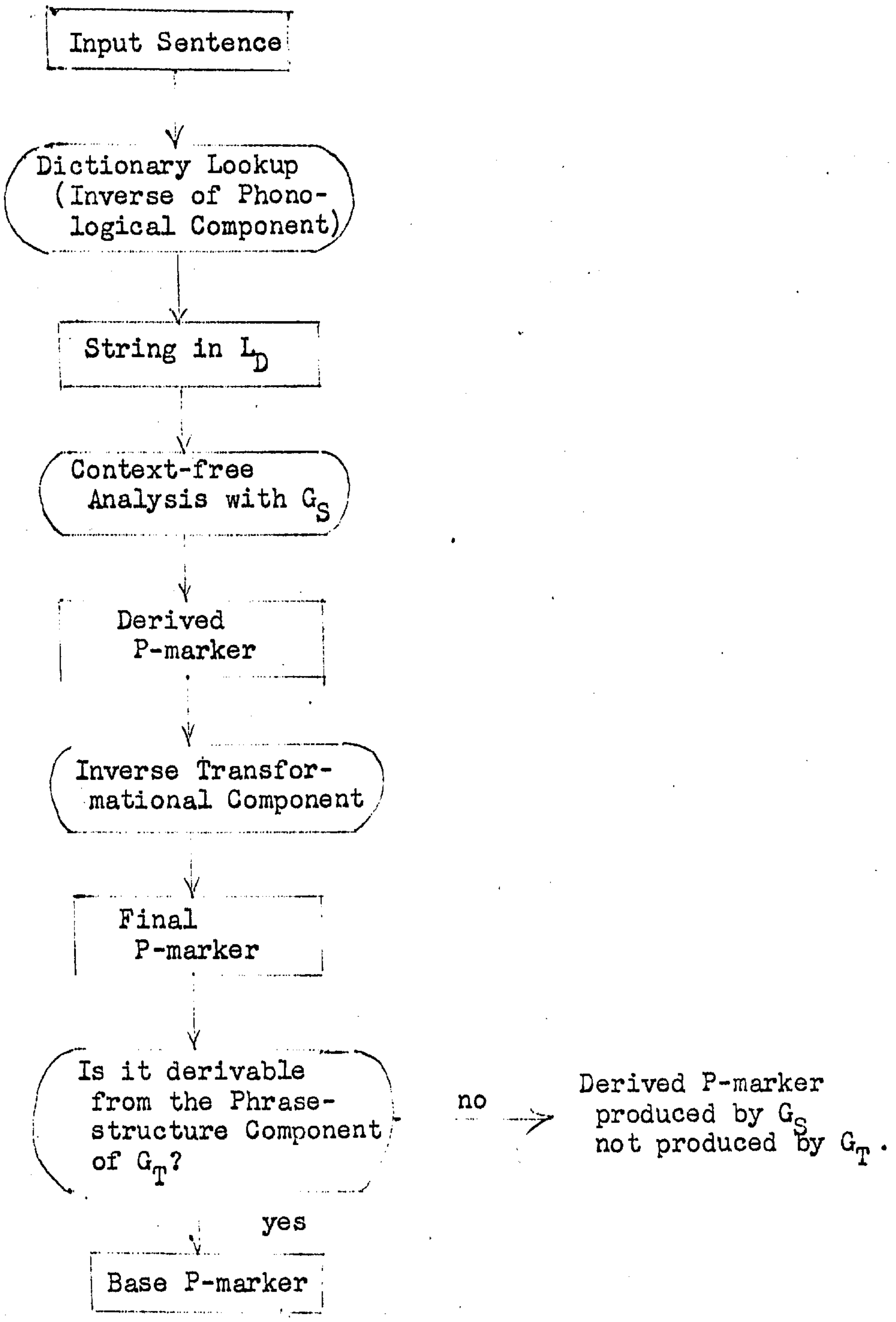
The analysis procedure works as follows (see Diagram 2).[***] Given a sentence in $L(G_T)$, the dictionary lookup program, which essentially plays the role of the inverse of a phonological component, converts the sentence into a string in L_D. A context-free analyzer with grammar G_S assigns one (or more if the string is ambiguous in G_S) derived P-marker(s) to the string. Then, each such P-marker is transferred to the inverse transformational component of G_T. A test is made to see which of the transformational rules could have been applied to map some previous P-marker into the current

[*] Private communication. The author is greatly indebted to Barbara C. Hall, who read a preliminary draft of this paper and gave him numerous valuable suggestions.

[**] Actually, the context-free grammars for derived P-markers in both Petrick's and the MITRE group's systems have been manually compiled. Hall's automatic procedure does not guarantee an optimal context-free grammar for derived P-markers of a given transformational grammar.

[***] The analysis procedure described here is that of the MITRE group, with some simplifications for the sake of clarity of explanation. Petrick's procedure is conceptually similar to, but actually deviates significantly from, the model described here.

Analysis Phase of $L(G_T)$ Recognizer


Input Sentence
Dictionary Lookup (Inverse of Phonological Component)
String in L_D
Context-free Analysis with G_S
Derived P-marker
Inverse Transformational Component
Final P-marker
Is it derivable from the Phrase-structure Component of G_T?
no
Derived P-marker produced by G_S not produced by G_T.
yes
Base P-marker

Transformational Language Recognizer (2)

Diagram 2

P-marker in the course of generation of the given sentence. If a rule is found whose derived constituent structure index* matches the P-marker, the inverse of the structural change specified by the rule is applied to the P-marker, and a new P-marker is obtained which matches the original structural index* of the rule. If no more transformational rules can be applied inversely to the current P-marker, either the P-marker is a base P-marker, or the P-marker assigned by G_S was not a final derived P-marker assigned to any sentence by G_T. The latter case is due to the condition that G_S accepts nonsentences as well as sentences in L_D and can give incorrect P-markers to sentences that are in L_D. In order to identify whether the P-marker under consideration is a real base P-marker or not, a test has to be made to see if the P-marker is obtainable by the phrase-structure component of G_T. If not, the original derived P-marker, which initiated the inverse transformational analysis path, is abandoned. If it is obtainable, the forward application of the transformational rules which were inversely applied confirms that it is in fact the base P-marker of the sentence under analysis. The base P-marker, the set of inversely applied transformational rules, and phonological rules contained in the dictionary entries constitute the analysis of the input sentence.

*Each transformational rule contains a structural index and a derived constituent structure index. The former specifies the condition that a P-marker has to fulfill in order for the rule to be applied to it. The latter specifies the structure of the P-marker into which the original P-marker is to be mapped by the transformation.

3. A Predictive Analyzer and Transformational Analysis

The system of transformational analysis which is proposed below aims at obtaining a set of base P-markers almost simultaneously as a set of surface P-markers is obtained. Rules of the analytical context-free grammar for the system have associated with them information pertaining to the transformational histories of their own derivation. For example, assume that the base P-marker of "I met a young prince" in a given transformational grammar is the one shown in Fig. 1, and that the transformational component of the grammar maps this base P-marker into the derived P-marker by a sequence of four transformations:

