



# An alternative conception of tree-adjointing derivation

## Citation

Yves Schabes and Stuart M. Shieber. An alternative conception of tree-adjointing derivation. In Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, pages 167-176, University of Delaware, Newark, Delaware, June 28-July 2 1992.

## Published Version

<http://www.aclweb.org/anthology/P92-1022>

## Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:2094441>

## Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP>

## Share Your Story

The Harvard community has made this article openly available.  
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

# An Alternative Conception of Tree-Adjoining Derivation\*

Yves Schabes  
Department of Computer and  
Information Science  
University of Pennsylvania  
Philadelphia, PA 19104

Stuart M. Shieber  
Aiken Computation Laboratory  
Division of Applied Sciences  
Harvard University  
Cambridge, MA 02138

## Abstract

The precise formulation of derivation for tree-adjoining grammars has important ramifications for a wide variety of uses of the formalism, from syntactic analysis to semantic interpretation and statistical language modeling. We argue that the definition of tree-adjoining derivation must be reformulated in order to manifest the proper linguistic dependencies in derivations. The particular proposal is both precisely characterizable, through a compilation to linear indexed grammars, and computationally operational, by virtue of an efficient algorithm for recognition and parsing.

## 1 Introduction

In a context-free grammar, the derivation of a string in the rewriting sense can be captured in a single canonical tree structure that abstracts all possible derivation orders. As it turns out, this *derivation tree* also corresponds exactly to the hierarchical structure that the derivation imposes on the string, the *derived tree* structure of the string. The formalism of tree-adjoining grammars (TAG), on the other hand, decouples these two notions of derivation tree and derived tree. Intuitively, the derivation tree is a more finely grained structure

than the derived tree, and as such can serve as a substrate on which to pursue further analysis of the string. This intuitive possibility is made manifest in several ways. Fine-grained syntactic analysis can be pursued by imposing on the derivation tree further combinatorial constraints, for instance, selective adjoining constraints or equational constraints over feature structures. Statistical analysis can be explored through the specification of derivational probabilities as formalized in stochastic tree-adjoining grammars. Semantic analysis can be overlaid through the synchronous derivations of two TAGs.

All of these methods rely on the derivation tree as the source of the important primitive relationships among trees. The decoupling of derivation trees from derived trees thus makes possible a more flexible ability to pursue these types of analyses. At the same time, the exact definition of derivation becomes of paramount importance. In this paper, we argue that previous definitions of tree-adjoining derivation have not taken full advantage of this decoupling, and are not as appropriate as they might be for the kind of further analysis that tree-adjoining analyses could make possible. In particular, the standard definition of derivation, due to Vijay-Shanker (1987), requires that elementary trees be adjoined at distinct nodes in elementary trees. However, in certain cases, especially cases characterized as linguistic modification, it is more appropriate to allow multiple adjunctions at a single node.

In this paper, we propose a redefinition of TAG derivation along these lines, whereby multiple auxiliary trees of modification can be adjoined at a single node, whereas only a single auxiliary tree of predication can. The redefinition constitutes a new definition of derivation for TAG that we will refer to as *extended derivation*. In order for such

---

\*The authors are listed in alphabetical order. The first author was supported in part by DARPA Grant N0014-90-31863, ARO Grant DAAL03-89-C-0031 and NSF Grant IRI90-16592. The second author was supported in part by Presidential Young Investigator award IRI-91-57996 from the National Science Foundation. The authors wish to thank Aravind Joshi for his support of the research, and Aravind Joshi, Anthony Kroch, Fernando Pereira, and K. Vijay-Shanker for their helpful discussions of the issues involved. We are indebted to David Yarowsky for aid in the design of the experiment mentioned in footnote 5 and for its execution.

a redefinition to be serviceable, however, it is necessary that it be both precise and operational. In service of the former, we provide a rigorous specification of our proposal in terms of a compilation of TAGs into corresponding linear indexed grammars (LIG) that makes the derivation structure explicit. With respect to the latter, we show how the generated LIG can drive a parsing algorithm that recovers, either implicitly or explicitly, the extended derivations of the string.

The paper is organized as follows. First, we review Vijay-Shanker’s standard definition of TAG derivation, and introduce the motivation for extended derivations. Then, we present the extended notion of derivation informally, and formalize it through the compilation of TAGs to LIGs. The original compilation provided by Vijay-Shanker and Weir and our variant for extended derivations are both described. Finally, we briefly mention a parsing algorithm for TAG that recovers extended derivations either implicitly or explicitly, and discuss some issues surrounding it. Space limitations preclude us from presenting the algorithm itself, but a full description is given elsewhere (Schabes and Shieber, 1992).

