19

1965 International Conference on Computational Linguistics

PUSHDOWN STORES AND SUBSCRIPTS

Jacob Mey

Lingvistisk Institutt
Universitetet i Oslo
P.B. 1012, Blindern,
Oslo 3, Norway.

PUSHDOWN STORES AND SUBSCRIPTS

Abstract

Various devices for the improvement of phrase structure grammars (PSG) have been suggested recently. In particular, the PSG model with a pushdown store (PSG/PDS) as described by V. Yngve, and the PSG with subscripts (PSG/S) as described by G. Harman are considered.
It is contended that such devices, even if they may do away with some of the difficulties of PSG, do not contain sufficient generative power to produce the structurally complicated sentences that are generated by other grammars (e.g., of transformational type).
The handling of multiple discontinuous constituents (DC) in PSG/PDS, as well as the use of deletion rules in PSG/S is examined and criticized. It is shown that the improvements on PSG will not allow the grammar to generate all the sentences of the language that a transformational grammar (TG) does; moreover, the improvements on PSG are obtained only at the cost of introducing too much power at the PS level, so that the improved grammars in some cases will exceed the requirements of the description, i.e. generate non-grammatical sentences.

0. Introduction

N.Chomsky has argued that a PSG is not sufficient to generate all the grammatical sentences of a language (Chomsky 1957:34 ff.). Recently, this conception of PSG has been criticized as being too primitive (Yngve 1960:445a, Harman 1963:604 ff.), and several ways of improving such a grammar have been suggested: a PDS has been connected with a PSG (Yngve 1960, 1961, 1962); the use of subscript notation has been recommended to give PSG a fair chance in competition with TG (Harman 1963).

1. PSG/PDS

1.1. PSG and DC

The problem of the so-called discontinuous constituents (for a detailed treatment, see Wells 1947:96 ff.) has always been a crux in IC analysis. One of the drawbacks of PSG as described by Chomsky, is that it is not able to handle these constituents in a way that satisfies both the formal criteria of the grammar and the intuitive feeling that call and up in, e.g., I called him up, belong together and should be treated accordingly in the analysis. Chomsky, in his discussion of PSG limitations, admits the possibility of "extending the notions of phrase structure to account for discontinuities" (Chomsky

1957:41), but, he adds, "...fairly serious difficulties arise in any systematic attempt to pursue this course."
An attempt in this direction is described by V.Yngve in several articles (see especially Yngve 1960); although the presence of DC is the most annoying of the complications under the PSG model (Yngve 1960:448a), the solution offered to this particular problem implies a wider claim, namely, that "any shortcomings /of PSG, JM/ can be overcome" (Ib.:445a). Accordingly, I will discuss below not only the problem of DC, but also the more general one of structure in a PSG/PDS.

1.2. DC and PDS

The crucial step in the derivation of DC by the automaton (for a full description, see Yngve 1960:448-9) is the question asked: Does the right half of the grammar rule in question (GR_i) contain the symbol "..." ? (where "..." stands for "discontinuity in rewriting the symbol on the left hand side of the rule") If the answer is Yes, we have to roll out the temporary memory (TM) tape one space (in a flow chart, one would symbolize this by the index notation 1 − 1 -> 1, where 1 stands for "leftmost": "rolling in" tape would then be indicated by 1 + 1 -> 1, see Fig. 1). During this operation, the original content of TM_1 (the leftmost location of TM) has to be kept in place, that is, the blank has to occur after the original TM_1 (on the right side, if the tape is thought of as moving from the left, see Fig. 1). If, however, the answer is No, we have to make sure that we have space for all the symbols on the right hand side of the rule and roll out tape accordingly. Let n be the number of symbols on the right hand side of GR_i: then we can symbolize the rolling out by the index formula $1 - (n - 1) \rightarrow 1$, as the first symbol always goes to the computing register.

Let further j be the subscript for right hand side symbols of GR_i. The rest of the operation is then performed as routine counting on GR_{i_j}, j being set at 2 (the first symbol has already been taken care of). There should, of course, be a proviso for the symbol "..." itself, so that it will not be copied onto the TM tape.

The method as described here will work neatly even in those cases where DC are "nested", that is, if the expansion of some DC turns out to be another DC (and so on, at least theoretically). As an example, one may try out the doubly discontinuous as far as the corner, where all the necessary rules are specified by Yngve himself (1960:449a).