Base P-marker: '#I met a # the prince was young # prince #

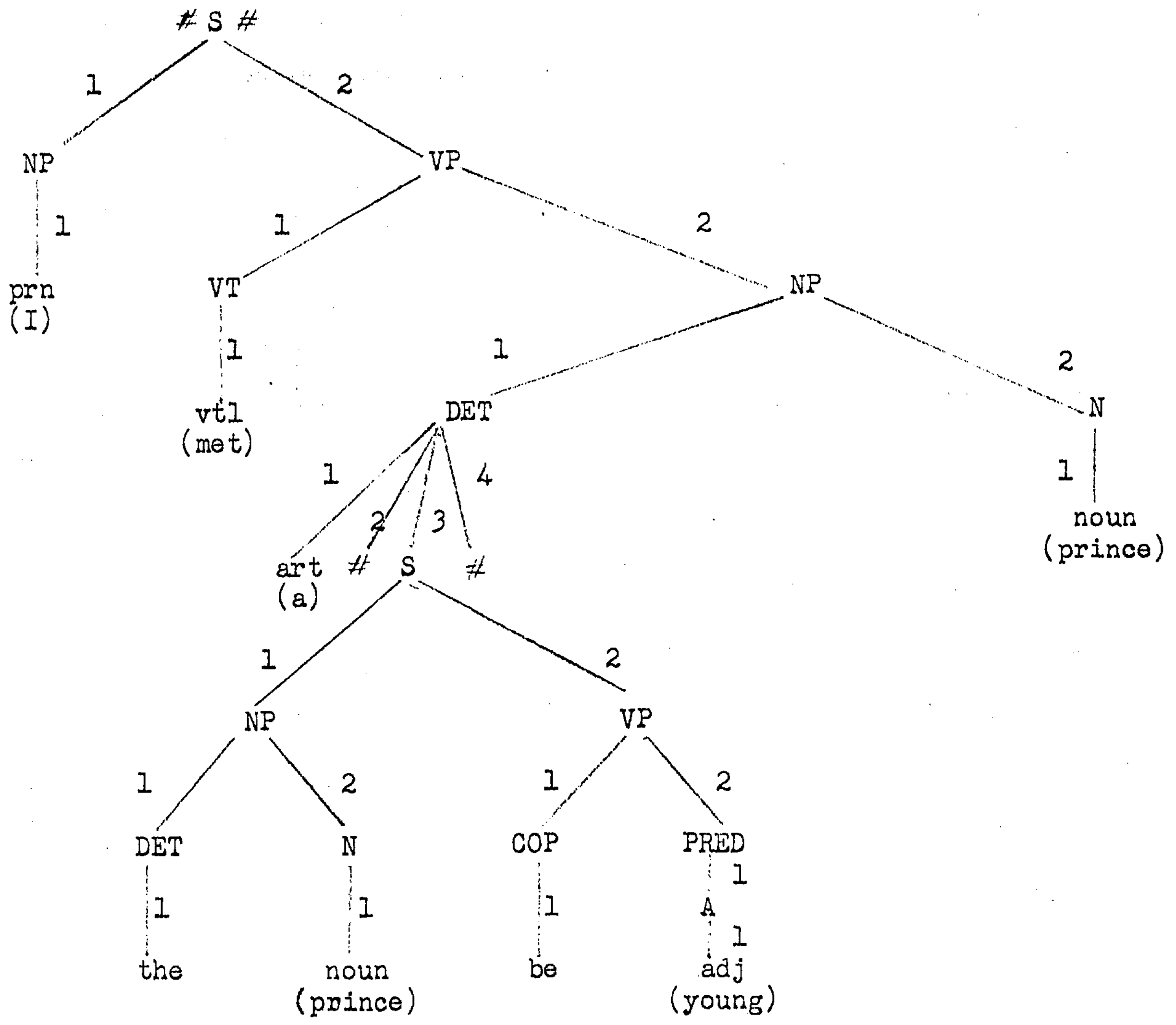
Intermediate P-marker: #I met a prince # the prince was young ##

Intermediate P-marker: #I met a prince who was young #

Intermediate P-marker: #I met a prince young#

Derived P-marker: #I met a young prince #

Then, the analytical context-free grammar for derived P-markers will have a rule which identifies a noun phrase consisting of an article (art), an adjective (adj), and a noun. To this rule, we can assign the information that the base P-marker image of this noun phrase is the subtree corresponding to "art # the noun be adj # noun" of Fig. 1. We can say that each such rule in the analytical context-free grammar draws a subtree of some base P-marker. When a derived P-marker of a sentence is obtained, the set of phrase-structure rules used for the analysis draws a set of subtrees which, when combined together, constitute the base P-marker corresponding to the derived P-marker.



Base P-marker for "I met a young prince."

Figure 1

The system is designed with the predictive analyzer[4,5] as its core. The predictive analyzer uses a predictive grammar G' whose rules (called "predictive rules") are of the following form:

$$\langle Z, c \rangle \mid Y_1 \cdots Y_m, \ m \geq 1$$

$$\langle Z, c \rangle \mid \lambda$$

where Z, Y_i are intermediate symbols (i.e., syntactic structures, also called predictions), c is a terminal symbol (i.e., syntactic word class) and λ denotes the absence of any symbol. $\langle Z, c \rangle$ is called an argument pair. $\langle SE, prn \rangle \mid VP \ PD$, for example, indicates that a sentence (SE) can be initiated by a prn (personal pronoun in the nominative case) if the prn is followed by a predicate (VP) and a period (PD). A fragment of our current English grammar is shown in Kuno and Oettinger.[4,5] It is proved by Greibach that G' is an exact inverse of a standard-form grammar G whose rules are of the form:

$$Z \longrightarrow cY_1 \cdots Y_m \text{ where } \langle Z, c \rangle \mid Y_1 \cdots Y_m \text{ is a rule in G', or}$$

$$Z \longrightarrow c \text{ where } \langle Z, c \rangle \mid \lambda \text{ is a rule in G'.}$$

Since Greibach has proved that every context-free language can be generated by a standard-form grammar, the predictive analyzer could accept any context-free language given a suitable predictive grammar.*

*Given a context-free grammar G'', we can automatically construct a standard-form grammar G which generates the same language as G'' does. However, it is to be noted that the structural descriptions assigned to a given sentence by G are not the same as those assigned to the same sentence by G''. In such a case, we say that G and G'' are weakly equivalent with respect to the structural description.

Consider a predictive grammar which does not contain more than one rule with the same argument pair, and an input string of words each of which is associated with a unique terminal symbol. The analysis of the sequence of terminal symbols $c_1 \cdots c_n$ is initiated with a pushdown store (PDS) containing some designated initial symbol ("SE" in the case of a natural language. See Fig. 2 for an example). At word k in the course of the analysis of the string, an argument pair $\langle Z_k, c_k \rangle$ is formed from the intermediate symbol Z_k topmost in the PDS and the current terminal symbol c_k. If a rule with this argument pair is not found in the grammar, the input string is ill-formed (ungrammatical). If it is found, we say that the prediction Z_k is <u>fulfilled</u> by the rule $\langle Z_k, c_k \rangle \mid Y_i \cdots Y_m$ (or $\langle Z_k, c_k \rangle \mid \lambda$), or simply that Z_k is fulfilled by c_k. A sequence of new intermediate symbols $Y_1 \cdots Y_m$ (or λ) then replaces the topmost intermediate symbol Z_k of the PDS and the analysis moves to word $k+1$. The input string is well-formed if the last terminal symbol c_n is processed yielding an empty PDS. A set of standard-form rules corresponding to the predictive rules used for the analysis of the string gives the derivational history of the string in the original standard-form grammar.

Actually, a grammar may have more than one rule with the same argument pair. Also, a word in an input string may be associated with more than one terminal symbol. Therefore, a mechanism for cycling through all possible combinations of these rules and terminal symbols must be superimposed on the simple pushdown store machine described in the previous paragraph. We are not concerned here, however, about how such a mechanism is designed in the current predictive analyzer (see Sec. 1 of Kuno[6] for the analysis algorithm). In the following discussions, only those analysis paths which lead to the end of the sentence are considered, and all abortive paths will be ignored in order to avoid unnecessary complications of the important question under discussion.

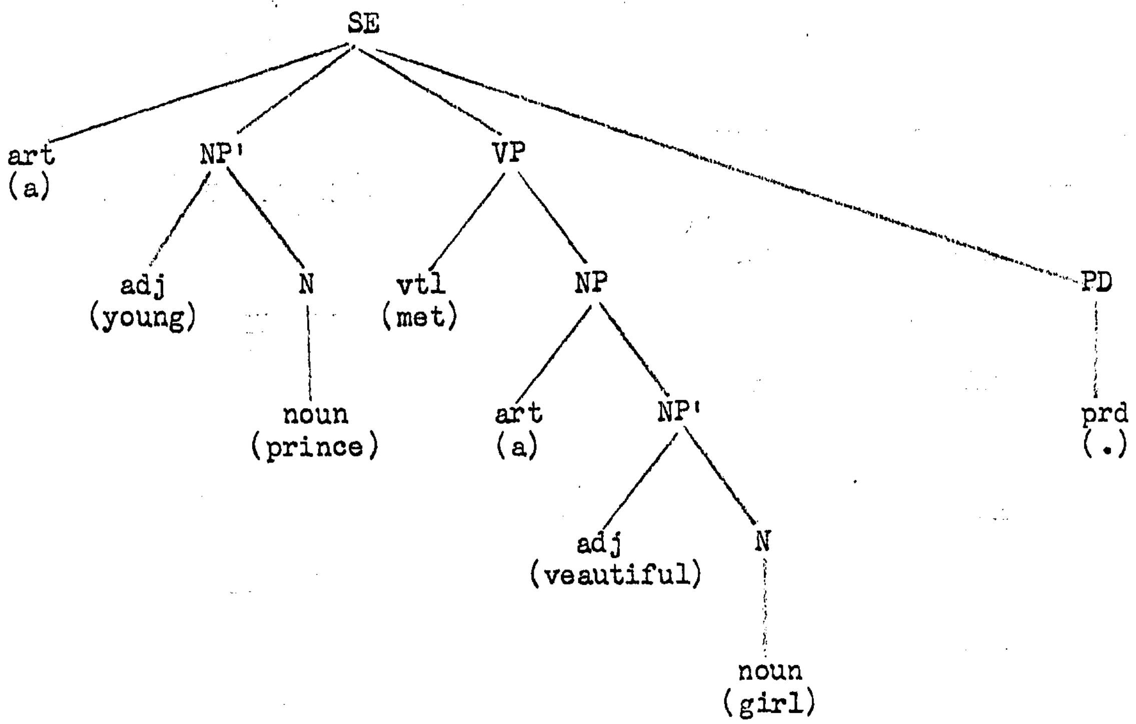
Assume that the input sentence "A young prince met a beautiful girl." is to be analyzed. Also assume that Rules 1 - 6 (see Fig. 2) have been used for the predictive analysis of the sentence. The configuration of the PDS prior to and immediately after the application of the rule at a given word position is shown in the preceding and succeeding lines of the column "PDS Configuration" of Fig. 2. The structural description (P-marker) assigned to this sentence by the set of standard-form rules corresponding to the utilized predictive rules is shown in Fig. 3.