## 2 The Standard Definition of Derivation

To exemplify the distinction between standard and extended derivations, we exhibit the TAG of Figure 1. This grammar derives some simple noun phrases such as “roasted red pepper” and “baked red potato”. The former, for instance, is associated with the derived tree in Figure 2(a). The tree can be viewed as being derived in two ways<sup>1</sup>

**Dependent:** The auxiliary tree  $\beta_{ro}$  is adjoined at the root node (address  $\epsilon$ ) of  $\beta_{re}$ . The resultant tree is adjoined at the root node (address  $\epsilon$ ) of initial tree  $\alpha_{pe}$ . This derivation is depicted as the derivation tree in Figure 3(a).

**Independent:** The auxiliary trees  $\beta_{ro}$  and  $\beta_{re}$  are adjoined at the root node of the initial tree  $\alpha_{pe}$ . This derivation is depicted as the derivation tree in Figure 3(b).

In the *independent* derivation, two trees are separately adjoined at one and the same node in the initial tree. In the *dependent* derivation, on the other hand, one auxiliary tree is adjoined to the

<sup>1</sup>As is standard in the TAG literature we disallow adjunction at the foot nodes of auxiliary trees.

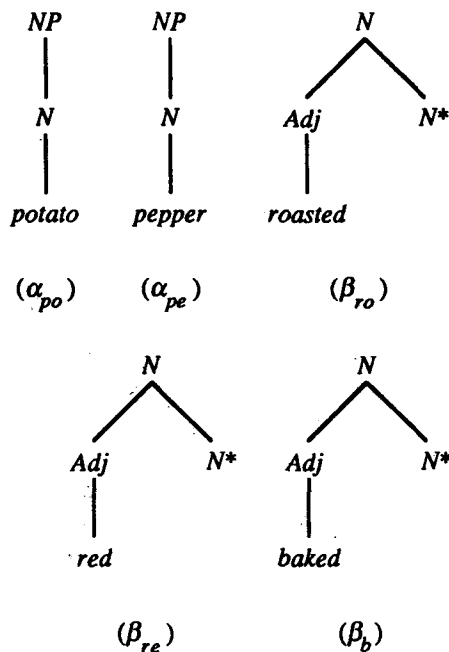


Figure 1: A sample tree-adjoining grammar

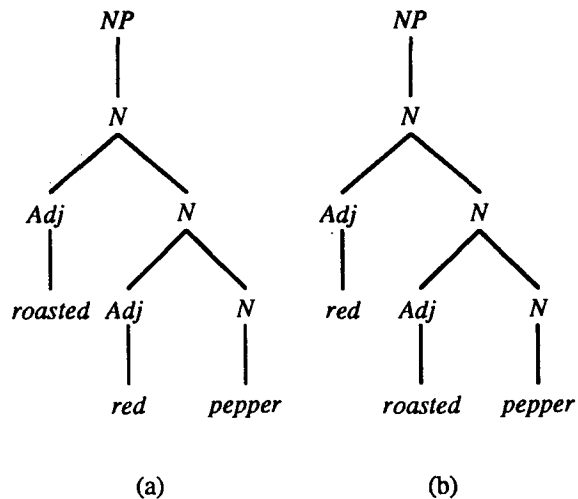


Figure 2: Two trees derived by the grammar of Figure 1

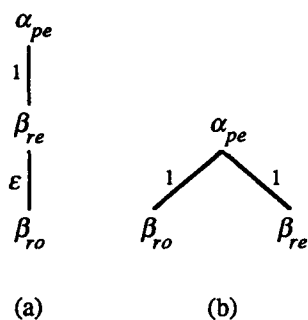


Figure 3: Derivation trees for the derived tree of Figure 2(a) according to the grammar of Figure 1

other, the latter only being adjoined to the initial tree. We will use this informal terminology uniformly in the sequel to distinguish the two general topologies of derivation trees.

The *standard* definition of derivation, as codified by Vijay-Shanker, restricts derivations so that *two adjunctions cannot occur at the same node in the same elementary tree*. The dependent notion of derivation is therefore the only sanctioned derivation for the desired tree in Figure 2(a); the independent derivation is disallowed. Vijay-Shanker's definition is appropriate because for any independent derivation, there is a dependent derivation of the same derived tree. This can be easily seen in that any adjunction of  $\beta_2$  at a node at which an adjunction of  $\beta_1$  occurs could instead be replaced by an adjunction of  $\beta_2$  at the root of  $\beta_1$ .

The advantage of this standard definition of derivation is that a derivation tree in this normal form unambiguously specifies a derived tree. The independent derivation tree on the other hand is ambiguous as to the derived tree it specifies in that a notion of precedence of the adjunctions at the same node is unspecified, but crucial to the derived tree specified. This follows from the fact that the independent derivation tree is symmetric with respect to the roles of the two auxiliary trees (by inspection), whereas the derived tree is not. By symmetry, therefore, it must be the case that the same independent derivation tree specifies the alternative derived tree in Figure 2(b).

