On Scrambling, Another Perspective

James Rogers
Dept. of Computer Science
Earlham College
Richmond, IN USA
jrogers@cs.earlham.edu

Abstract

We explore the possibility of accounting for scrambling patterns in German using multi-dimensional grammars. The primary desirable characteristic of this approach is that it employs elementary structures with a single uniform component and combining operations which operate at a single point in the derived structure. As a result, we obtain an analysis that is much closer in spirit to ordinary TAG and to the intuitions of TAG based linguistics. Ultimately, we obtain an account in which the variations in word order are consequences of variations of a small set of structural parameters throughout their ranges.

1 Introduction

The difficulty of accounting for the phenomenon of scrambling, the apparently arbitrary order in which arguments can occur in subordinate clauses in German, has been one of the primary motivations for exploration of extensions to TAG (Rambow, 1994; Kulick, 2000; Rambow et al., 1995; Rambow et al., 2001; Becker et al., 1991). The issue is not generation of the string sets—in most accounts these are actually context-free—but rather doing so within a derivational framework in which lexical heads and their arguments are introduced simultaneously. This idea that elementary structures should include all and only a single thematic domain (Frank, 2002) is generally taken to be the foundation of TAG based linguistic theories. Among other things, it insures that every elementary structure is semantically coherent and that derivations maintain that coherence. Under these assumptions, it has been shown that scrambling is beyond the generative power of ordinary TAG and, in full generality, beyond even set-local multicomponent TAG (Becker et al., 1992).

Generally, extensions to TAG intended to accommodate scrambling involve factorization of elementary structures either into tree sets or into trees with more or less independent regions accompanied by a modification of the combining operation to interleave these regions in the derived tree. In this paper, following the lead of our exploration of similar issues in TAG accounts of English raising phenomena (Rogers, 2002), we explore one illustrative pattern of scrambling using *multi-dimensional grammars* (Rogers, 2003). The primary desirable characteristic of this approach is that it employs elementary structures with a single uniform component and combining operations which operate at a single point in the derived structure. As a result, we obtain an analysis that is much closer in spirit to ordinary TAG and to the intuitions of TAG based linguistics. Ultimately, we obtain an account in which the variations in word order are consequences of variations of a small set of structural parameters throughout their ranges.

We should be clear at the outset that even though our primary motivation is a desire to preserve the fundamental tenets of standard TAG theories of syntax, our goal is not a linguistically complete account of scrambling, or even one that is linguistically motivated beyond the goal of maintaining semantically coherent elementary structures and derivations. Rather, we intend to show how the formal power of the multi-dimensional grammars can, potentially, support such an account. We look only at one particular case of scrambling, but we believe that this case illustrates the relevant formal issues. These results suggest that scrambling phenomena of any concrete degree of complexity can be handled at some level of the multi-dimensional grammar hierarchy. We close the paper with some speculation about what such a result might have to say about the nature of limits on the acceptability of scrambling as its complexity increases.

2 A Formalized Instance of Scrambling

The case we examine, taken directly from Joshi, Becker and Rambow (Joshi et al., 2000) (also (Becker et al., 1991)), involves scrambling within a matrix clause headed by a verb that subcategorizes for two NPs and an

TAG+7: Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms.
May 20-22, 2004, Vancouver, BC, CA.
Pages 178-185.

