March 1991

Unification-Based Tree Adjoining Grammars

K. Vijay-Shanker
University of Delaware

Aravind K. Joshi
University of Pennsylvania, joshi@cis.upenn.edu

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-25.

Unification-Based Tree Adjoining Grammars

Abstract

Many current grammar formalisms used in computational linguistics take a unification-based approach that use structures (called feature structures) containing sets of feature-value pairs. In this paper, we describe a unification-based approach to Tree Adjoining Grammars (TAG). The resulting formalism (UTAG) retains the principle of factoring dependencies and recursion that is fundamental to TAGs. We also extend the definition of UTAG to include the lexicalized approach to TAGs (see [Schabes *et al.*, 1988]). We give some linguistic examples using UTAG and informally discuss the descriptive capacity of UTAG, comparing it with other unificationbased formalisms. Finally, based on the linguistic theory underlying TAGs, we propose some stipulations that can be placed on UTAG grammars. In particular, we stipulate that the feature structures associated with the nodes in an elementary tree are bounded (there is an analogous stipulation in GPSG). Grammars that satisfy these stipulations are equivalent to TAG. Thus, even with these stipulations, UTAGs have more power than CFG-based unification grammars with the same stipulations.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-25.

Unification-Based Tree Adjoining Grammars

MS-CIS-91-25
LINC LAB 199

K. Vijay-Shanker
(University of Delaware)
Aravind K. Joshi
(University of Pennsylvania)

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

March 1991

[[To appear in "Unification-Based Grammars" (ed. Jurgen Wedekind), MIT PRESS 1991]]

Unification-Based Tree Adjoining Grammars*†

K. Vijay-Shanker
Dept. of Computer & Information Sciences
University of Delaware
Newark, DE 19711
U.S.A

Aravind K. Joshi
Dept. of Computer & Information Science
University of Pennsylvania
Philadelphia, PA 19104
U.S.A.

February 18, 1991

Abstract

Many current grammar formalisms used in computational linguistics take a unification-based approach that use structures (called feature structures) containing sets of feature-value pairs. In this paper, we describe a unification-based approach to Tree Adjoining Grammars (TAG). The resulting formalism (UTAG) retains the principle of factoring dependencies and recursion that is fundamental to TAGs. We also extend the definition of UTAG to include the lexicalized approach to TAGs (see [Schabes *et al.*, 1988]). We give some linguistic examples using UTAG and informally discuss the descriptive capacity of UTAG, comparing it with other unification-based formalisms. Finally, based on the linguistic theory underlying TAGs, we propose some stipulations that can be placed on UTAG grammars. In particular, we stipulate that the feature structures associated with the nodes in an elementary tree are bounded (there is an analogous stipulation in GPSG). Grammars that satisfy these stipulations are equivalent to TAG. Thus, even with these stipulations, UTAGs have more power than CFG-based unification grammars with the same stipulations.

1 Introduction

Tree Adjoining Grammars (TAG) were first introduced by Joshi, Levy, and Takahashi [1975]. The first study of this system, from the point of view of its formal properties and linguistic applicability,

*This work was partially supported by NSF grants MCS-82-19116-CER, DCR-84-10413,ARO grant DAA29-84-9-0027, and DARPA grant N0014-85-K0018.

†We want to thank Anne Abeillé, Sharon Cote, Tony Kroch, Megan Moser, Beatrice Santorini, Yves Schabes, Stuart Shieber, and David Weir for valuable discussion and comments on earlier versions of this paper, and Dawn Griesbach for significant editorial assistance.

was carried out by Joshi in [Joshi, 1985]. A detailed study of the linguistic relevance of TAGs was done by Kroch and Joshi in [Kroch and Joshi, 1986b]. Linguistic analyses of various constructions using the TAG formalism can also be found in [Kroch, 1986; Kroch and Joshi, 1986a; Kroch and Santorini, 1989].

In this paper we briefly define TAGs and show how TAGs can be embedded in a unification-based framework. By comparing the different operations used in TAGs and CFG-based formalisms, we state the reasons for embedding TAGs in the unification framework. These reasons are different from the reasons for embedding a CFG-based formalism in the unification framework. In Section 2, we define UTAG and compare the descriptive capacities of UTAG and TAG, focusing on the comparison of the implementation of adjoining constraints in the two systems. We then show how the lexicalized approach to TAGs [Schabes *et al.*, 1988] can be captured in UTAG.

In Section 3, we propose some stipulations on UTAG grammars in an attempt to capture some of the key features of the linguistic principles underlying TAGs. We also examine some of the consequences of these stipulations. A major consequence is that we can bound the size of the feature structures (in a manner similar to GPSG) associated with the nodes of the elementary trees of TAGs and still achieve greater descriptive and generative capacity than with CFG-based unification grammars having the same stipulations.