### 3 Motivation for Extended Derivations

In the absence of some further interpretation of the derivation tree nothing hinges on the choice

of derivation definition, so that the standard definition is as reasonable as any other. However, tree-adjoining grammars are almost universally extended with augmentations that make the issue apposite. We discuss three such variations here, all of which argue for the use of independent derivations under certain circumstances.

#### 3.1 Adding Adjoining Constraints

Already in very early work on tree-adjoining grammars (Joshi et al., 1975) constraints were allowed to be specified as to whether a particular auxiliary tree may or may not be adjoined at a particular node in a particular tree. The idea is formulated in its modern variant as *selective-adjoining constraints* (Vijay-Shanker and Joshi, 1985). As an application of this capability, we consider the remark by Quirk et al. (1985, page 517) that “direction adjuncts of both goal and source can normally be used only with verbs of motion”, which accounts for the distinction between the following sentences:

- (1)a. Brockway escorted his sister to the annual cotillion.
- b. #Brockway resembled his sister to the annual cotillion.

This could be modeled by disallowing through selective adjoining constraints the adjunction of the elementary tree corresponding to a *to* adverbial at the VP node of the elementary tree corresponding to the verb *resembles*.<sup>2</sup> However, the restriction applies even with intervening (and otherwise acceptable) adverbials.

- (2)a. Brockway escorted his sister last year.
- b. Brockway escorted his sister last year to the annual cotillion.
- (3)a. Brockway resembled his sister last year.
- b. #Brockway resembled his sister last year to the annual cotillion.

Under the standard definition of derivation, there is no direct adjunction in the latter sentence of the *to* tree into the *resembles* tree. Rather, it is dependently adjoined at the root of the elementary tree that heads the adverbial *last year*, the latter directly adjoining into the main verb tree. To restrict both of the ill-formed sentences, then, a restriction must be placed not only on adjoining

<sup>2</sup>Whether the adjunction occurs at the VP node or the S node is immaterial to the argument.

the goal adverbial in a *resembles* context, but also in the *last year* adverbial context. But this constraint is too strong, as it disallows sentence (2b) above as well.

The problem is that the standard derivation does not correctly reflect the syntactic relation between adverbial modifier and the phrase it modifies when there are multiple modifications in a single clause. In such a case, each of the adverbials independently modifies the verb, and this should be reflected in their independent adjunction at the same point. But this is specifically disallowed in a standard derivation.

It is important to note that the argument applies specifically to auxiliary trees that correspond to a modification relationship. Auxiliary trees are used in TAG typically for predication relations as well,<sup>3</sup> as in the case of raising and sentential complement constructions.<sup>4</sup> Consider the following sentences. (The brackets mark the leaves of the pertinent trees to be combined by adjunction in the assumed analysis.)

- (4)a. Brockway conjectured that Harrison wanted to escort his sister.
- b. [Brockway conjectured that] [Harrison wanted] [to escort his sister]
- (5)a. Brockway wanted to try to escort his sister.
- b. [Brockway wanted] [to try] [to escort his sister]
- (6)a. \* Harrison wanted Brockway tried to escort his sister.
- b. \* [Harrison wanted] [Brockway tried] [to escort his sister]

Assume (following, for instance, the analysis of Kroch and Joshi (1985)) that the trees associated with the various forms of the verbs “try”, “want”, and “conjecture” all take sentential complements, certain of which are tensed with overt subjects and others untensed with empty subjects. The auxiliary trees for these verbs specify by ad-

<sup>3</sup>We use the term ‘predication’ in its logical sense, that is, for auxiliary trees that serve as logical predicates over the trees into which they adjoin, in contrast to the term’s linguistic sub-sense in which the argument of the predicate is a linguistic subject.

<sup>4</sup>The distinction between predicative and modifier trees has been proposed previously for purely linguistic reasons by Kroch (1989), who refers to them as thematic and athematic trees, respectively. The arguments presented here can be seen as providing further evidence for differentiating the two kinds of auxiliary trees.

junction constraints which type of sentential complement they take: “conjecture” requires tensed complements, “want” and “try” untensed. Under this analysis the auxiliary trees must not be allowed to independently adjoin at the same node. For instance, if trees corresponding to “Harrison wanted” and “Brockway tried” (which both require untensed complements) were both adjoined at the root of the tree for “to escort his sister”, the selective adjunction constraints would be satisfied, yet the generated sentence (6a) is ungrammatical. Thus, the case of predicative trees is entirely unlike that of modifier trees. Here, the standard notion of derivation is exactly what is needed as far as interpretation of adjoining constraints is concerned.