Let us assume that the base P-marker that we want to have assigned to this sentence is not the one shown in Fig. 3, but the one in Fig. 4. Since a mapping of one P-marker into another P-marker involves shifting, removing, and adding of nodes in P-markers, it is important to have a device available to refer to any position in a P-marker. Names of branches in a P-marker are defined in the following way. If there are m branches emanating from a given node in a P-marker, the leftmost branch is named 1, the second leftmost branch 2, and so forth. The rightmost branch is named m (see Fig. 4). Given a node y in a P-marker, the branch number of y is obtained by the concatenation to the right of each successive number assigned to each successive branch which leads from the topmost node to node y. For example, the branch number of adj for "young" in Fig. 4 is 1211, the branch number of noun for "girl" is 22221, and so on. Similarly, if we are given

English Word	Rule Used	Argument Pair	New Predictions	PDS Configuration top <——— bottom
				SE
A	Rule 1	<SE, art>	NP' VP PD	NP' VP PD
young	Rule 2	<NP', adj>	N	N VP PD
prince	Rule 3	<N, noun>	λ	VP PD
met	Rule 4	<VP, vtl>	NP	NP PD
a	Rule 5	<NP, art>	NP'	NP' PD
beautiful	Rule 2	<NP', adj>	N	N PD
girl	Rule 3	<N, noun>	λ	PD
.	Rule 6	<PD, prd>	λ	λ

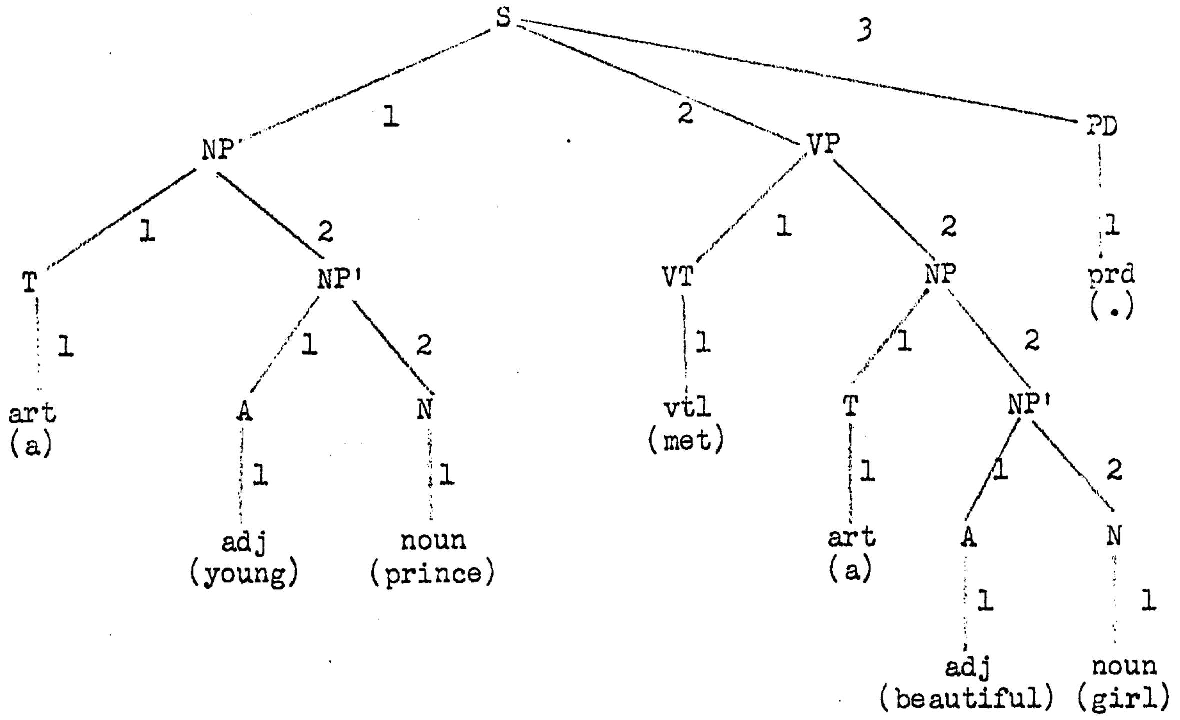
Predictive Analysis of a Sample Sentence

Figure 2



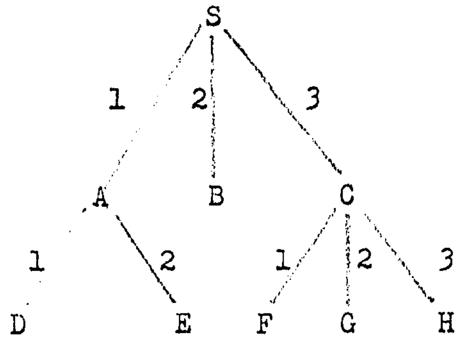
Structural Description Assigned by
the Predictive Analyzer

Figure 3



Desired Base P-marker

Figure 4



Ordered Pairs and a P-marker

Figure 5

a set of ordered pairs of (branch number, node) such as (1, A), (2, B), (3, C), (11, D), (12, E), (31, F), (32, G), (33, H), the P-marker shown in Fig. 5 can be automatically constructed given the initial symbol S.

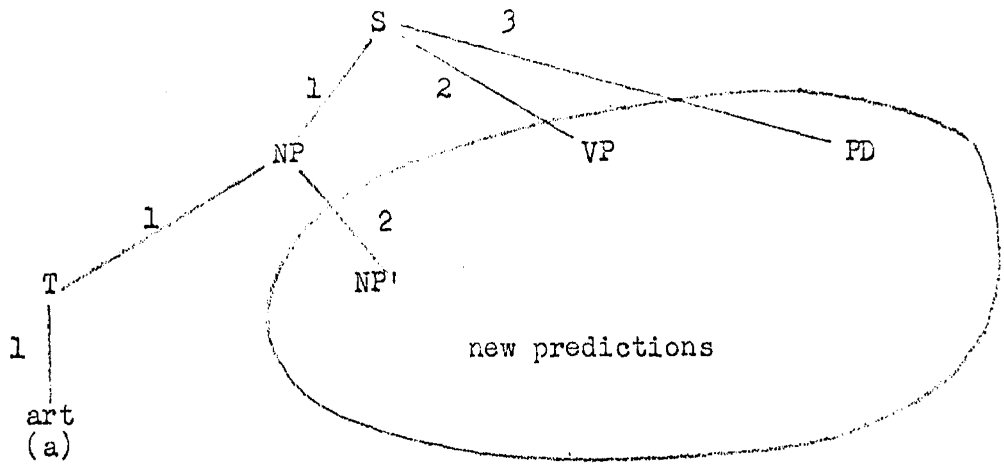
To each prediction in each rule of the predictive grammar is assigned a set of ordered pairs (x, y) where y indicates the name of a node and x the branch number of y in a P-marker. For example, Rule 1 will have the following sets of ordered pairs* assigned to its predictions:

Rule 1:	<SE, art>	NP'	VP	PD
	(1, NP) (11, T) (111, art)	(12, NP')	(2, VP)	(3, PD)

The set of ordered pairs assigned to the prediction of the argument pair in Rule 1 represents the names of nodes and branch numbers leading from the prediction of the argument pair to the final node "art". The set of ordered pairs associated with each new prediction shows the relationship of new predictions with the word class "art" of the argument pair (see Fig. 6). If in an ordered pair (x, y) associated with a prediction in a rule, y is not equal to the prediction itself (or to the word class of the argument pair in case the prediction is also in the argument pair), then the ordered pair plays the role of adding a new node y in a P-marker.

In the course of predictive analysis of a sentence, the set of ordered pairs associated with the argument pair's prediction is stored in

*In Rule 1, each of the new predictions NP', VP, and PD has a one-member set of ordered pairs. Examples of sets of more than one ordered pair will follow (e.g., Rule 3a).



