Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pp. 558--565

Wrapping of Trees

James Rogers
Department of Computer Science
Earlham College
Richmond, IN 47374, USA
jrogers@cs.earlham.edu

Abstract

We explore the descriptive power, in terms of syntactic phenomena, of a formalism that extends Tree-Adjoining Grammar (TAG) by adding a fourth level of hierarchical decomposition to the three levels TAG already employs. While extending the descriptive power minimally, the additional level of decomposition allows us to obtain a uniform account of a range of phenomena that has heretofore been difficult to encompass, an account that employs unitary elementary structures and eschews synchronized derivation operations, and which is, in many respects, closer to the spirit of the intuitions underlying TAG-based linguistic theory than previously considered extensions to TAG.

1 Introduction

Tree-Adjoining Grammar (TAG) (Joshi and Schabes, 1997; Joshi et al., 1975) is a grammar formalism which comes with a well-developed theory of natural language syntax (Frank, 2002; Frank, 1992; Kroch and Joshi, 1985). There are, however, a number of constructions, many in the core of language, which present difficulties for the linguistic underpinnings of TAG systems, although not necessarily for the implemented systems themselves. Most of these involve the combining of trees in ways that are more complicated than the simple embedding provided by the tree-adjunction operation.

The most widely studied way of addressing these constructions within TAG-based linguistic theory (Kroch and Joshi, 1987; Kroch, 1989; Frank, 2002) has been to assume some sort of multi-component adjoining (MCTAG (Weir, 1988)), in which elementary structures are factored into sets of trees that are adjoined simultaneously at multiple points. Depending on the restrictions placed on where this adjoining can occur the effect of such extensions range from no increase in complexity of either the licensed tree sets or the computational complexity of parsing, to substantial increases in both. In this paper we explore these issues within the framework of an extension of TAG that is conservative in the sense that it preserves the unitary nature of the elementary structures and of the adjunction operation and extends the descriptive power minimally.

While the paper is organized around particular syntactic phenomena, it is not a study of syntax itself. We make no attempt to provide a comprehensive theory of syntax. In fact, we attempt to simply instantiate the foundations of existing theory (Frank, 2002) in as faithful a way as possible. Our primary focus is the interplay between the linguistic theory and the formal language theory. All of the phenomena we consider can be (and in practice are (Group, 1998)) handled *ad hoc* with feature-structure based TAG (FTAG, (Vijay-Shanker and Joshi, 1991)). From a practical perspective, the role of the underlying linguistic theory is, at least in part, to insure consistent and comprehensive implementation of *ad hoc* mechanisms. From a theoretical perspective, the role of the formal language framework is, at least in part, to insure coherent and computationally well-grounded theories. Our overall goal is to find formal systems that are as close as possible to being a direct embodiment of the principles guiding the linguistic theory and which are maximally constrained in their formal and computational complexity.

2 Hierarchical Decomposition of Strings and Trees

Like many approaches to formalization of natural language syntax, TAG is based on a hierarchical decomposition of strings which is represented by ordered trees. (Figure 1.) These trees are, in essence, graphs representing two relationships—the left-to-right ordering of the structural components of the string and the relationship between a component and its immediate constituents.

The distinguishing characteristic of TAG is that it identifies an additional hierarchical decomposition of these trees. This shows up, for instance when a clause which has the form of a wh-question is embedded as an argument within another clause. In the

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pp. 558--565

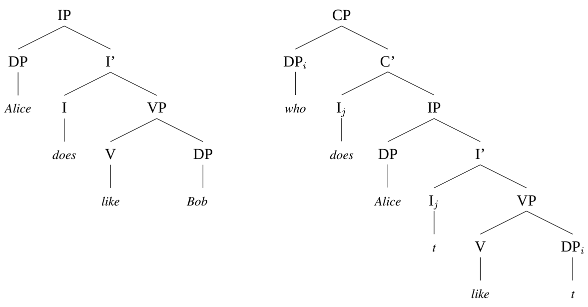

Figure 1: Wh-movement and subj-aux inversion.