An implicit assumption throughout the description of the mechanism is that DC can be represented by the simple formula A -> B + ... + C. It follows that there are two cases that cannot be handled directly by the machine: the first one can be symbolized by A -> B + ... + C + ... + D ("multiple discontinuous constituents"); this reduces easily to double discontinuity by a suitable manipulation of the input rules. The other case could be labeled "discontinuous multiple constituents": formula A -> B + C + ... + D (or some variation on this theme), which would imply that the blank has to occur two spaces from leftmost instead of one. Following the instructions given by Yngve we would not obtain the right string of symbols in this case (as examples, one may try: He's not that big a fool, or: As nice a little parlor as ever you did see, or the Spanish sentence: Habla más de lo que sabe 'He talks more than what he knows' (Bolinger 1957:63), where common sense would prefer the analyses that big ... fool, as nice ... parlor, más de ... que (see diagrams in Fig. 2), thus preserving analogy with constructions like such a fool etc. The program could be accommodated to perform this by combining a counting operation with the check on "...", whereafter the continuous part of GR_i's right hand side could be thrown in with the non-DC rules. Derivation being different, there would be no interference from constructions like that big fool, that are treated in the normal way by the machine.

A device like the one described here will, within its obvious limitations, be able to randomly generate sentences that are for the most part quite grammatical (Yngve 1962:70).

The question is: will it generate all, and only, the grammatical sentences of a lang_ uage? I will try to answer this question in the next paragraph.

1.3. <u>Limitations of PSG/PDS</u>

Although the model as proposed by Yngve in its original form only uses the PDS technique to solve a minor problem in syntactic analysis by the machine, the scope and use of PDS are by no means limited to this particular pro_ blem of DC (For a detailed discussion, see Oettinger 1961:126_7). The elegancy and sim_ plicity of PDS algorithms make them well_ suited for procedures of automatic syntactic analysis of languages.
There are, however, some inherent limitations. Common to all PDS techniques is the fact that information stored in this way only is access_ ible in accordance with the formula "last in, first out". Being essentially a linear array of information (Oettinger 1961:104), the user (the machine) will not be able to draw on other information than is given by the leftmost sym_ bol in a left_to_right production (the tempo_ rary memory tape in Yngve's machine, see Fig.1). Since, on the one hand, the machine output is past control (what is printed, is no longer available to the machine for inspection) and, on the other hand, the internal state of the machine is entirely determined by the current input symbol, one has to keep careful account

not only of the current derivational steps, but also of the "left_overs" from earlier steps. This is exactly what a PDS can do, and the problems in connection with this technique are, as shown above in the case of the so_called discontinuous multiple constituents, are mainly technical (providing indexes etc.)

The linear character of the memory, however, together with the finite state properties of the model itself give rise to another problem that seems unsolvable under the following assumptions for our machine: a finite number of states, a linear temporary memory, and a transition from one state into another by one_symbol input. The problem is the following: given any internal state of the machine that is determined by more than one symbol simultaneously, will the supplementary device of a PDS be able to supply the necessary instructions to the machine that are not contained in the current symbol?

The answer is in the negative, precisely because the memory is linear, and there is no "look_up" for items in the memory. What is stored in the memory can only be brought up to the surface by something outside the memory itself, that is, I have to create an "expectancy" that is specific for each item in the PDS. Only under these conditions the state of the machine can be defined as determined by the current symbol plus the contents of the temporary memory (Yngve 1960:449). This is essentially the procedure described by Harris for keeping track of nested constructions ("incurrence and discharge of requirements", Harris 1962: 53). The reason why the machine is able to handle DC is that this "nesting" occurs in one level, so that the symbols involved can be uniquely determined as belonging to the same dimension of analysis.

Where "surface structure" is explained only by underlying "deep structure" (Hockett 1959: 246 ff.), the machine will not be able to carry out the analysis correctly. The structure that underlies a symbol X_1 may be bound up with a special PS derivation, so that rules concerning structures like, say, $X_1 + X_2 + X_3$ will be ambiguous in their application. One could place restrictions (in Harris' sense) on (one of) the symbols, thus creating a multiple path through the derivation, possibly combined with a cycling device: this is what the subscript technique does, see 2.4 for a detailed discussion. Some of the difficulties are removed in this way, but others persist, like those cases where pairs of symbol formulae are involved (the so-called "generalized transformations" of early TG, Chomsky 1957:113); this point is also discussed below. While placing too many restrictions on the symbols has serious disadvantages (some of which will be discussed in section 2 of this paper), it certainly exceeds the capacity of the model as described by Yngve: his rules are all of the context- free form.