1.1 Introduction to Tree Adjoining Grammars

A Tree Adjoining Grammar (TAG) is specified by a finite set of *elementary* trees. Unlike the string rewriting formalisms which incorporate recursion into the rules that generate the phrase structure, a TAG factors recursion and dependencies into a finite set of elementary trees. The elementary trees in a TAG correspond to *minimal* linguistic structures that localize dependencies such as agreement, subcategorization, and filler-gap. There are two kinds of elementary trees: the *initial trees* and *auxiliary trees.* The initial trees roughly correspond to simple sentences (Figure 1). Thus, the root of an initial tree is labeled by the symbol S, and the nodes at the frontier are labeled by terminals.

The auxiliary trees (Figure 2) correspond roughly to minimal recursive constructions. Thus, if the root of an auxiliary tree is labeled by a nonterminal symbol, X, then there is a node (called the foot node) in the frontier of this tree which is labeled by X. The rest of the nodes in the frontier are labeled by terminal symbols.

We will now define the operation of adjunction. Let γ be a tree with a node labeled by X. Let β be an auxiliary tree, whose root and foot node are also labeled by X. Then, adjoining β at the node labeled by X in γ will result in the tree γ' illustrated in Figure 3.

So far, the only restriction we have placed on the set of auxiliary trees that can be adjoined at

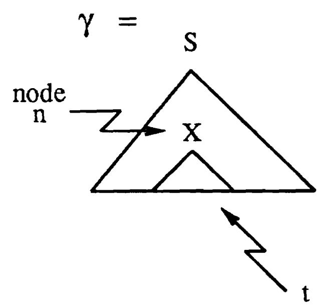


Figure 1: Initial Tree

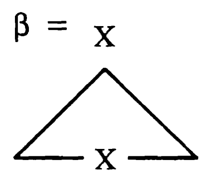


Figure 2: Auxiliary Tree

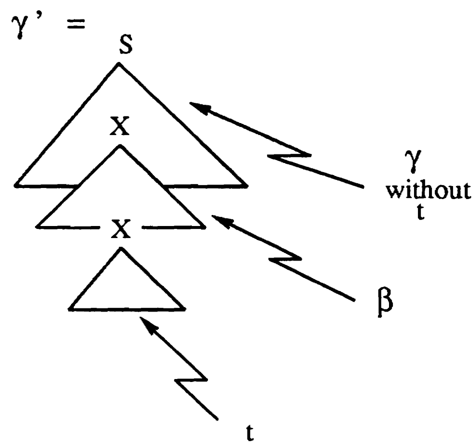


Figure 3: The operation of adjoining

a node is that the label of the node must be the same as the label of the root (and the foot) node of the auxiliary tree. Further restrictions can be placed on this set of auxiliary trees by enumerating the subset of auxiliary trees which can be adjoined at a particular node. This specification of a set of auxiliary trees, which can be adjoined at a node, is called the *Selective Adjoining* (SA) constraint. In the case where we specify the empty set, we say that the node has a *Null Adjoining* (NA) constraint. It is possible to specify that adjunction is mandatory at a node. In such a case, we say that the node has an *Obligatory Adjoining* (OA) constraint.

2 Unification-Based Tree Adjoining Grammars (UTAG)

The linguistic theory underlying TAGs is centered around the factorization of recursion and localization of dependencies[1] in the elementary trees. The elementary trees provide an *extended* domain of locality, such that the "dependent" items are available locally. Yet the extension of the domain of locality is only minimal in order that recursion is factored out and non-dependent items do not form a part of the same elementary tree. Thus, for example, the predicate and its arguments will be in the same tree, as will the filler and the associated gap. Our main goal in embedding TAGs in a unification framework is to capture this localization of dependencies.

2.1 Top and Bottom Feature Structures

In unification grammars, a feature structure is associated with a node, η, in a derivation tree. This feature structure is a description of that node and its relation to other nodes in the tree. This relationship with the other nodes (and the feature structure that describes it) may be broken into two parts.

1. The relation of η to its supertree (the siblings and ancestors of η). This view from above can be characterized by a feature structure (the *top* feature structure), say t_η.

2. The relation of η to its descendants. This view from below can be characterized by a feature structure say, b_η.

Although t_η and b_η are feature structures that make statements about the node η from different points of view, they both hold of the same node. In a CFG-based formalism, t_η and b_η have to be compatible. Consider an intermediate (non-leaf) node in some derivation tree of a CFG-based grammar. The feature structure t_η arises due to the rule that introduces η, whereas b_η arises due

[1]The types of dependencies we are interested in are agreement, subcategorization, filler-gap, etc.

to the rule that expands η. t_η (which relates the supertree to the node η) and b_η (which relates the subtree to the node η) must define the node in a consistent fashion since no new nodes (or treelets) can be introduced between supertree and subtree for this node. Hence, it is sufficient to associate just one feature structure (unification of the t and b feature structures) with such a node.

