Synchronous Tree-Adjoining Grammars

Stuart M. Shieber
Computer Science Department
Harvard University
Cambridge, MA USA

Yves Schabes
Computer and Information Sciences Department
University of Pennsylvania
Philadelphia, PA USA

Abstract

The unique properties of tree-adjoining grammars (TAG) present a challenge for the application of TAGs beyond the limited confines of syntax, for instance, to the task of semantic interpretation or automatic translation of natural language. We present a variant of TAGs, called synchronous TAGs, which characterize correspondences between languages. The formalism's intended usage is to relate expressions of natural languages to their associated semantics represented in a logical form language, or to their translates in another natural language; in summary, we intend it to allow TAGs to be used beyond their role in syntax proper. We discuss the application of synchronous TAGs to concrete examples, mentioning primarily in passing some computational issues that arise in its interpretation.

1 Introduction

Tree-adjoining grammars (TAG) constitute a grammatical formalism with attractive properties for the strong characterization of the syntax of natural languages, that is, characterization of the analysis trees of the expressions in the language (Kroch and Joshi, 1985; Kroch, 1989).[1] Among these properties are that

- The domain of locality in TAGs is larger than for formalisms that augment context-free grammars (such as lexical-functional, or generalized or head-driven phrase-structure grammar), and
- The statements of dependencies and recursion possibilities in a tree are factored, the former following from primitive dependencies in elementary trees, the latter a consequence of an operation of adjunction of trees.

These unique properties of TAGs present a challenge for the application of TAGs beyond the limited confines of syntax, for instance, to the task of semantic interpretation or automatic translation of natural language. The standard methods of moving beyond syntax to interpretation make use in one way or another of the compositional structure of the analysis tree that is manifested in the tree's derivation. Any version of compositional semantics, or syntax-directed translation relies on such a methodology to some extent. However, in the case of TAGs, the compositional structure of the tree is not mirrored by its derivational structure, so that a method for constructing semantics based on the compositional syntactic structure will be inherently nonderivational, that is, construction of the semantics will be independent of the derivation of the tree, and therefore subsequent.

On the other hand, a method mirroring the derivational structure will not necessarily be compositional with respect to the derived structures of expressions. Although such a method would be quite different from the primarily compositional methods previously postulated, it may have advantages, given that certain aspects of language seem to be noncompositional. (See Section 4.)

In this paper, we present a variant of TAGs, called synchronous TAGs, which characterize correspondences between languages. The formalism's intended usage is to relate expressions of natural languages to their associated semantics represented in a logical form language, or to their translations in another natural language; in summary, we intend the formalism to allow TAGs to be used beyond their role in syntax proper. We also discuss its application to concrete examples, and mention some computational issues that arise in its interpretation.

2 Synchronous TAGs—An Informal Description

Language interpretation tasks can be thought of as associating a syntactic analysis of a sentence with some other structure—a logical form representation or an analysis of a target language sentence, perhaps. Synchronous TAGs are defined so as to make such associations explicit. The original language and its associated structures are both defined by grammars stated in a TAG formalism; the two TAGs are synchronous in the sense that adjunction and substitution operations are applied simultaneously to related nodes in pairs of trees, one for each language. For convenience, we will call the two languages source and target languages, although the formalism is not inherently directional.

As an example, consider the task of relating a fragment of English with a simple representation of its predicate-argument structure. A synchronous TAG for this purpose is given in Figure 1. Each element of the

[1] We assume familiarity throughout the paper with previous work on TAGs. See, for instance, the introduction by Joshi (1987).

