
James T. Heringer, Jr.

* Sponsored in part by the National Science Foundation through Grant GN-534 from the Office of Science Information Service to the Information Sciences Research Center, The Ohio State University.
This article is concerned with the comparative strong and weak generative capacities of dependency grammars (dgs) and phrase-structure grammars (psgs), the latter being equivalent to Chomsky's context-free psgs. It is shown in the article that psgs and dgs are equal in weak generative capacity; they define the same set of strings. Since the structures assigned by these two types of grammars differ significantly, it is necessary to introduce notions of strong equivalence between grammars of the two types, based on structure assignment, before the strong generative capacities of the grammars can be compared. Different concepts of strong equivalence between dgs and psgs are defined in the article, and it is shown that in terms of these concepts, psgs have more strong generative capacity than dgs, in that for every dg there is a "naturally corresponding" psg, but not vice versa. The concept of degree of a psg is defined and used to characterize that property of psgs which causes them to have more power than dgs.

To show the analogies between dependency trees (d-trees) and phrase-structure trees (pss), Gaifman gives a descriptive definition of the former which he shows to be equivalent to the constructive definition used in RAND publications. Both definitions use the following formulation of a dg (or dependency system, in Gaifman's terminology). A dg consists of three sets: 1) a set of rules which gives for each category those categories which may derive directly from it with their relative positions, symbolized \( Y_1 \ Y_2 \ \ldots \ Y_k \ Y_{k+1} \ \ldots \ Y_n \), where \( k \) may equal zero or \( n \) and \( n \) may equal zero (meaning \( X \) may stand alone, without dependents; 2) a set of rules which give for each category the list of words belonging to it, the lists not necessarily being mutually exclusive; 3) a list of categories which may govern sentences. To define dependency trees, it is necessary to characterize the dependency relation associated with the dg, a two place relation. \( P \rightarrow Q \) means \( P \) depends directly on \( Q \); \( P \rightarrow Q \) means there is a sequence of dependencies relating \( P \) and \( Q \), where \( P \) and \( Q \) are occurrences of categories.

- 129 -
Gaifman defines $d$ as follows: 1) $d^*$ is irreflexive; 2) $d$ is many-one; 3) $d^*$ does not introduce discontinuities with respect to sentence word order; 4) $d$ is interconnective; 5) $d$ is specified by the first list of rules given above. In particular, if for no $P$ does $PdQ$ hold, then $X_j \ (\ast)$ must be a rule of the first list, where $X_j$ is the category which $Q$ is an occurrence of; 6) if for no $Q$ does $PdQ$ hold, then the category of $P$ is listed in the third list above; 7) the occurrences related by $d$ are occurrences of categories assigned to the words of the sentence by the rules of the second list above. Any set of occurrences related by $d$ is a $d$-tree. By theorem 1.1 of the article, this definition is equivalent to the recursive definition of $d$ given in RAND publications.

To characterize phrase-structure trees (phrase structures or pss, in the article) Gaifman introduces the notion of ramification, based on one meaning of the term parenthetical expression (pe) as Hayes uses it. For Gaifman, a pe in a string of words is a string of successive occurrences, i.e., a substring, taking position into account. A ramification over a string $\mathcal{X}$ is a set $\Gamma$ of pes in $\mathcal{X}$ satisfying the following: 1) $\mathcal{X}$ belongs to $\Gamma$. 2) If two pes are in $\Gamma$ and they are not equal then they are either disjoint or one is a proper substring of the other. 3) There is no pe in $\Gamma$ which is partially decomposable into other pes in $\Gamma$; either it is unanalyzable or decomposable as a whole.

A ps over a string of words is then defined as a set of ordered pairs $\langle \Pi, X \rangle$ in which all the $\Pi$'s form a ramification over the string and the $X$'s are categories. Since a psg is three sets: 1) a set of rules for expanding categories, 2) a set of rules associating words with some categories, 3) a set of initial categories, a ps may be assigned to a psg if the following hold: 1) Given a pe associated with a category in the ps, the decomposition of the pe must be associated with categories linked to the former category by a rule in the psg of type 1 above. 2) If a pe associated with a category is unanalyzable
then the word in the pe must be of that category by a rule in the psg of type 2 above. 3) The category assigned to the whole string is in the third set above.

