

Restricted Discontinuous Phrase Structure Grammar and Its Ramifications

Carl M. Vogel^{a*} and Tomaž Erjavec^{b†}

^aUniversity of Edinburgh, Centre for Cognitive Science, 2 Buccleuch Place, Edinburgh EH8 9LW, Scotland
 vogel@cogsci.ed.ac.uk

^bInstitute “Jožef Stefan,” Jamova 39, 61111 Ljubljana, Slovenia
 tomaz.erjavec@ijs.ac.mail.yu

This paper presents a phrase structure grammar called Restricted Discontinuous Phrase Structure Grammars (DPSG^R) designed for representing limited forms of discontinuous constituency (crossed branches). The formalism has only slightly greater than context free generative capacity, yet properly less power than the Mildly Context Sensitive Grammars (MCSGs). The existence DPSG^R suggests that there is not a discrete step between the Context Free Grammars (CFGs) and the MCSGs as suggested by the Joshi conjecture of the convergence of mildly context sensitive formalisms. DPSG^R is adequate for the description of certain discontinuity phenomena, but it is unable to generate important languages like those that express cross-serial dependencies.

1. Introduction

Arguments from linguistics motivate the inclusion of discontinuous constituency in grammatical representations. The expressive power of representations of linguistic structures is enhanced by admitting discontinuous constituency into the representation. Planar directed acyclic graphs (DAGs) are a primary tool of linguistic representation. A substantial literature in linguistics assumes the further restriction of DAGs to trees. This rules out two distinct but related forms of discontinuity from the representation: crossing branches, and multi-dominance. Linguistic motivations for these forms of discontinuity come from Bach (1979, 1980), Jacobson (1987) and McCawley (1982, 1987).

Bunt (1988, 1990, 1991) has proposed a formalism called Discontinuous Phrase Structure Grammar (DPSG) for representing crossing branches. Vogel and Bunt (1992) present a restricted version of DPSG, DPSG^R, which comes closer to fulfilling the linguistic motivations. This restricted formalism is mildly context sensitive, but appears to be less powerful than the other proper members of that class as outlined by Joshi, Vijay-Shanker, and Weir (1989). This paper describes DPSG^R, and indicates the reason for the conjecture that the formalism is less powerful than the other proper members of the mildly context sensitive language (MSCLs). If this result is correct, then DPSG^R constitutes a proof of

*Vogel is grateful to the Marshall Aid Commemoration Commission for supporting his stay in Edinburgh.

†Erjavec is supported by the Ministry of Science and Technology of Slovenia. Both authors are indebted to Fred Popowich and David Milward for constructive feedback, and to Thomas Plaß for technical assistance.

the existence of a proper subclass of the MSCLs. Since $DPSG^R$ is contained in the same class as the mildly context sensitive grammars it inherits a recognition procedure with complexity no worse than $O(n^7)$ (Pollard, 1984).

2. Definitions

As a first step to constructing discontinuous trees it is useful to begin with the usual notion of trees as generated by context free grammars. These are formalized in Defn. 1.

Defn. 1 Trees. *Let N be a nonempty set of nodes;*

1. *if t is a member of N and $[]$ is the empty list, then $\langle t, [] \rangle$ is an (atomic) tree;*
2. *if X_1, \dots, X_k are trees and t is a member of N , then $\langle t, [X_1, \dots, X_k] \rangle$ is a tree.*

In a non-atomic tree, the X_i are referred to as the daughters of the root t . Crossing branches never enter in through the recursive definition. The implicit assumption that none of the X_i that make up a tree are token identical eliminates the possibility for multi-dominance in a context free tree. A formal definition of the direct dominance relation (D) is given in Defn. 2.

Defn. 2 Direct Dominance. *Given a tree $\langle t, [X_1, \dots, X_k] \rangle$, t dominates each X_i (tDX_i). In the tree $\langle t, [] \rangle$, t dominates nothing.*

The transitive closure of the domination relation (D^*) is also defined. In a context free tree a node directly dominates all of its daughters; this is not true of the trees defined below.

Discontinuous Phrase Structure Grammars (DPSGs) were defined by Bunt (1988, 1990, 1991) to generate trees that can have crossing branches. Context free trees are a special case of this discontinuous variety. The definitions in this section revise those of Vogel and Bunt (1992) in generating Restricted DPSGs ($DPSG^R_S$). They proceed as before with recursive specification, but this happens in two stages.

Firstly, subtrees are defined in Defn. 3.