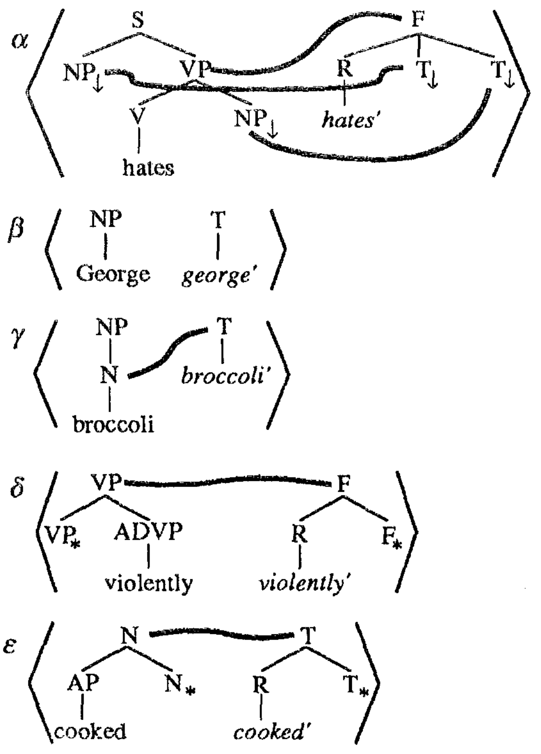


Figure 1: A sample synchronous TAG.

synchronous TAG is a pair consisting of two elementary trees, one from the source language (English) and one from the target (logical form [LF]). Nodes, one from each tree, may be linked;[2] such links are depicted graphically as thick lines. If we project the pairs onto their first or second components (ignoring the cross links), the projections are TAGs for an English fragment and an LF fragment, respectively. These grammars are themselves written in a particular variant of TAGs; the choice of this base formalism, as we will call it, is free. In the case at hand, we have chosen single-component lexicalized TAGs with adjunction and substitution (Schabes et al., 1988). Later examples are built on other bases.

The elementary operation in a synchronous TAG is supervenient on the elementary operations in the base formalism. A derivation step from a pair of trees $\langle \alpha_1, \alpha_2 \rangle$ proceeds as follows:

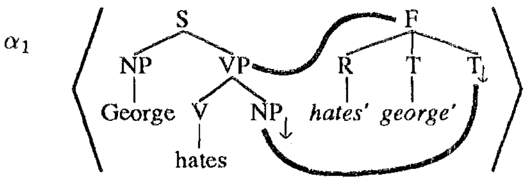
1. Nondeterministically choose a link in the pair connecting two nodes (say, n_1 in α_1 and n_2 in α_2).

2. Nondeterministically choose a pair of trees $\langle \beta_1, \beta_2 \rangle$ in the grammar.

3. Form the resultant pair $\langle \beta_1(\alpha_1, n_1), \beta_2(\alpha_2, n_2) \rangle$ where $\beta(\alpha, n)$ is the result of performing a primitive operation in the base formalism on α at node n using β (e.g., adjoining or substituting β into α at n).[3]

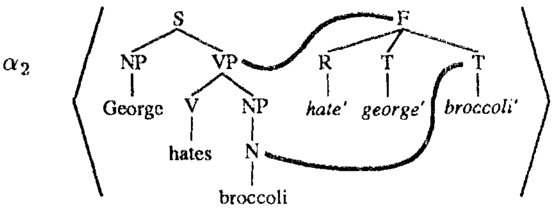
Synchronous TAG derivation then proceeds by choosing a pair of initial trees $\langle \alpha_1, \alpha_2 \rangle$ that is an element of the grammar, and repeatedly applying derivation steps as above.

As an example, suppose we start with the tree pair α in Figure 1.[4] We choose the link from the subject NP to T and the tree pair β to apply to its nodes. The resultant, by synchronous substitution, is the tree pair:



Note that the links from α are preserved in the resultant pair α_1 except for the chosen link, which has no counterpart in the result.

Using tree pair γ on the remaining link from NP to T in α_1 yields



This pairing manifests the correspondence between the sentence "George hates broccoli" and its logical form $hates'(george', broccoli')$ (as written in a more traditional notation). Here we see that the links in the operator trees (those in γ) are preserved in the resultant pair, accounting for the sole remaining link. The trees in γ are linked in this way so that other tree pairs can modify the N.

