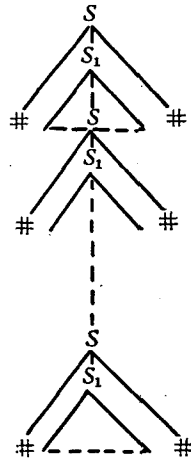


EBERHARD PAUSE

A CLASS OF TRANSFORMATIONAL
RECOGNITION GRAMMARS

1. PETRICK considers transformational grammars (T-grammars) of a special¹ form which essentially have the properties described by N. CHOMSKY (1965).

a) The base grammar is context-free. One recursive element S is distinguished. The base trees in general have the form



or in linear notation

$$S(\# S_1(\dots S(\# S_1(\dots S(\# S_1(\dots S(\# S_1(\dots \#) \dots) \#) \dots) \#) \dots) \#) \dots) \#).$$

Consequently each base tree consists of a finite set of subtrees or *kernel trees* each of the form $S(\# S_1(\underbrace{\dots}_x) \#)$ where x is a string over

¹ I shall assume the reader somewhat familiar with the theory of generative transformational grammar.

$V - \{\#\}$.² A string in which maximally n kernel trees stand one above the other has the *depth (of embedding)* n .

b) Transformational rules (T-rules) are singular or binary. The singular ones operate on a subtree, called constituent tree, whose initial node has the label S_1 and whose terminal string contains no sentence boundary symbol $\#$. The binary T-rules work on a constituent tree and the kernel tree (called the matrix tree) which dominates it directly.

c) The transformational rules are linearly ordered and are applied cyclically in this order from bottom to top. That is, a cycle is completed after the singular and binary rules have been applied to a constituent tree. (In the process the sentence boundary symbols of this subtree are removed).

Now Petrick defines a class of T-grammars generating recursive languages by first stating the condition of the "recoverability of deletions". That is, only a terminal node or a subtree which is identical with another subtree that remains in the resulting tree after the application of the T-rule can be deleted. He states further conditions so that there is an upper bound for the depth of embedding of a base tree underlying a sentence of the grammar. In his case that means, that a base tree that underlies a sentence of length n can have maximally depth n .

A recognition grammar for a T-grammar of this class is constructed by Petrick roughly in the following way: Transformational derivations are generated, and starting with the trees that constitute these derivations he determines context-free rules, so-called "auxiliary rules", reflecting the structure of the derived trees. By means of these auxiliary rules a given string can be attributed as much structure as is necessary for applying the "inverse" T-rules. If the string is a sentence of the given grammar one can obtain the underlying base tree(s) by this procedure.

2. The following problems arise in connection with the class of transformational grammars defined by Petrick:

a) To my knowledge there is no finite procedure to determine whether a given T-grammar has the defining properties of the class or not, which represents a solution to the problem whether there exists

² V means the entire vocabulary, Σ the terminal, Φ the nonterminal vocabulary of the base.

in principle some recognition procedure for the grammar. There also remains the question of whether this problem is recursively solvable at all.

b) In general there is no finite number of auxiliary rules implied by the derivations of a T-grammar of this form. For that reason Petrick can only construct auxiliary rules for finitely many derivations. These rules can then be used only for the analysis of sentences whose underlying base trees have maximally depth n (for a given n) whose length is therefore smaller than or equal to n . For sentences containing more than n words a new recognition grammar has to be constructed. There may also be sentences of length smaller than or equal to n which cannot even be analysed because the depth of their underlying base trees exceeds the specified boundary.

c) The set of trees accepted by the auxiliary rules is larger than the one generated by the original grammar. As a consequence spurious trees result at the end of the analysis process and these must be discarded by an additional synthesis phase.

These problems suggest that it might be reasonable to look at the following requirements when using a grammar for the recognition of sentences. It should belong to a (nontrivial) class of grammar having the following properties:

- a) All generated languages are recursive.
- b) It is decidable whether a given grammar belongs to the class or not.
- c) There should be a general procedure which constructs for each grammar of this type a practicable recognition grammar. That is, analysis with such a grammar should be as efficient as possible.

I do not wish to claim that these requirements have all been met for the type of grammar that I worked with.³ But I want to show now what results I obtained with a different approach to the matter and what difficulties one obviously faces when dealing with transformational recognition.

3. In what follows I will first of all illustrate a general method by which the set of base trees, given by some arbitrary context-free grammar can be decomposed into a finite set of kernel trees. This construction serves (in section 4) as a basis for the definition of a class of T-grammars that generate recursive languages. A hierarchy of types of

³ For details see my dissertation (E. PAUSE, 1972).

grammar will be characterized, included in this class, such that for each type membership of a given grammar is decidable. Section 5 deals with recognition grammars.