Thus, structure in a sufficiently powerful PSG is not only a matter of specifying the right rules, but also of choosing the right rules and combining them at the right places. There is still another factor that we have left out of consideration so far: the ordering of the rules. Yngve states that any order will do: an alphabetical order may be convenient (1960:445). Now this has two consequences: first, all of the rules have to be run through every time a symbol is expanded (perhaps only a minor drawback in a computer-oriented analysis), second, the advantages of ordered rules (economy, elegancy, accuracy) are lost ("forcing all kinds of low-level detail into the rules", Bach 1964:53). Besides, ordering of the rules is indispensable in cases where complicated high-level structural descriptions are involved: thus an immediate derivation of each non-terminal symbol all the way down to word level would not be permitted in any kind of PSG, not even the most context-sensitive ones. Being essentially context-free, Yngve's grammar will generate what is usually called "kernel sentences" (Chomsky 1963:132): unambiguous derivation of more complex structures (derived sentences) will only be feasible under a careful specification of the order in which the rules have to apply (as an example, cf. the discussion of wh-transformations as depending on the interrogative transformation in Chomsky 1963:140).

There is another way out of the difficulties that have been sketched in this section: phrase-structurizing at different levels, these being kept together by the representation relation (see Sgall 1964b). This solution is based on a somewhat different interpretation of PSG functions (not only syntactic, but also semantic rules are included); a PDS is coupled with the PSG of the lowest level. A detailed discussion of this system will have to wait for more details, but it seems that grammars based on dependency relations have received too little attention so far (for a comparison of IC and dependency theories, see Hays 1964: 519-22).

1.4. Grammar and psychology

Referring to experiments performed by G.A. Miller, Yngve establishes an analogy between the "depth" of memory in the human brain and the depth of sentence construction in the model (1960:452). The human brain is not capable of storing more than, say, seven plus minus two items at a time (for references, see Yngve ibid.). In other words, the human brain has a limited capacity, just like the temporary memory of Yngve's machine. One of the conditions to be put on a flawless handling of "deep" constructions is that the storage capacity is not exceeded by the number of symbols to be developed later on. In this connection Yngve makes the interesting observation that sentences and constructions in general actually do

have a sort of limited depth, i.e. the number of regressive nodes is bound by more or less the same upper limit as that for human memory's simultaneous storage capacity.
Now, I think that the analogy between the two kinds of "storage" should not be overstressed. It rests primarily on the tacit assumption that the model should, or could, be considered as a more or less true_to_life representation of human linguistic activity. As I have remark_ ed before, this supposition is altogether groundless, and will at best hamper an expla_ nation of such activity in truly linguistic terms. A remark made by Yngve in this connect_ ion may clarify the issue. Yngve says (1960: 452b; see also 1961:135_6 for an even more ex_ plicit commitment):
"The depth limitation does not apply to algebra, for example, because it is not a spoken langua_ ge. The user has paper available for temporary storage."
But so has the user of any other language, e.g, human everyday spoken language. The fact that we do not use paper actually when speaking has nothing to do with greater or lesser depth of sentences (or, if it does, the depth differences occur only to one side, namely that of decreas_ ing depth). One could pursue this analogy ad ab_ surdum by assuming two kinds of depth, one un_ limited, for written languages, and one limited, for spoken languages. The results would be dis_ astrous for any description of any language: sentences of the type: "That that that they are

both isosceles is true is obvious isn't clear" (Yngve 1960:458b) are as ungrammatical in written as they are in spoken English. Of course Yngve is perfectly right in attributing the difference between the above non_grammatical (deep regressive) that_clause and its grammatical (progressive) counterpart: "It isn't clear that it is obvious that it is true that they are both isosceles" to excess depth. So, there is a depth limitation and this limitation is gramatically relevant. But this linguistically fruitful concept should not be confounded with hypotheses from descriptive psychology.

That the claim for descriptive similarity between psychology and linguistics is latent in Yngve's model can be seen from another instance. His second assumption for the model (1960: 445) is that "the model should share with the human speaker ... the property that words are produced one at a time in the proper time sequence, that is, in left_to_right order ..." (the first assumption, viz. that any shortcomings of the PS model can be overcome, has partly been dealt with above, and will be treated at length in the second half of this paper). This restriction, I think, on a model (or a grammar, insofar as the grammar is based on the model) is unnecessary and self_contradictory. It is unnecessary, since the model should only copy relevant traits in the speech production of the individual; and even though it may be true that words are produced in a linear sequence (as already Saussure has remarked), it has not yet been shown how this linearity is to be interpreted in human speech production: I think it is only weakly relevant, that is to say, linearity alone will never suffice to give a complete picture of the speech event. For a full_fledged description of speech I suppose the assumption that we speak in senten_ces rather than in words will have many advant_ages.