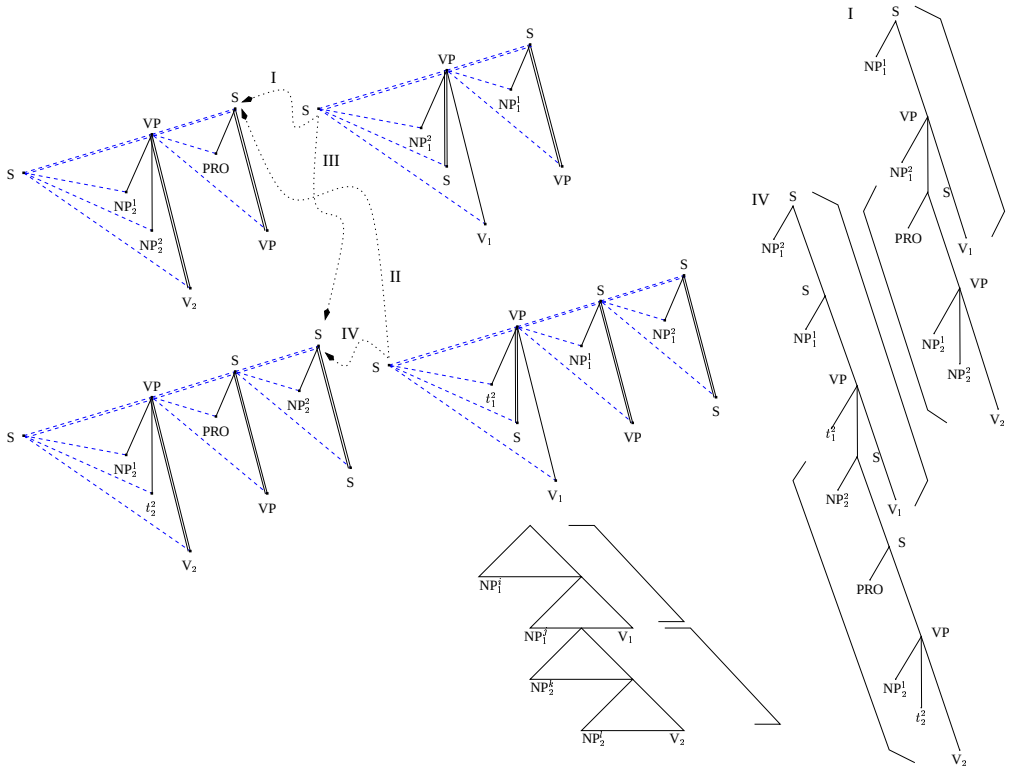


Figure 1: Class A) $\mathrm{NP}_1^i \mathrm{NP}_1^j \mathrm{NP}_2^k \mathrm{NP}_2^l \mathrm{V}_2 \mathrm{V}_1$

S with that embedded S headed by a verb that subcategorizes for three NPs, one of which is a PRO subject. This formalizes as

$$\{\sigma(\mathrm{NP}_1^1, \mathrm{NP}_1^2, \mathrm{NP}_2^1, \mathrm{NP}_2^2)\mathrm{V}_2\mathrm{V}_1 \mid \sigma \text{ a permutation}\}$$

where NP_1^1 and NP_1^2 are the first and second argument of the matrix verb and NP_2^1 and NP_2^2 are the arguments, other than PRO, of the embedded clause.

There are 24 permutations of the four NP arguments. We divide these into six classes (where i, j and k, l each vary over 1,2):

A) $\mathrm{NP}_1^i \mathrm{NP}_1^j \mathrm{NP}_2^k \mathrm{NP}_2^l \mathrm{V}_2 \mathrm{V}_1$
B) $\mathrm{NP}_2^k \mathrm{NP}_1^i \mathrm{NP}_1^j \mathrm{NP}_2^l \mathrm{V}_2 \mathrm{V}_1$
C) $\mathrm{NP}_2^k \mathrm{NP}_2^l \mathrm{NP}_1^i \mathrm{NP}_1^j \mathrm{V}_2 \mathrm{V}_1$
D) $\mathrm{NP}_1^i \mathrm{NP}_2^k \mathrm{NP}_2^l \mathrm{NP}_1^j \mathrm{V}_2 \mathrm{V}_1$
E) $\mathrm{NP}_2^k \mathrm{NP}_1^i \mathrm{NP}_2^l \mathrm{NP}_1^j \mathrm{V}_2 \mathrm{V}_1$
F) $\mathrm{NP}_1^i \mathrm{NP}_2^k \mathrm{NP}_1^j \mathrm{NP}_2^l \mathrm{V}_2 \mathrm{V}_1$.

3 Class A)—CF Structures

Class A) is the canonical structure with, potentially, the arguments extracted within their own clauses. Standard TAG accounts treat the matrix clause as an auxiliary tree adjoining at the root of the embedded clause. Following Rogers (1998) and Rogers (1999) we interpret TAG as a sort of Context-Free Grammar over trees. Just as CFG productions can be interpreted as *local* (height one) trees expanding a root node to a string yield with the derivation process splicing these together to form derivation trees, TAG auxiliary trees can be interpreted as local three-dimensional structures expanding a root node to a tree yield with the TAG derivation process splicing these together to form three-dimensional derivation structures.

These derivation structures correspond exactly to the derivation trees normally associated with TAG, with the exception that the derived structure (in this case a tree) can be obtained from it by restricting to nodes of maximal depth (in the third dimension) in a way analogous to taking the string yield of a CFG derivation tree. The intuition behind these structures is that TAG expresses a hierarchical decomposition of trees analogous to the hierarchical decomposition of strings that those trees represent.

It is important not to misconstrue this notion of "dimension." While it may be convenient to visualize these structures as having actual extent in the third dimension, they are, fundamentally, just graphs with multi-sorted edges and, hence, dimensionless. The three dimensions correspond to linear precedence, ordinary domination and domination of the "adjoining" sort. These are not arbitrary or independent. As they represent recursive hierarchical decomposition, each edge relation is "tree-