The whole matter is discussed with respect to T-grammars of a generalized form:

- a) Every context-free grammar can serve as base grammar.
- b) Transformational rules are unordered. There is no distinction between singular and binary rules.⁴

We may assume that a context-free grammar $P = (V, \Sigma, R, S)$ where $V - \Sigma = \Phi$, the sentence symbol S in Φ , is reduced and has a standard form.⁵ That is, R contains only rules of the form

- (I) $S \rightarrow \lambda$,
- (II) $A \rightarrow D_1 \dots D_n, n \geq 2, A$ in Φ, D_i in $\begin{cases} \Phi - \{S\}, & \text{if } S \rightarrow \lambda \text{ is in } R^6 \\ \Phi & \text{otherwise,} \end{cases}$
- (III) $A \rightarrow a, a$ in Σ .

There is no loss of generality on considering only context-free grammars of this form since for every context-free grammar an equivalent standard form grammar could effectively be given.

A *recursion sequence* (of length k) of a standard form context-free grammar P is defined as a sequence of rules from R

$$\begin{aligned} A_0 &\rightarrow u_1 A_1 v_1 \\ A_1 &\rightarrow u_2 A_2 v_2 \\ &\dots \\ &\dots \\ A_{k-1} &\rightarrow u_k A_k v_k \end{aligned}$$

where A_i is in Φ, u_i, v_i in V^* ,⁷ $A_0 = A_k$ and $A_i \neq A_j$ (for $i \neq j$) otherwise. Each symbol A_i occurring in the sequence is a recursive element in P . Since there are only finitely many elements in Φ every recursion

⁴ In E. PAUSE (1972) I also investigate T-grammars that contain generalized transformations of roughly the same form as discussed by N. CHOMSKY (1957). Concerning the form of the T-rules I refer the reader to the sample grammar in section 4.

⁵ For details see for instance S. GINSBURG (1966).

⁶ λ denotes the empty string.

⁷ V^* denotes the free semigroup (generated by V).

sequence of P can maximally have length $m = |\Phi|^8$ and since R is a finite set, the number of recursion sequences of P is finite.

Two recursion sequences are equivalent if they could be identified by cyclical permutation of the rules. This leads to the concept of the equivalence class or *recursion cycle* of a recursion sequence. Exactly k recursion sequence of length k which could be transformed into one another constitute a recursion cycle. Therefore each recursion cycle is uniquely determined by specifying one recursion sequence belonging to it.

Now all recursion cycles of a standard form context-free grammar $P = (V, \Sigma, R, S)$, where $t = |\Phi|$, could be enumerated in the following way: Let us consider all sequences of rules from R of the form

$$\begin{aligned}
 (*) \quad r_{i1}: S = A_0 &\rightarrow u_1 A_1 v_1 \\
 r_{i2}: A_1 &\rightarrow u_2 A_2 v_2 \\
 &\vdots \\
 &\vdots \\
 r_{ik}: A_{k-1} &\rightarrow u_k X_k v_k
 \end{aligned}$$

where r_{im} is in R , A_j in Φ and $A_j \neq A_m$ (for $j \neq m$), then after maximally $k \leq t$ steps either

- a) X_k is in Σ or
- b) X_k is equal to A_j for some j smaller than k .

In case (b) the sequence of rules r_{i1}, \dots, r_{ik} , noted as above, obviously represents some recursion sequence of P . Clearly, there are only finitely many sequences as given above in P and it is easy to see that in all these sequences (which correspond in some sense to all derivations in P of length smaller than or equal to t) there is to be found at least one representative of each recursion cycle of P .

Now equivalent recursion sequences are identified until there remains exactly one representative for each cycle. Furthermore, one arbitrary recursive element of each recursion cycle is chosen. Let $\{C_1, \dots, C_s\}$ be the set of these symbols each of which is called a *base symbol* of P together with the sentence symbol S . Then consider all occurrences of base symbols on the right hand side of the rules of P as terminal elements which could not further be expanded by some rule. Construct

⁸ $|\Phi|$ denotes the cardinality of Φ .

derivation trees by taking the set $Z = \{C_1, \dots, C_s\} \cup \{S\}$ as start symbols: you will obtain a finite set of trees B with terminal string either in Σ^* or in $\{\Sigma \cup Z\}^*$ having the following properties:

a) every base tree corresponding to some sentence derivation in P can be uniquely decomposed into elements of B (in the obvious manner).

b) If elements of B are embedded into one another by identifying each time some initial node with some terminal node having as label the same base symbol the resulting tree is always a subtree of a tree associated with some sentence derivation in P .