Partial P-marker

Figure 6

the output work area. The set of ordered pairs associated with each new prediction is stored in the PDS together with the prediction.*

The branch number of an ordered pair in a rule does not have to be a constant as is the case with all the ordered pairs of Rule 1. For example, see Rule 2.

*The expression "argument pair's prediction" is used as distinct from the expression "fulfilled prediction". The former is prediction Z of $\langle Z, c \rangle$, while the latter refers to the prediction which is topmost in the PDS and fulfilled by the rule $\langle Z, c \rangle \mid Y_1 \cdots Y_m$ (or $\mid \lambda$). The fulfilled prediction was a new prediction of a rule which was used at some preceding word position, and has associated with it in the PDS a set of ordered pairs. Although the fulfilled prediction itself at a given word position is always the same as the argument pair's prediction of the rule used at the same word position, it is convenient to distinguish the two for our subsequent discussions because the set of ordered pairs associated with the fulfilled prediction in the PDS is different from the set of ordered pairs associated with the argument pair's prediction in the rule (see explanation of Rule 2).

Rule 2:

<NP', adj>	N
{(x, y)} (x1, A) (x11, adj)	(x2, N)

This rule is used for the processing of "young" and "beautiful" of the example "A young prince met a beautiful girl." (see Fig. 2). The branch number that the node NP' which dominates "young" is to receive is different from the branch number that the node NP' which dominates "beautiful" is to receive in the base P-marker. Since NP' can be a recursive symbol, there is no way of assigning all the possible branch numbers that NP' can be associated with in any finite number of rules. Instead, we use a variable x whose value is determined by the branch number of the immediately dominating node in a P-marker. The notation {(x, y)} is used to indicate that the prediction appearing above the notation is to be assigned the same set of ordered pairs as the fulfilled prediction used to have in the PDS. In our example, the first NP' ("young") has {(12, NP')} due to Rule 1 when it becomes topmost in the PDS. In the case of the second NP' ("beautiful"), it will be shown later that it has {(222, NP')}.

Similarly, the branch number that the node adj for "young" is to receive in a base P-marker is different from the branch number that the node adj for "beautiful" is to receive. In fact, each of the two branch numbers depends upon the branch number which its respective immediately dominating node NP' is associated with (see Fig. 4). Yet, if NP' is to be regarded as the initial node, the branch numbers to be associated with A and adj for "young" and N and noun for "prince" are exactly the same as those to be associated with A and adj for "beautiful" and N and noun for "girl", respectively. Therefore, in Rule 2, the branch numbers dominated by NP' are given as constants, and branch numbers emanating from the initial symbol and leading to NP' are given as variables. The notation (x1, A), for example, indicates that whatever the branch number from the initial node to NP' might be, A is to receive 1 as the rightmost digit for the entire branch number from the initial node to A. It is to be noted that ordered pairs with variables appear only in rules in the grammar whose argument pairs do not contain the initial prediction SE. Once a rule is used for the analysis of a sentence, all the variables for branch numbers in the set of ordered pairs associated with this rule will be changed into some numerical branch numbers.

In general, $(x^\frown m, Z)$ $(m \geq 1)$ not in a pair of braces indicates the following:

> Take the maximum value* of branch numbers (max x) in $\{(x, y)\}$ of the fulfilled prediction. (Remember that the branch numbers of ordered pairs associated with the fulfilled prediction are all numerical, and do not contain any variables. Regard numeric branch numbers as integers to obtain the "maximum value".)
>
> Concatenate m to the right of max x.
>
> Form an ordered pair with Z.

The concatenation mark is suppressed where no confusion can result. When $(x^\frown m, Z)$ appears in a pair of braces, all the values of x in $\{(x, y)\}$ of the fulfilled prediction, not the maximum value, are used to form a set of new ordered pairs with m concatenated to the right of each value of x (see Rule 5a for example).

*Why max x is used among values of x in $\{(x, y)\}$ will be explained in Sec. 5.

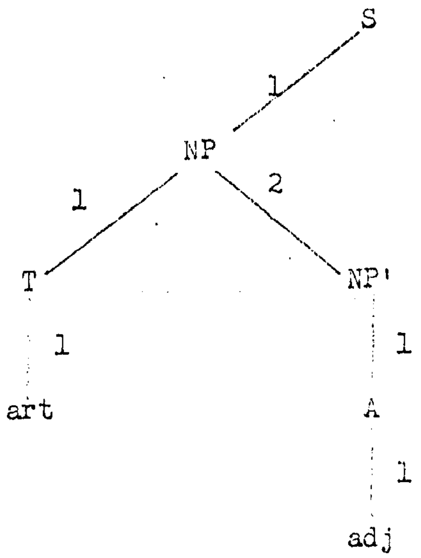
Ordered pairs with variables (x, y), $(x^\frown m, y)$ can be regarded as a notation for some function whose value depends upon the previously obtained value of the same function. It is this recursive nature of ordered pairs in the grammar that allows the proposed system to work for an infinite number of sentences in the language.

In the case under discussion, the fulfilled prediction NP' corresponding to "young" has {(12, NP')} associated with it in the PDS. Therefore, max x = 12. So, (x1, A) and (x11, adj) are changed into $(12^\frown 1, A)$ and $(12^\frown 11, adj)$, respectively, and the latter two are stored in the output work area. As explained in the previous paragraph, {(x, y)} associated with the argument pair's prediction is replaced by (12, NP'), which also is stored in the output work area. The new prediction N of Rule 2 is assigned the ordered pair $(12^\frown 2, N)$. N, (122, N) replaces the fulfilled prediction NP' and its ordered pair (12, NP') in the PDS. Now the output work area contains (1, NP), (11, T), (111, art) due to Rule 1 and (12, NP'), (121, A), (1211, adj) due to Rule 2. This set of ordered pairs corresponds to a partial P-marker shown in Fig. 7.

Rule 3 is shown below with ordered pairs:

$$\text{Rule 3:}\quad \begin{array}{l|l} \underline{\langle N,\ noun\rangle} & \underline{\lambda\qquad} \\ \{(x,\ y)\} & \\ (x1,\ noun) & \end{array}$$

When Rule 3 is used for the processing of the third word "prince" of the example, the fulfilled prediction has associated with it the ordered pair (122, N). Therefore, (122, N) and $(122^\frown 1, noun)$ are stored in the output



Partial P-marker Constructed

Figure 7

work area. Rules 4-6 are shown below in the new form; Fig. 8 shows the analysis of the same sentence using the new rules.

Rule 4:	<VP, vt1>	NP
	{(x, y)} (x1, VT) (x11, vt1)	(x2, NP)
Rule 5:	<NP, art>	NP'
	{(x, y)} (x1, T) (x11, art)	(x2, NP')
Rule 6:	<PD, prd>	λ
	{(x, y)} (x1, prd)	

It is to be noted that the set of ordered pairs in the output work area in Fig. 8 is isomorphic to the P-marker shown in Fig. 4.

Let us go back to the transformational grammar previously mentioned which assigns the base P-marker of Fig. 1 to "I met a young prince.".

English Word	Argument Pair	Contribution to Output Work Area	PDS Configuration top <- bottom
A	<SE, art>	(1,NP),(11,T),(111,art)	SE λ
young	<NP', adj>	(12,NP'),(121,A),(1211,adj)	NP' (12,NP') VP (2,VP) PD (3,PD)
prince	<N, noun>	(122,N),(1221,noun)	N (122,N) VP (2,VP) PD (3,PD)
met	<VP, vtl>	(2,VP),(21,VT),(211,vtl)	VP (2,VP) PD (3,PD)
a	<NP, art>	(22,NP),(221,T),(2211,art)	NP (22,NP) PD (3,PD)
beautiful	<NP', adj>	(222,NP'),(2221,A),(22211,adj)	NP' (222,NP') PD (3,PD)
girl	<N, noun>	(2222,N),(22221,noun)	N (2222,N) PD (3,PD)
.	<PD, prd>	(3,PD),(31,prd)	PD (3,PD) λ

Analysis of the Sample Sentence

Figure 8

The following set of rules, in the framework of the same mechanism as was introduced above, can give the desired base P-marker.

Rule 1a:

⟨SE, prn⟩	VP	PD
(1, NP) (11, prn)	(2, VP)	λ[*]

Rule 2a:

⟨VP, vtl⟩	NP
{(x, y)} (x1, VT) (x11, vtl)	(x2, NP)

Rule 3a:

⟨NP, art⟩	A	N
{(x, y)} (x1, DET) (x11, art) (x12, #) (x13, S) (x14, #) (x131, NP) (x1311, DET) (x13111, the) (x132, VP) (x1321, COP) (x13211, be) (x1322, PRED)	(x13221, A)	(x2, N) (x1312, N)

Rule 4a:

⟨A, adj⟩	λ
{(x, y)} (x1, adj)	

[*] λ associated with a prediction in a rule performs the function of eliminating the node for the prediction from a P-marker.