On the other hand, in a TAG formalism, due to the adjunction operation at an intermediate node (say η), an auxiliary tree *replaces* the node, introducing a new set of nodes (those of the auxiliary tree) between the supertree and the subtree of the node, η (where adjunction took place). Thus, after adjunction at η, t_η now relates the supertree of η to a node (the root of the auxiliary tree) that is different from the node (the foot of the auxiliary tree) that is related to the subtree of η by b_η. This approach of associating two feature structures (rather than one) with an intermediate node in an elementary tree is in the spirit of TAGs especially when we consider OA (obligatory adjoining) constraints in TAGs. A node with OA constraints *cannot* be viewed as a single node and must be considered as something that has to be replaced by an auxiliary tree. t and b are restrictions on the auxiliary tree that must be adjoined at this node. Note that if the node does not have an OA constraint, then we should expect t and b to be compatible. For example, in the final sentential tree (where there are no nodes with OA constraints), this node will be viewed as a single entity.

2.1.1 Feature Structures Associated with Nodes of Elementary Trees

In this section, we will discuss the feature structures that are associated with all the nodes of elementary trees in a UTAG. We have just stated that the need for associating two feature structures with a node arises due to the adjunction operation. In a TAG, adjunction can take place at a nonterminal node. Hence, with such nodes, we will associate the t and b feature structures. Since adjunction cannot take place at terminal nodes, we will associate only one structure with terminal nodes.[2]

2.1.2 The Adjoining Operation in UTAG

Let us now consider adjoining as shown in Figure 4. The notation we use for the trees in a UTAG is to write the t and b feature structures alongside each node, (using the standard matrix notation), with the t structure written above the b structure. Let us say that t_{root}, b_{root} and t_{foot}, b_{foot} are the t and b structures of the root and foot nodes respectively, of the auxiliary tree used for adjunction at the node η. Based on what t and b stand for, it is obvious that, upon adjunction, the

[2]It is possible to allow adjunction at nodes corresponding to pre-lexical items. In that case, we will have to associate two feature structures with pre-lexical nodes too.

statements t_η and t_{root} hold of the node corresponding to the root of the auxiliary tree. Similarly, the statements b_η and b_{foot} hold of the node corresponding to the foot of the auxiliary tree. Thus, upon adjunction, we unify t_η with t_{root}, and b_η with b_{foot}. In fact, this adjunction is permissible only if t_{root} and t_η are compatible and b_{foot} and b_η are compatible. If we do not adjoin at the node, η, then we unify t_η with b_η. In a TAG, at the end of a derivation, the tree generated must not have any nodes with OA constraints. We check this by unifying the t and b feature structures of every node at the end of a derivation.