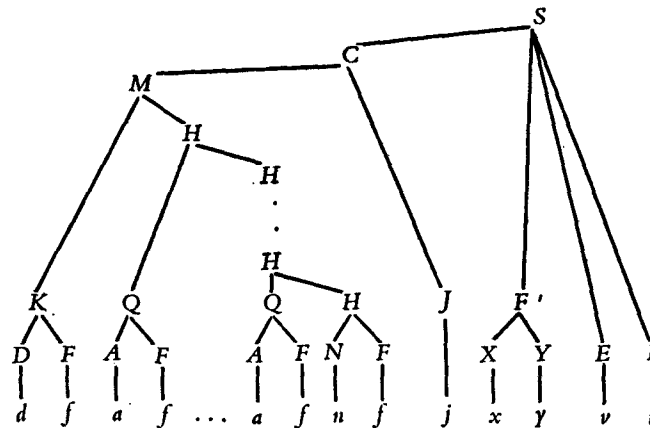
Hence the elements of B are kernel trees in almost the same sense as are those occurring in base trees considered by Petrick. I will not give here a proof of my main statements but will illustrate them by an example.

Let $P = (V, \Sigma, R, S)$ where $\Sigma = \{a, d, f, i, j, n, v, x, \gamma\}$,

$\Phi = \{C, D, E, F, F', H, I, J, K, M, N, Q\}$, and

$R = \{S \rightarrow CF'EI, F' \rightarrow XY, C \rightarrow MJ, M \rightarrow KH, K \rightarrow DF, H \rightarrow QH, Q \rightarrow AF, H \rightarrow NF, A \rightarrow a, D \rightarrow d, F \rightarrow f, E \rightarrow v, I \rightarrow i, J \rightarrow j, N \rightarrow n, X \rightarrow x, Y \rightarrow \gamma\}$.

Taken P as base grammar the base trees in general have the form



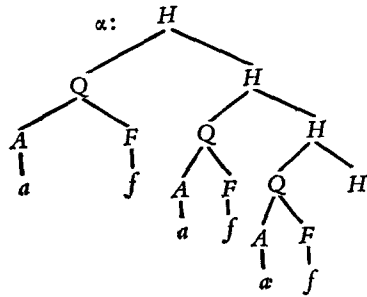
Now all rule sequences of length smaller than or equal to $12 = |\Phi|$, (of the form $(\overset{*}{\downarrow})$), starting with the sentence symbol S are

- | | | |
|--|---|--|
| (1) $S \rightarrow CF'EI$
$C \rightarrow MJ$
$M \rightarrow KH$
$K \rightarrow DF$
$D \rightarrow d(\text{or } F \rightarrow f)$ | (2) $S \rightarrow CF'EI$
$C \rightarrow MJ$
$M \rightarrow KH$
$H \rightarrow QH$ | (3) $S \rightarrow CF'EI$
$C \rightarrow MJ$
$M \rightarrow KH$
$H \rightarrow NF$
$N \rightarrow n(\text{or } F \rightarrow f)$ |
| (4) $S \rightarrow CF'EI$
$C \rightarrow MJ$
$J \rightarrow j$ | (5) $S \rightarrow CF'EI$
$F' \rightarrow XY$
$X \rightarrow x(\text{or } Y \rightarrow y)$ | (6) $S \rightarrow CF'EI$
$E \rightarrow v(\text{or } I \rightarrow i)$ |

It can be immediately observed that there is only one recursion sequence, namely the rule $H \rightarrow QH$ in (2) which represents simultaneously the only recursion cycle of P . Now $\{H, S\}$ is taken as the set of base symbols of P . Further following the given construction finally the kernel trees (a), (b), (c) result:

- (a) $S(C(M(K(D(d)F(f))H)J(j))F'(X(x)Y(\gamma))E(v)I(i))$
- (b) $H(Q(A(a)F(f))H)$
- (c) $H(N(n)F(f))$

It is easy to see that for instance taking the embedding sequence α



there is a tree associated with some sentence generation in P such that the sequence is a subtree of it. From the construction equally follows that each base tree can be decomposed into elements of B in the opposite manner as constructing trees using the members of B .