In summary, the interpretation of adjoining constraints in TAG is sensitive to the particular notion of derivation that is used. Therefore, it can be used as a litmus test for an appropriate definition of derivation. As such, it argues for a nonstandard, independent, notion of derivation for modifier auxiliary trees and a standard, dependent, notion for predicative trees.

### 3.2 Adding Statistical Parameters

In a similar vein, the statistical parameters of a stochastic lexicalized TAG (SLTAG) (Resnik, 1992; Schabes, 1992) specify the probability of adjunction of a given auxiliary tree at a specific node in another tree. This specification may again be interpreted with regard to differing derivations, obviously with differing impact on the resulting probabilities assigned to derivation trees. (In the extreme case, a constraint prohibiting adjoining corresponds to a zero probability in an SLTAG. The relation to the argument in the previous section follows thereby.) Consider a case in which linguistic modification of noun phrases by adjectives is modeled by adjunction of a modifying tree. Under the standard definition of derivation, multiple modifications of a single NP would lead to dependent adjunctions in which a first modifier adjoins at the root of a second. As an example, we consider again the grammar given in Figure 1, that admits of derivations for the strings “baked red potato” and “baked red pepper”. Specifying adjunction probabilities on standard derivations, the distinction between the overall probabilities for these two strings depends solely on the adjunction probabilities of  $\beta_{re}$  (the tree for *red*) into  $\alpha_{po}$  and  $\alpha_{pe}$  (those for *potato* and *pepper*, respectively), as the tree  $\beta_b$  for the word *baked* is adjoined in both cases at the root of  $\beta_{re}$  in both standard

derivations. In the extended derivations, on the other hand, both modifying trees are adjoined independently into the noun trees. Thus, the overall probabilities are determined as well by the probabilities of adjunction of the trees for *baked* into the nominal trees. It seems intuitively plausible that the most important relationships to characterize statistically are those between modifier and modified, rather than between two modifiers.<sup>5</sup> In the case at hand, the fact that potatoes are more frequently baked, whereas peppers are roasted, would be more determining of the expected overall probabilities.

Note again that the distinction between modifier and predicative trees is important. The standard definition of derivation is entirely appropriate for adjunction probabilities for predicative trees, but not for modifier trees.

### 3.3 Adding Semantics

Finally, the formation of synchronous TAGs has been proposed to allow use of TAGs in semantic interpretation, natural language generation, and machine translation. In previous work (Shieber and Schabes, 1990), the definition of synchronous TAG derivation is given in a manner that requires multiple adjunctions at a single node. The need for such derivations follows from the fact that synchronous derivations are intended to model semantic relationships. In cases of multiple adjunction of modifier trees at a single node, the appropriate semantic relationships comprise separate modifications rather than cascaded ones, and this is reflected in the definition of synchronous TAG derivation.<sup>6</sup> Because of this, a parser for synchronous TAGs must recover, at least implicitly, the extended derivations of TAG derived trees.

<sup>5</sup>Intuition is an appropriate guide in the design of the SLTAG framework, as the idea is to set up a linguistically plausible infrastructure on top of which a lexically-based statistical model can be built. In addition, suggestive (though certainly not conclusive) evidence along these lines can be gleaned from corpora analyses. For instance, in a simple experiment in which medium frequency triples of exactly the discussed form “(adjective) (adjective) (noun)” were examined, the mean mutual information between the first adjective and the noun was found to be larger than that between the two adjectives. The statistical assumptions behind the experiment do not allow very robust conclusions to be drawn, and more work is needed along these lines.

<sup>6</sup>The importance of the distinction between predicative and modifier trees with respect to how derivations are defined was not appreciated in the earlier work; derivations were taken to be of the independent variety in all cases. In future work, we plan to remedy this flaw.

Note that the independence of the adjunction of modifiers in the syntax does not imply that semantically there is no precedence or scoping relation between them. As exemplified in Figure 4, the derived tree generated by multiple independent adjunctions at a single node still manifests nesting relationships among the adjoined trees. This fact may be used to advantage in the semantic half of a synchronous tree-adjoining grammar to specify the semantic distinction between, for example, the following two sentences:<sup>7</sup>

- (7)a. Brockway paid for the tickets twice intentionally.
- b. Brockway paid for the tickets intentionally twice.

We hope to address this issue in greater detail in future work on synchronous tree-adjoining grammars.

## 4 Informal Specification of Extended Derivations

We have presented several arguments that the standard notion of derivation does not allow for an appropriate specification of dependencies to be captured. An extended notion of derivation is needed that

1. Differentiates predicative and modifier auxiliary trees;
2. Requires dependent derivations for predicative trees;
3. Requires independent derivations for modifier trees; and
4. Unambiguously specifies a derived tree.