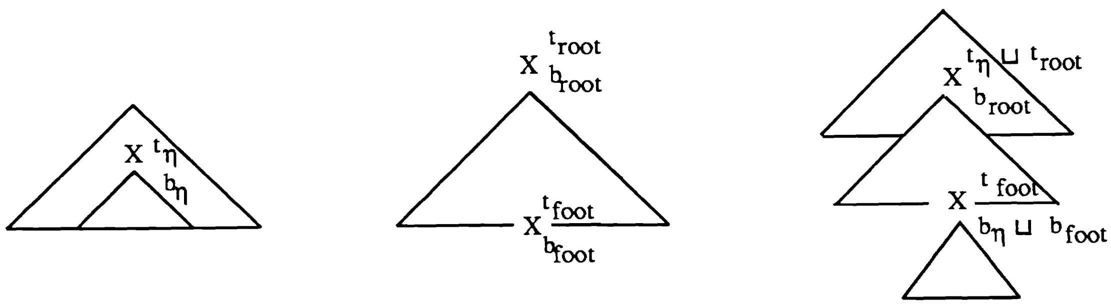


Figure 4: Feature structures and adjunction

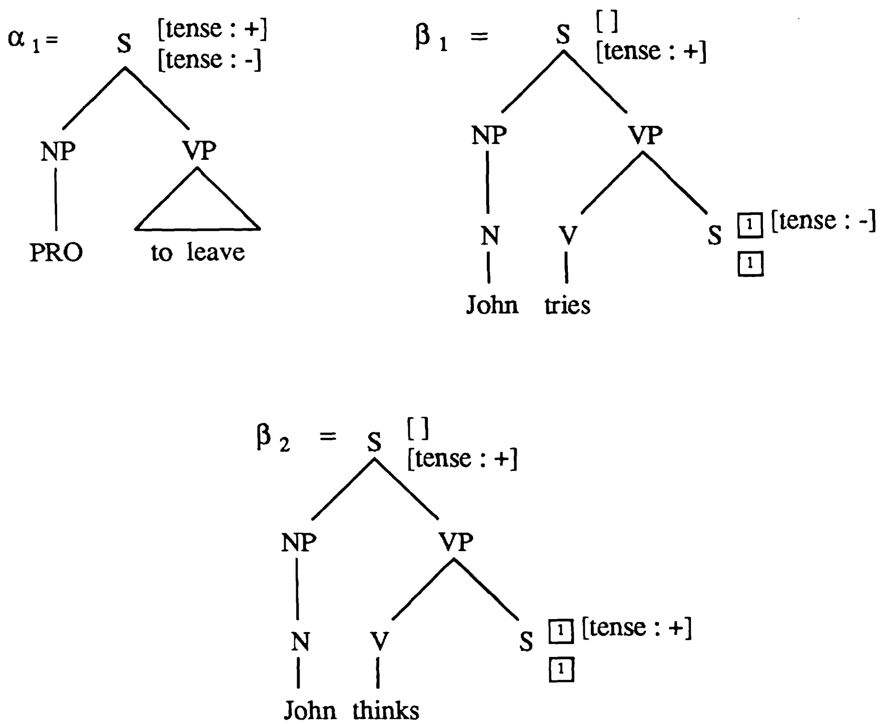
We note here that, just as in a TAG, the elementary trees which are the domain of co-occurrence restrictions are available as single units during each step of the derivation. Thus, most of these co-occurrence constraints can be checked even before the tree is used in a derivation, and this checking need not be linked to the derivation process.

2.2 Unification and Adjoining Constraints

We will now discuss how the adjoining constraints are implemented in UTAG. As we have already shown, t_η and t_{root}, and b_η and b_{foot} must be compatible for adjunction to occur. We can thus specify feature-values in these t, b statements to state the local constraints such that

1. if an auxiliary tree should not be adjoined at a node (because of its SA (selective adjoining) constraint) then some unification (t_η with t_{root}, or b_{foot} with b_η) involved in our attempt to adjoin this auxiliary tree will fail, and
2. if a node has an OA constraint, we should ensure that an appropriate auxiliary tree *does* get adjoined at that node. This is ensured if t_η is incompatible with b_η.

The example, given in Figure 5, illustrates the implementation of both the OA and SA constraints. In this paper, we do not distinguish S from $\overline{S}$, in order to simplify the discussion. In all the examples in this paper, we have shown only the relevant features in the t and b feature structures. Also we have shown the t and b feature structures only for the nodes that are relevant to the discussion.



[Note: coindexing by $\boxed{1}$ in β_1 means that t_{foot} is the same as b_{foot} in β_1, similarly in β_2].

Figure 5: Illustration of implementation of SA and OA constraints

The view of the root node of α_1 from below suggests that b statement for this node should assert that the value of the *tense* attribute is $-$ (or untensed). However, the t statement should assert *tense*:+ (since every complete sentence must be tensed).[3] Thus, an auxiliary tree whose root node corresponds to a tensed sentence and whose foot node dominates an untensed sentence can be adjoined at this node. Therefore, only those auxiliary trees whose main verb subcategorizes for an untensed sentence (or an infinitival clause) can be adjoined at the root node of this initial tree. This shows why only an auxiliary tree such as β_1 can be adjoined (because all the relevant unifications

[3]Note that we said that t structure is a statement about the node while viewing the node from the top, and hence it is a statement concerning the entire subtree below this node (i.e., including the part due to an auxiliary tree adjoined at the node). Thus, although the root node being considered has no supertree, we can still make the assertion *tense*:+ in the t feature structure of this node.

involved in adjoining will succeed), whereas an auxiliary tree (β_2) corresponding to *John thinks S* cannot be adjoined (because b_{root} and b_{foot} will fail to unify), since *thinks* subcategories for a tensed sentence. This example also serves to illustrate the implementation of the OA constraint at the root of α_1, since the t and b feature structures for this node are not unifiable.

2.2.1 Comparing the Implementation of Adjoining Constraints

In the TAG formalism, local constraints are specified by enumeration. However, specification by enumeration is not linguistically desirable. In a UTAG we associate two feature structures with each node, which are declarations of linguistic facts about the node. That only appropriate trees get adjoined is a corollary of the fact that only trees compatible with these declarations are acceptable trees in a UTAG. As a result, in a UTAG, constraints are dynamically instantiated and are not pre-specified as in a TAG. This can be advantageous (in terms of economy of grammar specification). For example, consider the derivation of the sentence (obtained by adjoining β_3 and then β_4 to the derived tree):

What do you think Mary thought John saw

In the TAG formalism, we are forced to replicate some auxiliary trees. Consider the auxiliary tree β_3 in the TAG fragment in Figure 6.