We can continue the derivation, using δ and ϵ to generate the pair given in Figure 2 thereby associating the meaning

$$violently'(hates'(george', cooked'(broccoli')))$$

with the sentence "George hates cooked broccoli violently."

A subtle issue arises with respect to link updating in the resultant pair if two links impinge on the same node. When one of the links is chosen and an adjunction performed at the node, the other link must appear in the resultant. The question as to whether that link should now end at the root or foot of the adjoined tree can be resolved in several ways. Although the choice of method does not affect any of the examples in this paper, we mention our current resolution of this problem here. If the remaining link is connected initially to the top of

[2] We will generalize the links later to allow sets of nodes from one tree to be linked to sets from the other.

[3] The definition allows for the operations performed on the first and second trees to differ, one being a substitution and the other an adjunction, for example.

[4] We use standard TAG notation, marking foot nodes in auxiliary trees with '*' and nodes where substitution is to occur with '↓'. The nonterminal names in the logical form grammar are mnemonic for Formula, Relation (or function) symbol, Term, and Quantifier.

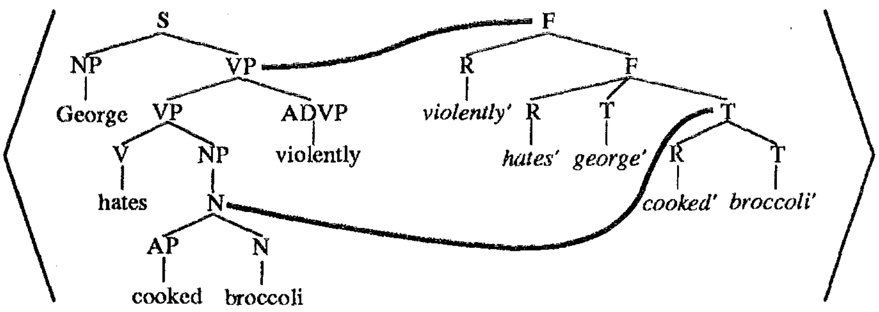


Figure 2: Derived tree pair for "George hates cooked broccoli violently."

the node serving as the adjunction site, it will connect to the top of the root node of the adjoined auxiliary tree after the adjunction has been performed; conversely, if it is connected initially to the bottom of the node, it will connect to the bottom of the foot node of the auxiliary tree. In all of the examples in this paper, the links may be thought of as connecting to the tops of nodes. The issue has important ramifications. For instance, the link updating process allows for different derivations of a single derivation in the source language to correspond to derivations of different derivations in the target language; that is, derivation order in synchronous TAGs is in this respect crucial, unlike in the base TAG formalisms. We rely on this property in the analysis of quantifier scope in Section 4.2.

3 Why Synchronous TAGs?

We turn to the question of why, in augmenting TAGs for the purposes of encoding semantic information, it is preferable to use the synchronous TAG method over more conventional methods, such as semantic rules involving logical operations (as in Montague grammar or generalized phrase-structure grammar) or complex-feature-structure encodings (as in unification-based or logic grammar formalisms).

First, the arguments for factoring recursion and dependencies as TAGs do for the syntax of natural language have their counterparts in the semantics. The structure of TAGs allows syntactic dependencies—agreement, subcategorization, and so forth—to be localized in the primitives of a grammar, the elementary trees. This is most dramatically evident in the case of long-distance dependencies, such as that between a wh-phrase and its associated gap. Similarly, using TAGs to construct logical forms allows the localization of semantic dependencies in the logical forms of natural language expressions, dependencies such as the signature requirements (argument type and arity) of function and relation symbols, and even the long-distance dependencies between a wh-quantifier and its associated bound variable. With other methods of semantics, these dependencies cannot be localized; the semantic aspects of filler-gap dependencies must be passed among the features of various nodes in a parse tree or otherwise distributed over the entire derivation.