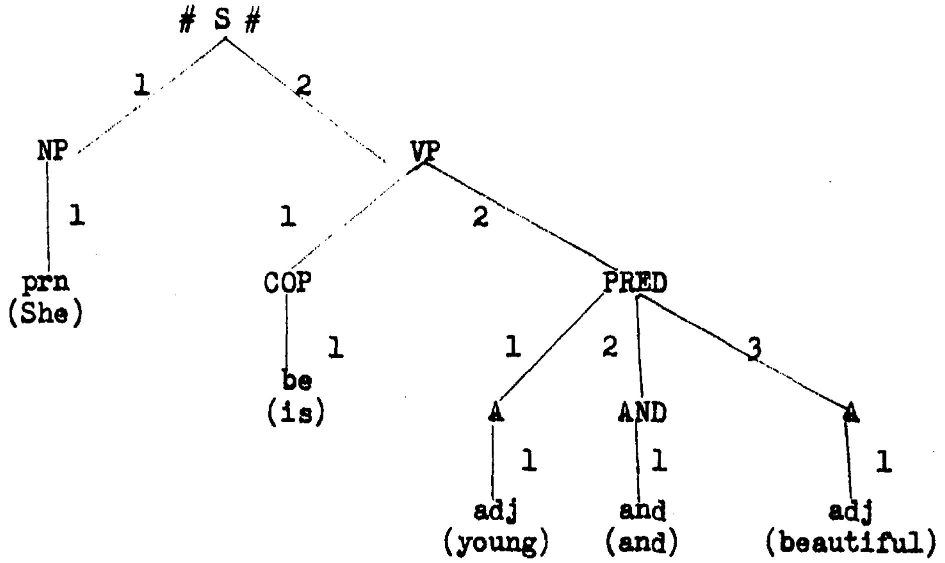
$$\text{Rule 5a:}\quad \begin{array}{c|c} \langle N, \text{noun}\rangle & \lambda \\ \hline \{(x,\ y)\} & \\ \{(x1,\ \text{noun})\} & \end{array}$$

$$\text{Rule 6a:}\quad \begin{array}{c|c} \langle PD, \text{prd}\rangle & \lambda \\ \hline \lambda & \end{array}$$

The argument pair's prediction N of Rule 5a has associated with it a set of ordered pairs $\{(x1, \text{noun})\}$. It is to be noted that x in $\{(xm, Z)\}$ represents all the values of x in $\{(x, y)\}$ of the fulfilled prediction, not the maximum value of x (see the explanation of Rule 2). When Rule 5a is used to process "prince," the fulfilled prediction N has associated with it ordered pairs (222, N) and (221312, N). Therefore, $\{(x1, \text{noun})\}$ is changed into (2221, noun) and (2213121, noun).

In comparing Rule 3a, for example, with Fig. 1, one may wonder why (x2, N) and (x1312, N) are associated with the new prediction N, and not with the argument pair's prediction NP. If the latter alternative were chosen, N would have no ordered pairs in Rule 3a. Then, when Rule 5a is used for the processing of "prince," there would be no way of obtaining desired branch numbers for the noun in $\{(x1, \text{noun})\}$.

The concatenation operation $x^\frown m$ introduced in the previous paragraphs is not enough to deal with coordinate structures. Assume that the base P-marker of Fig. 9 is to be assigned to "She is young and beautiful.".



Base P-marker for "She is young and beautiful."

Figure 9

Rule 7:

<PRED, adj>	AND	A
{(x, y)} (x1, A) (x11, adj)	(x2, AND)	(x3, A)

Rule 8:

<AND, and>	λ
{(x, y)}	

Rule 9:

<A, adj>	λ
{(x, y)} (x1, adj)	

Rule 7 is capable of assigning numbers 1, 2, and 3 to the three branches emanating from PRED and leading to A, AND, and A, respectively.

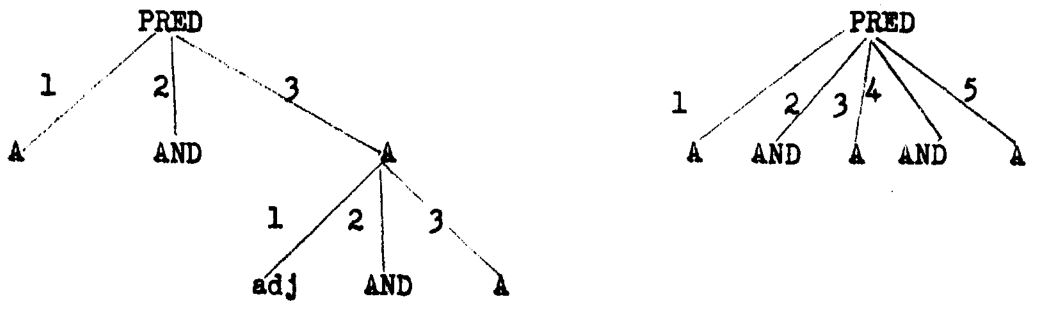
However, if the predicate has three adjectives "young and beautiful and intelligent," the inadequacy of a context-free grammar manifests itself. The P-marker that we want to obtain is not that of Fig. 10(a), but of Fig. 10(b). Yet, we cannot include in the predictive grammar a rule such as

⟨PRED, adj⟩	AND	A	AND	A
{(x, y)} (x1, A) (x11, adj)	(x2, AND)	(x3, A)	(x4, AND)	(x5, A)

because we will face the same problem for coordinate predicates with more than three coordinated members, and because we cannot have an infinite number of rules pertaining to i-member coordinate structures where $i = 2,3,\ldots,\infty$.[*]

In order to obtain P-markers of the type shown in Fig. 10(b) with a finite set of rules, a new operation "+" is introduced. If a prediction in a rule has (x+m, u), max x is chosen among the values of x of {(x, y)} associated with the fulfilled prediction: m is numerically

[*] Actually, the difficulty under discussion is not only of a context-free analyzer, but also of the phrase-structure component of a transformational grammar. A base P-marker of the type shown in Fig. 10(b) cannot be obtained by any phrase-structure grammar if an infinite number of coordinated members is to be accounted for. One solution for a transformational <u>generative</u> grammar is to have in its phrase-structure component a rewriting schema such as PRED ⟶ A (AND A)*, where (AND A)* can be repeated any number of times (including zero). This is done in the MITRE procedure in both the generative phrase structure component of G_T and the context-free analysis component G_S. In the generative component, the only starred rule is S' ⟶ S (AND S)*; in the recognition component, all compoundable intermediate symbols have rules of this type.



(a)

(b)

Base P-markers for Coordinate Structures
Figure 10

added to the rightmost position of max x. (If more than nine constituents are to be accepted in a construct, it is necessary to use more than one digit for the name of each branch, but this does not cause any additional complexities.) For example, when the second adjective "beautiful" of the example "She is young and beautiful and intelligent." fulfills the prediction A, Rule 10 is used:

$$\text{Rule 10:}\quad \begin{array}{l|l|l} \langle \text{A, adj}\rangle & \text{AND} & \text{A} \\ \hline \begin{matrix}\{(\text{x, y})\} \\ (\text{x1, adj})\end{matrix} & (\text{x+1, AND}) & (\text{x+2, A}) \end{array}$$

{(x, y)} for the fulfilled prediction is (223, A); therefore, (x+1, AND) and (x+2, A) are changed into (224, AND) and (225, A), respectively (224 = 223 + 1, 225 = 223 + 2), and are stored in the PDS with the corresponding predictions AND and A. If the predicate has four adjectives as in "young and beautiful and intelligent and bright," Rule 10 will be used again for the processing of "intelligent." This time, max x = 225. Therefore, new predictions AND and A will be stored in the PDS with the new ordered pairs (226, AND) and (227, A), respectively.

It should now be noted that the concatenation operation $x^\frown m$ plays the role of generating a subtree whose initial node has the branch number max x, while x+ m plays the role of adding a branch to the right of a branch whose branch number is x, and whose immediately dominating node also dominates the added branch.