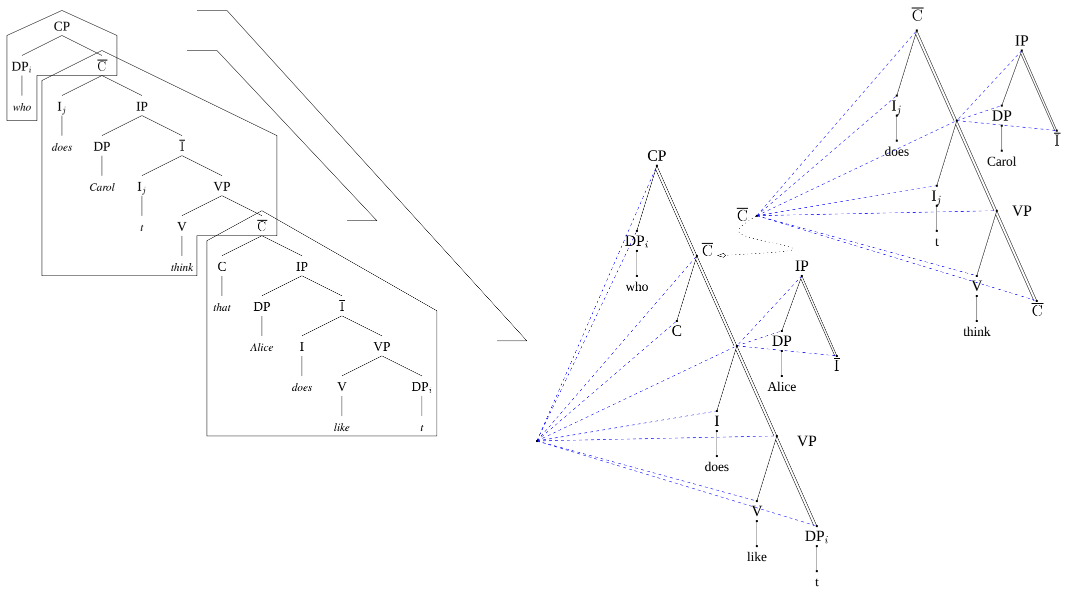

Figure 2: Bridge verbs and subj-aux inversion.

wh-form (as in the right-hand tree of Figure 1), one of the arguments of the verb is fronted as a wh-word and the inflectional element (*does*, in this case) precedes the subject. This is generally known in the literature as *wh-movement* and *subj-aux inversion*, but TAG does not necessarily assume there is any actual transformational movement involved, only that there is a systematic relationship between the wh-form and the canonical configuration. The 't's in the trees mark the position of the corresponding components in the canonical trees.[1]

When such a clause occurs as the argument of a *bridge verb* (such as *think* or *believe*) it is split, with the wh-word appearing to the left of the matrix clause and the rest of the subordinate clause occurring to the right (Figure 2). Standardly, TAG accounts analyze this as insertion of the tree for the matrix clause between the upper an lower portions of the tree for the embedded clause, an operation known as tree-adjunction. In effect, the tree for the embedded clause is *wrapped* around that of the matrix clause. This process may iterate, with adjunction of arbitrarily many instances of bridge verb trees:

> Who does Bob believe . . . Carol thinks that Alice likes.

One of the key advantages of this approach is that the wh-word is introduced into the derivation within the same elementary structure as the verb it is an argument of. Hence these structures are semantically coherent—they express all and only the structural relationships between the elements of a single functional domain (Frank, 2002). The adjoined structures are similarly coherent and the derivation preserves that coherence at all stages.

Following Rogers (2003) we will represent this by connecting the adjoined tree to the point at which it adjoins via a third, "tree constituency" relation as in the right hand part of Figure 2. This gives us

[1]This systematic relationship between the wh-form and the canonical configuration has been a fundamental component of syntactic theories dating back, at least, to the work of Harris in the '50's.

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pp. 558--565

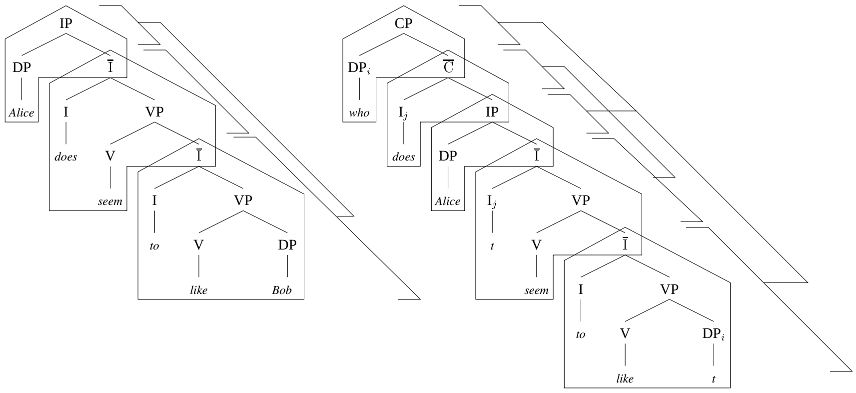

Figure 3: Raising verbs.