Second, the use of the synchronous TAG augmentation allows an even more radical reduction in the role of features in a TAG grammar. Because of the extended domain of locality that TAGs possess, the role of features and unification is reduced from its role in context-free based systems. Only finite-valued features are needed, with the possible exception of a feature whose value encodes an expression's logical form. In removing the construction of logical forms from the duties delegated to features, we can maintain a strictly finite-valued—and therefore formally dispensable—feature system for TAGs.

As a side note, we mention a ramification of the synchronous TAG analysis concerning the claim of Kaplan and Zaenen (1989) that the paths over which long-distance dependencies operate (in the f-structure of lexical-functional grammatical theory) form a regular language. Vijay-Shanker and Joshi (1989) provide an argument that this claim follows from several assumptions concerning how a feature system for TAGs might be constrained. Vijay-Shanker (personal communication) has noted that by placing a simple assumption on the elementary trees in the logical form component of a synchronous TAG, the proof of this claim becomes immediate. Any TAG in which all foot nodes are immediate children of their associated root generates a tree path language that is regular.[5] Thus, a synchronous TAG (like the grammar presented in Figure 1) whose semantic component forms a TAG with this property necessarily obeys the regular language constraint on long-distance semantic dependencies.

4 Applications

To exemplify the formalism's utility, we briefly and informally describe its application to the semantics of idioms and quantifiers. A companion paper (Abeillé et al., 1990) uses a mapping between two TAGs for automatic translation between natural languages, and constitutes a further application of the synchronous TAG concept.

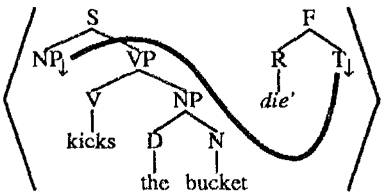
[5] This is a folk theorem whose straightforward proof is left as an exercise for the reader.

More expansive descriptions of these analyses will be forthcoming in joint work with Anne Abeillé (idioms and translation) and Anthony Kroch (quantifiers).

4.1 Idioms

Abeillé and Schabes (1989) note that lexicalized TAGs are an appropriate representation language for idiomatic constructions, as their expanded domain of locality can account for many syntactic properties of idioms. It seems natural to generalize beyond syntax, as they do, to the claim that lexicalized TAGs allow one to deal with semantic noncompositionality. Their argument to this claim is based on an intuition that semantics depends on the TAG derivation structure, an intuition that synchronous TAGs makes precise. For example, the idiomatic construction "kick the bucket" cashes out as the following tree pair, under its idiomatic interpretation:

α_3



whereas the literal usage of "kick" is associated with a tree pair similar to that of "hates" in Figure 1. Two derivations of the sentence "George kicked the bucket" are possible, each using a different one of these two elementary tree pairs, but both yielding identical derived constituency trees for the English. They will be associated, of course, with two different readings, corresponding to the idiomatic ($die'(george')$) and literal ($kick'(george', bucket')$) interpretations, respectively.

All of the arguments for the TAG analysis of idioms and light verb constructions can then be maintained in a formalism that allows for semantics for them as well. In particular,

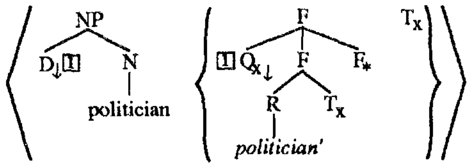
- Discontinuous syntactic constituents can be semantically localized.
- Nonstandard long-distance dependencies are statable without resort to reanalysis.
- Both frozen and flexible idioms can be easily characterized.

4.2 Quantifiers

In order to characterize quantifier scoping possibilities, we use a synchronous TAG whose base formalism is multi-component TAGs (Joshi, 1987), in which the primitive operation is incorporation (by multiple substitutions and adjunctions) of a set of elementary trees at once. In synchronous multi-component TAGs, the links between trees connect, in general, a set of nodes in one tree with a set in another. In particular, an NP will be linked both to a formula in the semantics (the quantifier's scope) and a term (the position bound by the quantifier). We will begin a derivation with just such a pair of elementary trees, depicted as α_1 in Figure 3.