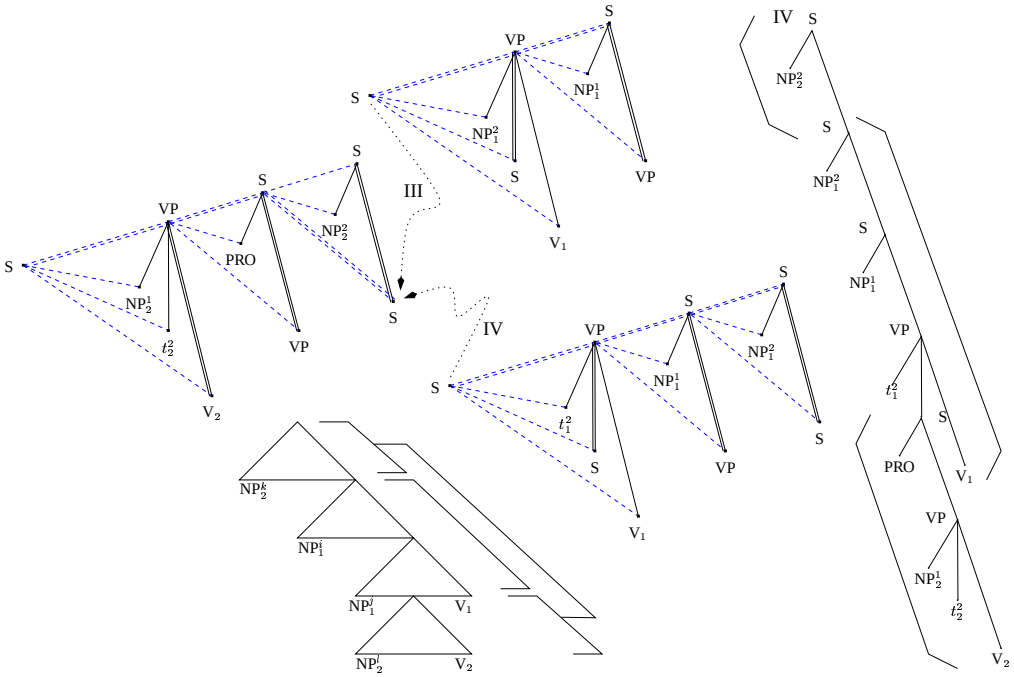


Figure 2: Class B) $\mathrm{NP}_2^k \mathrm{NP}_1^i \mathrm{NP}_1^j \mathrm{NP}_2^l \mathrm{V}_2 \mathrm{V}_1$

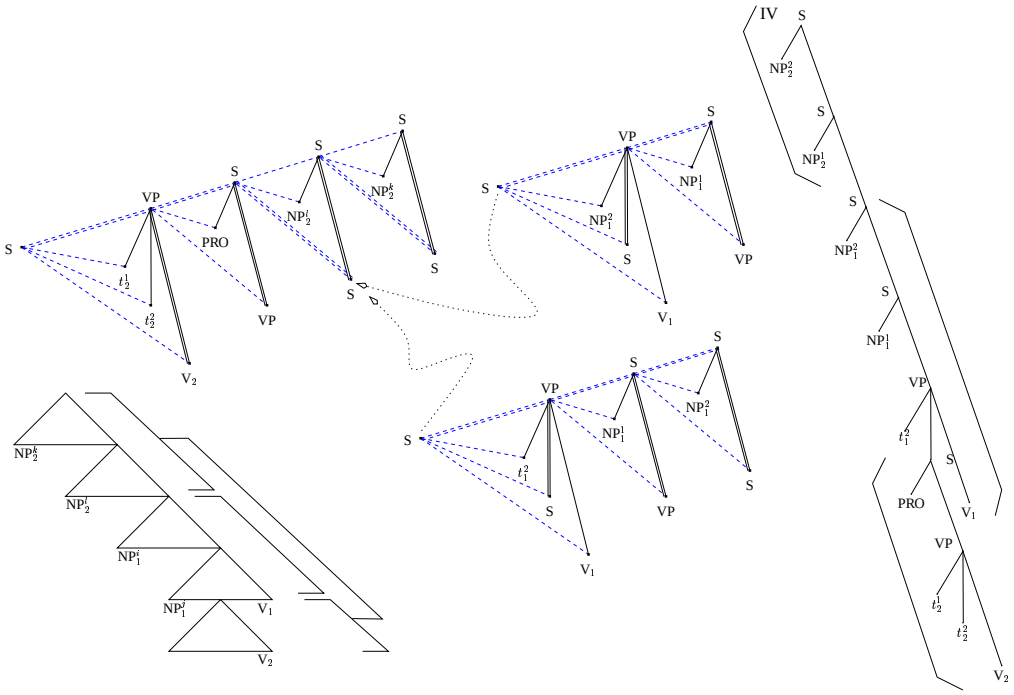


Figure 3: Class C) $\mathrm{NP}_2^k \mathrm{NP}_2^l \mathrm{NP}_1^i \mathrm{NP}_1^j \mathrm{V}_2 \mathrm{V}_1$

like" and descendants of a node along a given dimension inherit the relationships of that node in all smaller dimensions in the same way that linear precedence is inherited by the nodes in a subtree. (See Rogers (2003).)