Moreover, the claim that the model should du_plicate the property of left_to_right product_ion in the human speaker cannot be brought to harmonize with the model. In fact, the model can only examine one symbol at a time: the machine may erase or delete or read only that section of the memory tape that is closest to the roll, i.e. the leftmost symbol only (Yngve 1960:446). Now, the limitation of human memory is on re_producing more than a certain number of items <u>at the same time</u>. The analogy clearly does not hold between human memory and machine storage: the explanation is that the machine produces symbols, whereas the speech of humans is struct_ured. In other words, a left_to_right product_ion may in many cases be explained by a linear structure in the producer; the pushdown store is a linear memory device. But there are other left_to_right productions that are structured in such a way that a PDS or other left_to_right arrangements will not suffice. It is of course true that a structural description is not alto_gether absent from a PSG/PDS: Yngve's machine produces as its output a string of symbols containing both syntactical markers ("flattened_out trees") and terminal symbols. This will suffice to "infer the derivational history of each string from that string in a single way" (Sgall 1963:41), but only insofar as the structure can be described in one_level terms, cf. discussion above (see also Sgall 1963; 1964a). The question will be treated at length in part two of this paper.

2. PSG/S

2.1. The subscript notation

The subscript method referred to here is not in the first place thought of as a machine program (even though its close affinity with the computer language COMIT is asserted, see Harman 1963:608fn.). Accordingly, it has a more general scope: namely, to offer a full_fledged alternative, in PS form, to other grammars (e.g. of transformational obedience), thereby proving that "transformational grammar has no advantage over the phrase structure grammar" (Harman 1963:598).
Subscripts are added to the PSG rules in two ways: first, to introduce restrictions on such rules, second, to specify where those restrictions apply. An example of the first kind is the rule S -> S1/NUMBER_SG (Harman 1963:609), and, in general, any rule of the type A -> B/J + ... The second case obtains e.g. in the following rule: NP/NOT_WH -> DETERMINER + NOUN, and, of course, in all rules where subscripts

are "lost" during expansion. I think there will be a third type as well, even though this is not expressly mentioned in the article, namely, subscripts that do both: introduce new subscripts at places indicated by old ones; but this is only a minor point. More important is the observation that subscripts can take care of all sorts of constituents, both continuous and discontinuous. For the latter, the generation rules are adapted from rules suggested by Victor Yngve (Harman 1963:606; the reference quoted is Yngve 1960). Like in Yngve's model, the rules of PSG/S are unordered: all necessary information about when and where to apply a rule is contained in the subscripts (which, by the way and perhaps a fortiori, are said to occur in an unordered sequence). But, as will be seen from the following paragraphs, this "when" and "where" is not only a notational problem: in fact, it is one of the big underlying differences between PSG and TG. (On the difficulty of ordering rules in a PSG, see Chomsky 1957:35). A further important difference from other PSG interpretations is the admission of deletion rules, that is rules of the form A -> Ø (Harman 1963:603); also this point will be discussed at length below.

2.2. Subscripts And Transformations

In general, one cannot deny the possibility of incorporating (by means of subscripts or other devices) some of the information that is contained in a transformational grammar into a

one_level grammar of PS type.
But the grammar thus constructed will never generate all and only the grammatical sentences of the language. Either it will generate too little (the normal case for PSG without subscripts or similar devices) or, if it generates more, it will also generate some non_grammatical sentences (Harman 1963:611:"... not all sentences constructed in accordance with this grammar are well_formed.")
A very simple example will show this. Suppose we want to transform optionally a sentence into its question counterpart. To do this in the PSG/S according to Harman, we have to choose an appropriate expansion of the symbol S2 (the same paths hold for number_ and mode_restricted S : S1, resp. S2, Harman:600), namely either the second or the fourth rule in 3., the set of expansion rules for S2. We choose the second rule (normal question, the fourth rule concerns wh_questions):
S2 --> VP/TYPE_QUES,NOT_WH + NP/CASE_NOM,NOT_WH.
Now, note two things: in order to conform to the rules for this grammar, we have already added some of the subscripts from Rules 1 and 2 to the symbol S2 (e.g., NUMBER_SG and MODE_ACT). These subscripts, together with the new ones, are to appear on every symbol that is contained in every rule from now on (unless a delete subscript is introduced, cf. below). This is necessary, since we cannot let any information that is conveyed by the subscripts be lost, even if