To distinguish two separate links from a single link among several nodes, we use a coindexing—rather than graphical—notation for links. Thus, the subject NP node on the left is linked with both the F and first T node on the right, as indicated by the boxed index 1. The interpretation of such "hyper-links" is that when a pair is chosen to operate at the link, it must have sets of the correct sizes as its left and right component (1 and 2 in the case at hand) and the sets are simultaneously used at the various nodes as in a multi-component TAG. For instance, a quantifiable noun will be paired with a set of two trees:[6]

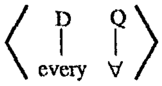
β_1



Applying the latter multi-component tree pair β_1 to the initial tree pair α_1, we derive the next stage in the derivation α_2. We have highlighted the link being operated on at this and later steps by using thick lines for the index boxes of the selected link.

The determiner can be introduced with the simple pair

β_2



leading to the derivation step α_3. Completing the derivation using analogous elementary tree pairs, we might generate the final tree pair α_4 of Figure 3. This final pairing associates the meaning

$$\exists y : vegetable'(y).\forall x : politician'(x).hates'(x, y)$$

with the sentence "Every politician hates some vegetable." It should be clear that in a structure such as this with multiple NPs, the order of substitution of NPs determines the relative scope of the quantifiers, although it has no effect whatsoever on the syntactic structure. Developing this line of reasoning has led to several detailed predictions of this analysis of quantifier scope, which is beyond this paper's purview. In summary, however, the analysis is slightly more restrictive than that of Hobbs and Shieber (1987), making predictions regarding the scope of topicalized or wh-moved constituents, relative scope of embedded quantifiers, and possibly even syntactic structure of complex NPs.

5 Using Synchronous TAGs

The synchronous TAG formalism is inherently nondirectional. Derivation is not defined in terms of constructing

[6] The subscript x on certain nodes is the value of a feature on the nodes corresponding to the variable bound by the quantifier. The technique of using metavariables to encode object variables is familiar from the logic and unification-based grammar literatures. Variable renaming with respect to these variables proceeds as usual.