Recall that a derivation tree is a tree with unordered arcs where each node is labeled by an elementary tree of a TAG and each arc is labeled by a tree address specifying a node in the parent tree. In a standard derivation tree no two sibling arcs can be labeled with the same address. In an extended derivation tree, however, the condition is relaxed: No two sibling arcs *to predicative trees* can be labeled with the same address. Thus, for any given address there can be at most one predicative tree and several modifier trees adjoined at

<sup>7</sup>We are indebted to an anonymous reviewer for raising this issue crisply through examples similar to those given here.

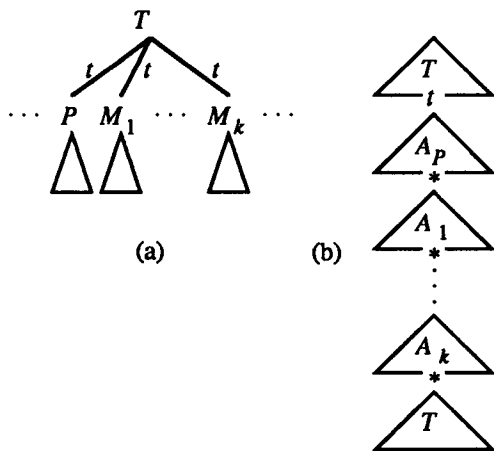


Figure 4: Schematic extended derivation tree and associated derived tree

that node. So as to fully specify the output derived tree, we specify a partial ordering on sibling arcs by mandating that arcs corresponding to modifier trees adjoined at the same address are treated as ordered left-to-right. However, all other arcs, including those for predicative adjunctions are left unordered.

A derivation tree specifies a derived tree through a bottom-up traversal (as is standard since the work of Vijay-Shanker (1987)). The choice of a particular traversal order plays the same role as choosing a particular rewriting derivation order in a context-free grammar — leftmost or rightmost, say — in eliminating spurious ambiguity due to inconsequential reordering of operations. An extended derivation tree specifies a derived tree in exactly the same manner, except that there must be a specification of the derived tree specified when several trees are adjoined at the same node.

Assume that in a given tree  $T$  at a particular address  $t$ , the predicative tree  $P$  and the  $k$  modifier trees  $M_1, \dots, M_k$  (in that order) are directly adjoined. Schematically, the extended derivation tree would appear as in Figure 4(a). Associated with the subtrees rooted at the  $k + 1$  elementary auxiliary trees in this derivation are  $k + 1$  derived auxiliary trees ( $A_P$  and  $A_1, \dots, A_k$ , respectively). (The derived auxiliary trees are specified inductively; it is this sense in which the definition corresponds to a bottom-up traversal.)

There are many possible trees that might be entertained as the derived tree associated with the derivation rooted at  $T$ , one for each permutation

of the  $k + 1$  auxiliary trees. Since the ordering of the modifiers in the derivation tree is essentially arbitrary, we can fix on a single ordering of these in the output tree. We will choose the ordering in which the top to bottom order in the derived tree follows the partial order on the nodes in the derivation tree. Thus  $A_1$  appears higher in the tree than  $A_2$ ,  $A_2$  higher than  $A_3$  and so forth. This much is arbitrary.

The choice of where the predicative tree goes, however, is consequential. There are  $k + 1$  possible positions, of which only two can be seriously maintained: outermost, at the top of the tree; or innermost, at the bottom. We complete the (informal) definition of extended derivation by specifying the derived tree corresponding to such a derivation to manifest outermost predication as depicted in Figure 4(b).

Both linguistic and technical consequences argue for outermost, rather than innermost, predication. Linguistically, the outermost method specifies that if both a predicative tree and a modifier tree are adjoined at a single node, then the predicative tree attaches “higher” than the modifier tree; in terms of the derived tree, it is as if the predicative tree were adjoined at the root of the modifier tree. This accords with the semantic intuition that in such a case, the modifier is modifying the original tree, not the predicative one. (The alternate “reading”, in which the modifier modifies the predicative tree, is still obtainable under an outermost-predication standard by having the modifier auxiliary tree adjoin at the root node of the predicative tree.) In contrast, the innermost-predication method specifies that the modifier tree attaches higher, as if the modifier tree adjoined at the root of the predicative tree and was therefore modifying the predicative tree, contra semantic intuitions.

From a technical standpoint, the outermost-predication method requires no changes to the parsing rules to be presented later, but only a single addition. The innermost-predication method induces some subtle interactions between the original parsing rules and the additional one, necessitating a much more complicated set of modifications to the original algorithm. (In fact, the complexities in generating such an algorithm constituted the precipitating factor that led us to revise our original, innermost-predication, attempt at redefining tree-adjoining derivation.)