structures that we usually conceptualize as three-dimensional trees, but which can simply be regarded as graphs with three sorts of edges, one for each of the hierarchical relations expressed by the structures. Within this context, tree-adjunction is a process of concatenating these structures, identifying the root of the adjoined structure with the point at which it is adjoined.[2]

The resulting complex structures are formally equivalent to the derivation trees in standard formalizations of TAG. The derived tree is obtained by concatenating the tree yield of the structure analogously to the way that the string yield of a derivation tree is concatenated to form the derived string of a context-free grammar. Note that in this case it is essential to identify the point in the frontier of each tree component at which the components it dominates will be attached. This point is referred to as the *foot* of the tree and the path to it from the root is referred to as the (principal) *spine* of the tree. Here we have marked the spines by doubling the corresponding edges of the graphs.

Following Rogers (2002), we will treat the subject of the clause as if it were "adjoined" into the rest of the clause at the root of the $\bar{\text{I}}$. At this point, this is for purely theory-internal reasons—it will allow us to exploit the additional formal power we will shortly bring to bear. It should be noted that it does not represent ordinary adjunction. The subject originates in the same elementary structure as the rest of the clause, it is just a somewhat richer structure than the more standard tree.

3 Raising Verbs and Subj-Aux Inversion

A problem arises, for this account, when the matrix verb is a *raising* verb, such as *seems* or *appears* as in

Alice seems to like Bob

Who does Alice seem to like

Here the matrix clause and the embedded clause share, in some sense, the same subject argument. (Figure 3.) Raising verbs are distinguished, further, from the *control* verbs (such as *want* or *promise*) in the fact that they may realize their subject as an expletive *it*:

It seems Alice likes Bob.

Note, in particular, that in each of these cases the inflection is carried by the matrix clause. In order to maintain semantic coherence, we will assume that the subject originates in the elementary structure of the embedded clause. This, then, interprets the raising verb as taking an $\bar{\text{I}}$ to an $\bar{\text{I}}$, adjoining at the $\bar{\text{I}}$ between the subject and the inflectional element of the embedded clause (as in the left-hand side of Figure 3).

For the declarative form this provides a nesting of the trees similar to that of the bridge verbs; the embedded clause tree is wrapped around that of the matrix clause. For the wh-form, however, the wrapping pattern is more complex. Since *who* and *Alice* must originate in the same elementary structure as *like*, while *does* must originate in the same elementary structure as *seem*, the trees evidently must factor and be interleaved as shown in the right-hand side of the figure. Such a wrapping pattern is not possible in ordinary TAG. The sequences of labels occurring along the spines of TAG tree sets must form context-free languages (Weir, 1988). Hence the "center-embedded" wrapping patterns of the bridge verbs and the declarative form of the raising verbs are possible but the "cross-serial" pattern of the wh-form of the raising verbs is not.

[2] Context-free derivation can be viewed as a similar process of concatenating trees.