Here, the adjunction of the matrix tree at the root of the embedded tree attaches the root of the three dimensional structure representing the former at the S node in the (tree) yield of the latter. (See Figure 1.) The upper pair of structures represent the base structures, the lower pair the structures with a locally extracted argument. Each of the four variations of Class A) is obtained from one of the four combinations of these four structures. We have followed Rogers (2002) in treating the subject as if it were adjoined, in some sense, at the root of the VP. This is done, here, for purely formal reasons—it will provide the structural flexibility we will need to derive the more complex scrambling patterns. We carry this structural configuration through in treating extraction; we assume that extracted arguments attach in a similar fashion to the root of the S.

Note that there is potential ambiguity in taking the tree yield of these structures in that the yield of one component might attach at any of the leaves of the yield of the component which immediately dominates it. In TAG, of course, this is resolved by designating one of the leaves as the *foot* with all splicing being done at the foot. Here we designate the foot by marking the *spine* of the components with doubled lines. We carry this through in higher dimensions, as well; each elementary structure, in each dimension, has a spine leading from its root to a foot node in its yield. Two of the four resulting tree yields are shown on right of the figure.

Since adjunction at the root has the same effect as substitution, this is effectively a context-free structure. As shown schematically in the figure, the (two dimensional) yields of the two structures are simply concatenated. Note that, as in the standard TAG accounts, additional recursion is accommodated by adjoining additional subordinating structures at the root of what is here the matrix structure.

4 Classes B) and C)—Ordinary Adjunction

In Classes B) and C), the arguments of V_2 are wrapped around those of V_1, as shown at the bottom of Figures 2 and 3. This is the pattern corresponding to adjunction proper. Class B) can be obtained by extracting either NP_2^1, NP_2^2 then adjoining the matrix structure at the foot of the yield of the extracted NP structure (the point between the extracted element and the original S node). As usual, this has the effect of splitting the tree yields of the subordinate structure into two factors and inserting the tree yield of the matrix structure between them. Note that all that distinguishes this class from Class A) is the third-dimensional foot node of the embedded structure, which is, itself, determined by the form of the extracted NP structure.

Class C) is nearly identical. We extract both of the arguments of V_2 and adjoin, again, at the point between the extracted elements and the original S node.

Note that in both these cases, the scrambling can apply recursively by attaching additional auxiliary structures at the tree yield of the first. If this is attached at the node corresponding to the root of the yield of that structure, in the manner of Class A), the effect is only to move the arguments scrambled out of the more deeply embedded clauses across the new clause. If, on the other hand, it is attached at the foot of the yield of the extracted NP structure, in the manner of Classes B) and C), then the effect will be to scramble additional arguments out of the intermediate clause.

5 Class D)—Higher-Order Adjunction

Class D) is the first of the configurations that cannot be obtained by ordinary adjunction. Here the arguments of V_1 and V_2 don't simply nest one inside the other, but, rather interleave in the way shown schematically on the top left of Figure 4. Since the sequences of labels along the spines of TAG tree sets must form CF languages, such "cross-serial" configurations cannot be generated by TAGs. They can, however, be generated if we add another level of hierarchical decomposition. Grammars at this level yield tree sets with TAL spine languages (corresponding to the third level of Weir's Control Language Hierarchy (Weir, 1992)). A schematic representation of the general embedding pattern provided at this level is given in Figure 5.

To employ this nesting pattern we adopt four-dimensional structures and take the matrix structure to, again, attach between the extracted structure and the original S, but now along the fourth dimension. (Figure 4.) This has the result of splitting the three-dimensional yield of the embedded structure into two factors and inserting the three-dimensional yield of the matrix structure between them. Given the third-dimensional foot of the matrix structure, the "upper" factor of the embedded structure is, effectively, adjoined between the two arguments of the matrix structure which is, in turn adjoined at the root of the "lower" factor. (As shown at the bottom of the figure.) The effect on the tree yield is exactly as if the (two-dimensional) matrix tree had been factored into two components, one adjoining at the root of the embedded tree and one properly along its spine. (As shown at the right.)

It should be noted that with this configuration we can account for all the variations of Classes A) through D) by varying the position of the foot of the matrix structure and