it be irrelevant to the symbol in question (such as, say, a MODE restriction on a NP). One can easily imagine that rewrite rules of this type soon become very unwieldy (even if we do not allow ourselves to be frightened by the prospects of "millions of rules", Harman:605). Thus, in rule 7 of this comparatively simple grammar we already have 6 subscripts to each symbol. This number is substantially increased in the more elaborate version of the grammar (see Appendix to Harman's article). This is certainly not what one would call simplicity of description. But objections of this kind can be met by the following consideration: even if the multiplication of entia, i.c. symbols and subscripts, seems without rationale for humans, one can conceive of it as a necessity for computer data handling, and the computer certainly does not mind going through all the subscripts, adding some, deleting others, etc., every time a symbol is mentioned or expanded. So, if one has a working program in which these restrictions can be written out as subroutines, and if the computer space needed does not exceed that available, the objection just made does not hold (cf. Harman:610fn.: "Many of these grammars are in the form of computer programs for generating actual sentences.")

The other question is far more important. It can be split up into two parts:

1. Can all the data of the grammar be put into the subscript_restriction schema?
2. Will the subscript_restriction schema not

put more data into my grammar than wanted? The first question concerns the adequate re_ presentation of the structure, the other ex_ presses the fear that I may add structure to my grammar, thus producing sentences that are not grammatical (see Chomsky 1962:514ff.) Adopting a distinction made by Chomsky, I make the following assertion: A PSG/S will serve as a more or less adequate observational and descriptive representation of the facts covered by a normal PSG; as far as TG is concernd, the structure of the transformational model (how trees map into trees) will not be represented adequately on the descriptive (and perhaps not even on the observational) level by a PSG/S. In no case the PSG/S will attain the level of explanatory adequacy.

The first part of my assertion can easily be proved from the observation that a normal PSG and a PSG/S are strongly equivalent grammars, the only difference being the notation. (On the notion of equivalence, cf. also Hays 1965:519). In fact, it makes no difference whether one ex_ pands a symbol on the basis of a rule to be af_ fixed to the constituent by means of a sub_ script, or on the basis of a rule contained so_ mewhere else in the grammar. The essential is that generation proceeds from left to right, and one symbol is produced at a time. (See discuss_ ion above, 1.2).

To prove the other half of the assertion made above, I will try to give an answer to the two_ fold question about representation of structure. Let's go back to the elementary example of the optional T_q, and try to imagine how this is handled in a PSG/S. The main difference between PSG and TG is that the rules in PSG operate on symbols, in TG on strings of symbols. When I put a subscript on a symbol that is part of a string, and I want to mark off a structure that is based on several symbols occurring in a certain order, I will have to mark all the symbols of my string in the same way, and this way of marking must be unique, i.e. define a unique path through the rules. This path may, in due course, require additions, deletions, permutations and the like. Now, in TG these operations are carried out after the PS derivation has been completed. In PSG/S, however, the cleavage between affirmative and interrogative sentences occurs already in the third rule, where S2 is expanded into NP + VP, VP + NP, respectively (omitting the subscripts). The two derivations follow separate paths through the rules: in terms of tree diagrams, what is left in the one is right in the other of the two trees, In this way, many PSG rules are unnecessarily duplicated (see above); moreover, the relationship between interrogative and declarative sentences, as defined in TG, is reduced to a remote common source of derivation, namely S2. It is not true that "Sentences are 'transformationally related' to the extent that the same choice of restrictions is made in their derivations and if the same lexical choices are made where possible" (Harman:608; single quotes are his), unless one takes "'transformationally related'" in a sense rather different from Chomsky's, namely: sentences that have a (partial) path through the rules in common. This is, in fact, the only 'transformational relation' that it is possible to define in a PSG/S, but unfortunately, it is not transformational. Even in the case that two paths coincide, and coincide altogether, we do not have 'transformational relatedness', but "grammatical similarity" (Harman:608). Lexical choices have nothing to do with this relation: both in PSG and in TG the choice on the lexical level is made after the application of expansion, respectively transformational rules. (This is not altogether correct: lexical choices may be made earlier and thus affect the derivation, but this is beside the point; complex symbols (see Klima 1964) are not taken into consideration here, but they could be built into a PSG as well as into any other generative grammar. I think, e.g., that some complex symbol could be devised to prevent sentences like The man walks the men, that could easily be generated in accordance with the rules described on pp. 609_10 of Harman's article.)