4. Salient Features of the Proposed System for Transformational Analysis

What are the salient differences between the transformational analysis system (see Sec. 2(ii) of this paper) proposed by the MITRE group and Petrick (to be referred to as M-P system) and the one proposed in the present paper (to be referred to as K-system)? The M-P system is based on the condition that a transformational grammar is given. A context-free analysis component is automatically constructed* on the basis of the transformational grammar; the context-free analysis component assigns one or more derived P-markers to a sentence to be analyzed; transformational rules are applied inversely to each P-marker step by step until the base P-markers of the sentence are obtained. For example, after a derived P-marker is assigned to "He met a beautiful girl.", the M-P system will compare the P-marker with the derived

*See the second footnote on page 4 .

constituent structure indices of transformational rules, and find that this derived P-marker is the result of the transformational rule which places an adjective in front of a noun. Therefore, by applying this rule inversely, an intermediate P-marker corresponding to "#He met a girl beautiful#" is obtained. Next, this new P-marker is compared with derived constituent structure of transformational rules, and it is found that this is the result of the transformational rule which deletes a relative pronoun and a copula. Therefore, by applying this rule inversely, an intermediate P-marker corresponding to "#He met a girl who was beautiful#" is obtained. Next, this intermediate P-marker is compared with the derived constituent structure indices of transformational rules again and is identified as being the result of a relativization rule. Therefore, the rule is applied inversely, and a new P-marker corresponding to "#He met a girl # the girl was beautiful#" is obtained, which in turn is identified as originating from a rule which places an embedded #S# dominated by DET after the noun. A new P-marker corresponding to "#He met a # the girl was beautiful #girl#" is thus obtained. After comparing this P-marker again with rules in the transformational component, it is found that there is no rule whose derived constituent structure index matches the P-marker. It is also found that the P-marker is derivable from the phrase-structure component of the transformational grammar. Thus, the P-marker is identified as being a base P-marker, and forward application of the transformations which were inversely applied confirms that it is in fact the base P-marker of the sentence under analysis.

With regard to the K system, on the other hand, a predictive grammar which accepts all the sentences of a given transformational grammar G_T (and probably nonsentences in addition) is manually compiled. A derived P-marker assigned to a given sentence by the predictive grammar is usually not equal to the derived P-marker which is assigned to the same sentence by G_T. The mapping of such a distorted P-marker into the base P-marker is not performed step by step through intermediate P-markers as is the case with the M-P system. Instead, it is performed in one step by means of ordered pairs. For example, the fact that the predictive rule

$$\langle NP, art \rangle \mid A \quad N$$

has been used for assigning a distorted P-marker to the sentence "He met a beautiful girl." indicates immediately that an embedded sentence which constitutes a relative clause is involved here, that the subject of the embedded sentence is the same as a noun ("girl" in our example) which fulfills N of the predictive rule, and that the adjective ("beautiful") which fulfills A is the predicate adjective of the embedded sentence. The predictive rule has associated with it a set of ordered pairs which draws a subtree of the base P-marker image of this NP. The summation of such subtrees drawn by all the rules used for obtaining the distorted P-maker yields the base P-maker of the sentence.

The K system does not achieve this one-step mapping without cost. The sacrifice is paid in the simplicity of the context-free analysis component. For example, in order to obtain desired base P-markers for

(i) Look at the girl who is dancing the mazurka.

(ii) This is the girl whom everyone likes.

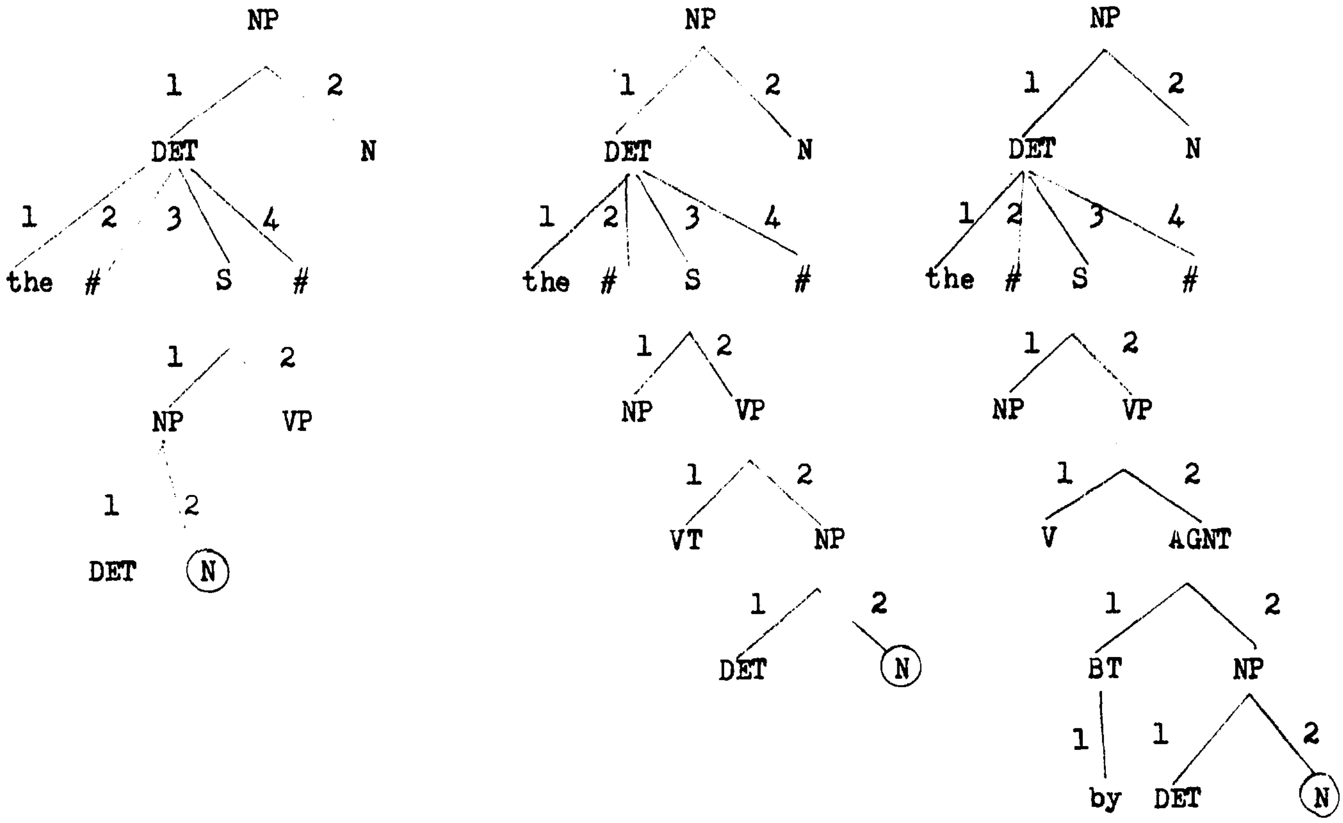
(iii) This is the girl by whom he was ruined.

the predictive grammar must have three different rules pertaining to a noun phrase initiated by the definite article "the." Each rule specifies a different position, in the embedded sentence, of the predicted N (see circled N's in Fig. 11).

Rule (i):	⟨NP, the⟩	N	REL_{sbj}
	{(x, y)} (x1, DET) (x11, the) (x12, #) (x13, S) (x14, #) (x131, NP) (x1311, DET) (x13111, the)	(x2, N) (x1312, N)	(x13, R)

Rule (i-a)	⟨REL_{sbj}, who⟩	VP
	λ	(x2, VP)

Rule (ii)	⟨NP, the⟩	N	REL_{obj}
	{(x, y)} (x1, DET) (x11, the) (x12, #) (x13, S) (x14, #) (x1322, NP) (x13221, DET) (x132211, the)	(x2, N) (x13222, N)	(x13, R)



for sentence (i)　　　　for sentence (ii)　　　　for sentence (iii)

Position of Predicted N in Self-embedded Sentence

Figure 11

Rule (ii-a)	⟨REL$_{obj}$, whom⟩	NP	VP'
	λ	(x1, NP)	(x2, VP)

Rule (ii-b)	⟨VP', vt1⟩	λ
	{(x, y)} (x1, VT) (x11, vt1)	

Rule (iii)

⟨NP, the⟩	N	REL_{pass}
{(x, y)} (x1, DET) (x11, the) (x12, #) (x13, S) (x14, #) (x222, NP) (x2221, DET) (x22211, the)	(x2, N) (x2222, N)	(x13, R)

Rule (iii-a)

⟨REL_{pass}, by⟩	WHOM	NP	VP_{pass}
(x22, AGNT) (x221, BY) (x2211, by)	λ	(x1, NP)	(x2, VP) (x21, V)

Moreover, in order to deal with sentences such as

(iv) Look at the girl dancing the mazurka.

(v) Look at the dancing girl.

(vi) This is the girl liked by everybody.

additional rules have to be recognized which have the same argument pair ⟨NP, the⟩ but which have different sequences of new predictions and the different sets of ordered pairs from those in Rules (i), (ii) and (iii). Depending upon the nature of the original transformational grammar G_T, the number of such rules with the same argument pair can become very large. However, when a given sentence with a noun phrase is analyzed, only one of these rules will lead to the end of the sentence (unless the sentence is ambiguous with respect to the noun phrase), and all the other rules of ⟨NP, the⟩

will come to an impasse before the end of the noun phrase is reached. Moreover, once an analysis of the sentence is obtained, the derived P-marker can be unambiguously mapped into the corresponding base P-marker.