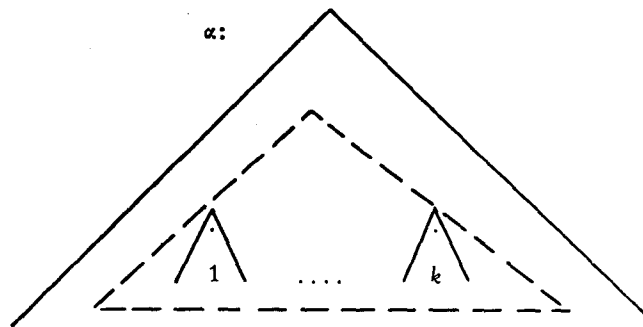
Since there are three kernel trees in α embedded into one another we speak in exactly the same sense as above of the depth of a tree. Thus α has depth 3. In the following we will also call every tree built up only of kernel trees a *complex*.

*

4. In this section, I will first deal with some relevant implications of the foregoing construction and introduce the required terminology. Suppose, some T-rule τ of a T-grammar G is applied to a base tree α :

Let β be the resulting tree. Then β can also be decomposed into kernel trees, taking as cutting-points again those nodes labeled with base symbols. Some of the resulting kernel trees will perhaps have been deleted or will have been modified by the application of τ . In the latter case we get derived kernel trees. In any case, each tree that could be generated in G can be decomposed into (base or derived) kernel trees. I will call the set of all those kernel trees, thus obtained of all trees occurring in the derivations of G , the *decomposition set* B^* of G .

Suppose further that there are k terms in the structural description of τ which have been related to k nodes of α . Now the sequence of all and only those kernel trees (from left to right) in which these nodes lie will be called the *characteristic domain* of τ .



The complex which is the smallest subtree of α such that it contains exactly the nodes, characterized by τ , (see the dashed line) is said to be a *minimal complex* of τ (relative to α).

Obviously only the minimal complex of a T-rule (relative to some tree) is relevant for the observation of the structural change produced by the rule, because all structure beyond it (above and below in the tree) is not directly concerned. To examine certain properties of transformational derivations, it could therefore be sufficient to construct derivations consisting of such complexes rather than of the whole trees. This can be done nearly in the following way:

a) Relative to all T-rules of G , at first all possible minimal complexes of depth 1 are built up of the elements of B . The set of kernel trees, obtained after the decomposition of the complexes resulting from

the application of the rules, yields together with set B set $B_{1,1}$. This procedure can be repeated for minimal complexes over B of depth 2, 3, ... which gives the sets $B_{2,1}, B_{3,1}, \dots$

b) Now continuations of the foregoing derivations (of length 1) could be constructed taking as basis each time the corresponding set $B_{i,1}, B_{i,2}, \dots$. According to the resulting derivations of length 2, 3, ... the sets $B_{1,2}, B_{1,3}, \dots, B_{2,2}, B_{2,3}, \dots$ of kernel trees will be obtained.

By this procedure, which I cannot describe in detail here, the decomposition set B^* of G can be enumerated as the union set of the sets $B_{p,q}$ ($p, q \geq 1$).

For our purposes, it is relevant to consider what happens with some kernel tree γ or some node of a tree in the course of a derivation: performing some transformation, γ is either deleted or there are some derived kernel trees (at least one) in the resulting tree which are the images of γ , either identical or modified by the rule. Again the associated images of these in the following derived tree (which are also images of γ) could be identified, and so on. This procedure could be carried out along the whole derivation starting with an element of B , called the *origin*, in the base tree. The same is valid for a node in a tree. It is either deleted or copied or only transferred by applying a T-rule, where in the latter cases it has some images (at least one) in the resulting tree β called the *occurrences of the node* in β .

Now let us consider some derivation in G . It starts with a base tree α , and we are looking at some kernel tree γ that is a subtree of α , and some derived tree β occurring in the derivation:

a) All occurrences of terminal nodes of γ in β , labeled with terminal elements, or occurrences of nodes in β , inserted in γ or an image of it by T-rules introducing new terminals (or morphemes), are called the *rest-nodes* of γ in β .

b) Each terminal node of γ labeled with a base symbol is called a *base-node*, and all occurrences of such nodes of γ in β are said to be *base-nodes* of γ in β .

After these remarks we are ready to outline the class of *structurally bounded T-grammars*. For a grammar G to be structurally bounded in particular the following conditions must hold:

a) The condition of recoverability of deletions.

b) If a rest-node of some kernel tree α in the characteristic domain of a T-rule τ relative to a tree β is deleted, the number of rest-nodes of α in β must be greater than 1.

c) If a subtree γ is deleted by performing τ , then each base-node

occurring in γ , and lying in some kernel tree of β , may be an occurrence or a base-node of the origin of γ .

d) Let the number of terminal nodes of the origin of a kernel tree occurring in the characteristic domain of τ be m . Then this subtree, following its changes under the rules up to β by considering its "trace" along its images in the given derivation, may not have occurred more than $m-1$ times in the characteristic domain of τ .