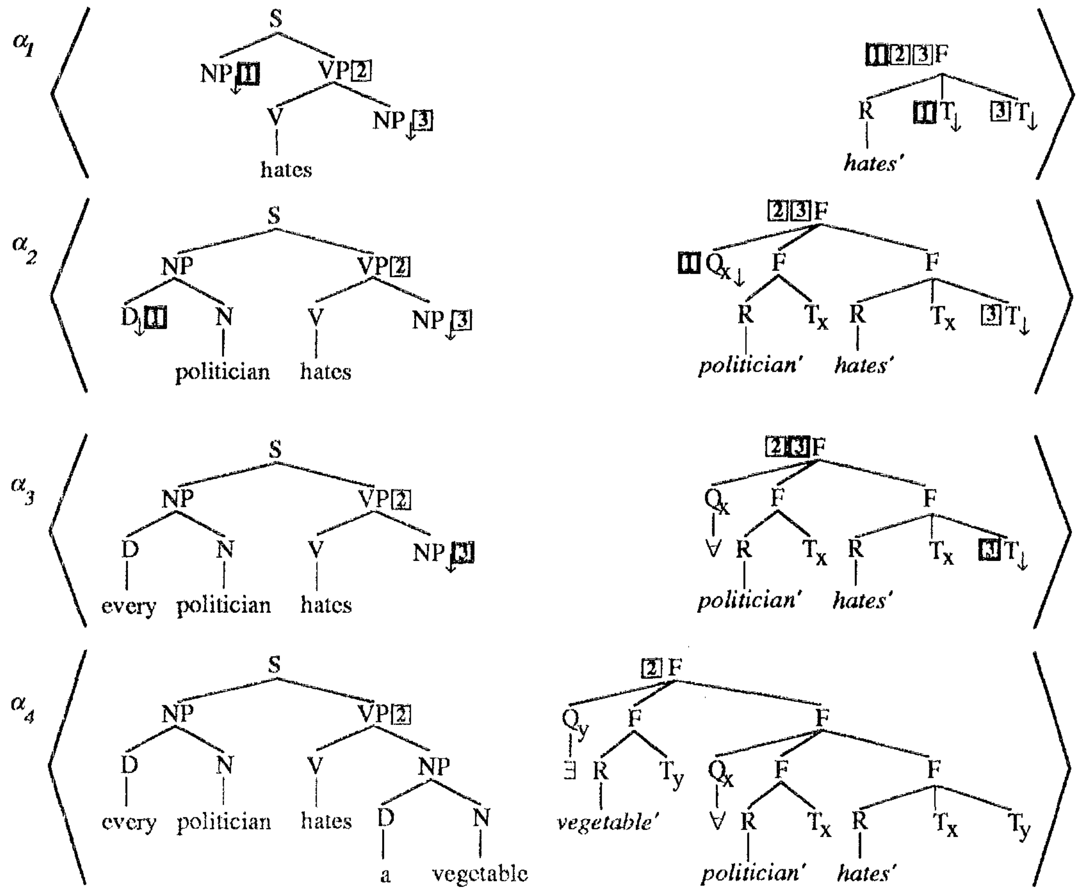


Figure 3: Sample synchronous TAG derivation steps for "Every politician hates a vegetable."

a target expression from a source or vice versa. Thus, it can be used to characterize both of these mappings. Furthermore, the existence of a parsing algorithm for the base formalism of a synchronous TAG is a sufficient condition for interpreting a synchronous TAG grammar. Schabes and Joshi (1988) and Vijay-Shanker and Joshi (1985) provide parsing algorithms for TAGs that could serve to parse the base formalism of a synchronous TAG.

Given such an algorithm, semantic interpretation can be performed by parsing the sentence according to the source grammar; the pairings then determine a derivation in the target language for the logical form. Generation from a logical form proceeds by the converse process of parsing the logical form expression thereby determining the derivation for the natural language sentence. Machine translation proceeds along similar lines by mapping two TAGs directly (Abeillé et al., 1990).

In previous work, one of us noted that generation according to an augmented context-free grammar can be made more efficient by requiring the grammar to be *semantically monotonic* (Shieber, 1988); the derived semantics for an expression must include, in an appropriate sense, the semantic material of all its subconstituents. It is interesting to note that synchronous TAGs are inherently semantically monotonic. Furthermore, it is reasonable to require that the semantic component of a synchronous TAG be *lexicalized* (in the sense of Schabes et al. (1988)), allowing for more efficient parsing according to the semantic grammar and, consequently, more efficient generation. In the case of augmented context-free grammars, the semantic monotonicity requirement precludes "lexicalization" of the semantics. It is not possible to require nontrivial semantics to be associated with each lexical item. In summary, just as lexicalization of the syntactic grammar aids parsing (Schabes and Joshi, 1990), so lexicalization of the semantic grammar aids generation.

The description of parsing and generation above may seem to imply that these processes cannot be performed incrementally, that is, an entire source derivation must be recovered before the corresponding target derivation can be computed. The issue deserves clarification.

In the case where the synchronous TAG is order-independent (that is, the order of derivation in one TAG does not effect the result in the other, as when no two links share an endpoint) there is a one-to-one mapping between the source and target derivation. When partial source derivations are recognized by the parser, the corresponding partial target derivation (for example semantic interpretation) can be incrementally computed: as the input is read from left to right, interpretations of the partial target derivations corresponding to partial source derivations can be combined in one step to build a larger partial target derivation.