Defn. 3 Subtrees. *Let N be a nonempty set of nodes;*

1. *if t is a member of N and $[]$ is the empty list, then $\langle t, [] \rangle$ is an (atomic) subtree;*
2. *if X_1, \dots, X_k are subtrees or lists of subtrees and t is a member of N , then $\langle t, [X_1, \dots, X_k] \rangle$ is a subtree.*

This is very much like the definition of context free trees, except that the daughter of a node can be a tree or a sequence of trees. If X_i is a sequence of subtrees, then each subtree in that sequence is a *context daughter* of t ; otherwise, the X_i is a *real daughter* of t . We also assume a notion of direct dominance for subtrees as in Defn. 2. Subtrees generalize context free trees by admitting context daughters that are not directly dominated by their context mothers. A useful way of looking at a subtree is in terms of its *node sequence*, which is formalized in Defn. 4.

Defn. 4 Node Sequences.

1. The node sequence of an atomic tree $\langle t, [] \rangle$ is $[t]$.
2. If $\langle t, X \rangle$ is a subtree, then its node sequence is $[t]$ concatenated to the front of the node sequence of X .
3. If X is a list of subtrees, then its node sequence is the sequential concatenation of the node sequences of each subtree in the list.

A form of multi-dominance is allowed in these grammars by the generalization of the dominance relation which stipulates that a node *generally dominates* (*GD*) its real and context daughters. This is formalized in Defn. 5.

Defn. 5 General Dominance (*GD*).

1. In an atomic subtree $\langle t, [] \rangle$ no node is generally dominated by t .
2. In the subtree $\langle t, [X_1, \dots, X_n] \rangle$, t generally dominates each X_i that is a subtree or a member of a sequence of subtrees.

This gives a weak variety of multidominance since a node can be generally dominated by both its real mother and its context mother.

For example, in (1) (cf. Figure 1 for a graphical representation and (2) for the node sequence generated by the tree), the node *paren* is a context daughter of *s*, and it is not *dominated* by *s*.

$$\langle s, [\langle np, [] \rangle, \langle paren, [] \rangle, \langle vp, [] \rangle] \rangle \quad (1)$$

$$[s, np, paren, vp] \quad (2)$$

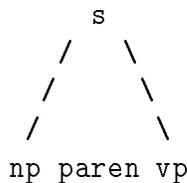


Figure 1. A Discontinuous Subtree.

However, *s* does *generally dominate* *paren*. A node *dominates* only its real daughters, but *generally dominates* all of its daughters, contextual and real. In the second stage of the definition of discontinuous trees (see Defn. 8 below) it will be stipulated that all nodes except the root must be directly dominated (not simply generally dominated) by some other node in the tree. By the way subtrees and node sequences are defined, this means that a discontinuous element will appear twice: the occurrences are token identical.

Another important notion for context free trees is the linear precedence relation on nodes. Definition 6 states the precedence relation in terms of the linear notation for subtrees, and Defn. 7 formalizes the notion of leftmost descendant.

Defn. 6 Precedence (\prec). *If X and Y are subtrees in a subtree R , then $X \prec Y$ iff the leftmost descendant of X precedes the leftmost descendant of Y in the node sequence of R .*

Defn. 7 Leftmost Descendant.

1. *If X is an atomic subtree $\langle t, [] \rangle$ then $Lm(X) = t$.*
2. *If X is a subtree of the form $\langle t, [X_1, \dots, X_n] \rangle$, then $Lm(X) = Lm(X_1)$.*
3. *If X is a list of subtrees, $[X_1, \dots, X_n]$, then $Lm(X) = Lm(X_1)$.*

Proposition 1 \prec is not a partial ordering.

Essentially, the relation lacks the property of antisymmetry. Because of the token-identical duplicate appearance of contextual nodes, a contextual node can both precede and be preceded by portions of the subtree that it skips by virtue of a crossed branch.

These definitions facilitate the second stage of the definition of restricted discontinuous trees given in Defn. 8.

Defn. 8 Restricted Discontinuous trees. *A restricted discontinuous tree is a singly rooted subtree which satisfies the following: Any node c which occurs as a context daughter of some node t must also be directly dominated by some node s , distinct from t , such that sGD^*t and c does not precede t ($c \not\prec t$).*

Unlike the root of a subtree, the root of a discontinuous tree does dominate (D^*) all of its descendants. Discontinuous trees are singly rooted—a sequence of subtrees does not itself constitute a tree. Because subtrees may contain undominated daughters which are dominated in a containing tree, discontinuous trees admit crossing branches. Under this definition discontinuous constituents can cross only to the left. Restricting the direction of movement is computationally desirable, yet the actual direction is arbitrary.