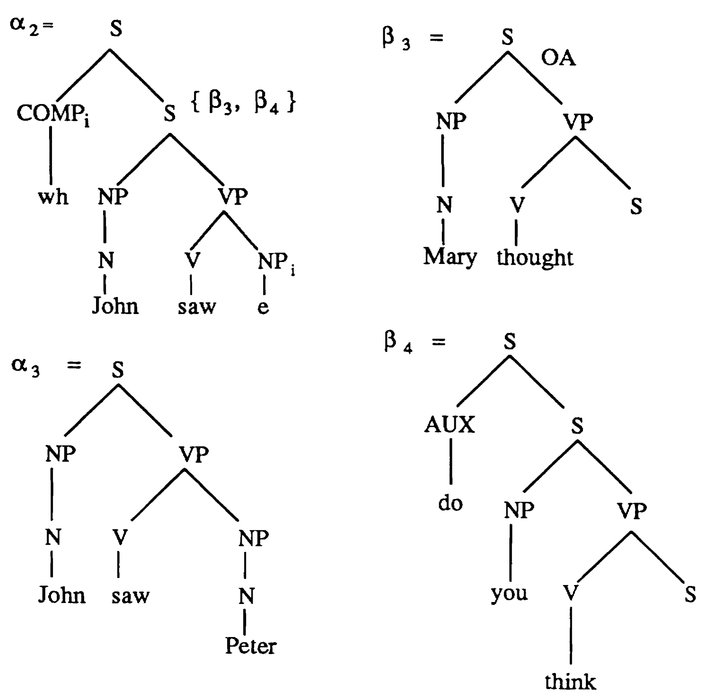


Figure 6: A TAG fragment

Since the intermediate phrase *what Mary thought John saw* (obtained by adjoining β_3 to α_2 at the internal S node) is not a complete sentence, we will have to use OA constraints at the root of the auxiliary tree β_3. However, this root node should not have OA constraints when it is used in some other context; as in the case of (obtained by adjoining β_3 to α_3 at the root of α_3):

Mary thought John saw Peter

Thus we will need another auxiliary tree, say β_5, (not shown in Figure 6) with exactly the same tree structure as β_3 except that the root of β_5 will not have an OA constraint. Further, the root nodes in α_2 and α_3 will need SA constraints that allow for adjunction only by β_3 and β_4 respectively.

We will now show that by using the fact that constraints are dynamically instantiated in a UTAG, we need only one tree, say β_6 (see Figure 7).

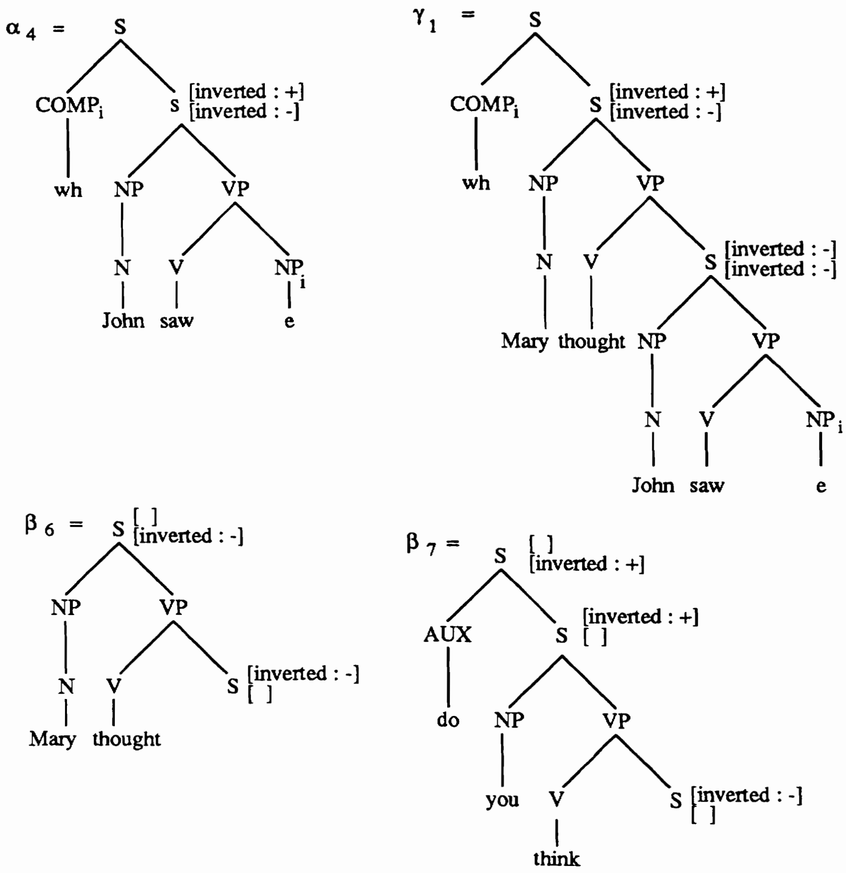


Figure 7: A UTAG fragment

When used in the derivation of (β_6 adjoined to α_4 at the internal S node, giving γ, and then β_7 adjoined to γ_1 at the internal S node just below the root of γ_1):

What do you think Mary thought John saw

t_{root} of β_6 inherits the feature *inverted*:+ which it otherwise does not have, and b_{root} of β_6 inherits the feature *inverted*:−.