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pp. 558--565

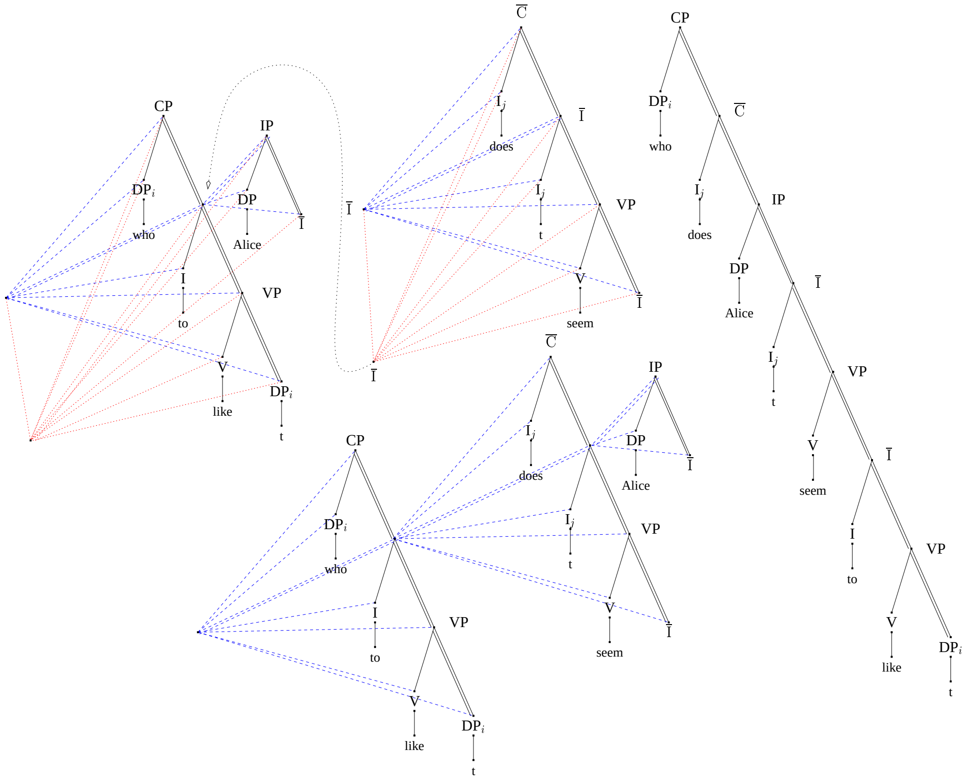

Figure 4: An higher-order account.

4 Higher-order Decomposition

One approach to obtaining the more complicated wrapping pattern that occurs in the wh-form of the raising verb trees is to move to a formalism in which the spine languages of the derived trees are TALs (the string languages derived by TAGs), which can describe such patterns. One such formalism is the third level of Weir's *Control Language Hierarchy* (Weir, 1992) which admits sets of derivation trees generated by CFGs which are filtered by a requirement that the sequences of labels on the spines occur in some particular TAL.[3] The problem with this approach is that it abandons the notion of semantic coherence of the elementary structures.

It turns out, however, that one can generate exactly the same tree sets if one moves to a formalism in which another level of hierarchical decomposition is introduced (Rogers, 2003). This now gives structures which employ four hierarchical relations—the fourth representing the constituency relation encoding a hierarchical decomposition of the third-level structures. In this framework, the *seem* structure can be taken to be inserted between the subject and the rest of the *like* structure as shown in Figure 4. Again, spines are marked by doubling the edges.

The third-order yield of the corresponding derived structure now wraps the third-order *like* structure around that of the *seem* structure, with the fragment of *like* that contains the subject attaching at the third-order "foot" node in the tree-yield of the *seem* structure (the $\bar{\text{I}}$) as shown at the bottom of the figure. The center-embedding wrapping pattern of these third-order spines guarantees that the wrapping pattern of spines of the tree yield will be a TAL, in particular, the "cross-serial" pattern needed by raising of wh-form structures.

The fourth-order structure has the added benefit of clearly justifying the status of the *like* structure as a single elementary structure despite of the apparent extraction of the subject along the third relation.

5 Locality Effects

Note that it is the $\bar{\text{I}}$ to $\bar{\text{I}}$ recursion along the third-order spine of the *seem* structure that actually does the raising of the subject. One of the consequences of this is that *that-trace* violations, such as

$*$ Who$_i$ does Alice$_j$ seem that t_j does like t_i.

cannot occur. If the complementizer originates in the *seem* structure, it will occur under the $\overline{\text{C}}$. If it originates in the *like* tree it will occur in a similar position between the CP and the $\bar{\text{I}}$. In either case,

[3] TAG is equivalent to the second level of this hierarchy, in which the spine languages are Context-Free.

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pp. 558--565

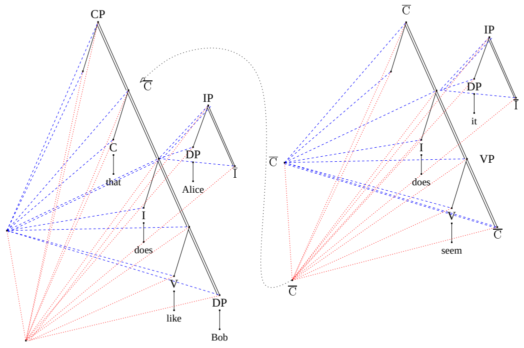

Figure 5: Expletive *it*.

the complementizer must precede the raised subject in the derived string.