This definition sufficiently constrains the relationship between the principle nodes involved in discontinuous constructions—a context daughter, its context mother, and the real mother of the context daughter—to define a modified string-rewriting grammar formalism to generate members of $DPSG^R$. The RHS of a grammar rule specifies those nodes which may appear as context and real daughters of the symbol on the LHS. A set of rules in this form which generates those trees defined in Defn. 8 is called a $DPSG^R$.

A simple example of a $DPSG^R$ is given in (3).

$$\begin{aligned} S &\rightarrow X + c & (3) \\ X &\rightarrow a + [c] + b \end{aligned}$$

These rules license either of the trees depicted in Figure 2, and given in linear notation in (4 and 5). The coindexing in the discontinuous case (4) denotes token identity of the subtree with the node c . Note that c , the context daughter of X is directly dominated by S , which generally dominates X , and $X \prec c$ without $c \prec X$. The context free case satisfies these same conditions vacuously.

$$\langle S, [\langle X, [\langle a, [] \rangle, [\langle c, [] \rangle^1], \langle b, [] \rangle] \rangle, \langle c, [] \rangle^1 \rangle \quad (4)$$

$$\langle S, [\langle X, [\langle a, [] \rangle, \langle b, [] \rangle] \rangle, \langle c, [] \rangle] \rangle \quad (5)$$

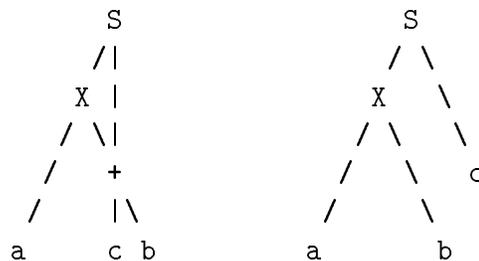
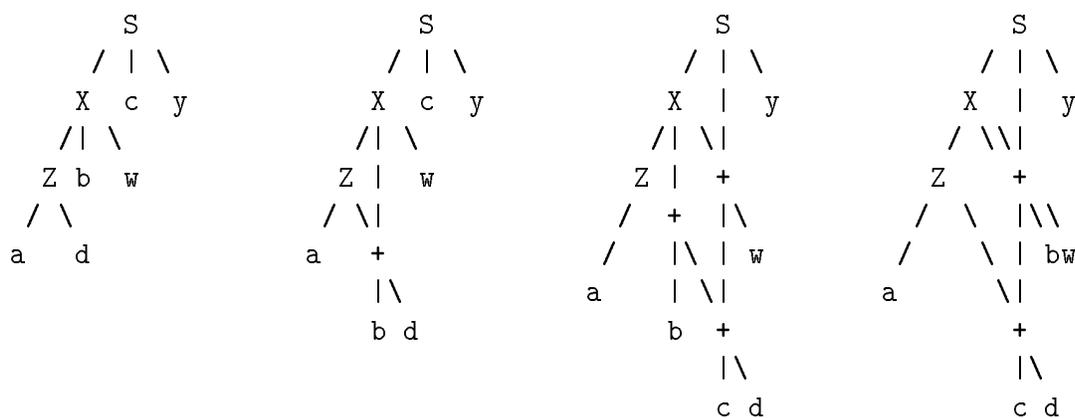


Figure 2. A Discontinuous Tree and Related Context Free Tree

3. More Examples

Figure 3. “Movements” Allowed by an $DPSG^R$: A Node May Realize Some or All of Its Context Daughters.

Consider the $DPSG^R$ given in (6); it generates all of the trees in Figure 3.

$$\begin{aligned}
 S &\rightarrow X + c + y & (6) \\
 X &\rightarrow Z + b + w \\
 Z &\rightarrow a + [b] + [c] + d
 \end{aligned}$$

The third of these is the most complex. Both b and c are realized as context daughters of Z . The node b is directly dominated by X , and since X directly dominates Z , X generally dominates Z . The node c is directly dominated by S which also generally dominates Z . But, it is important to note that the grammar does not generate the tree whose terminal sequence is $\langle abdcwy \rangle$ since c cannot occur as a context daughter to X . A node (and its constituency) can “move” to only those positions where it is specified as a potential context daughter. In this example, the context mother had to specify exactly the number

of daughters that could appear in its context. Using recursive contexts, allowing a context daughter to have its own context daughters, avoids this specificity.

