

Structure Sharing in Lexicalized Tree-Adjoining Grammars*

K. Vijay-Shanker

Dept. of Computer & Information Science
University of Delaware
Newark, DE 19716, USA
vijay@udel.edu

Yves Schabes

Dept. of Computer & Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389, USA
schabes@unagi.cis.upenn.edu

Abstract

We present a scheme for efficiently representing a lexicalized tree-adjoining grammar (LTAG). The proposed representational scheme allows for structure-sharing between lexical entries and the trees associated with the lexical items. A compact organization is achieved by organizing the lexicon in a hierarchical fashion and using inheritance as well as by using lexical and syntactic rules.

While different organizations (Flickinger, 1987; Pollard and Sag, 1987; Shieber, 1986) of the lexicon have been proposed, in the scheme we propose, the inheritance hierarchy not only provides structure-sharing of lexical information but also of the associated elementary trees of extended domain of locality. Furthermore, the lexical and syntactic rules can be used to derive new elementary trees from the default structures specified in the hierarchical lexicon.

In the envisaged scheme, the use of a hierarchical lexicon and of lexical and syntactic rules for lexicalized tree-adjoining grammars will capture important linguistic generalizations and also allows for a space efficient representation of the grammar. This will allow for easy maintenance and facilitate updates to the grammar.

1 Motivations

Lexicalized tree-adjoining grammar (LTAG) (Schabes et al., 1988; Schabes, 1990) is a tree-rewriting formalism used for specifying the syntax of natural languages. It combines elementary lexical trees with two operations, adjoining and substitution. In a LTAG, lexical items are associated with complex syntactic structures (in the form of trees) that define the various phrase structures they can participate in. LTAG allows for unique syntactic and semantic properties:

- The domain of locality in LTAG is larger than for other formalisms, and
- Most syntactic dependencies (such as filler-gap, verb-subject, verb-objects) and some semantic

dependencies (such as predicate-argument) have been localized within the elementary trees stated in the lexicon.

These unique properties of LTAGs have been shown to be linguistically very useful (Kroch and Joshi, 1985; Kroch, 1987; Kroch, 1989; Abeillé, 1988; Shieber and Schabes, 1990). However these same aspects can cause many practical problems. This is because there is considerable redundancy of information among the elementary trees that provide the enlarged domain of locality. So far, the lexicon of a LTAG has been organized in a completely flat manner, not allowing for any sharing of syntactic or semantic properties of the lexical items. Also, in the current organization there is no structure sharing among the different trees associated with the different lexical items as they are stated independently of each other. For example, Figure 1 shows some of the trees associated with the lexical item 'eat'.

In Figure 1, the tree α_1 corresponds to a declarative sentence, α_2 to a WH-question on its subject and α_3 to a relative clause in which the subject has been relativized. This example illustrates the redundancy among the elementary trees found in the lexicon. For example, the effect of the rule $S \rightarrow NP VP$ is found in all trees associated with a verb. Similarly, $VP \rightarrow V NP$ is found in all trees associated with a transitive verb. The current implementation of the LTAG for English (Abeillé et al., 1990) comprises over 800 sentential tree frames (trees without lexical items). Each one of these frames includes a part that corresponds to the rule $S \rightarrow NP VP$. This problem of replication of information has reached an acute stage and any practical natural language processing system based on LTAG must address this issue.

An equally serious problem is one of maintaining and updating such a grammar. It arises due to the lack of structure-sharing and of statements of various independent principles that make up the elementary trees. Any small change to some aspect of the design of the grammar could necessitate making changes to possibly hundreds of trees manually. For instance, an addition of a constraint equation associated with the rule $S \rightarrow NP VP$ would affect the description of every tree associated with a verb; a change to the way wh-questions are formed must be propagated to every tree for wh-question. Furthermore, one can only manually verify that such an update does not conflict with

*The first author is partially supported by NSF Grant IRI90-16591. The second author is partially supported by DARPA Grant N0014-90-31863, ARO Grant DAAL03-89-C-0031 and NSF Grant IRI90-16592.