If we fill the subject position of the *seem* structure with expletive *it*, as in Figure 5, the $\overline{\mathrm{I}}$ position in the yield of the structure is occupied and we no longer have $\overline{\mathrm{I}}$ to $\overline{\mathrm{I}}$ recursion. This motivates analyzing these structures as $\overline{\mathrm{C}}$ to $\overline{\mathrm{C}}$ recursion, similar to bridge verbs, rather than $\overline{\mathrm{I}}$ to $\overline{\mathrm{I}}$. (Figure 5.) More importantly the presence of the expletive subject in the *seem* tree rules out *super-raising* violations such as

$*$ Alice does it seems t does like Bob.

$*$ Alice does appear it seems t does like Bob.

No matter how the *seem* structure is interpreted, if it is to raise *Alice* then the *Alice* structure will have to settle somewhere in its yield. Without extending the *seem* structure to include the $\overline{\mathrm{C}}$ position, none of the possible positions will yield the correct string (and all can be ruled out on simple structural grounds). If the *seem* structure is extended to include the $\overline{\mathrm{C}}$, the raising will be ruled out on the assumption that the structure must attach at $\overline{\mathrm{C}}$.

6 Subject-Object Asymmetry

Another phenomenon that has proved problematic for standard TAG accounts is extraction from nominals, such as

Who did Alice publish a picture of t.

Here the wh-word is an argument of the prepositional phrase in the object nominal *picture of*. Apparently, the tree structure involves wrapping of the *picture* tree around the *publish* tree. (See Figure 6.) The problem, as normally analyzed (Frank, 2002; Kroch, 1989), is that the the *publish* tree does have the recursive structure normally assumed for auxiliary trees. We will take a somewhat less strict view and rule out the adjunction of the *publish* tree simply on the grounds that it would involve attaching a structure rooted in $\overline{\mathrm{C}}$ (or possibly CP) to a DP node.

The usual way around this difficulty has been to assume that the *who* is introduced in the *publish* tree, corresponding, presumably, to the as yet missing DP. The *picture* tree is then factored into two components, an isolated DP node which adjoins at the wh-DP, establishing its connection to the argument trace, and the *picture* DP which combines at the object position of *publish*.

This seems to at least test the spirit of the semantic coherence requirement. If the *who* is not extraneous in the *publish* tree then it must be related in some way to the object position. But the identity of who is ultimately not the object of *publish* (a picture) but rather the object of the embedded preposition (the person the picture is of).

If we analyze this in terms of a fourth hierarchical relation, we can allow the *who* to originate in the *picture* structure, which would now be rooted in CP. This could be allowed to attach at the root of the *publish* structure on the assumption that it is a C-node of some sort, providing the wrapping of its tree-yield around that of the *publish*. (See Figure 6.) Thus we get an account with intact elementary structures which are unquestionably semantically coherent.

One of the striking characteristics of extraction of this sort is the asymmetry between extraction from the object, which is acceptable, and extraction from the subject, which is not:

$*$ Who did a picture of t illustrate the point.

In the account under consideration, we might contemplate a similar combination of structures, but in this case the *picture* DP has to somehow migrate up to combine at the subject position. Under our assumption that the subject structure is attached to the *illustrate* tree via the third relation, this would require the subject structure to, in effect, have two