For example, (7) gives a grammar which utilizes a recursive context on X s: X appears as a context daughter to another instance of X which can itself appear in moved or unmoved position. It is important that token identity in the linear notation for a tree constructed from this grammar is only between the contextual occurrence of X and the position it moves from; there is no coindexing between X s that are in contextual mother-daughter relations to each other.

$$\begin{aligned}
 S &\rightarrow [] & (7) \\
 S &\rightarrow a + S + X + c \\
 X &\rightarrow b + [X] \\
 X &\rightarrow b
 \end{aligned}$$

The grammar given in (7) is closely related³ to the non-context free language $a^n b^n c^n$. A tree which realizes the structure of $a^n b^n c^n$ is depicted in Figure 4.

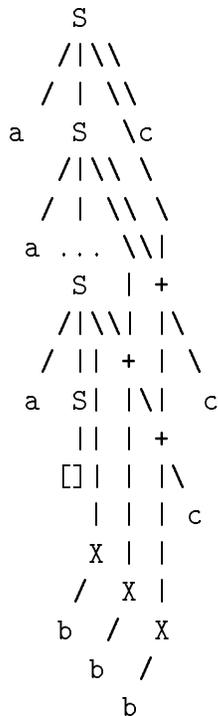


Figure 4. $a^n b^n c^n$

There are limits to the generative capacity of $DPSG^R$. The trees shown in Figure 5 are not generated by any $DPSG^R$. Plausible rule sets can be given as in (8–10):

³The language it generates is actually, $a^n b^i (c^j (bc)^k)^l$ where $(j+k) * l = n, i+k * l = n, l \geq 1$. The intersection of this language with the regular language $a^* b^* c^*$ yields $a^n b^n c^n$ (Vogel & Bunt, 1992).

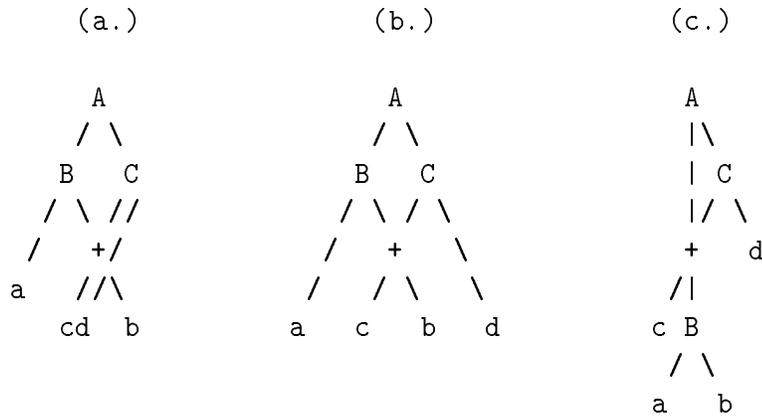


Figure 5. These Are Not DPSG^R Trees.

$$\begin{array}{lll}
 A \rightarrow B + C & (8) & A \rightarrow B + C & (9) & A \rightarrow B + C & (10) \\
 B \rightarrow a + [c, d] + b & & B \rightarrow a + [c] + b & & B \rightarrow a + b & \\
 C \rightarrow c + d & & C \rightarrow c + [b] + d & & C \rightarrow c + [B] + d &
 \end{array}$$

But the trees ‘generated’ are ruled out: in (a.) (grammar 8) the real mother of the context daughters does not generally dominate the context mother; the same is true of both nodes c and b in (b.) (grammar 9); in (c.) (grammar 10) B is a context daughter of C , and B is directly dominated by A which generally dominates B , but the precedence restriction is violated since $B \prec C$. This last case demonstrates that “movement” can happen in only one direction. A real daughter can occur contextually to the left of the position where it is expected as a real daughter, but it may not occur to the right of that position.

A final negative example is given in (11) as a potential grammar for generating the structure shown in Figure 6.

$$S \rightarrow a + [c] + b + c \quad (11)$$

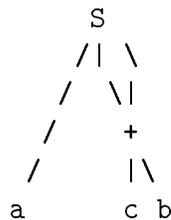


Figure 6. Within-Level Shuffling Is Outside DPSG^R .

This is ruled out because the real and context mothers involved, the single node S , are

not distinct nodes. However, the limits on the generative capacity of $DPSG^R$ are exactly what make the formalism interesting as a greater than context free formalism.