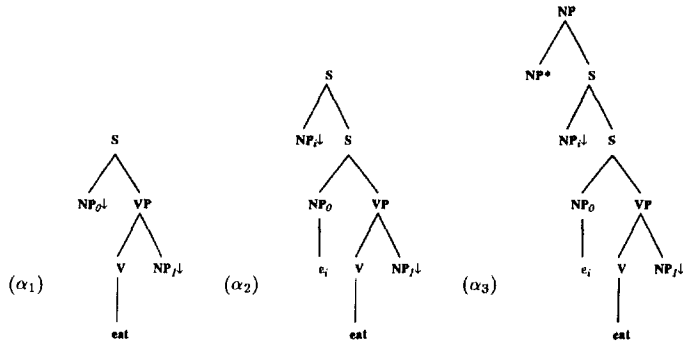


Figure 1: Sample of Elementary Trees in a LTAG

any other principle already instantiated. Given the size of the grammar, this is not a feasible task.

2 Goals of the Proposed Work

The problems mentioned above indicate an urgent need for addressing the issue of organization of a LTAG. For a LTAG, much of this effort would have to deal with the organization of the lexicon and the elementary trees.

Proposals for a compact representation of the lexicon and grammars have been suggested. For example, Flickinger (1987) and Pollard and Sag (1987) use a hierarchical lexicon and rules for implementing Head-driven Phrase Structure Grammars. Shieber (1986) proposed the use of default inheritance combined with templates and of transformation rules in the PATR-II system for organizing a unification based grammar. Lexical redundancy rules have been used in LFG (Bresnan and Kaplan, 1983) to capture relations among lexical items. Gazdar et al. (1985) proposed the use of meta-rules for expressing transformational relationships.

There has been suggestions for compacting the size of a tree-adjoining grammar lexicons (Becker, 1990; Habert, 1991). However, they only partially solve the problem since they fail to combine in a uniform way a compact representation of the lexicon and, at the same time, of their associated elementary trees.

In this paper, we present a scheme to efficiently represent a LTAG and illustrate this scheme by examples. We examine the information that needs to be associated with the classes in our hierarchical organization in order that we can represent the elementary trees of a LTAG. Our main concern in this paper is the proposal for organizing a LTAG. In order to be concrete, we consider the representation of a particular grammar for English (Abeillé et al., 1990). While the elegance (and the correctness) of the mechanisms used to capture linguistic generalizations is an important issue in such an enterprise, these linguistic concerns are beyond

the scope of this work. We give no linguistic motivations for the grammar that is being represented, nor for some of the methods used to represent it. The linguistic aspects of the work presented in this paper are meant to be suggestive. Also, while our scheme borrows heavily from Flickinger (1987), it is differentiated from similar enterprises in that we consider the representation of syntactic structures (in the form of trees) associated with lexical items. For this reason, we concentrate on the representation of the tree structures.

The representation we propose allows for structures sharing between lexical entries and for syntactic and lexical rules while being lexically sensitive. Lexical items as well as the elementary trees found in a lexicalized tree-adjoining grammar are organized in a hierarchical lexicon using inheritance. Furthermore, the lexical rules relate the information associated with lexical entries together. In particular, they derive new elementary trees of extended domain of locality from the one found in the hierarchical lexicon. Lexical idiosyncrasies are specified in the particular lexical entries.

3 Lexical Organization

The lexical entries (LEs) are organized in hierarchical fashion. The value of an attribute of lexical entry in the lexicon is either obtained by inheritance or by local specification. We allow for overwriting inherited attributes by assuming that the local specification has a higher precedence. Figure 2 shows a fragment of the hierarchy for verbs. The lexicon associates lexical items with a set of classes.

Entries specify relationships and properties of sets of nodes in trees which will be associated with the lexical items. The framework for describing the tree that will be associated with the lexical item is very similar to unification based tree-adjoining grammar (Vijay-Shanker, 1992) in which the trees are described with partial descriptions of their topology (Rogers and Vijay-Shanker, 1992) using statements of domination

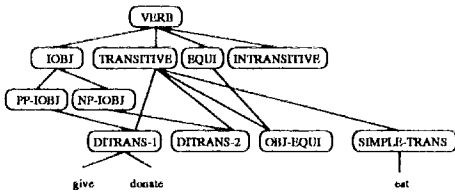


Figure 2: Fragment of the Lexicon

and linear precedence. We do not discuss the description language in which these trees are stated. Instead, we will pictorially represent these partial descriptions of trees.