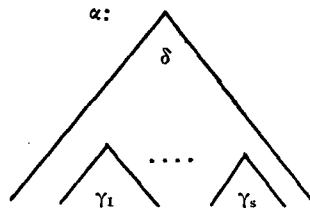
For this class of grammars, say G_1 , the following theorem can be proved:

(A) If G is structurally bounded, then the length of each sentence derived from some base tree α of depth n cannot be shorter than n , the number of kernel trees contained in α could maximally be $\sum_{i=1}^n k^{i-1}$ (where k is a constant depending on G).

Suppose G is structurally bounded, and m is the maximum of subtrees which could be deleted on account of their identity by one T-rule of G . Let j be the number of T-rules of G , and r the length of the longest terminal string of members in the set B of G , then let $j.m.r = k$. $n = 1$: If α is a base tree of depth 1, then every tree β that could be derived from α may contain a rest-node of α . Thus the length of the terminal string of β is greater than or equal to $n = 1$, and the number of kernel trees in α is $1 = \sum_{i=1}^n k^{i-1}$.

Now suppose (A) is true for all trees of depth h smaller than n (h greater than 1).

Let α be a base tree of depth n . Without loss of generality, we may assume that α has the form,



where δ is some kernel tree, $\gamma_1, \dots, \gamma_s$ subtrees maximally of depth $n-1$, and let us further assume that s does not exceed k . Then $k-1$ of these subtrees could be deleted maximally. Some rest node of δ in a tree β derived from α must also survive. From this and by induction it

follows that the length of the terminal string of β must be greater or equal to $n-1 + 1 = n$. Hence the number of kernel trees in α has as maximum value $k \cdot \left(\sum_{i=1}^{n-1} k^{i-1} \right) + 1 = \sum_{i=1}^n k^{i-1}$.

Obviously a decision procedure for the language generated by some grammar of G_1 could be given. However, it can be proved that there is no recursive solution to the problem whether in general a given grammar belongs to G_1 or not.

In the sequel I shall go on to sketch some properties of grammars in G_1 for which this problem is decidable.

In considering the construction of the decomposition set B^* of a given T-grammar G and of the corresponding derivations of complexes (see page 52 and 53), there might exist some point where the following requirements are fulfilled:

Suppose that all derivations starting with complexes up to a certain depth p have been constructed, that the length of these derivations has as yet reached some q , and that they are "structurally bounded". Suppose further that by continuing the construction for some p' greater than p no new derivation (beginning with complexes over B) could be started. Then G is structurally bounded and is said to be in class $G_{p,q}$, if the given derivations become periodical. That is, if they have the same continuations in the $q + 1$ -th step as in the $q - n + 1$ -th step (for some n), in the $q + 2$ -th step as in the $q - n + 2$ -th step, ..., in step $q + n$ as in step q , and so on. This in particular means, that at those points in the derivations always the same minimal complexes appear periodically, and that therefore the decomposition set B^* of G is finite.⁹

Since there are grammars for which these properties hold for arbitrary p and q , there exists an infinite hierarchy G_2 of types of grammars $G_{p,q}$ ($p, q \geq 1$) in G_1 . For each p and q obviously membership of a given grammar in $G_{p,q}$ could be determined in a finite number of steps.

The following example should help in understanding these brief remarks:

Let G be a T-grammar with base given by the above-mentioned context-free grammar P , and let G contain the following obligatory T-rules:

⁹ There is great evidence in the assumption that every «kernel sentence» of a sentence of a natural language can undergo only a finite number of transformations.

$$\begin{aligned}
\tau_1: (\mathcal{M}, \gamma, \nu, i) 1, 2, 3 \rightarrow 3, 2, 0^{10} \\
\tau_2: (\mathcal{M}, j, x, i, \nu) 1, 2, 3, 4 \rightarrow 0, 1, 3, 4 \\
\tau_3: (K, H, F', \nu) 1, 2, 3, 4 \rightarrow 1 + 3, 2, 0, 4 \\
\tau_4: (d, f, F', \mathcal{M}) 1, 2, 3, \rightarrow 1, 3, 3, \\
\tau_5: (\mathcal{M}, F', F', n, f, \mathcal{M}) 1, 2, 3, 4, \rightarrow 1, 0, 3, 2 \\
\tau_6: (\mathcal{M}, F', F', Q, H, \mathcal{M}) 1, 2, 3, 4, \rightarrow 1, 3, 3, \\
\tau_7: (\mathcal{M}, a, f, F', \mathcal{M}) 1, 2, 3 \rightarrow 1, 3, 3
\end{aligned}$$