4. Application

$DPSG^R$ provides adequate descriptive means for capturing a range of discontinuous constituent structures that involve optional “movement” as in verb-particle constructions. The formalism offers descriptions that are reminiscent of Bach’s (1979, 1980) right-wrap operation which he used in analyzing transitive verb phrases (TVPs) with arguments about the simplicity of restrictions on passivization when, for instance, “John” is seen as a discontinuous constituent in the intransitive verb phrase, “persuade John to go” (in which “persuade to go” is a TVP). But, $DPSG^R$ is not equivalent to right-wrap since Bach’s operation is restricted to apply only to a single discontinuous element, while there is no such a priori restriction on the number of context daughters a node in an $DPSG^R$ may have. Also, Bach’s right-wrap is obligatory, while the $DPSG^R$ equivalent cannot be forced, but this optionality seems appropriate given that wrapping does not occur in cases of heavy NPs. However, the restriction in $DPSG^R$ on direction of movement inhibits fully satisfactory representation of sentence adjuncts since their optional movement is in the opposite direction to that required by verb-particles and TVPs. The directional restriction seems adequate for the latter phenomena but not for sentential adjunct positioning.

5. Complexity

$DPSG^R$ has greater than context free generative capacity (Vogel & Bunt, 1992) but is unable to capture the string duplicating language that idealizes cross-serial dependencies. If the intuition is correct then $DPSG^R$ generates a proper subclass of the mildly context sensitive languages (cf. Joshi et al., 1989; Vijay-Shanker, Weir, & Joshi, 1987). Vijay-Shanker et al. (1987) characterize an infinite hierarchy of languages beginning with the context free languages from the point of view of generators, automata, and grammars. In this hierarchy, context free languages are level 1 languages, and Tree Adjoining Languages (TALs) are level 2. Vijay-Shanker et al. (1987) show characteristic languages for the i^{th} class in the hierarchy as i -copying $\{w|w \in \{a,b\}^*\}$ and $2i$ -counting $\{a_1^n \dots a_i^n | n \geq 0\}^4$. Vijay-Shanker, Weir, and Joshi (1986) label the languages in the second level of this hierarchy the Mildly Context Sensitive Languages (MCSLs) and show the convergence of a number of grammar formalisms in generating these languages.

Yet, $DPSG^R$ is difficult to place in this hierarchy since it is not context free, yet neither can it generate the second level copying language. To appreciate the ground for this intuition consider that any $DPSG^R$ devised which takes advantage of crossing branches to generate the string ww will also generate a tree in which those branches are uncrossed. Thus, in most cases, as well as generating ww , a $DPSG^R$ will overgenerate to a lot of other strings that do not fit that form as well. If the grammar does not utilize crossing branches, then the $DPSG^R$ will be equivalent to a context free grammar, and clearly the string duplicating languages (over alphabets of more than one symbol) are not context

⁴Context free languages can count to 2, $(a_1^n a_2^n)$ and TAL can count to 4, $(a_1^n a_2^n a_3^n a_4^n)$, while Indexed Languages (ILs) can count to arbitrary i .

free. Thus, in order to generate exactly ww , it is necessary to enrich the formalism. In particular, a stronger notion of multidominance as in phrase-linking grammars may be required to capture the cross-serial dependencies.

It is possible to give a characterization of $DPSG^R$ in terms of the hierarchy used by Vijay-Shanker et al. (1987), but it requires recognition of intermediate levels in the hierarchy. Level 2 control grammars can of course be given that yield the same languages for corresponding $DPSG^R$ s. However, a level 2 control grammar has more descriptive power. For a simple example, consider again the $DPSG^R$ in (3). By giving an appropriate control grammar as in (12) it is possible to define a set of labeled context free productions and a companion context free grammar whose strings are *control words* that select productions from the prior grammar to apply at any given point in the analysis.⁵

$$\begin{array}{l} \text{Base CFG} \quad \{l_1 : S \rightarrow Xc, l_2 : S \rightarrow X, l_3 : X \rightarrow acb, l_4 : X \rightarrow ab\} \\ \text{Control CFG} \quad \{S' \rightarrow l_1l_4, S' \rightarrow l_2l_3\} \end{array} \quad (12)$$

The two control words are thus l_1l_4 and l_2l_3 . This means that the strings of the controlled CFG are those that use the production labeled by l_1 followed by the production labeled l_4 and the strings that use the production labeled by l_2 followed by the production labeled l_3 —respectively, abc and acb . This corresponds to the language generated by the grammar in (3), however it would equally have been possible to specify the controlled CFG as in (13).