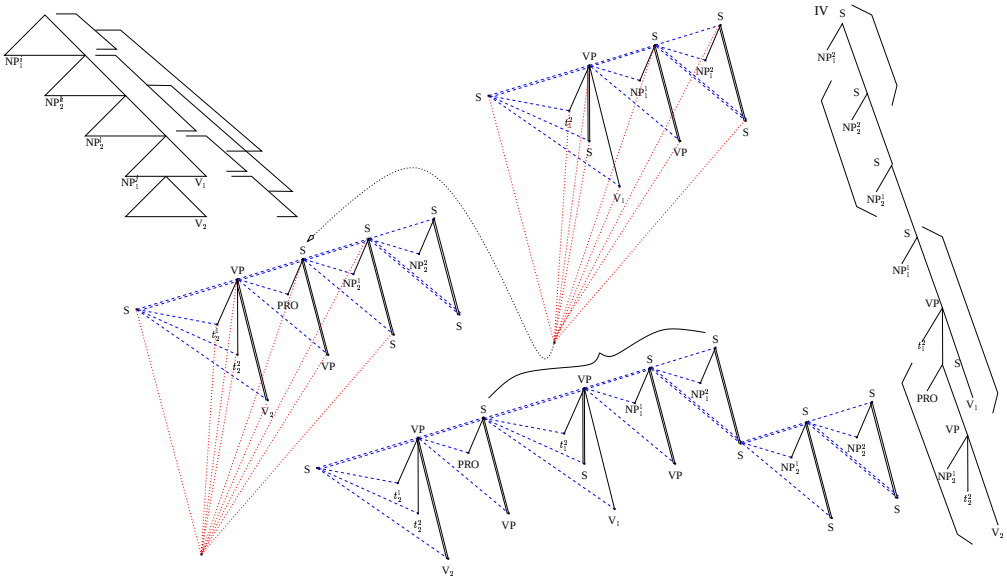


Figure 4: Class D) $\mathrm{NP}_1^i \mathrm{NP}_2^k \mathrm{NP}_2^l \mathrm{NP}_1^j \mathrm{V}_2 \mathrm{V}_1$

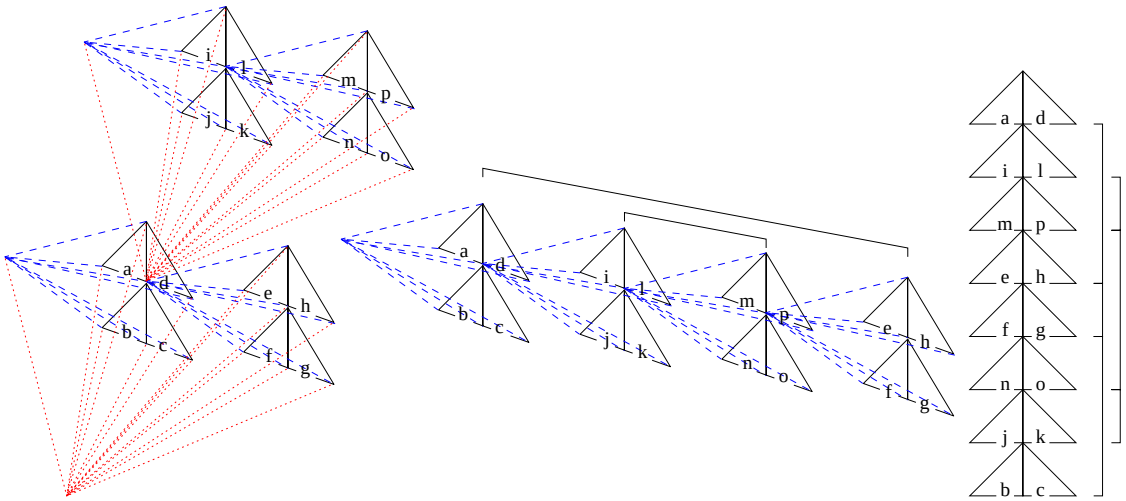
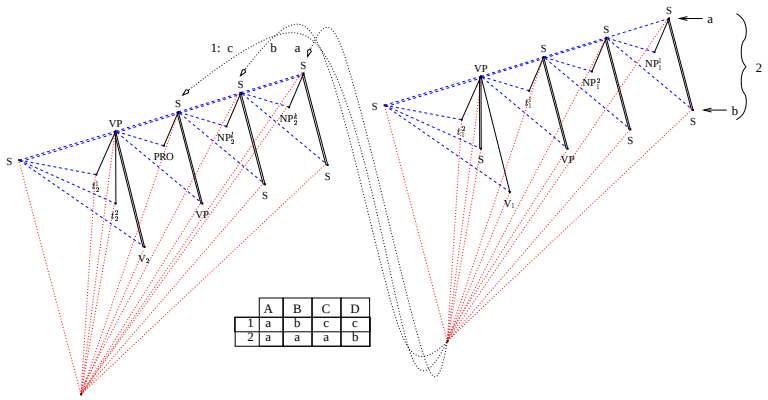


Figure 5: 4^{th}-level nesting



	A	B	C	D
1	a	b	c	c
2	a	a	a	b

Figure 6: A unified account of Classes A) through D)