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pp. 558--565

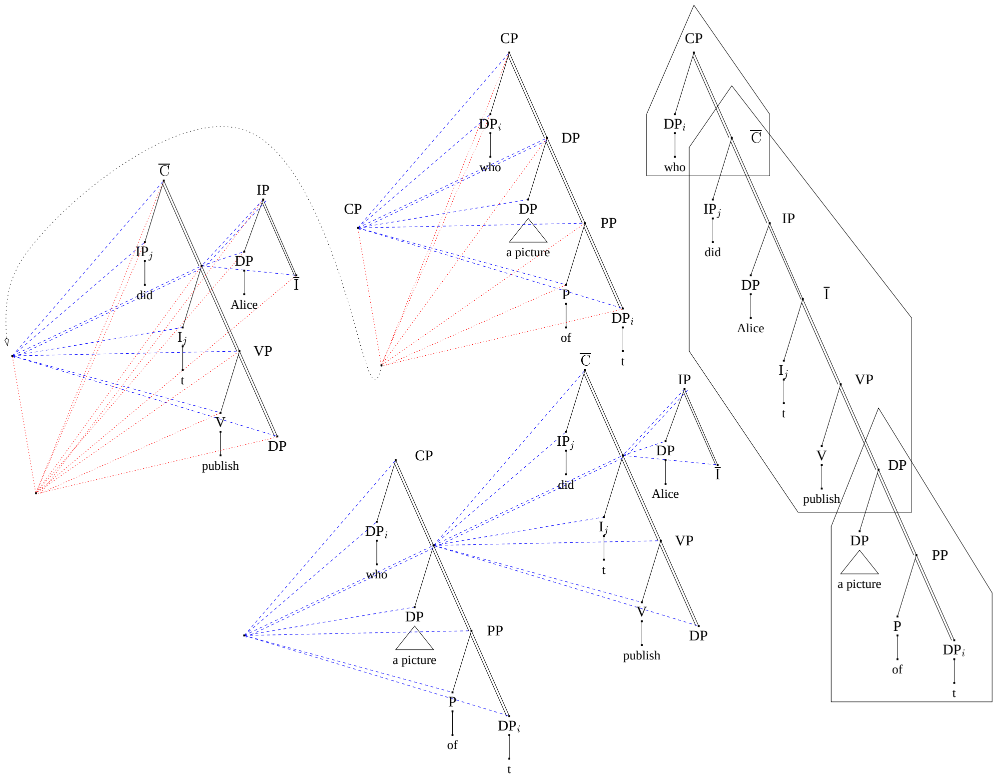

Figure 6: Extraction from object nominal.

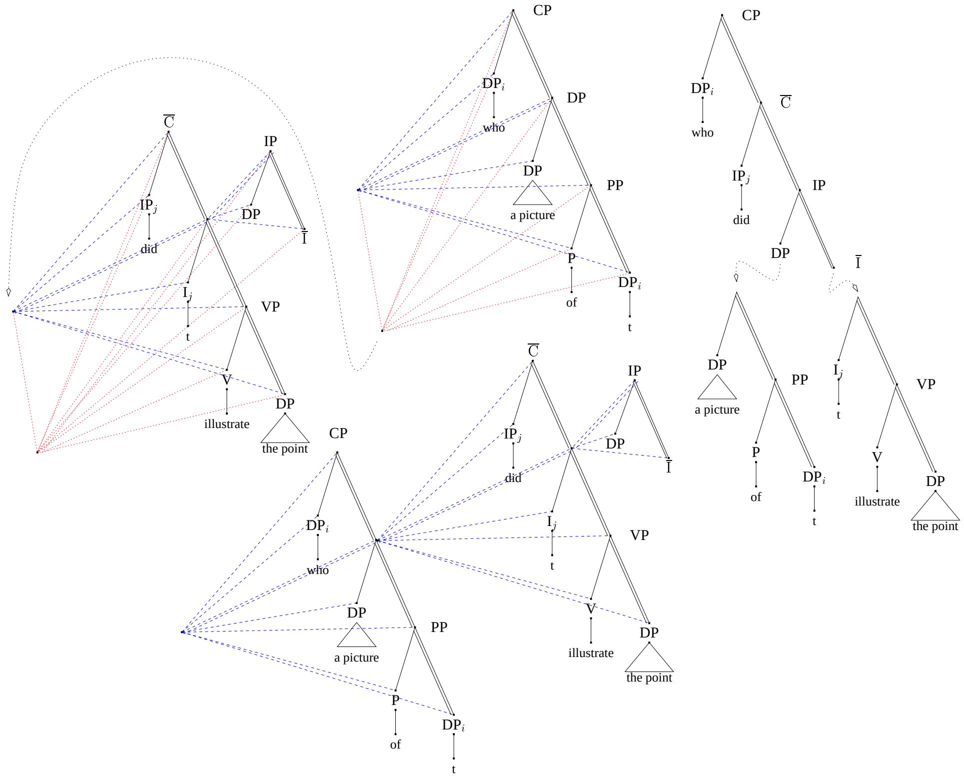

Figure 7: Extraction from subject nominal.

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04), pp. 558--565

feet, an extension that strictly increases the generative power of the formalism. Alternatively, we might assume that the *picture* structure attaches in the yield of the *illustrate* structure or between the main part of the structure and the subject tree, but either of these would fail to promote the *who* to the root of the yield structure.