In my opinion, a PSG/S will never be able to show transformational relationships as formally defined and described by Chomsky and others; hence such a grammar, even though it may attain a certain descriptive adequacy, will never give an explanation of the fact that precisely this, and not some other sentence, is transformed into another structure.

2.3. Deletion in a PSG

Another difficulty in PSG/S concerns the problem of deletion rules. In normal PSG, no deletes are permitted (Chomsky 1961:9). Harman gives as reason for this restriction that trees must be uniquely recoverable in a PSG (p.603). This is, however, only part of the motivation. Deletes are not symbols: they cannot be expanded (un_less one chooses to expand them into deletes, which is obviously useless in a description). Whenever a deletion rule occurs, the structure of the derivate is altered in such a way that rules may apply which originally should not. One could say that deletes are extremely con_text_sensitive: in Harman's PSG/S, which in reality is a highly restricted PSG, the number of rules having the form A -> Ø is very limited indeed, even though the author advocates their use (p.605). In passing, I would like to remark that nearly all of the deletion rules have to do with the expansion of NP/WH (this subscript occurs only once in the smaller grammar, p.609, and should therefore be rejected by the machine, since there are no constituents on which the rule could apply.)
The real reason why a delete cannot be admitted in a PSG (especially a highly context_sensitive

one) is that the rules following the deletion rules should be modified or altered completely , otherwise it would not be possible to keep the distinction between not_rewritten and re_written symbols clear: the rules following de_letion might thus operate on symbols originally belonging to the context. (Note, by the way, that in the case of wh_words the context pro_blem is somewhat simplified by the fact that these words normally stand at the beginning of a sentence, so that the left context can be thought of as zero.) In our example, the transformational rule for interrogative sent_ences to be generated from declarative ones operates on a string of symbols that may be symbolized $X_1 - X_2 - X_3$ (Chomsky 1957:112), carrying it into the shape $X_2 - X_1 - X_3$. Now, suppose that in the course of the deriv_ation to non_terminal symbols (the kernel string) we have a deletion rule operating, say, on X_1. Suppose moreover that the non_terminal symbol following X_3 qualifies for the condi_tions originally put on X_3. The transformatio_nal rule will then operate on a string $X_2 - X_3 - X_4$, and carry it into $X_3 - X_2 - X_4$, thus generating a non_grammatical sentence. I do not pretend that the actual PSG/S as proposed and described by Harman in his article will generate these sentences: as already said, the grammar makes a very cautious use of deletions, so that sentences like the ones mentioned will not occur. This does not, however, invalidate the criticism.

Subscripts may not only be added in a PSG/S, but also deleted. In this manner a restriction that has been put on a certain rule can be removed (this deletion of subscripts is of course quite another matter than the deletion of symbols discussed above). Subscripts may be superfluous, such as in Rule 8.1 (p.609), where the subscript AUX_MODAL is removed from the constituent INFINITIVE by the subscript —AUX_MODAL, even though the lexicon would offer no ambiguous rewrites in the case of a non-removal of the superfluous subscript. One could perhaps wonder why this precaution is taken, since in many other instances superfluous subscripts persist all the way through the derivation (see discussion above). In other cases, the removal of subscripts can be motivated by the desire to prevent ungrammatical "loops", i.e. endless recursive expansions that have no justification in the grammar. Thus in Rule 8.1 the symbol VP3/AUX_MODAL is expanded into INFINITIVE/... + VP3/AUX_HAVE,—AUX_MODAL, thus preventing another expansion by the same rule of VP3. If, on the other hand, we wish the symbol in question to be expanded recursively (and according to the latest development in TG there should be no difficulty in admitting recursivity for all symbols, S not excluded: see Klima 1964), we can restart the cycle by wiping our slate, i.e. deleting all the subscripts by means of the instruction ERASE.OTHERS, to be incorporated as a subscript on the right hand side of the rule. Naturally, we would expect a subscript of this kind to occur in those cases where a whole sentence is to be embedded into another by means of what in early TG was called "generalized transformations" (Chomsky 1957:113). The nominalizing transformation is an instance in kind: under 3g in the extended PSG/S (p.613), we find, among others, the entry:

NP8 —> S1/CLAUSE.TYPE:NOMINALIZATION,SUBJ.IN.
GENITIVE,B,C,D,E,Z,Y,ERASE.OTHERS

This means that all the subscripts originally found on NP8 are to be deleted; the new subscripts deal exclusively with the derivation of the embedded clause (as can easily be verified from the rules of the PSG/S as given in the Appendix of the article). Whereas TG keeps track of the changes to be made by means of a structural description of the pair of kernel sentences involved, together with a formula for structural change, in PSG/S we have only a constituent NP to be expanded by means of PS rules. How this NP fits into the structure of the original kernel sentence (being essentially its path through the PS derivation) can be followed in PSG by tracing back the nodes of the tree representation. In PSG/S, this path is marked by the subscripts added to the NP in question. Now, all this information is struck from the record by the removal of the subscripts in accordance with the instruction ERASE OTHERS. Structural description of the sentence as a whole is not available: the expansion of NP8 destroyed our bridge back to the original S. It is as if we had been expanding a constituent while forgetting what it was we were expanding.

3. Conclusion

Of the two models discussed here, the first one (PSG/PDS) has not actually been proposed as a full-scale grammatical model, but I have tried to show that the implications of the claim that any shortcomings of PSG can be overcome lead to difficulties of about the same nature as those encountered in the second model (PSG/S). Descriptive adequacy is not attained in those cases where structural descriptions are relevant for the operation of the rules: neither PSG/PDS nor PSG/S permits one structural description to be carried over into another. As one will have noticed, the argument in both cases runs along the same lines. Moreover, of the several devices proposed by Harman to boost the power of PSG, the deletion rule was explicitly rejected on the ground that it would add too much power to the grammar. On the other hand, the use of subscripts, no matter how carefully chosen, will not help enlarge the descriptive power of the grammar (Harman 1963:605) enough to account for all the grammatical sentences of the language. Thus, one-level grammars like the ones discussed above will not attain explanatory adequacy in any case, and in some cases not even descriptive adequacy. "Dieser Versuch /namely, the defense of phrase structure, JM/ verfehlt den entscheidenden Punkt aber in zweifacher Hinsicht: Erstens überschreiten die Regeln Harmans die Kapazität einer PSG. Und zweitens lösen auch sie nicht das Problem einer geigneten Zuordnung von Stammbäumen." (Bierwisch 1964: 49fn.11)

References

Bach 1964: E.Bach, An Introduction to Transformational Grammars, New York etc., 1964

Bierwisch 1964: M.Bierwisch, Aufgaben und Form der Grammatik (Preprint IId International Symposium "Zeichen und System der Sprache, Magdeburg, Germany, September 1964)

Bolinger 1953: Dwight L.Bolinger, Addenda to the Comparison of Inequality in Spanish Lg. 29 (1953), 62-6.

Chomsky 1957: N.A.Chomsky, Syntactic Structures The Hague, 1957

Chomsky 1961: Id., On the Notion Rule of Grammar, in: Structure of Language and its Mathematical Aspects, PSAM XII (1961), 6-24, Providence, R.I., 1961

Chomsky 1962: Id., A Transformational Approach to Syntax, in: Third Texas Conference on Problems of Linguistic Analysis (1958), Austin, Tex., 1963, 124-58

Harman 1963: Gilbert H.Harman, Generative Grammar without Transformation Rules: A Defense for Phrase Structure Grammar, Lg. 39 (1963), 597-616

Harris 1962: Zellig S.Harris, String Analysis of Sentence Structure, The Hague, 1962

Hays 1964: David G.Hays, Dependency Theory: A Formalism and some observations, Lg 40 (1964), 511-25

Hockett 1959: Charles F.Hockett, A Course in Modern Linguistics, New York, 1959

Klima 1964: E.S.Klima, Current developments in Generative Grammar (in press). A MS copy of this paper, which was originally read before the 1964 Colloquium on Algebraic Linguistics, Prague, Czechoslovakia, was kindly put at my disposition by the author.

Oettinger 1964: A.G.Oettinger, Automatic Syntactic Analysis and the Pushdown Store, in: PSAM XII (1961), 104_29

Sgall 1963: P.Sgall, The Intermediate Language in Machine Translation and the Theory of Grammar, in: American Documentation Institute, 26th Annual Meeting, Chicago, Ill., 1963, 41_2.