For the purposes of this paper, in our representation scheme, we will focus on the descriptions of associated elementary trees.

Each class comprises of the following attributes (among others):

- *superclasses*, the set of immediate ancestor classes from which the current class inherits.
- *nodes*, the set of entities involved in the lexical entry.
- *description*, a partial description of a tree. This description consists of partial statements of domination, immediate domination and linear precedence over the set of nodes given in the previous attribute. In the following, we will ignore the linear precedence relationship. The immediate domination relationship will be illustrated by a plain line and the domination relationship by a dotted line. The language of this description and its semantic is given by Rogers and Vijay-Shanker (1992). The dashed line between tree nodes does not mean they are necessarily different nodes. It is used to indicate the two nodes in question could be the same or if they are different then one of them dominates the other in the manner indicated.
- *constraint equations* are unification equations that hold between the set of nodes. These equations specify feature structures associated with the set of nodes. Attributes such as agreement (*agr*) or case (*case*) are found in these equations.
- *completion*; $y = completion(x)$ specifies that y is the lowest node in the tree which does not require any argument of the predicative element x . This will be used, for example, in defining how the tree for wh-question is obtained.
- *head-daughter*; $x = head-daughter(y)$. This will be used in propagation of features by an implicit assumption of head-feature convention.
- *argument node*; *arg* specifies the node for the argument being introduced by the entry. This will be used to identify nodes that are mentioned in

different classes e.g. in NP-IOBJ or used in the syntactic rules such as for WH-movement.

- *linear precedence* (LP) statements which define precedence among nodes within the framework of ID/LP TAG proposed by Joshi (1987).
- *anchor*; *anchor* = x specifies that the node x is the anchor node of the tree being described.

For each entity in the hierarchy, attributes (such as arg) of some its ancestors can be referred to for further specifying the description while inheriting the description of its ancestors.

We can now consider an example. The following entry can be associated with the class VERB:¹ In this entry, as well as in the following entries, we do not give the full specification but specify only that part which is relevant to the discussion.

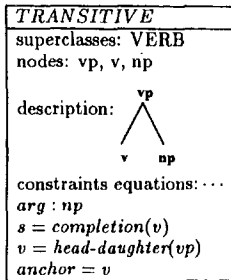
VERB	
nodes:	s, np, vp, v
description:	
constraints equations:	$\begin{cases} s. < cat > = S \\ np. < agr > = vp. < agr > \\ np. < case > = nom \\ \vdots \end{cases}$
<i>arg</i> :	np
$s = completion(v)$	
$vp = head-daughter(s)$	
<i>anchor</i> =	v
$np < vp$	

This entry specifies partially the tree structure for every verb, indicating that (by default) each verb must have a subject. It is important to note that despite the pictorial representation used, s, np, vp, v are used to refer to node and not to their labels.

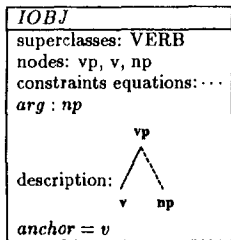
The following entry is associated with the class of transitive verbs (TRANSITIVE):²

¹The tree described below could have been predicted from general principles such as HPSG's rule stated on page 149 in Pollard and Sag (1987).

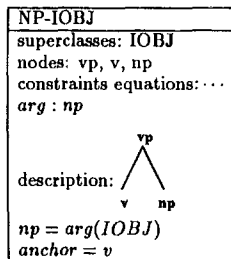
²Similarly, the tree described below could have been predicted from HPSG's rule stated on page 151 in Pollard and Sag (1987).



The following entry is associated with the class of verbs taking an NP as indirect objects(IOBJ) which may be possibly found within a prepositional phrase or not:

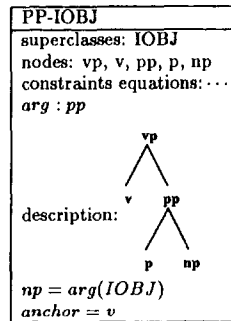


The following entry is associated with the class of verbs taking an NP as indirect objects (NP-IOBJ):

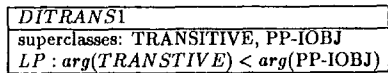


The equality $np = arg(IOBJ)$ used in the above frame forces the NP argument introduced in *IOBJ* (a superclass of *NP-IOBJ*) to be immediately dominated by the VP node, thus disallowing it being embedded in a prepositional phrase.