Thus, the node which corresponds to root of β_6, by the dynamic instantiation of the feature structure, gets an OA constraint. Note that there will not be any OA constraint on the nodes of the final tree γ_1 corresponding to:

What do you think Mary thought John saw.

Also, the root of the auxiliary tree, corresponding to *Mary thought S*, does not get an OA constraint, when this tree is used in the derivation of the sentence (obtained by adjoining β_6 to α_3 in Figure 6):

Mary thought John saw Peter.

2.3 Extending UTAG to Include Lexicalized Approach to TAGs

In Section 1.1, we defined the elementary trees which are elaborated up to the lexical items, with the adjoining operation as the only operation of composition of trees. In a more recent approach, due to Schabes, Abeillé, and Joshi [1988], where lexicalization of TAGs is considered, this is no longer the case. The crucial aspect of lexicalized TAGs relevant to this paper is that a finite set of elementary trees are associated with each lexical item, which will usually be the head (or the functional head) of the structure. Following the terminology of Schabes, Abeillé, and Joshi, we call this lexical item the lexical *anchor* of the associated trees. It is not necessary to consider these trees to be fully expanded to lexical items, but only that they be elaborated to include the lexical anchor. This leaves the possibility that certain nodes in the frontier will be labeled by nonterminal nodes in contrast to the original definition of TAGs. These nodes are marked for substitution (which is, of course, mandatory) by *initial* trees with the same root symbol.[4] We now give an example of such trees (Figure 8). Further details of this lexicalized approach to TAGs may be found in [Schabes *et al.*, 1988].

In Section 2.1, we have defined the interpretations of the t and b feature structures that are associated with nodes of trees in a UTAG grammar. We stated that adjunction takes place at nodes labeled by nonterminal symbols, and hence we associated both t and b feature structures with the nonterminal nodes. However, in the case of the nodes marked for substitution, we have to associate only one feature structure. The reason is that we can only define the constraints on the possible subtree rooted at this node due to its relation with the supertree (i.e., the t feature structure). Associating the b feature structure is not appropriate, consistent with intuition underlying the introduction of substitution nodes where we expect substitution first before any adjunction.

[4]Initial trees no longer need to be labeled by S only.

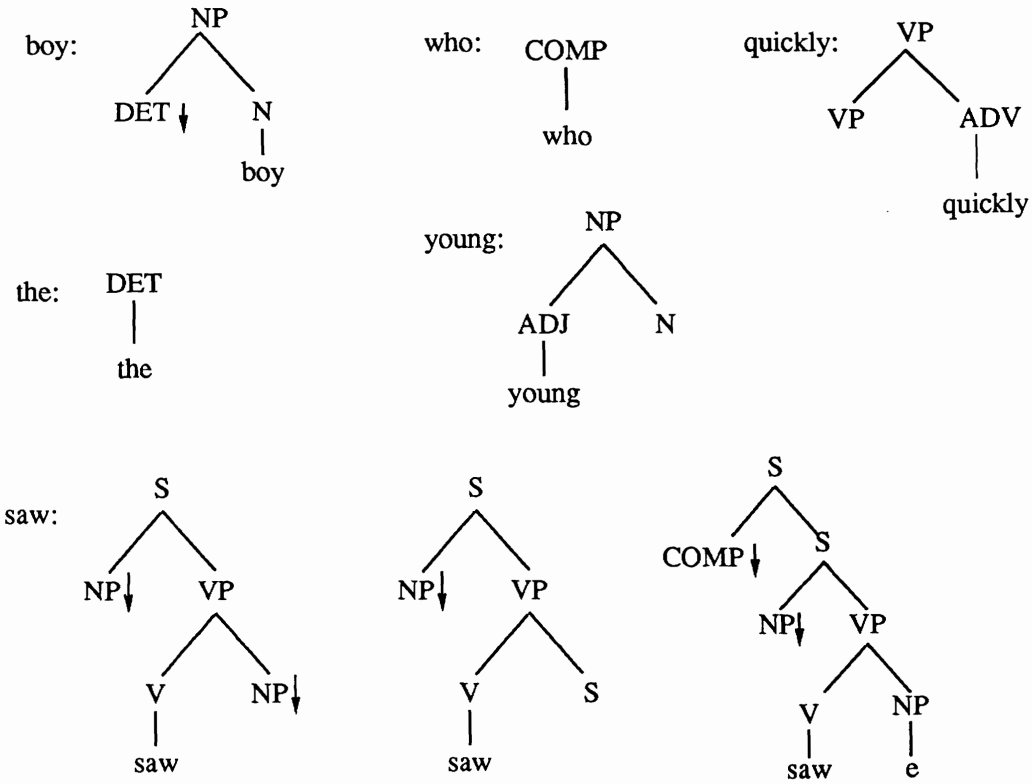


Figure 8: Example of Elementary Trees in Lexicalized TAGs

The operation of substitution in UTAG is defined as follows. Note that the root of the initial tree used in the substitution operation will have two feature structures, say t_{root} and b_{root}. The node, η, where substitution takes place has only one feature structure (say t_η) as defined above. Upon substitution, t_η has to be unified with t_{root}, since both these statements now hold of this node, viewing it from above. On the other hand, when we consider it from below, only b_{root} holds. In Figure 9, we give an example of the use of substitution in UTAG.