$$\begin{array}{l} \text{Base CFG} \quad \{l_1 : S \rightarrow Xc, l_2 : S \rightarrow X, l_3 : X \rightarrow acb, l_4 : X \rightarrow ab, \} \\ \text{Control CFG} \quad \{S' \rightarrow l_2l_3\} \end{array} \quad (13)$$

In that case, the controlled CFG expresses what $DPSG^R$ cannot: obligatory movement. Since the languages in the second level of this infinite hierarchy of languages properly contain the languages generated by $DPSG^R$ s, it suggests examining ways in which intermediate levels in the hierarchy can be specified, particularly between the first and second levels.

A current theme in our work on $DPSG^R$ is the formal articulation of a complete normal form and a pumping lemma to use in adversarial arguments that show particular languages outside the capacity of $DPSG^R$. Although by the result claimed here we might be able to hope for a relatively straightforward pumping lemma, we expect the final version to be by cases as is Roach's (1987) for Head Grammars (HGs).

MCSLs are identified with the second level of the infinite hierarchy of languages that begins with CFLs at the first level. Vijay-Shanker et al. (1986) have proven the containment of HGs in Tree Adjoining Grammars and equivalence for a trivial modification of HGs. Both of these formalisms are able to generate the string duplicating language. Since $DPSG^R$ would have to be enriched to capture the language, this suggests that $DPSG^R$ is a member of a class of languages that is larger than the context free but smaller than the mildly context sensitive. As a member of the mildly context sensitive languages $DPSG^R$ inherits a recognition procedure that is no worse than $O(n^7)$, but since the containment appears proper, a better result may be available.⁶

⁵Grammars that actually utilize recursion are of course more interesting than the one in (12).

⁶A parser for $DPSG^R$ has been implemented in Prolog, but we make no claims about its efficiency.

References

- Bach, E. (1979). Control in Montague Grammar. *Linguistic Inquiry*, 10, 515–531.
- Bach, E. (1980). In Defense of Passive. *Linguistics and Philosophy*, 3, 297–341.
- Bunt, H. (1988). DPSG and its Use in Sentence Generation from Meaning Representations. In *Advances in Natural Language Generation*. Pinter Publishers, London.
- Bunt, H. (1990). Disco-Trees and their Use in Parsing. In *Proceedings of the Symposium on Discontinuous Constituency*, pp. 7–19. Institute for Language Technology and Artificial Intelligence, Tilburg University, Holland.
- Bunt, H. (1991). DPSG and its Use in Parsing. In *Current Issues in Parsing Technology*. MIT Press, Cambridge, MA.
- Jacobson, P. (1987). Phrase Structure, Grammatical Relations, and Discontinuous Constituents. In Huck & Ojeda (Eds.), *Discontinuous Constituency*, pp. 27–69. Academic Press.
- Joshi, A. K., Vijay-Shanker, K., & Weir, D. (1989). The Convergence of Mildly Context-Sensitive Grammar Formalisms. Tech. rep. MS-CIS-89-14; LINC LAB 144, Department of Computer and Information Science University of Pennsylvania, Philadelphia, PA.
- McCawley, J. D. (1982). Parentheticals and Discontinuous Constituent Structure. *Linguistic Inquiry*, 13, 91–106.
- McCawley, J. D. (1987). Some Additional Evidence for Discontinuity. In Huck & Ojeda (Eds.), *Discontinuous Constituency*, pp. 27–69. Academic Press.
- Pollard, C. (1984). *Generalized Phrase Structure Grammars, Head Grammars, and Natural Languages*. Ph.D. thesis, Stanford University, CA.
- Roach, K. (1987). Formal Properties of Head Grammars. In *Mathematics of Language*. J. Benjamins Pub. Co., Amsterdam.
- Vijay-Shanker, K., Weir, D., & Joshi, A. (1986). Tree Adjoining and Head Wrapping. In *11th International Conference on Computational Linguistics*. Bonn University, West Germany.
- Vijay-Shanker, K., Weir, D., & Joshi, A. (1987). On The Progression from Context-Free to Tree Adjoining Languages. In *Mathematics of Language*. J. Benjamins Pub. Co., Amsterdam.
- Vogel, C. & Bunt, H. (1992). Representing Discontinuous Constituency: Motivations and Complexity. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Sept 16-18, 1992*. Seoul, South Korea.