## 5 Formal Specification of Extended Derivations

In all three application areas of TAGs, the need is evidenced for a modified notion of derivation that retains the dependent notion of derivation for predicative trees but mandates independent adjunction for modifier trees. A formal definition of extended derivation can be given by means of a compilation of tree-adjointing grammars into linear indexed grammars. We discuss such a compilation in this section. This compilation is especially useful as it can be used as the basis for a parsing algorithm that recovers the extended derivations for strings. The design of the algorithm is the topic of Section 6.

Linear indexed grammars (LIG) constitute a grammatical framework based, like context-free, context-sensitive, and unrestricted rewriting systems, on rewriting strings of nonterminal and terminal symbols. Unlike these systems, linear indexed grammars, like the indexed grammars from which they are restricted, allow stacks of marker symbols, called *indices*, to be associated with the nonterminal symbols being rewritten. The linear version of the formalism allows the full index information from the parent to be used to specify the index information for only one of the child constituents. Thus, a linear indexed production can be given schematically as:

$$N_0[..\beta_0] \rightarrow N_1[\beta_1] \cdots N_{s-1}[\beta_{s-1}] \\ N_s[..\beta_s] \\ N_{s+1}[\beta_{s+1}] \cdots N_k[\beta_k]$$

The  $N_i$  are nonterminals, the  $\beta_i$  strings of indices. The “..” notation stands for the remainder of the stack below the given string of indices. Note that only one element on the right-hand side,  $N_s$ , inherits the remainder of the stack from the parent. (This schematic rule is intended to be indicative, not definitive. We ignore issues such as the optionality of the inherited stack, how terminal symbols fit in, and so forth. Vijay-Shanker and Weir (1990) present a complete discussion.)

Vijay-Shanker and Weir (1990) present a way of specifying any TAG as a linear indexed grammar. The LIG version makes explicit the standard notion of derivation being presumed. Also, the LIG version of a TAG grammar can be used for recognition and parsing. Because the LIG formalism is based on augmented rewriting, the parsing algorithms can be much simpler to understand and easier to modify, and no loss of generality is in-

curred. For these reasons, we use the technique in this work.

The compilation process that manifests the standard definition of derivation can be most easily understood by viewing nodes in a TAG elementary tree as having both a top and bottom component, identically marked for nonterminal category, that dominate (but may not immediately dominate) each other. (See Figure 5.) The rewrite rules of the corresponding linear indexed grammar capture the immediate domination between a bottom node and its child top nodes directly, and capture the domination between top and bottom parts of the same node by optionally allowing rewriting from the top of a node to an appropriate auxiliary tree, and from the foot of the auxiliary tree back to the bottom of the node. The index stack keeps track of the nodes that adjunction has occurred on so that the recognition to the left and the right of the foot node will occur under identical assumption of derivation structure. In summary, the following LIG rules are generated:

1. *Immediate domination dominating foot:* For each auxiliary tree node  $\eta$  that dominates the foot node, with children  $\eta_1, \dots, \eta_s, \dots, \eta_n$ , where  $\eta_s$  is the child that also dominates the foot node, include a production

$$b[..\eta] \rightarrow t[\eta_1] \cdots t[\eta_{s-1}] t[..\eta_s] t[\eta_{s+1}] \cdots t[\eta_n]$$

2. *Immediate domination not including foot:* For each elementary tree node  $\eta$  that does not dominate a foot node, with children  $\eta_1, \dots, \eta_n$ , include a production

$$b[\eta] \rightarrow t[\eta_1] \cdots t[\eta_n]$$

3. *No adjunction:* For each elementary tree node  $\eta$  that is not marked for substitution or obligatory adjunction, include a production

$$t[..\eta] \rightarrow b[..\eta]$$

4. *Start root of adjunction:* For each elementary tree node  $\eta$  on which the auxiliary tree  $\beta$  with root node  $\eta_r$  can be adjoined, include the following production:

$$t[..\eta] \rightarrow t[..\eta\eta_r]$$

5. *Start foot of adjunction:* For each elementary tree node  $\eta$  on which the auxiliary tree  $\beta$  with