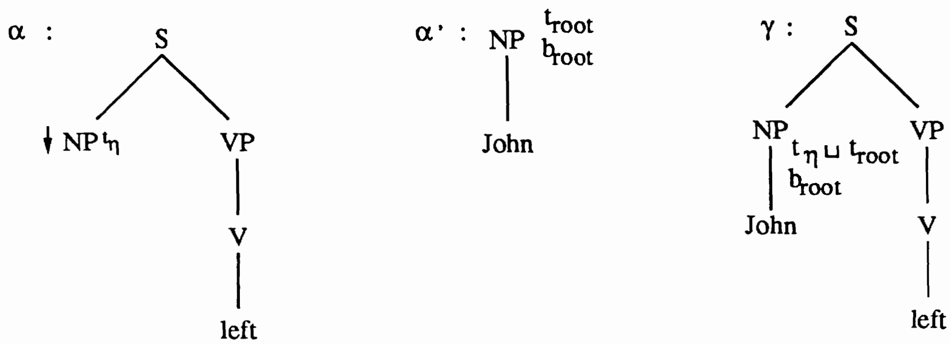


Figure 9: Substitution operation in UTAG

3 Some Possible Linguistic Stipulations on UTAG

In this section, we will discuss some possible stipulations for a UTAG grammar, which are linguistically motivated. We could have included these stipulations in our unification-based approach to TAGs right from the beginning. However, we have chosen to define UTAG in the most general manner and then consider the stipulations one by one.

The current linguistic theory underlying TAGs assumes that every foot node has a NA constraint and ensures that the adjunction operation does not alter the grammatical relations defined by the intermediate tree structures. For example, consider a derivation of the sentence, *Mary thought John saw Bill hit Jill*, where there is an intermediate tree in the derivation corresponding to *Mary thought Bill hit Jill* obtained by adjoining an auxiliary tree corresponding to *Mary thought* S to an initial tree corresponding to *Bill hit John.* Here we have the relation of Mary thinking that "Bill hit Jill." This relation will be altered if we adjoin an auxiliary tree corresponding to *John saw* S at the node corresponding to the foot node of the auxiliary tree corresponding to *Mary thought* S. If a NA constraint is stipulated for the foot node of every auxiliary tree, then the above derivation will be blocked. The only derivation that is possible is the one with an intermediate tree corresponding to *John saw Bill hit Jill* and then to *Mary thought John saw Bill hit Jill.*

One way to implement this stipulation is to insist that only one feature structure be associated with the foot node, i.e., the t_{foot} and b_{foot} are combined. The definition of adjunction then needs to be modified so that adjunction at a node with only one feature structure will be disallowed.[5]

The second stipulation involves the complexity of the feature structure associated with the nodes. So far, we have not placed any restrictions on the growth of these feature structures. One of the possible stipulations with linguistic relevance is to put a bound on the information content in these feature structures. This results in a bound on the size of feature structures (i.e., the number of possible features) associated with a node, as well as on the size of the values of these features. This stipulation is comparable to a restriction on feature structures in GPSG. A UTAG grammar, which incorporates this stipulation, will be equivalent to a TAG from the point of view of generative capacity, but it will have an enhanced descriptive capacity because TAGs are more powerful than CFGs and belong to the class of mildly context-sensitive grammars.

Unbounded feature structures have been used to capture the subcategorization phenomenon by having feature structures that act like stacks (and hence are unbounded in size). However, in a TAG, the elementary trees specify the subcategorization domain. As noted earlier, the elements subcategorized by the main verb in an elementary tree are part of the same elementary tree.

[5]The current implementations of the TAG parsers have this stipulation.

Thus, with the feature structures associated with the elementary trees we can just point to the subcategorized elements and do not need any further devices. Thus, any stack-based mechanism that might be needed for subcategorization is provided by the TAG formalism itself. This follows from the fact that the tree sets generated by TAGs have context free paths (unlike the trees of CFGs which have regular paths). This additional power provided by the TAG formalism has been used in giving an account of West Germanic verb-raising [Kroch and Santorini, 1989].

A UTAG grammar with these two stipulations will be called an FTAG (feature structure-based TAG). The TAGs for English and French developed so far are in the framework of FTAGs.