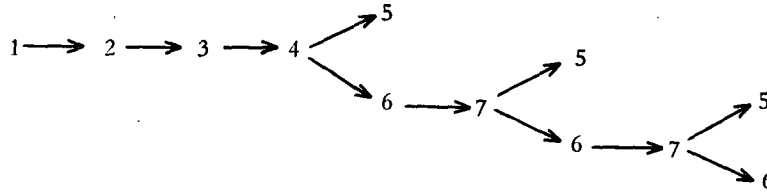
Now the following derivations could be constructed:

$$\begin{aligned}
& \textcircled{1} \alpha_1 = S(C(M(K(D(d)F(f))H)J(j))F'(X(x)Y(\gamma))E(\nu)I(i))^{11} \\
& \textcircled{2} \alpha_2 = S(C(M(K(D(d)F(f))H)J(j))F'(X(x)Y(i))E(\nu)) \\
& \textcircled{3} \alpha_3 = S(C(M(K(D(d)F(f))H))F'(X(j)Y(i))E(\nu)) \\
& \textcircled{4} \alpha_4 = S(C(M(K(D(d)F'(X(j)Y(i))H)))E(\nu)) \\
& \textcircled{4} \alpha_5 = S(C(M(\underbrace{K(D(d)F'(X(j)Y(i)))}_{\gamma_1}))F'(X(j)Y(i))H)E(\nu)) \\
& \rightarrow \left\{ \begin{aligned} \alpha_6 &= S(C(M(\gamma_1 F'(X(j)Y(i))H(N(n)F(f))))E(\nu)) \\ \alpha_7 &= S(C(M(\gamma_1 H(N(n)F(F'(X(j)Y(i))))))E(\nu)) \end{aligned} \right. \\
& \rightarrow \left\{ \begin{aligned} \alpha_8 &= S(C(M(\gamma_1 F'(X(j)Y(i))H(Q(A(a)F(f))H)))E(\nu)) \\ \alpha_9 &= S(C(M(\gamma_1 H(Q(A(a)F(f))F'(X(j)Y(i))H)))E(\nu)) \\ \alpha_{10} &= S(C(M(\gamma_1 H(\underbrace{Q(A(a)F(F'(X(j)Y(i)))}_{\gamma_2}))F'(X(j)Y(i))H)))E(\nu)) \end{aligned} \right. \\
& \rightarrow \left\{ \begin{aligned} \alpha_{11} &= H(\gamma_2 F'(X(j)Y(i))H(N(n)F(f))) \\ \alpha_{12} &= H(\gamma_2 H(N(n)F'(X(j)Y(i)))) \end{aligned} \right. \\
& \rightarrow \left\{ \begin{aligned} \alpha_{13} &= H(\gamma_2 F'(X(j)Y(i))H(Q(A(a)F(f))H)) \\ \alpha_{14} &= H(\gamma_2 H(Q(A(a)F(f))F'(X(j)Y(i))H)) \\ \alpha_{15} &= H(\gamma_2 H(Q(A(a)F(F'(X(j)Y(i))))F'(X(j)Y(i))H)) \end{aligned} \right.
\end{aligned}$$

There is no difficulty in verifying that the continuations which could follow always start with the minimal complexes α_{11} and α_{13} . G is obviously structurally bounded and lies in $G_{2,6}$. The following graph representing the order in the possible applications of the T-rules of G makes perhaps the periodicity of the derivations more transparent:

¹⁰ The symbol \mathcal{M} means almost the same as variables like X in the usual notation of T-rules.

¹¹ \textcircled{i} means that the T-rule i is applied to the foregoing tree. The arrow \rightarrow denotes some continuation of the derivation where it leads out.



5. As indicated above, there is a recognition grammar for each grammar G of the class $G_{p,q}$, namely G itself, using the general decision procedure for the language generated by G . This is, however, obviously not an efficient procedure but represents some general way of analysis by synthesis.

I will now sketch a way of constructing recognition grammars for the class of grammars considered here which in most cases are more practicable. This will mainly be done by examining our sample grammar. I will finish with the discussion of some problems involved in the described procedure.