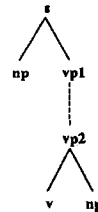
However, the following entry is associated with the class of verbs taking a prepositional phrase (PP-IOBJ):



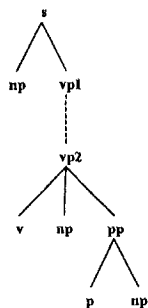
The following entry is associated with the class of ditransitive verbs taking a noun phrase as direct object and a prepositional phrase as an indirect object. The entry only specifies that the NP direct object must precede the NP introduced by the prepositional phrase.



The description of the default tree for *DITRANSI* is inherited from *VERB*, *TRANSITIVE*, *IOBJ*, *PP-IOBJ*. From the descriptions given in *VERB* and in *TRANSITIVE* we obtain the following structures:



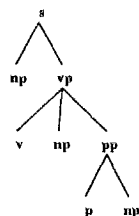
Note that the VP node in *VERB* dominates the verb node whereas the one introduced in *TRANSITIVE* immediately dominates the verb node. This results in the VP node introduced in verbs dominating (hence the dashed line) the VP node introduced in the *TRANSITIVE* frame. This kind of reasoning that leads to the formation of complex tree structures is given by Rogers and Vijay-Shanker (1992). Proceeding with the description of the tree structure inherited from *IOBJ* and *PP-IOBJ* we get:



which is used as default structure for verbs that belong to *DITRANS1*.³

In general this method for building a tree to be associated with a lexical item can be described as follows. First the nodes described in each superclass of the lexical entry are collected along with the statements of relationships specified between the nodes. This may require renaming of nodes in case nodes in different classes are given the same name. For instance, when we collect the nodes specified in *VERB* and *TRANSITIVE*, the *VP* nodes specified in them must be renamed (say as *vp1* and *vp2*) as must the *NP* nodes (say, the node for the subject that is specified in *VERB* gets renamed as *np1* and the object specified in *TRANSITIVE* gets renamed as *np2*). Next we must add an extra statement to explicitly equate the anchor nodes specified. Now if we additionally inherit the descriptions from *IOBJ* and *PP-IOBJ* the two *NP* nodes introduced get renamed but identified as a result of the identification suggested in *PP-IOBJ*. Notice that the identification of the *VP* nodes in *TRANSITIVE*, *IOBJ*, and *PP-IOBJ* does not get occur at this point. Such an identification gets done when we pass the tree descriptions collected to the machinery described by Rogers and Vijay-Shanker (1992). Since the anchors specified in these three classes get identified, the three *VP* nodes specified (in *TRANSITIVE*, *IOBJ*, and *PP-IOBJ*) as the parents of these anchor nodes must also get identified. Using this type of reasoning about the structural properties of trees, the structure given above gets created. To complete the discussion of the inheritance of the tree descriptions, the *head-daughter* relations are noted in order that they can be used for feature sharing. Also the set of *arg* nodes are also collected and called the *args* of the lexical entry. For example, the *args* in the case above would be *np1* (from *VERB*), *np2* (from *TRANSITIVE*), *np3* (from *IOBJ*), and *pp* (from *PP-IOBJ*). Later, in the syntactic rule, *WH-QUESTION*, we use the *args* of a lexical entry to indicate the set of possible nodes that can be moved.

In TAG the structure we derived above for *DITRANS1* is represented in the form of the following tree:

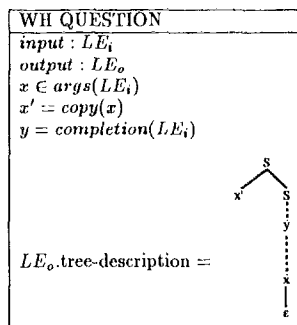


with two feature structures (top and bottom) associated with the *VP* node to indicate the collapsing of two *VP* nodes linked by domination. This process is also described in (Rogers and Vijay-Shanker, 1992).