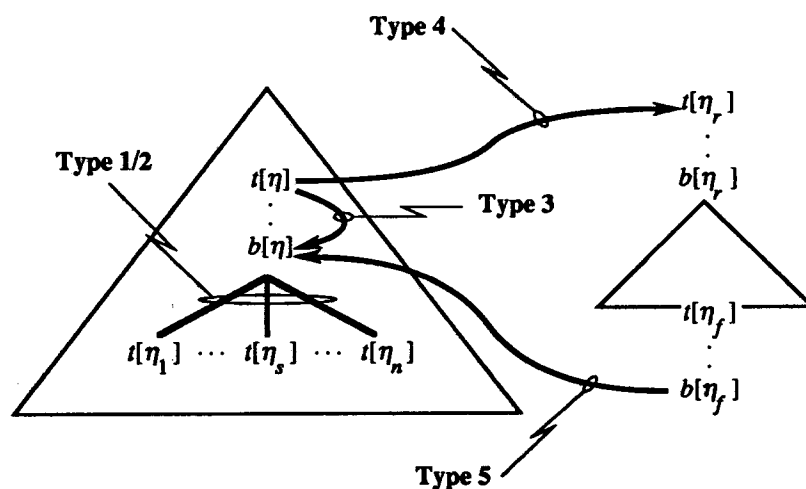


Figure 5: Schematic structure of adjunction with top and bottom of each node separated

foot node  $\eta_f$  can be adjoined, include the following production:

$$b[..\eta_f] \rightarrow b[..\eta]$$

6. *Start substitution*: For each elementary tree node  $\eta$  marked for substitution on which the initial tree  $\alpha$  with root node  $\eta_r$  can be substituted, include the production

$$t[..\eta] \rightarrow t[..\eta_r]$$

We will refer to productions generated by Rule  $i$  above as Type  $i$  productions. For example, Type 3 productions are of the form  $t[..\eta] \rightarrow b[..\eta]$ . For further information concerning the compilation see the work of Vijay-Shanker and Weir (1990) and Schabes (1991). For present purposes, it is sufficient to note that the method directly embeds the standard notion of derivation in the rewriting process. To perform an adjunction, we move (by Rule 4) from the node adjoined at to the top of the root of the auxiliary tree. At the root, additional adjunctions might be performed. When returning from the foot of the auxiliary tree back to the node where adjunction occurred, rewriting continues at the bottom of the node (see Rule 5), not the top, so that no more adjunctions can be started at that node. Thus, the dependent nature of predicative adjunction is enforced because *only a single adjunction can occur at any given node*.

In order to permit extended derivations, we must allow for multiple modifier tree adjunctions at a single node. There are two natural ways this might be accomplished, as depicted in Figure 6.

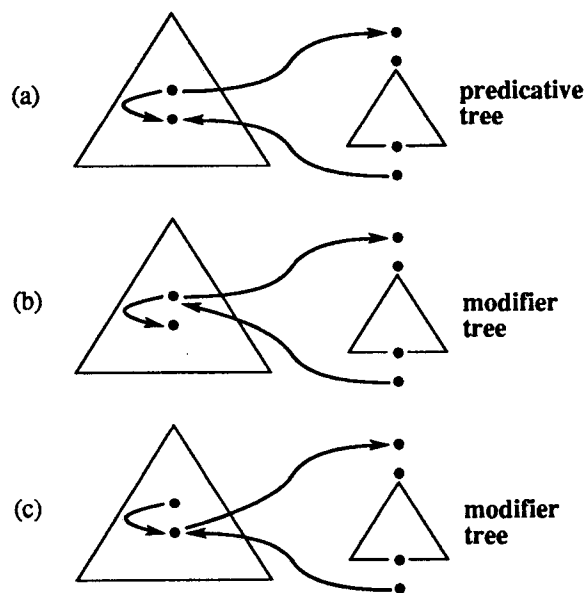


Figure 6: Schematic structure of possible predicative and modifier adjunctions with top and bottom of each node separated

1. *Modified start foot of adjunction rule:* Allow moving from the bottom of the foot of a modifier auxiliary tree to the top (rather than the bottom) of the node at which it adjoined (Figure 6b).
2. *Modified start root of adjunction rule:* Allow moving from the bottom (rather than the top) of a node to the top of the root of a modifier auxiliary tree (Figure 6c).

As can be seen from the figures, both of these methods allow recursion at a node, unlike the original method depicted in Figure 6a. Thus multiple modifier trees are allowed to adjoin at a single node. Note that since predicative trees fall under the original rules, at most a single predicative tree can be adjoined at a node. The two methods correspond exactly to the innermost- and outermost-predication methods discussed in Section 4. For the reasons described there, the latter is preferred.

In summary, independent derivation structures can be allowed for modifier auxiliary trees by starting the adjunction process from the bottom, rather than the top of a node for those trees. Thus, we split Type 4 LIG productions into two subtypes for predicative and modifier trees, respectively.