We now have two sorts of structure on strings of words, that defined by dependency relations and that defined by ramification. One way of comparing the structures is to examine the ramification induced by a d-tree, defined thusly: since every occurrence P in a d-tree, together with all occurrences depending directly or indirectly on it form a pe of the string, we may let this pe, \( \Pi(P) \), be a member of the induced ramification \( \Pi' \). If we add no more pes the set \( \Pi' \) will fulfill conditions 1 and 2 for ramifications given above, by properties 3) and 6) of the d-relation. However, to get complete decomposition of pes, we must add to \( \Pi' \) as pes all the single occurrences P which govern some other occurrence. As an example, consider the d-tree

\[
\begin{array}{ccc}
P_1 & & P_2 \\
& P_3 & \\
P_2 & & P_3
\end{array}
\]

We have \( (P_1) = P_2 P_3, (P_2) = P_2, (P_3) = P_3 \) and the set consisting of these pes can be described by:

\[
\begin{array}{ccc}
\Pi(P_2) & & \Pi(P_3) \\
(\ast) & & (\ast) \\
\end{array}
\]

But note \( \Pi(P_1) \) is not completely decomposed. To obtain a ramification, we must add \( P_1 \) as a pe, deriving \( (**)(*)(*) \). It is obvious that the induced ramification does not reflect all the structure of the d-tree, for

\[
\begin{array}{ccc}
P_1 & & P_2 \\
& P_3 & \\
P_2 & & P_1
\end{array}
\]

would induce the same ramification as given above.

To get a pes from the induced ramification, Gaifman associates a category with each pe in that ramification. First, with every \( \Pi(P) \) is associated the category assigned to P in the d-tree. Second, if P governs other occurrences, P itself will be a pe distinct from \( \Pi(P) \). These pes must be assigned different categories or recursion will be introduced into the psg associated with the pes which is not present in the dg. Gaifman does this by associating the category \( x^w \)
(X restricted to a single word) to P if X is associated with (P) \( \neq P \).

Gaifman now induces a psg from a dg by finding a grammar giving all the pss induced by all the d-trees given by a dg. Actually the algorithm derives the psg directly from the rules of the dg in an obvious manner. Gaifman claims that, because of distinguishing between X's and \( X^w \)'s, "although the induced ramification did not express fully the d-relation, the induced ps does." Namely, given an induced ps, the d-tree can be effectively reconstructed by algorithm. What he does not point out, however, is that the reconstruction involves analysis of the supposedly unitary symbol \( X^w \) as a complex symbol in order to identify it with the symbol X. Thus, the reconstruction algorithm uses information extraneous to the psg and involves a strictly local transformation.

To compare psgs and dgs, Gaifman defines a psg and a dg to be equivalent if they have exactly the same ramifications (if the ramifications of all the pss are the same as the induced ramifications of all the d-trees). Gaifman notes this is a qualified sort of equivalence, since induced ramifications do not preserve all the structure of the d-trees. For every dg there is an equivalent psg, the induced psg defined above, expressing, in a certain sense, the full detail of the dg. However, for some psgs there are no equivalent dgs.

A necessary and sufficient condition for a psg to be equivalent to some dg involves the notion of degree of a psg. A category X in a psg is of degree zero, "deg (X) = 0," if only single words are of category X. \( \deg (X) = n \) if deg (X) is not less than n and for every rule of the form \( X \rightarrow Z_1 \ldots Z_k \) there is a \( Z_i \), \( 1 \leq i \leq k \), such that deg (\( Z_i \)) is less than or equal to \( n-1 \). \( \deg (X) = \infty \) if for no \( n \) \( \deg (X) = n \). The degree of a psg is the maximum of the degrees of the categories in the psg. Intuitively, if \( \deg (X) = n \), this means that starting from a node of category X in any tree of the psg, it is impossible to travel downwards through more than \( n \) nodes before
reaching a single word, and, at the same time, the psg must be capable of generating at least one tree where there are n nodes between X and the closest single word lower down in the tree. Gaifman formalizes this in several lemmas. Since degree is actually a property of ramifications (one aspect of tree structure), two equivalent psgs must have the same degree in order to have the same ramifications. It is fairly obvious that if a psg is equivalent to some dg then it must be of degree 0 or 1, for the induced ramifications of d-trees never display any nodes which would have degree greater than one in the induced psg. This implication holds in the other direction also, a result that follows from the main theorem of the paper.

Since equivalence erases some d-tree information, it may be argued, according to Gaifman, that it is not significant that dgs are equivalent to a special class of psgs (those with degree 0 or 1). To counter this, he compares the two grammar types by the notion of correspondence, originally introduced by Hayes, based on the concept of subtrees and complete subtrees in d-trees. A subtree is any set of occurrences connected by the d-relation. A complete subtree is the \( \Pi(P) \) we defined before; an occurrence P with all those occurrences depending directly or indirectly on it. Hayes has the following definition of a ramification corresponding to a d-tree. Every complete subtree is a pe of the ramification (as with Gaifman's induced ramification) and every pe is a subtree. In other words, to construct the corresponding ramification, we begin with the pes defined by the complete subtrees of the d-tree and add to these pes pes which are subtrees until the set of pes constitutes a ramification. In general, there will be more than one ramification corresponding to a d-tree. Gaifman makes the obvious extension of the notion of correspondence to the relation between d-trees and psgs by calling them corresponding if the corresponding d-tree ramification is the same as the ps ramification. If all the pss of a psg correspond respectively to all the d-trees of a dg, the psg and the dg correspond.