When the synchronous TAG is order-sensitive, however, there may be a many-to-many correspondence between source derivations and target derivations. This is the case, for instance, in a grammar in which alternative quantifier scopings may be generated for a single sentence. In this case, it is unclear what should even be meant by incremental computation. For instance, midway in parsing a sentence, at a point at which a single quantified NP has been analyzed, the incremental interpretation could not possibly represent all possible scopings that that quantifier might end up taking, as it is not known what the quantifier might be required to scope with respect to. At the point in the parse where the scoping decision can be made, it is not clear whether an incrementality requirement would mean that the variant scopings must all be explicitly generated at that point, or only implicitly generable.

With respect to synchronous TAGs, these considerations are reflected in choice of parsing algorithm. Efficiency of parsing necessitates that only one canonical derivation (say leftmost or rightmost) need to be computed; all other derivations yield the same object. Standard parsing algorithms for both TAGs and CFGs rely on this optimization. If incrementality requires that we generate explicit representations of all possible interpretations (i.e., target derivations) of the string seen so far, then this optimization cannot be used, and parsing will be highly inefficient. If the representation can be left implicit, the optimization can be maintained, but retrieval of explicit representations will be combinatorially more complex.

6 Conclusion

The use of tree-adjoining grammars for natural-language-processing tasks requires the ability to move beyond a characterization of syntactic structure. Synchronous TAGs provide a simple mechanism that can be used to graft such an ability onto a base TAG formalism.

Acknowledgements

This research was partially funded by ARO grant DAAG29-84-K-0061, DARPA grant N00014-85-K0018, and NSF grant MCS-82-19196 at the University of Pennsylvania. We are indebted to Aravind Joshi for his support of this research and to Anne Abeillé and Anthony Kroch for their collaboration in the genesis of these ideas and their comments on earlier versions. K. Vijay-Shanker and Marilyn Walker also provided valuable comments. All remaining errors are the authors' responsibility alone.

Bibliography

Anne Abeillé and Yves Schabes. 1989. Parsing idioms in tree adjoining grammars. In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, England.

Anne Abeillé, Yves Schabes, and Aravind K. Joshi. 1990. Using lexicalized tree adjoining grammars for machine translation. To appear in the 13^{th} International Conference on Computational Linguistics (COLING'90).

Jerry Hobbs and Stuart M. Shieber. 1987. An algorithm for generating quantifier scopings. *Computational Linguistics*, 13(1-2):47–63.

Aravind K. Joshi and K. Vijay-Shanker. 1989. Unbounded dependencies in tags and lfg: functional uncertainty is a corolary in tags. In *Proceedings of the 27^{th} Meeting of the Association for Computational Linguistics*, Vancouver.

Aravind K. Joshi. 1987. An introduction to tree adjoining grammars. In A. Manaster-Ramer, editor, *Mathematics of Language*. John Benjamins, Amsterdam.

Ron Kaplan and Annie Zaenen. 1989. Long-distance dependencies as a case of functional uncertainty. In M. Baltin and A. Kroch, editors, *Alternative Conceptions of Phrase Structure*. University of Chicago Press.

Anthony Kroch and Aravind K. Joshi. 1985. The 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.

Yves Schabes and Aravind K. Joshi. 1988. An Earley-type parsing algorithm for tree adjoining grammars. In *Proceedings of the 26^{th} Meeting of the Association for Computational Linguistics*, Buffalo, June.

Yves Schabes and Aravind K. Joshi. 1990. Parsing with lexicalized tree adjoining grammar. In Masaru Tomita, editor, *Current Issues in Parsing Technologies*. Kluwer Accademic Publishers.

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

Stuart M. Shieber. 1988. A uniform architecture for parsing and generation. In *Proceedings of the 12^{th} International Conference on Computational Linguistics*, Budapest, August.

K. Vijay-Shanker and Aravind K. Joshi. 1985. Some computational properties of tree adjoining grammars. In 23^{rd} *Meeting of the Association for Computational Linguistics*, pages 82–93, Chicago, Illinois, July.