4 Lexical and Syntactic Rules

The second mechanism we adopt for structure-sharing is the use of lexical and syntactic rules to capture inflectional and derivational relationships among lexical entries. The mechanism is very similar than the one proposed by Flickinger (1987), however it differs from it since we derive elementary trees of extended domain of locality. Lexical and syntactic rules relate an input lexical entry to an output lexical entry. The output lexical entry gets its information from the input lexical entry, lexical and syntactic rules, and possibly additional idiosyncratic information specified in the lexicon.

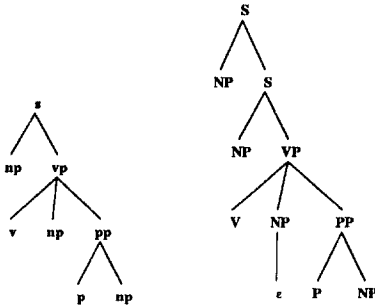
We illustrate the use of lexical and syntactic rules by examples. In the following, we will focus our attention to the derivational relationships and also to the output tree description. Consider the rule for *wh-question*.



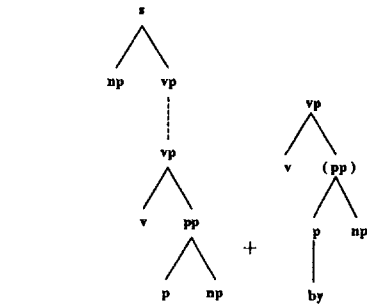
We treat formation of structure for *wh-question* as relation between two lexical entries specified here as LE_i and LE_o . The tree description in LE_o indicates that an argument node (x) in the tree described in LE_i can be moved leaving a trace. Here the relationship between x and y is obtained from the description in LE_i . $\text{Copy}(x)$ indicates that a copy of the description of entire sub-tree rooted at node x needs to be recorded in output description. In the resulting description, the filler is shown to C-command the gap.

³If needed, the value of the preposition can be specified by additional information at the lexical entry.

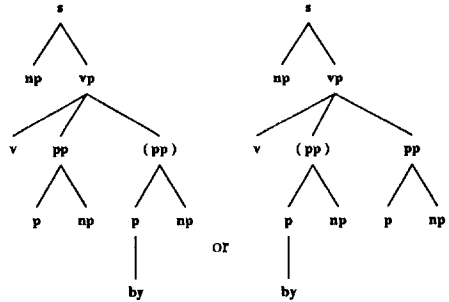
Thus, if LE_i stood for *DITRANS1* and say we consider x to be the node *NP* direct object, the trees described in LE_i and LE_o are:



Before illustrating the passive rule, we need to introduce the so-called *CHANGE ARITY* relation introduced by Flickinger (1987). We say that $CE = CHANGE-ARITY(CI)$ if CI is the immediate superclass of CE distinct from *TRANSITIVE*. We can now state the passive rule:



and by equating the two *V* nodes and by collapsing the two *VP* nodes as before, we get:



From the tree description and constraint equations of the passive class, we will inherit information to place the two feature structures on the *VP* nodes on top *form:passive* and on the bottom *form:ppart*. Since these feature structures cannot be unified, the auxiliary verb "be" is required to adjoin on the *VP* node.

As in any similar enterprise, we have to provide means to handle exceptions. For example, we will have to provide such mechanisms to handle verbs that are exceptions to the use of *PASSIVE* or *DATIVE* rule. Like in (Flickinger, 1987) we allow overwriting and state explicitly that certain rules are not applicable in the entries of some lexical items. However, considerable more machinery would need to be added to capture semantic constraints on the application of such rules. At present, little work has been done to incorporate specification of semantic constraints in conjunction with TAG.

<i>PASSIVE</i>
input : LE_i
output : LE_o
$PASSIVE \in LE_o.CLASSES$
$CHANGE-ARITY(LE_i.CLASS) \in LE_o.CLASSES$
$LE_i.CLASS \in LE_o.CLASSES$

Suppose we let $LE_i.class$ to be *DITRANS1*. Thus from the definition $CHANGE-ARITY(DITRANS1)$ is *PP-IOBJ*. The tree description inherited from *PP-IOBJ* differs from that of *DITRANS1* only in that we do not postulate the presence of the node for *NP* direct object. Thus the tree description we arrive at is:

5 Conclusion

While a number of proposals (Flickinger, 1987; Pollard and Sag, 1987; Shieber, 1986) have been made for a hierarchical organization of a lexicon and grammar, in our approach the hierarchical lexicon, the syntactic and lexical rules additionally specify partial descriptions of trees of extended domain of locality which capture syntactic and semantic dependencies. The description of elementary trees has been obtained by collecting par-

tial description of trees and then realizing the least tree satisfying these constraints. The syntactic and lexical rules enable us to derive new entries from existing ones. Overwriting allows us to be sensitive to lexical idiosyncrasies.