7 Processing

As with any computationally oriented formalism, the ability to define the correct set of structures is only one aspect of the problem. Just as important is the question of the complexity of processing language relative to that definition. Fortunately, the languages of the Control Language Hierarchy are well understood and recognition algorithms, based on a CKY-style dynamic programming approach, are know for each level. The time complexity of the algorithm for the k^{th} level, as a function of the length of the input (n), is $O(n^{3 \cdot 2^{k-1}})$ (Palis and Shende, 1992). In the case of the fourth-order grammars, which correspond to the third level of the CLH, this gives an upper bound of $O(n^{12})$.

While, strictly speaking, this is a feasible time complexity, in practice we expect that approaches with better average-case complexity, such as Early-style algorithms, will be necessary if these grammars are to be parsed directly. But, as we noted in the introduction, grammars of this complexity are not necessarily intended to be used as working grammars. Rather they are mechanisms for expressing the linguistic theory serving as the foundation of working grammars of more practical complexity.

Since all of our proposed use of the higher-order relations involve either combining at a root (without properly embedding) or embedding with finitely bounded depth of nesting, the effect of the higher-dimensional combining operations are expressible using a finite set of features. Hence, the sets of derived trees can be generated by adding finitely many features to ordinary TAGs and the theory entailed by our accounts of these phenomena (as expressed in the sets of derived trees) is expressible in FTAG. Thus, a complete theory of syntax incorporating them would be (not necessarily not) compatible with implementation within existing TAG-based systems. A more long term goal is to implement a compilation mechanism which will translate the linguistic theory, stated in terms of the hierarchical relations, directly into grammars stated in terms of the existing TAG-based systems.

8 Conclusion

In many ways the formalism we have working with is a minimal extension of ordinary TAGs. Formally, the step from TAG to add the fourth hierarchical relation is directly analogous to the step from CFG to TAG. Moreover, while the graphs describing the derived structures are often rather complicated, conceptually they involve reasoning in terms of only a single additional relation. The benefit of the added complexity is a uniform account of a range of phenomena that has heretofore been difficult to encompass, an account that employs unitary elementary structures and eschews synchronized derivation operations, and which is, in many respects, closer to the spirit of the intuitions underlying TAG-based linguistic theory than previously considered extensions to TAG.

While it is impossible to determine how comprehensive the coverage of a more fully developed theory of syntax based on this formalism will be without actually completing such a theory, we believe that the results presented here suggest that the uniformity provided by adding this fourth level of decomposition to our vocabulary is likely to more than compensate for the added complexity of the fourth level elementary structures.

References

Robert Evan Frank. 1992. *Syntactic Locality and Tree Adjoining Grammar: Grammatical, Acquisition and Processing Perspectives*. Ph.D. dissertation, Univ. of Penn.

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

The XTAG Research Group. 1998. A lexicalized tree adjoining grammar for english. Technical Report IRCS-98-18, Institute for Research in Cognitive Science.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In *Handbook of Formal Languages and Automata*, volume 3, pages 69–123. Springer-Verlag.

Aravind K. Joshi, Leon Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of the Computer and Systems Sciences*, 10:136–163.

Anthony Kroch and Aravind K. Joshi. 1985. The linquistic relevance of tree adjoining grammar. Technical Report MS-CS-85-16, Dept. of Computer and Information Sciences.

Anthony S. Kroch and Aravind K. Joshi. 1987. Analyzing extraposition in a tree adjoining grammar. In *Syntax and Semantics*, pages 107–149. Academic Press. Vol. 20.

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04), pp. 558--565

Anthony Kroch. 1989. Asymmetries in long distance extraction in a tree adjoining grammar. In Mark Baltin and Anthony Kroch, editors, *Alternative Conceptions of Phrase Structure*, pages 66–98. University of Chicago Press.

Michael A. Palis and Sunil M. Shende. 1992. Upper bounds on recognition of a hierarchy of non-context-free languages. *Theoretical Computer Science*, 98:289–319.

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. Syntactic structures as multi-dimensional trees. *Research on Language and Computation*, 1(3–4):265–305.

K. Vijay-Shanker and Aravind K. Joshi. 1991. Unification based tree adjoining grammars. In J. Wedekind, editor, *Unification-based Grammars*. MIT Press, Cambridge, MA.

David J. Weir. 1988. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. Ph.D. thesis, University of Pennsylvania.

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