5. Practical Applications

The mechanism introduced in Sec. 3 for transformational analysis is quite effective for obtaining pairs (or triples, etc.) of words which are in certain syntactic relationships in a sentence. Assume that "The young prince made the beautiful girl his wife." is to be analyzed and that we are interested in obtaining word-triples "prince - made - girl," "prince - (be) - young," "girl - (be) - wife," and "girl - (be) - beautiful." We can achieve this aim by the following set of rules:

Rule 1':

<SE, the>	NP'	VP	PD
λ	(1, z)	(2, z)	λ

Rule 2':

<NP', adj>	N
(x3, z) (x2, be)	{(x, y)} (x1, z)

Rule 3':

<N, noun>	λ
{(x, y)}	

Rule 4':

<VP, vt3>	NP	NP
{(x, y)} ((x+1)2, be)	(x+1, z) ((x+1)1, z)	((x+1)3, z)

Rule 5':

<NP, the>	NP'
λ	{(x, y)}

Rule 5'a:

<NP, the>	N
λ	{(x, y)}

Rule 6':

<PD, prd>	λ
λ	

"z" as the second coordinate of an ordered pair means that when the ordered pair is stored in the work area (not in the PDS), z should be changed into whatever word form has fulfilled the prediction. For example, the second word "young" of the sentence is processed with Rule 2', which has two ordered pairs (x2, be) and (x3, z) associated with the argument pair's prediction NP'. NP' in the PDS has (1, z) due to Rule 1. Therefore, max x = 1, and z = young. So, (12, be) and (13, young) are stored in the output work area.

When the fourth word "made" is processed with Rule 4', the fulfilled prediction VP has (2, z) associated with it in the PDS. Therefore, max x = 2. Ordered pair ((x+1)2, be) indicates that 1 is to be numerically added to max x, and 2 is to be concatenated to the right of the sum. Therefore, $((x+1)^\frown 2$, be) = $((2+1)^\frown 2$, be) = (32, be) is obtained, which is stored in the output word area as well as (2, made) obtained from (2, z). In the same way, the two sets of ordered pairs for the two new predictions of Rule 4' will be changed into:

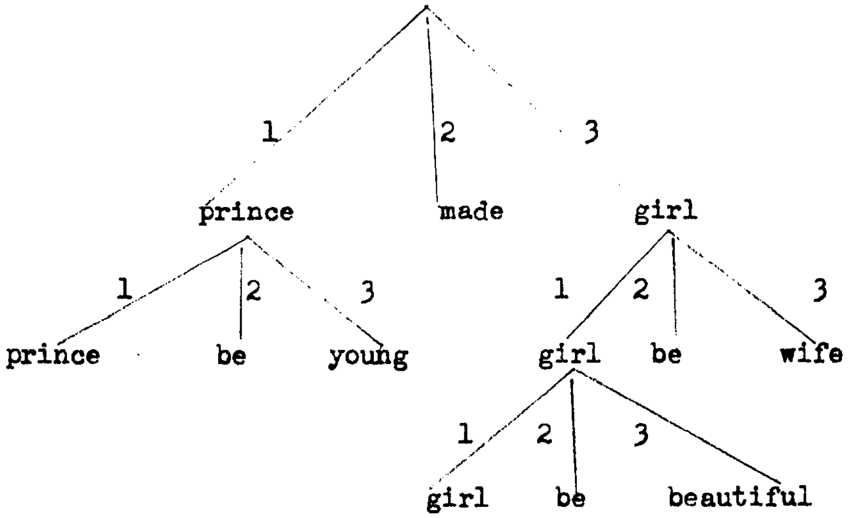
NP	N
(3, z) (31, z)	(33, z)

When Rule 5' is used for the processing of the fifth word "the," the fulfilled prediction NP has associated with it two ordered pairs (3, z) and (31, z). The argument pair's prediction has no ordered pairs; the new prediction NP' is assigned the same set of ordered pairs as was assigned to the fulfilled prediction NP. Therefore, when Rule 2' is used for the processing of the sixth word "beautiful," the fulfilled prediction has ordered pairs (3, z) and (31, z). Max x is equal to 31. Therefore, (x2, be) and (x3, z) are changed to (312, be) and (313, beautiful), respectively, which are then stored in the output work area. The new prediction N is assigned (3, z), (31, z), and (311, z) due to the set of ordered pairs $\{(x, y)\}$ and (x1, z) of the prediction. The reason that max x is to be used among all the values of x in $\{(x, y)\}$ is that, whatever the branch number of the noun ("girl") which fulfills N may be, we want to have the word triple corresponding to N ("girl") - be - adj ("beautiful") emanate as the lowest-order subtree dependent upon the lowest-order occurrence of N ("girl"). Otherwise, the branch numbers of N ("girl"), be, adj ("beautiful") would be confused with branch numbers of N ("girl"), be, N ("wife") (see Fig. 12).

When the analysis of the sentence is obtained, the ordered pairs (with no variable component in the branch number) in the output work area are sorted with the right-adjusted branch numbers as the sorting key. The result of the sorting is:

```
(  1, prince)
(  2, made)
(  3, girl)
( 11, prince)
( 12, be)
( 13, young)
( 31, girl)
( 32, be)
( 33, wife)
(311, girl)
(312, be)
(313, beautiful)
```

Each set of ordered pairs whose branch numbers differ from each other only at the rightmost position forms a word pair (or triple, etc.). The set of all the ordered pairs can also be regarded as constituting a tree of the structured information shown in Fig. 12.



Kernel Sentences for the Sample Sentence

Figure 12

Observe that the addition operation of "x+m," which was introduced originally to deal with coordinated structures (see Sec. 3), has been used for a different purpose in Rule 4'. The first of the two new NP predictions in Rule 4' has associated with it the ordered pair (x+1, z). This places the NP (which is eventually fulfilled by "girl") on the same level in a tree as the prediction VP which has been fulfilled by "made."

When P-markers of the type shown in Fig. 12 are desired, neither the addition operation nor the concatenation operation is satisfactory in dealing with sentences with coordinate structures, for which a new device has to be introduced. Assume that the sentence to be analyzed is "He met Mary and Jane and Karen.", and that three word-triples

he - met - Mary
he - met - Jane
he - met - Karen

are to be identified in the sentence. In order to accomplish this object, the notion of a decimal point is used. The notation x.m in an ordered pair indicates that m should be concatenated to the right of x as the rightmost fraction digit. For example, if x = 32.3 and m = 1, x.m = $32.3^\frown 1$ = 32.31. If x = 3, and m = 1, x.m = 3.1. The concatenation and addition operations described in Sec. 3 are performed on the units digit of a given branch number. For example, if x = 32.3 and m = 1,

$x \frown m = (32 \frown 1).3 = 321.3$; and $x+m = (32+1).3 = 33.3$. As is the case with $x \frown m$ and $x+m$, x usually indicates the maximum value of x in the set of ordered pairs of the fulfilled prediction. However, $\{(x.m, y)\}$ indicates that all the ordered pairs associated with the fulfilled prediction should be assigned to the corresponding prediction with a fraction digit m concatenated to the right of each branch number (see Rule 13 for an example).

Rule 11:	<SE, prn>	VP	PD
	(1, z)	(2, z)	λ

Rule 12:	<VP, vt1>	NP
	$\{(x, y)\}$	(x+1, z)

Rule 13:	<NP, noun>	AND	NP
	$\{(x, y)\}$	λ	$\{(x.k, y)\}$

Rule 14:	<NP, noun>	λ
	$\{(x, y)\}$	

The fraction digit to be concatenated can be a variable itself. The variable "k" in (x.k, y) stands for the units digit of max x. For example,

if x = 13 , then k = 3 and $x.k = 13.\frown 3 = 13.3$
if x = 13.21, then k = 3 and $x.k = 13.21 \frown 3. = 13.213$

Similarly, $\{(x.k, y)\}$ in Rule 13 indicates the same operation should be performed for each x of the set of ordered pairs $\{(x, y)\}$. Whenever the fraction variable k appears in a rule utilized at a given word position, the following modification of the contents of the output work area and the PDS is performed: for each (x.k, y), look for ordered pairs (in the output work area or PDS) whose branch number is different from (x, y) only with regard to the units digit. For each such pair in the output work area or PDS, form a new ordered pair by concatenating the value of k as a fraction digit to the right of its branch number. Store the new ordered pair in the work area or PDS, respectively.