Let us consider the T-grammar G given above: Analysis should start with some given string over V , say "d j i n j i v". Now by reversing the generation process, the last transformation that has been applied in generating this sentence, $\tau_5: (\mathcal{M}, F', F', n, f, \mathcal{M}) 1, 2, 3, 4 \rightarrow 1, 0, 3, 2$, has now to be performed first. The inverse transformation τ_5^{-1} (which can mechanically be constructed from τ_5) would have the structural description $(\mathcal{M}, F', n, F', \mathcal{M})$. However, there are no rules to attribute some structure to the given string such that there were nodes labeled with F which could be related to the corresponding terms in this inverse structural description. Base rules cannot do this work because the base structure has been modified by the application of T-rules.

Now by inspecting the derivations of complexes a term A of a structural description σ could be 'expanded' in this way: the sequence of labels of the nodes which are dominated by a node related to A , such that the string of these labels is accepted by the inverse base rules,¹² is substituted for A in σ . This will be done, if possible (see below) and necessary, for each term and each T-rule relative to the different trees to which it has been applied in all derivations. In general, for each T-rule more than one 'expanded' rule results.

¹² For each base rule of the form $A \rightarrow x$, the rule $x \rightarrow A$ is the inverse.

Since there are only finitely many different derived (minimal) complexes in the derivations of a grammar G of $G_{p,q}$, one obtains a grammar G' (called *reversible*) containing a finite set of expanded T-rules. G' is by construction equivalent to G . A recognition grammar G is obtained from G' by computing the corresponding inverse T-rules. Then analysis will be performed by 'intermixed parsing', that is, by alternating the application of base rules and T-rules.¹³

In order to prevent the expanded T-rules being applicable at some point in a derivation, where the original rules could not have been applied, auxiliary symbols are inserted in the rules to control their correct application.

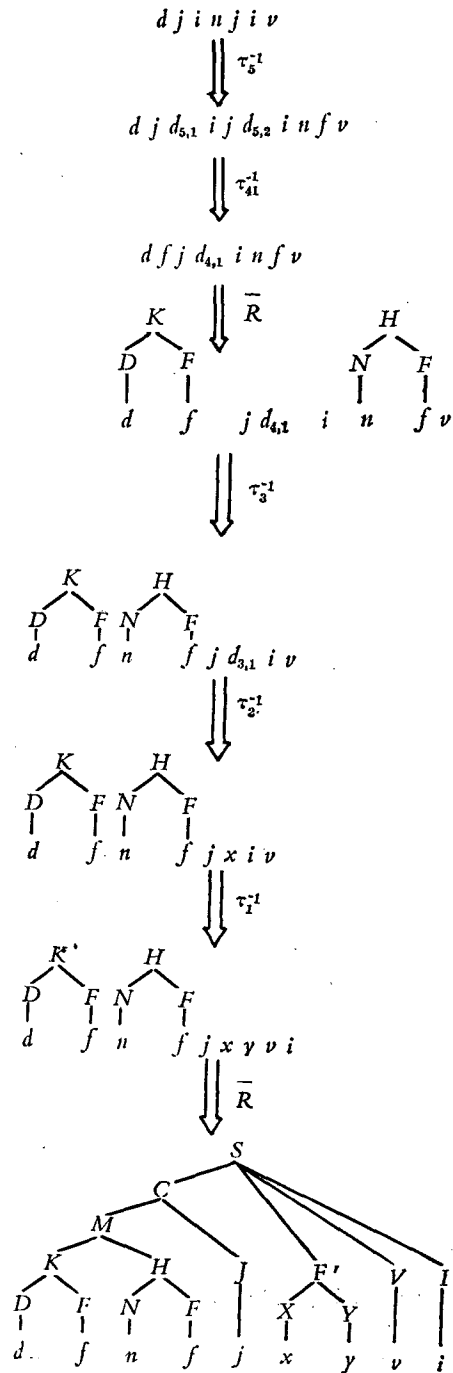
To illustrate these remarks, let us look at the expanded inverse T-rules of our sample grammar G :