As mentioned earlier, the linguistic examples given here were meant only to indicate the potential of our approach. In general, we envisage the use of a hierarchical lexicon, of syntactic and lexical rules for lexicalized tree-adjointing grammars that capture important linguistic generalizations and provide for a space efficient representation of the grammar. Equally important, such a scheme would facilitate the automation of the process of updating and maintaining the grammar, an urgent need felt during the development of a large lexicalized tree-adjointing grammar.

We are currently investigating the possibility of defining parsing strategies that take advantage of the type of hierarchical representation we proposed in this paper. Many other related topics will be explored in the future. A much more elaborate organization will be considered, which in turn may suggest the need for additional machinery. We will implement the inheritance machinery described above and the process of building trees from these descriptions. We would also like to consider the treatment of idioms and the integration of syntactic and semantic specifications in the context of LTAG.

References

Anne Abeillé, Kathleen M. Bishop, Sharon Cote, and Yves Schabes. 1990. A lexicalized tree adjoining grammar for English. Technical Report MS-CIS-90-24, Department of Computer and Information Science, University of Pennsylvania.

Anne Abeillé. 1988. Parsing french with tree adjoining grammar: some linguistic accounts. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING'88)*, Budapest, August.

T. Becker. 1990. Meta-rules on tree adjoining grammars. In *Proceedings of the 1st International Workshop on Tree Adjoining Grammars*, Dagstuhl Castle, FRG, August.

J. Bresnan and R. Kaplan. 1983. Lexical-functional grammar: A formal system for grammatical representation. In *The Mental Representation of Grammatical Relations*. M.I.T. Press, Cambridge, MA.

Daniel Paul Flickinger. 1987. *Lexical Rules in the Hierarchical Lexicon*. Ph.D. thesis, Stanford University.

G. Gazdar, E. Klein, G. K. Pullum, and I. A. Sag. 1985. *Generalized Phrase Structure Grammars*. Blackwell Publishing, Oxford. Also published by Harvard University Press, Cambridge, MA.

Benoît Habert. 1991. Using inheritance in object-oriented programming to combine syntactic rules and lexical idiosyncrasies. In *Proceedings of the second International Workshop on Parsing Technologies*, Caicuu, Mexico, February.

Aravind K. Joshi. 1987. Word-order variation in natural language generation. In *AAAI 87, Sixth National Conference on Artificial Intelligence*, pages 550-555, Seattle, Washington, July.

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. 1987. Unbounded dependencies and subadjacency in a tree adjoining grammar. In A. Manaster-Ramer, editor, *Mathematics of Language*. John Benjamins, Amsterdam.

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.

Carl Pollard and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics. Vol 1: Fundamentals*. CSLI.

Jim Rogers and K. Vijay-Shanker. 1992. Reasoning with descriptions of trees. In *30th Meeting of the Association for Computational Linguistics (ACL'92)*.

Yves Schabes, Anne Abeillé, and Aravind K. Joshi. 1988. Parsing strategies with 'lexicalized' grammars: Application to tree adjoining grammars. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING'88)*, Budapest, Hungary, August.

Yves Schabes. 1990. *Mathematical and Computational Aspects of Lexicalized Grammars*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, August. Available as technical report (MS-CIS-90-48, LINC LAB179) from the Department of Computer Science.

Stuart Shieber and Yves Schabes. 1990. Synchronous tree adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING'90)*, Helsinki.

Stuart M. Shieber. 1986. *An Introduction to Unification-Based Approaches to Grammar*. Center for the Study of Language and Information, Stanford, CA.

K. Vijay-Shanker. 1992. Using description of trees in a tree adjoining grammar. To appear in *Computational Linguistics*.