Sgall 1964a: P.Sgall c.s., Cesty moderní jazykovědy, Praha 1964

Sgall 1964b: P.Sgall, Zum Verhältnis von Grammatik und Semantik (Preprint IId Intern. Symp., Magdeburg, Germ,, 1964)

Yngve 1960: V.H.Yngve, A Model and an Hypothesis for Language Structure, in: Proceedings of the American Philosophical Society 104 (1960), 444_66

Yngve 1961: Id., The Depth Hypothesis, in: PSAM XII (1961), 130_8

Yngve 1962: Id.,, Random Generation of English Sentences, in: 1961 International Conference on Machine Translation of Languages and Applied Language Analysis, London 1962, 65_81

Wells 1947: Rulon S.Wells, Immediate Constituents, Lg 23 (1947), 81_117

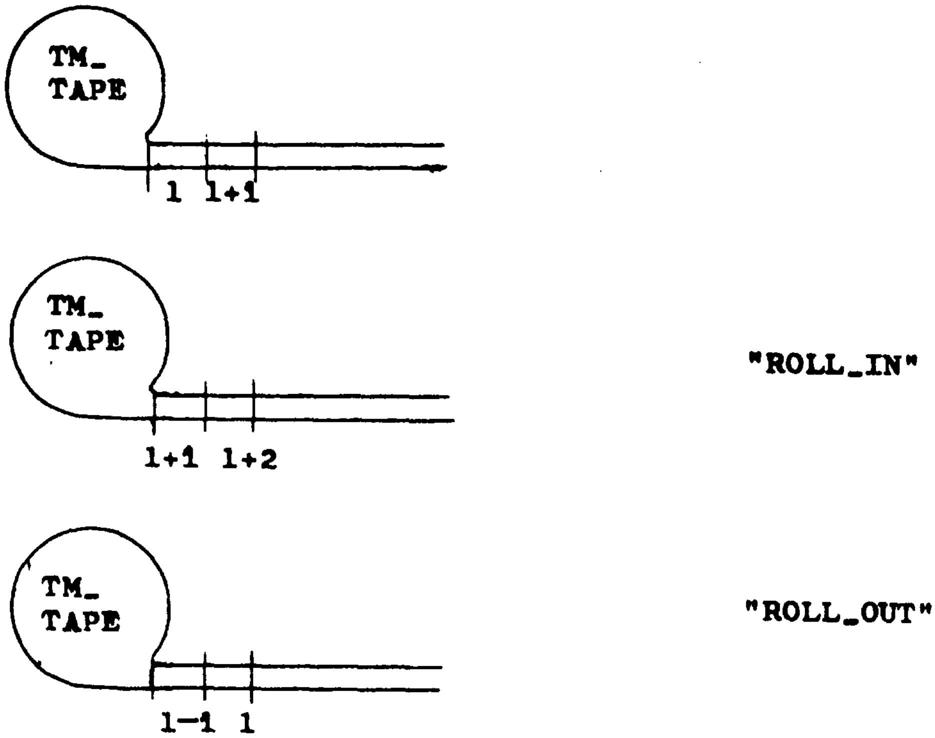

FIG.1. THE TEMPORARY MEMORY

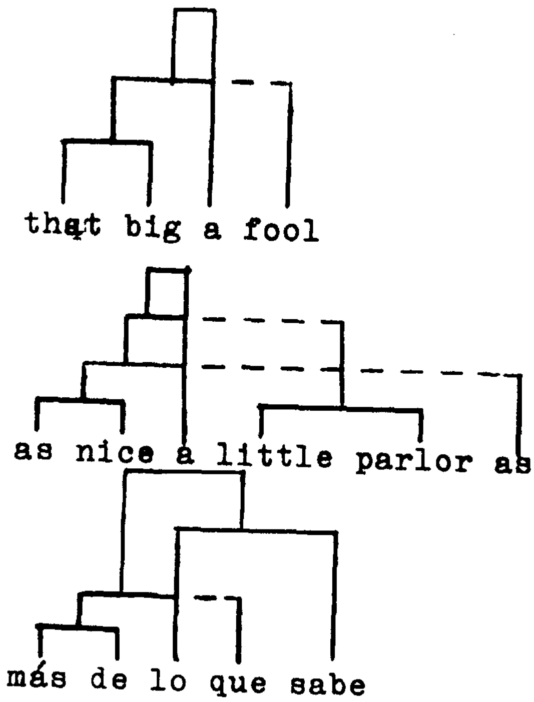

FIG. 2. DISCONTINUOUS MULTIPLE CONSTITUENTS