4 Conclusion

We have shown a method of embedding TAGs in a feature structure based framework. This system takes advantage of the extended domain of locality of TAGs and allows linguistic statements about co-occurence of features of dependent items to be stated within the scope of elementary trees. The specification of local constraints in a TAG is by enumeration, which is not satisfactory from a linguistic point of view. We show that in UTAG, we can avoid such specifications. Instead, the declarative statements made about nodes are sufficient to ensure that only the appropriate trees get adjoined at a node. We also illustrate how duplication of information can be avoided in UTAGs in contrast to TAGs.

Some linguistic analyses require extensions of TAGs to TAGs with multi-component adjoining [Joshi, 1987; Weir, 1988] (simultaneous adjunction of a set of trees into distinct nodes of an elementary tree) [Kroch and Joshi, 1986a; Kroch, 1986]. It can be shown that these analyses can be easily accommodated in UTAGs.

The Earley-style parser, described by Schabes and Joshi [1988], has been extended to parse UTAGs. The reason such an extension is possible is because the t and b feature structures for every node in UTAG are compatible with the characterization of a node in terms of two substrings in the parsing algorithm described in [Schabes and Joshi, 1988].

We have proposed a restricted version of UTAG. In a manner similar to GPSG, we place a bound on the information content of feature structures associated with the nodes of trees used in the grammar. The resulting system has the same generative power as TAGs; however it provides increased descriptive and generative capacity, as compared to CFGs (and therefore GPSGs), due to the extended domain of locality of TAGs.

In a later paper [Vijay-Shanker, 1991], a fixed-point semantics for UTAG will be developed using the work of Rounds and Kasper [1986] and Johnson [1987] on the logical formulation of feature structures and the work of Rounds and Manaster-Ramer [1987] on the representation of unification-based grammars.

References

[Johnson, 1987] M.E. Johnson. *Attribute-value logic and the theory of grammar.* PhD thesis, Stanford University, Stanford, CA, 1987.

[Joshi, 1987] A.K. Joshi. Word-order variation in natural language generation. In *Proceedings of the Annual Conference of the American Association for Artificial Intelligence (AAAI-87)*, Seattle, July 1987.

[Joshi *et al.*, 1975] A.K. Joshi, L.S. Levy, and M. Takahashi. Tree adjunct grammars. *Journal of Computer Systems Science*, 10(1), 1975.

[Joshi, 1985] A. K. Joshi. How much context-sensitivity is necessary for characterizing structural descriptions — Tree Adjoining Grammars. In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Processing — Theoretical, Computational and Psychological Perspective*, Cambridge University Press, New York, NY, 1985. Originally presented in 1983.

[Kroch, 1986] A. Kroch. Asymmetries in long distance extraction in a TAG grammar. In *New Conceptions of Phrase Structure*, 1986.

[Kroch and Joshi, 1986a] A.S. Kroch and A.K. Joshi. Analyzing extraposition in a tree adjoining grammar. In G. Huck and A. Ojeda, editors, *Syntax and Semantics: Discontinuous Constituents*, Academic Press, New York, NY, 1986.

[Kroch and Joshi, 1986b] A.S. Kroch and A.K. Joshi. *Linguistic Relevance of Tree Adjoining Grammars.* Technical Report MS-CIS-85-18, Dept. Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1986. To appear also in *Linguistics and Philosophy.*

[Kroch and Santorini, 1989] A. Kroch and B. Santorini. The derived constituent structure of West Germanic verb raising construction. In R. Freidin, editor, *Proceedings of the Princeton Workshop on Comparative Grammar*, MIT Press, Cambridge, MA, 1989.

[Rounds and Manaster-Ramer, 1987] W. Rounds and A. Manaster-Ramer. A logical version of functional grammar. In *Proceedings of the* 25^{th} *Annual Meeting of the Association of Computational Linguistics*, Stanford, 1987.

[Rounds and Kasper, 1986] W. C. Rounds and R. Kasper. A complete logical calculus for record structures representing linguistic information. In *IEEE Symposium on Logic and Computer Science*, 1986.

[Schabes and Joshi, 1988] Y. Schabes and A. K. Joshi. An Earley-type parsing algorithm for tree adjoining grammars. In 26^{th} *Meeting of the Association for Computational Linguistics*, Buffalo, 1988.

[Schabes *et al.*, 1988] Y. Schabes, A. Abeillé, and A. K. Joshi. New parsing strategies for tree adjoining grammars. In 12^{th} *International Conference on Computational Linguistics*, Budapest, Hungary, 1988.

[Vijay-Shanker, 1991] K. Vijay-Shanker. *Semantics for Feature Structure Based Tree Adjoining Grammars.* 1991. forthcoming.

[Weir, 1988] D. Weir. *Characterizing Mildly Context-Sensitive Grammatical Formalisms.* PhD thesis, Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1988.