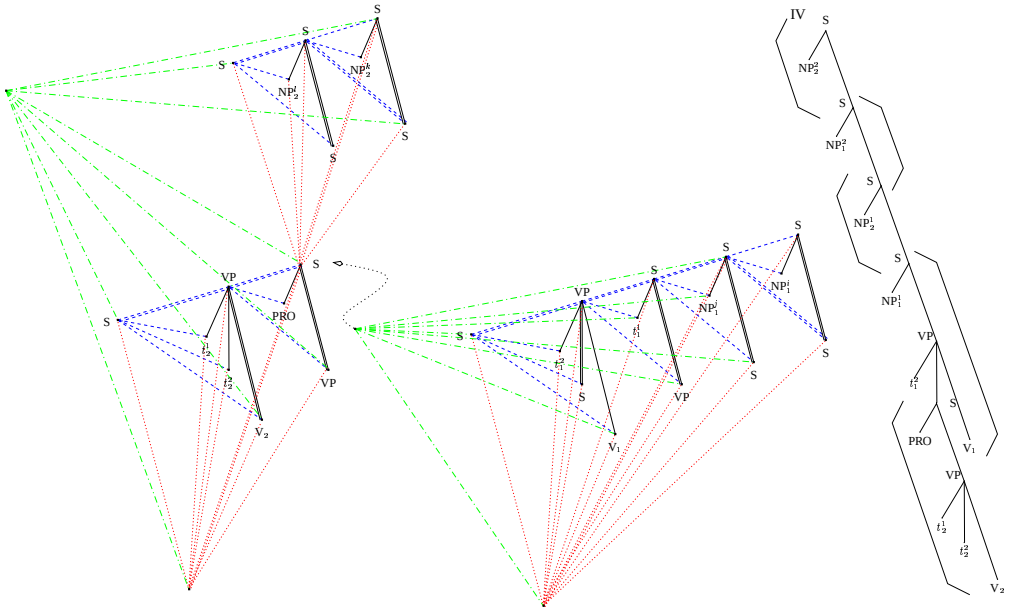


Figure 7: Class E) $\mathrm{NP}_2^k \mathrm{NP}_1^i \mathrm{NP}_2^l \mathrm{NP}_1^j \mathrm{V}_2 \mathrm{V}_1$

the point at which it attaches to the embedded structure. (See Figure 6.)

6 Classes E) and F) and a Unified Account

The nesting pattern of Class E) (Figure 7) requires the embedded tree to be factored into three components, not just two. This can still be obtained with multi-component adjoining in a manner similar to that of Class D), with both components of the matrix structure adjoining properly along the spine of the embedded tree. While this pattern is also obtainable in the four-dimensional grammars, it necessarily uses the upper half of the VP component, which, for the sake of consistency, we would rather not do. Consequently, we again add another hierarchical relationship and lift to the fifth level, taking the two arguments of V_2 to be extracted via the fourth relation rather than the third. Class F) can be treated similarly, with the exception that one of the arguments is extracted along the third relation, the other along the fourth. (Figure 8.)

This variation between Classes E) and F) leads to an account in which all six classes are derivable within a single basic structure, shown in Figure 9. Here there are six parameters of variation:

1. The position of the fourth-dimensional foot of the matrix structure.

2. Whether one or both arguments of V_1 are extracted along the fourth relation.

3. and 4. The position of the three-dimensional feet of the matrix and embedded structures.

5. and 6. And, finally, the relative nesting of the extracted NPs in each structure.

This gives 96 combinations but as the word-order variations are exhaustive, they generate only the 24 distinct configurations of the six classes of structures.

7 Arbitrarily Complex Scrambling

While no level of the multi-dimensional grammar hierarchy can capture scrambling of arbitrary complexity, there is no bound on the number of tree factors that can be interleaved at some level of this hierarchy. In general, grammars at the k^{th} level factor the tree yields of the elementary structures into 2^{k-2} fragments, with the tree yield of the result of adjoining one into another being split into $2^{k-3}+1$ regions from the initial structure interleaved with 2^{k-3} regions from the auxiliary structure. (Figure 10 gives the pattern for the fifth level.) Consequently, scrambling of any concrete degree of complexity can be captured at some level of the hierarchy, although it is not clear that this can necessarily be done in an plausibly "uncontrived" way.

In Joshi et al. (2000), Joshi, Becker and Rambow note that the boundary of general acceptability in scrambling, roughly two levels of embedding, coincides with what can be handled by tree-local MCTAG. This leads them to suggest that this boundary may actually be competence based, rather than performance based as is usually assumed. Here we have additional flexibility. In choosing the level of the competence grammar in the multi-dimensional hierarchy, we set the boundary on the complexity of the scrambling we admit. On the other hand,