- 4a. *Start root of predicative adjunction:* For each elementary tree node  $\eta$  on which the predicative auxiliary tree  $\beta$  with root node  $\eta_r$  can be adjoined, include the following production:

$$t[..\eta] \rightarrow t[..\eta\eta_r]$$

- 4b. *Start root of modifier adjunction:* For each elementary tree node  $\eta$  on which the modifier auxiliary tree  $\beta$  with root node  $\eta_r$  can be adjoined, include the following production:

$$b[..\eta] \rightarrow t[..\eta\eta_r]$$

Once this augmentation has been made, we no longer need to allow for adjunctions at the root nodes of modifier auxiliary trees, as repeated adjunction is now allowed for by the new rule 4b. Consequently, Rules 4a and 4b must treat all modifier auxiliary tree root nodes as if they have adjoining constraints that forbid modifier tree adjunctions that do not correspond to modification of the tree itself.

This simple modification to the compilation process from TAG to LIG fully specifies the modified notion of derivation. The recognition algorithms for TAG based on this compilation, however, must be adjusted to allow for the new rule types.

## 6 Recognition and Parsing

Following Schabes (1991), the LIG generated by compiling a TAG can be used as the basis for Earley recognition. Schabes's original method must be modified to respect the differences in compilation engendered by extended derivations. Such parsing rules, along with an extension that allows building of explicit derivation trees on-line as a basis for incremental interpretation, have been developed, and are presented in an extended version of this paper (Schabes and Shieber, 1992). In summary, the algorithm operates as a variant of Earley parsing on the corresponding LIG. The set of extended derivations can subsequently be recovered from the set of Earley items generated by the algorithm. The resultant algorithm can be further modified so as to build an explicit derivation tree incrementally as parsing proceeds; this modification, which is a novel result in its own right, allows the parsing algorithm to be used by systems that require incremental processing with respect to tree-adjoining grammars.

As a proof of concept, the parsing algorithm just described was implemented in Prolog on top of a simple, general-purpose, agenda-based inference engine. Encodings of explicit inference rules are essentially interpreted by the inference engine. The Prolog database is used as the chart; items not already subsumed by a previously generated item are asserted to the database as the parser runs. An agenda is maintained of potential new items. Items are added to the agenda as inference rules are triggered by items added to the chart. Because the inference rules are stated explicitly, the relation between the abstract inference rules described in this paper and the implementation is extremely transparent. Because the prototype was implemented as a meta-interpreter it is not particularly efficient. (In particular, the implementation does not achieve the theoretical  $O(n^6)$  bound on complexity, because of a lack of appropriate indexing.) Code for the prototype implementation is available for distribution electronically from the authors.

## 7 Conclusion

The precise formulation of derivation for tree-adjoining grammars has important ramifications for a wide variety of uses of the formalism, from syntactic analysis to semantic interpretation and statistical language modeling. We have argued that the definition of tree-adjoining derivation

must be reformulated in order to take greatest advantage of the decoupling of derivation tree and derived tree by manifesting the proper linguistic dependencies in derivations. The particular proposal is both precisely characterizable, through a compilation to linear indexed grammars, and computationally operational, by virtue of an efficient algorithm for recognition and parsing.

## References

Aravind K. Joshi, L. S. Levy, and M. Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10(1).

Anthony Kroch and Aravind K. Joshi. 1985. Linguistic relevance of tree adjoining grammars. Technical Report MS-CIS-85-18, Department of Computer and Information Science, University of Pennsylvania, April.

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

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman.

Philip Resnik. 1992. Lexicalized tree-adjoining grammar for distributional analysis. To appear in *Proceedings of the 14<sup>th</sup> International Conference on Computational Linguistics*.

Yves Schabes and Stuart M. Shieber. 1992. An alternative conception of tree-adjoining derivation. Technical Report 08-92, Harvard University.

Yves Schabes. 1991. Computational and mathematical studies of lexicalized grammars. Manuscript in preparation based on the author's PhD dissertation (University of Pennsylvania, August 1990).

Yves Schabes. 1992. Stochastic lexicalized tree-adjoining grammars. To appear in *Proceedings of the 14<sup>th</sup> International Conference on Computational Linguistics*.

Stuart M. Shieber and Yves Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13<sup>th</sup> International Conference*

*on Computational Linguistics (COLING'90)*, Helsinki.

K. Vijay-Shanker and Aravind K. Joshi. 1985. Some computational properties of Tree Adjoining Grammars. In *23<sup>rd</sup> Meeting of the Association for Computational Linguistics*, pages 82–93, Chicago, Illinois, July.

K. Vijay-Shanker and David J. Weir. 1990. Polynomial parsing of extensions of context-free grammars. In Masaru Tomita, editor, *Current Issues in Parsing Technologies*, pages 191–206. Kluwer Academic Publishers.

K. Vijay-Shanker. 1987. *A Study of Tree Adjoining Grammars*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.