- 133 -
The major theorem of the article is this: "A psg corresponds to some dg if it is of finite degree. Moreover, if this condition is fulfilled, then given the psg a dg can be constructed effectively having the following property: Given any d-tree of the dg a unique [italics Gaifman's] ps of the psg which corresponds to it can be constructed effectively and vice versa. (Thus the dg expresses fully the psg and all the passages from one to the other can be done in an automatic way by a computer.)" Unfortunately, the theorem as stated is wrong, for, given a d-tree, in general there will not be a unique corresponding ps, and, given a ps, there will not be a unique corresponding d-tree. Thus there is no algorithm for passage back and forth between dgs and psgs of finite degree.

The first part of the proof of the major theorem shows that if a psg corresponds to some dg then it is of finite degree. This is done by showing that rule length in the corresponding dg must increase as the degree of the psg increases. Since rules must be of finite length, if a psg is of infinite degree, it does not correspond to any dg.

The second part of the proof presents an algorithm to construct for any psg of finite degree a corresponding dg fulfilling the requirements of the theorem. It is this algorithm that fails. Space limitations prohibit presenting the algorithm here. However the dg constructed with the given algorithm will generate several d-trees for a given ps whenever the psg has a rule containing more than one category on the right-hand-side which is of smaller degree than the category on the left-hand-side, and this rule was employed in generating the ps. This contradicts the theorem.

As a concrete example, consider the psg $S \rightarrow NP \ VP$, $NP \rightarrow Det \ N \ VP \rightarrow V \ NP$. Both $NP$ and $VP$ are of degree 1, $S$ is of degree 2. To form a dg from this psg, the decision must be made as to whether the subject noun or the verb is to dominate the sentence and as to whether the determiner of the noun is to dominate the
noun phrase. Whatever information is relevant to these decisions, i.e., government relations, is not given in the psg. How to make these decisions is not discussed in the algorithm presentation.

Also, a procedure is given for deriving the corresponding d-tree from a psg. This procedure also fails, for it uses information irrelevant to the operation of the dg and which was derived in obtaining the dg from a psg. As an example, one of the dgs corresponding to the psg given above is $S(NP \ast NP)$, $NP(Det \ast)$, $Det(\ast)$, where words formerly of category V are now of category S, and words formerly of category N are now of category NP. The first rule yields several possible sets of corresponding psg rules: $S \rightarrow NP X_1 NP$, $X_1 \rightarrow V$ or $S \rightarrow NP X_1$, $X_1 \rightarrow V NP$ (this is the original) or $S \rightarrow X_1 NP$, $X_1 \rightarrow NP V$. Gaifman circumvents this difficulty and chooses a unique corresponding set of rules by using subscripts on categories indicating at what level they appeared in the original pss. Thus, instead of $S(NP \ast NP)$, he has $S(NP_1 \ast NP_2)$, which signals that the subject NP must be introduced before the object NP in the corresponding psg. But this means the unique corresponding psg must be given before it may be algorithmically derived, certainly an unreasonable prerequisite.

It is my own view that the notion of dominance in d-trees and that of level of structure in psgs are incomparable. Gaifman's concept of correspondence fails just as equivalence did in this respect. However, the former concept allows interesting comparison between dgs and psgs. Starting from the following six rule psg $S \rightarrow NP VP$, $NP \rightarrow N (S)$, $NP \rightarrow NP + NP$, $VP \rightarrow V NP (S)$, and using Gaifman's algorithm, one derives a corresponding dg with 119 rules producing non-unique corresponding d-trees. If the verb is taken to dominate the sentence, the dg with unicity has 39 rules. If the noun (or with conjoined NPs) dominates, the dg with unicity has 51 rules. If redundant categories and rules are removed (Gaifman's algorithm introduces redundant categories, which are avoidable with a simple change in the algorithm),
the dg without unicity has 40 rules, the V-dominant dg has 14 rules, and the N-dominant dg has 20 rules. The multiplicity of rules results from the necessity for dependency theory to view the NP in NP → N (S) and the NP in NP → NP + NP as two distinct categories, one resulting in an occurrence of N and one resulting in an occurrence of + in the related d-trees.

The article concludes with an outline of a proof of the theorem that psgs and dgs are equal in weak generative capacity. The proof is based on the fact that every set of sentences definable by a psg is definable by a psg of finite degree. This, together with the major theorem, provides the desired result.

In conclusion, it should be noted that the results of the article have no linguistic relevance. Since we know that psgs fail to characterize natural language in part because of insufficient strong generative capacity, we might think that now the same argument could be used against dgs, and with even more force. However, consider psgs of infinite degree. They are characterized by their containing rules, or sequences of rules, which yield X⁺XX: X⁺XXX, or any other finite number of XXs. Rules of this form are not used in grammars characterizing natural languages. A rule schema of the form S → (S)* subsumes these rules, but, through inclusion of r.h.s.'s which are arbitrarily long, transcends the strong generative capacity of psgs. Actually, then, the degree of a psg is irrelevant to its inability to characterize natural language; restricting ourselves to use of psgs of finite degree, either empirically or logically (by using dgs) neither helps nor hinders this inability.