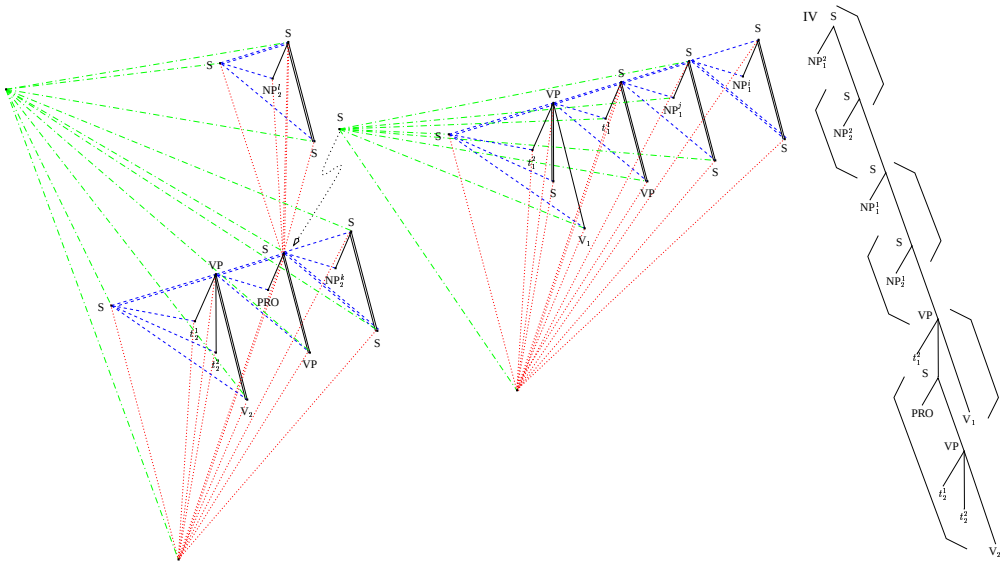


Figure 8: Class F) $\mathrm{NP}_1^i \mathrm{NP}_2^k \mathrm{NP}_1^j \mathrm{NP}_2^l \mathrm{V}_2 \mathrm{V}_1$

given that the level of the grammar corresponds to the number of hierarchical relations we use in encoding the structure of the utterances, one could make a plausible argument that the level of the grammar might be determined by performance considerations, such as working memory limitations. In this way one might arrive at an account of the limits on the complexity of scrambling that was simultaneously performance based—a consequence of bounds on working memory—and competence based—a consequence of the complexity of the grammars which can be processed within those bounds.

References

Tilman Becker, Aravind K. Joshi, and Owen Rambow. 1991. Long-distance scrambling and tree adjoining grammars. In *Fifth Conference of the European Chapter of the Association for Computational Linguistics (EACL'91)*, pages 21–26. ACL.

Tilman Becker, Owen Rambow, and Michael Niv. 1992. The derivational generative power of formal systems or scrambling is beyond LCFRS. Technical report, Institute for Research in Cognitive Science, Univ. of Pennsylvania, Philadelphia, PA.

Robert Frank. 2002. *Phrase Structure Composition and Syntactic Dependencies*. MIT Press.

Aravind K. Joshi, Tilman Becker, and Owen Rambow. 2000. Complexity of scrambling: A new twist to the competence-performance distinction. In Anne Abeillé and Owen Rambow, editors, *Tree Adjoining Grammars*, chapter 6, pages 167–181. CSLI.

Seth Kulick. 2000. *Constraining Non-Local Dependencies in Tree Adjoining Grammar: Computational and Linguistic Perspectives*. Ph.D. thesis, Univ. of Pennsylvania.

Owen Rambow, K. Vijay-Shanker, and David Weir. 1995. D-tree grammars. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL'95)*, pages 151–158, Cambridge, MA.

Owen Rambow, K. Vijay-Shanker, and David Weir. 2001. D-tree substitution grammars. *Computational Linguistics*, 27(1):87–121.

Owen Rambow. 1994. *Formal and Computational Aspects of Natural Language Syntax*. Ph.D. thesis, Univ. of Pennsylvania.

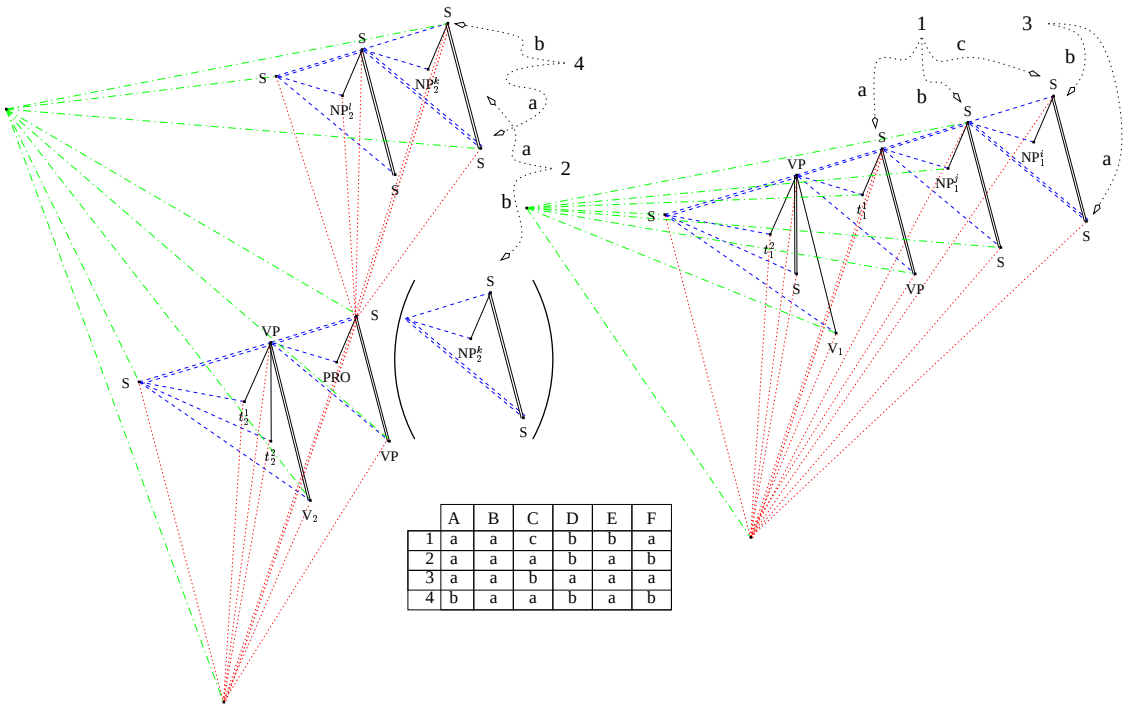
James Rogers. 1998. A descriptive characterization of tree-adjoining languages. In *Proc. of the 17th International Conference on Computational Linguistics (COLING'98) and the 36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*, Montreal. ACL. Project Note.