$$\begin{aligned}
 \tau_1^{-1}: (\mathcal{M}, i, v) 1, 2 &\rightarrow \gamma, 2 + 1 \\
 \tau_2^{-1}: (\mathcal{M}, j, d_{3,1}, i, v) 1, 2, 3, 4 &\rightarrow 1, x, 3, 4 \\
 \tau_3^{-1}: (K, j, d_{4,1}, i, H, v) 1, \dots, 6 &\rightarrow 1, 0, 0, 0, 5, 2 + d_{3,1} + 4 + 6 \\
 \tau_{41}^{-1}: (d, j, d_{5,1}, i, j, d_{5,2}, i, \mathcal{M}) &\left. \vphantom{\tau_{41}^{-1}} \right\} 1, \dots, 7 \rightarrow 1, 0, 0, f, 5, d_{4,1}, 7 \\
 \tau_{42}^{-1}: (d, j, d_{6,1}, i, j, d_{6,2}, i, \mathcal{M}) &\left. \vphantom{\tau_{42}^{-1}} \right\} \\
 \tau_5^{-1}: (\mathcal{M}, j, i, n, j, i, \mathcal{M}) 1, 2, 3, 4, 5 &\rightarrow 1, d_{5,1} + 2, 4 + d_{5,2} + \\
 &\quad + 5 + 3, 0, f \\
 \tau_6^{-1}: (\mathcal{M}, j, i, Q, j, d_{7,1}, i, H, \mathcal{M}) 1, \dots, 7 &\rightarrow 1, d_{6,1} + 2, 4 + d_{6,2} + \\
 &\quad + 6 + 3, 0, 0, 0, 7 \\
 \tau_{71}^{-1}: (\mathcal{M}, a, j, d_{5,1}, i, j, d_{5,2}, i, \mathcal{M}) &\left. \vphantom{\tau_{71}^{-1}} \right\} \\
 \tau_{72}^{-1}: (\mathcal{M}, a, j, d_{6,1}, i, j, d_{6,2}, i, \mathcal{M}) &\left. \vphantom{\tau_{72}^{-1}} \right\} 1, \dots, 7 \rightarrow 1, 0, 0, f, 5, d_{7,1}, 7
 \end{aligned}$$

The construction should be clear in considering the original rules and the derivations given above. The symbols $d_{n,m}$ denote auxiliary symbols. For the rules τ_4 and τ_7 two expanded rules have been constructed because τ_4 and τ_7 precede the application of either the rule τ_5 or τ_6 .

The recognition procedure will be illustrated by the analysis of the sample sentence "djnjiv". Let $\downarrow \tau_i$, $\downarrow \bar{R}$ denote the application of the rule τ_i or of the inverse base rules respectively. Notice that the inverse T-rules are not true inverse rules, since they apply to sequences of trees, so-called "terminal rest-trees".

¹³ It can be shown that the set of context-sensitive languages is properly included in the set of languages generated by grammars of G_1 . Furthermore, for each context-sensitive grammar a reversible T-grammar can effectively be given.



In addition the following condition is important: the final derived tree of the input string (with respect to the generation process) could be reconstructed in the course of the analysis process. It must then be checked if some obligatory T-rule of G' is applicable to this tree. If this is not the case, the input string is a sentence of the grammar.

The problems which arise out of the foregoing construction are the following:

a) There are cases in which no reversible grammar G' for a given grammar G could be constructed, because, for instance, some auxiliary symbol could not be removed in the course of a sentence derivation.

b) Another source of not obtaining a reversible grammar is, that there could occur subtrees of arbitrary depth which have to be considered for the expansion of some term in a structural description. In this case a partially reversible grammar could be constructed. The recognition grammar then works with base rules, T-rules, and so-called "predictions" associated to the T-rules.¹⁴ Analysis with these grammars will without doubt be much less efficient.

The advantage in using reversible grammars appears to be

a) that there is no additional synthesis phase necessary when the analysis step has been carried out,

b) that the different possible paths which must be pursued in parsing a sentence could be reduced to a minimum, in using auxiliary symbols whenever it is possible,

c) and that the recognition grammar could effectively be given for the whole set of generated sentences.¹⁵

¹⁴ See my dissertation (E. PAUSE, 1972).

¹⁵ At page 54, Theorem (A), the number k must correctly be taken as the maximum number of base symbols occurring in a kernel tree of the set B.

REFERENCES

- N. CHOMSKY, *Syntactic Structures*, The Hague, 1957.
- N. CHOMSKY, *Aspects of the Theory of Syntax*, Cambridge (Mass.), 1965.
- S. GINSBURG, *The mathematical Theory of Context-Free Languages*, New York, 1966.
- S. GINSBURG, B. PARTEE, *A mathematical model of transformational grammars*, in «Information and Control», XV (1969), pp. 297-334.
- A. KRATZER, E. PAUSE, A. v. STECHOW, *Einführung in Theorie und Anwendung der generativen Syntax*, Frankfurt a. M., 1973.
- G. H. MATTHEWS, *Analysis by Synthesis in the Light of recent developments in the theory of grammar*, in «Kybernetica», I (1965), pp. 271-280.
- E. PAUSE, *Transformationssyntaxen, generative Kraft-Entscheidbarkeit-Analyse*, Diss., Konstanz, 1972.
- ST. R. PETRICK, *A Recognition Procedure for Transformational Grammars*, Ph. D. Thesis, M.I.T., 1965.