For example, when "Mary" of "He met Mary and Jane and Karen." is processed with Rule 13, the fulfilled prediction NP has associated with it the ordered pair (3, z). Therefore, k is set to 3, and $\{(x.k, y)\}$ for the new prediction NP is changed to (3.3, z). At this point, the search is made in the output work area and the PDS (see Fig. 13) for ordered pairs whose branch number is different from "3" only with regard to the units digit. Ordered pairs (1, he) and (2, met) in the output work area satisfies the stated condition. Therefore, new ordered pairs (1.k, he) = (1.3, he) and (2.k, met) = (2.3, met) are formed, and are stored in the output work area.

When the second noun "Jane" is fulfilled, again with Rule 13, the fulfilled prediction NP has associated with it the ordered pair (3.3, z). Therefore, k is set to 3, and $\{(x.k, y)\}$ for the new prediction NP is changed to (3.3k, Jane) = (3.33, Jane), which is stored in the PDS with NP. The search is made for ordered pairs whose branch number is different from 3.3 only with regard to the units digit. This time, the

Output Work Area	PDS Configuration
(1, he)	PD
(2, met)	λ

Contents of Output Work Area and the PDS at "Mary"
Figure 13

output work area and the PDS contain the ordered pairs shown in Fig. 14. Ordered pairs (1.3, he) and (2.3, met) satisfies the stated condition; therefore, new ordered pairs (1.33, he) and (2.33, met) are formed and stored in the output work area. The third noun "Karen" is processed with Rule 14. Since Rule 14 does not contain any ordered pairs whose branch number is of the form x.k,

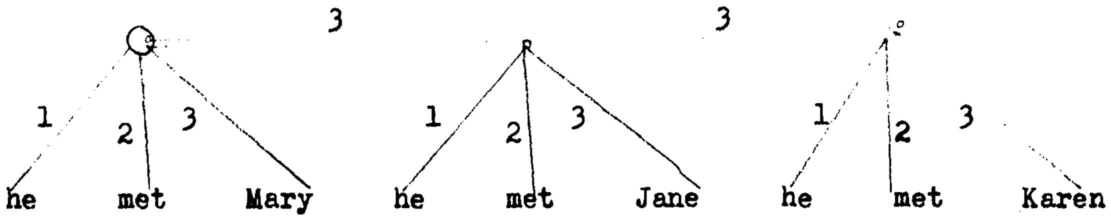
Output Work Area	PDS Configuration
(1, he)	PD
(2, met)	λ
(3, Mary)	
(1.3, he)	
(2.3, met)	

Contents of Output Work Area and the PDS at "Jane"
Figure 14

no modification of the contents of the output work area or PDS is performed. After the processing of the period, the output work area contains the following set of ordered pairs:

(1, he)
(2, met)
(3, Mary)
(1.3, he)
(2.3, met)
(3.3, Jane)
(1.33, he)
(2.33, met)
(3.33, Karen)

The ordered pairs are sorted first on left-adjusted decimal part, and then on right-adjusted integral part of the branch numbers. A set of ordered pairs whose branch numbers are different among themselves only with regard to the units digits forms a word-pair (or triple, etc.). Two or more word-pairs (or word-triples, etc.) whose branch numbers are different from each other only with regard to fraction digits are in the relationship of coordination. In the example above, "he - met - Mary," "he - met - Jane," and "he - met - Karen" satisfy the latter condition. Therefore, these word-triples are in coordination. The set of ordered pairs shown above can be represented in a tree diagram of Fig. 15. It should be noted that tree diagrams of this form are isomorphic to sets of ordered pairs in the following way. The number for a single-line branch should be interpreted in the same way as before (see Fig. 12, for example). The number for a double-line branch is a fraction digit. In a path leading from the starting point (a circle in Fig. 15) to a given node in the tree, the number for a double-line branch is concatenated to the right of fraction digits, while the number for a single branch is concatenated to the right of nonfraction digits. Therefore, "he" of "he met Jane" in Fig. 15 has the branch number 1.3, "Jane" 3.3, and "met" of "he met Karen" 2.33, and so on.



Tree Representation of Coordinated Word Triples
Figure 15

Figure 16 shows the word-triples identified in the sentence "Tom and Jim and Bill met Mary and Jane and Karen and liked Mary and Karen and disliked Jane.". Two new rules are needed for the processing of the sentence.

Rule 15:

<SE, noun>	AND	NP	VP	PD
(1, z)	λ	(1.1, z)	(2. z) (2.1, z)	λ

Rule 16:

<VP, vtl>	NP	AND	VP
{(x, y)}	(x+1, z)	λ	{(x.k, y)}

Figure 17 shows the word-triples identified in the sentence "A young and handsome prince met a beautiful and attractive girl and made the girl his wife." Three new rules are needed for the processing of this sentence.

(1, Tom) (2, met) (3, Mary)	(1.1, Jim) (2.1, met) (3.1, Mary)	(1.11, Bill) (2.11, met) (3.11, Mary)	(1.112, Bill) (2.112, liked) (3.112, Mary)	(1.1122, Bill) (2.1122, disliked) (3.1122, Jane)
(1.1123, Bill) (2.1123, liked) (3.1123, Karen)	(1.113, Bill) (2.113, met) (3.113, Jane)	(1.1133, Bill) (2.1133, met) (3.1133, Karen)	(1.12, Jim) (2.12, liked) (3.12, Mary)	(1.122, Jim) (2.122, disliked) (3.122, Jane)
(1.123, Jim) (2.123, liked) (3.123, Karen)	(1.13, Jim) (2.13, met) (3.13, Jane)	(1.133, Jim) (2.133, met) (3.133, Karen)	(1.2, Tom) (2.2, liked) (3.2, Mary)	(1.22, Tom) (2.22, disliked) (3.22, Jane)
(1.23, Tom) (2.23, liked) (3.23, Karen)	(1.3, Tom) (2.3, met) (3.3, Jane)	(1.33, Tom) (2.33, met) (3.33, Karen)		

Identified Word-triples (1)

Figure 16

(1, prince) (2, met) (3, girl)	(11, prince) (12, be) (13, young)	(31, girl) (32, be) (33, beautiful)	(111, prince) (112, be) (113, handsome)
(311, girl) (312, be) (313, attractive)	(1.2, prince) (2.2, made) (3.2, girl)	(31.2, girl) (32.2, be) (33.2, wife)	

Identified Word-triples (2)

Figure 17

Rule 17:	<NP', adj>	AND	NP'
	(x2, be) (x3, z)	λ	{(x, y)} (x1, z)

Rule 18:

<NP', adj>	N
(x2, be)	{(x, y)}
(x3, z)	(x1, z)

Rule 19:

(NP, art)	N
λ	{(x, y)}

6. Conclusion

An experimental program has been written in SNOBOL III[21] for the system of transformational analysis described above.* It is still to be seen whether the proposed system can be used for an arbitrary transformational grammar. A study is now being made to see if, given a transformational grammar, there is any mechanical procedure for obtaining a predictive grammar with associated ordered pairs which will assign the same base P-markers to a given sentence as would the original transformational grammar.

For the purpose of structure matching in information retrieval systems and of a crude semantic compatibility test between subject and complement, subject and verb, etc., the type of output described in Sec. 5 seems to be most practically manageable. Applications of the proposed system in these two fields are now being studied.

*The author is greatly indebted to Karen Brassil who has programmed for the proposed system and also compiled a small sample grammar of English for testing the system.

REFERENCES

1. Bobrow, D. G., "Syntactic Analysis of English by Computer - A Survey," AFIPS Conference Proceedings, Vol. 24, Spartan, Baltimore (1963).

2. Robinson, J., Preliminary Codes and Rules for the Automatic Parsing of English, Memo RM-3339-PR, The RAND Corporation, Santa Monica, California (December 1962).

3. Described in Hays, D., "Automatic Language-Data Processing," in Borko, H. (ed.), Computer Applications in the Behavioral Sciences, Prentice-Hall, Englewood Cliffs, N. J. (1962).

4. Kuno, S. and Oettinger, A. G., "Multiple-path Syntactic Analyzer," Information Processing-62, North-Holland, Amsterdam (1963).

5. Kuno, S. and Oettinger, A. G., "Syntactic Structure and Ambiguity of English," AFIPS Conference Proceedings, Vol. 24, Spartan, Baltimore (1963).

6. Kuno, S., "The Predictive Analyzer and a Path Elimination Technique," to appear in The Communication of the ACM.

7. Robinson, J., Automatic Parsing and Fact Retrieval: A Comment on Grammar, Paraphrase, and Meaning, Memo RM-4005-PR, The RAND Corporation, Santa Monica, California (February 1964).

8. Olney, J., "An Experiment in the Use of Discourse Analysis Procedures for Reducing Syntactic and Semantic Ambiguity," reported at the 1964 Annual Meeting of the Association for Machine Translation and Computational Linguistics, Indiana University, Bloomington (July 29-30, 1964), paper in preparation.

9. Carmody, B. T. and Jones, P. E., Jr., "Automatic Derivation of Constituent Sentences," ibid.