James Rogers. 1999. Generalized tree-adjoining grammar. In *Sixth Meeting on Mathematics of Language*, pages 189–202.

James Rogers. 2002. One more perspective on semantic relations in tag. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammars and Related Frameworks*, Venice, IT, May.

James Rogers. 2003. wMSO theories as grammar formalisms. *Theoretical Computer Science*, 293(2):291–320.

David J. Weir. 1992. A geometric hierarchy beyond context-free languages. *Theoretical Computer Science*, 104:235–261.



	A	B	C	D	E	F
1	a	a	c	b	b	a
2	a	a	a	b	a	b
3	a	a	b	a	a	a
4	b	a	a	b	a	b

Figure 9: Unified account

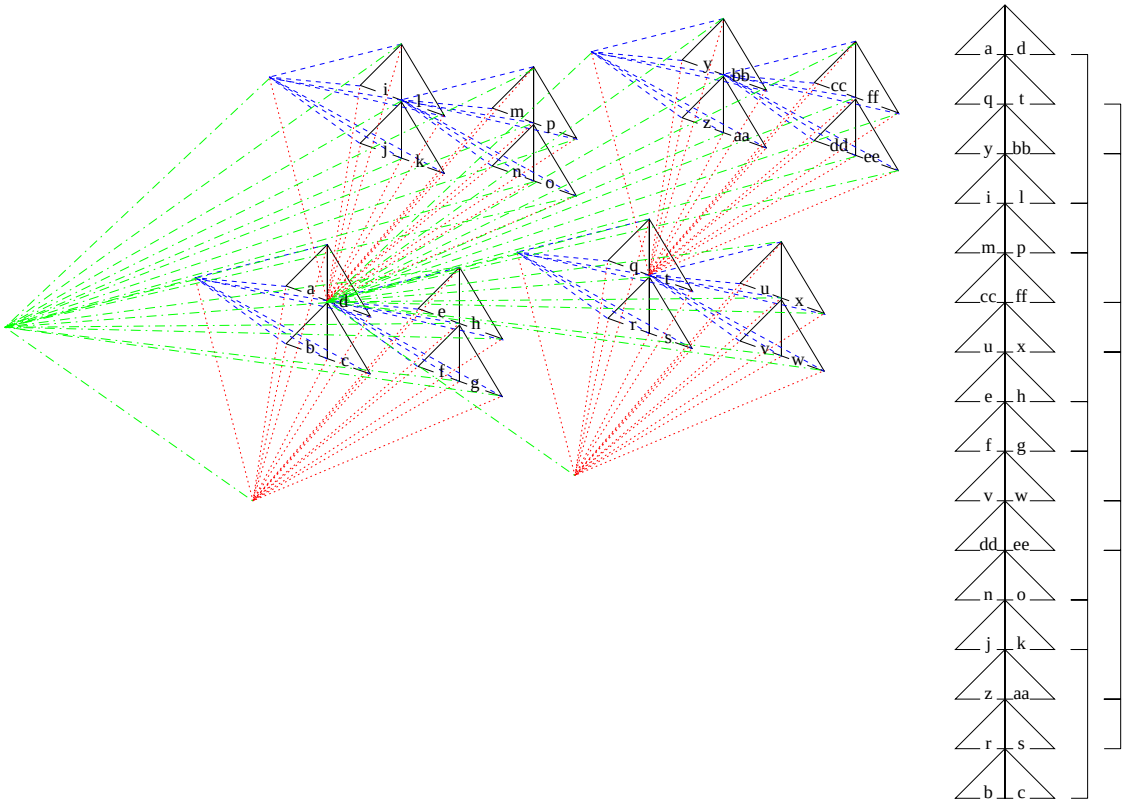


Figure 10: 5^{th}-level nesting