EXPLORATIONS OF A DOMAIN OF LOCALITY: Lexicalized Tree-Adjoining Grammar (LTAG)

Aravind K. Joshi
Department of Computer and Information Science, and
Institute for Research in Cognitive Science
University of Pennsylvania
joshi@linc.cis.upenn.edu

Abstract

Each grammar formalism specifies a domain of locality, i.e., a domain over which various dependencies (syntactic and semantic) can be specified. It turns about that the various properties of a formalism (syntactic, semantic, computational, and even psycholinguistic), follow, to a large extent, from the initial specification of the domain of locality. In this paper, we will briefly explore the extended domain of locality provided by the Lexicalized Tree-Adjoining Grammar (LTAG) in the context of some linguistic, computational, and psycholinguistic properties. This extended domain is achieved by specifying the elementary objects as structured objects (trees or directed acyclic graphs) instead of strings and two universal combining operations. Using lexicalized elementary structured objects it is possible to study directly many aspects of strong generative capacity which are more relevant to the linguistic descriptions.

1 Introduction

Tree-adjoining grammar (TAG) is a formal tree rewriting system. TAG and Lexicalized Tree-Adjoining Grammar (LTAG) have been extensively studied both with respect to their formal properties and to their linguistic relevance. TAG and LTAG are formally equivalent, however, from the linguistic perspective LTAG is the system we will be concerned with in this paper. We will often use these terms TAG and LTAG interchangeably.

The motivations for the study of LTAG are both linguistic and formal. The elementary objects manipulated by LTAG are structured objects (trees or directed acyclic graphs) and not strings. Using structured objects as the elementary objects of the formal system, it is possible to construct formalisms whose properties relate directly to the study of strong generative capacity (i.e., structural descriptions), which is more relevant to the linguistic descriptions than the weak generative capacity (sets of strings).

Each grammar formalism specifies a domain of locality, i.e., a domain over which various dependencies (syntactic and semantic) can be specified. It turns out that the various properties of a formalism (syntactic, semantic, computational, and even psycholinguistic) follow, to a large extent, from the initial specification of the domain of locality.

1.1 Domain of locality of CFGs

In a context-free grammar (CFG) the domain of locality is the one level tree corresponding to a rule in a CFG (Fig. 1). It is easily seen that the arguments of a predicate (for example, the two arguments of *likes*) are not in the same local domain. The two arguments are distributed over the two rules (two domains of locality)– $S \rightarrow NP\ VP$ and $VP \rightarrow V\ NP$. They can be brought together by introducing a rule $S \rightarrow NP\ V\ VP$. However, then the structure provided by the VP node is lost. We should also note here that not every rule (domain) in the CFG in (Fig. 1) is lexicalized. The three rules on the right are lexicalized, i.e., they have a lexical anchor. The rules on the left are not lexicalized. The second and the third rules on the left are almost lexicalized, in the sense that they each have a preterminal category (V in the second rule and ADV in the third rule), i.e., by replacing V by *likes* and ADV by *passionately* these two rules will become lexicalized. However, the first rule on the left ($S \rightarrow NP\ VP$) cannot be lexicalized. Can a CFG be lexicalized, i.e., given a CFG, G, can we construct another CFG, G', such that every rule in G' is lexicalized and $T(G)$, the set of (sentential) trees

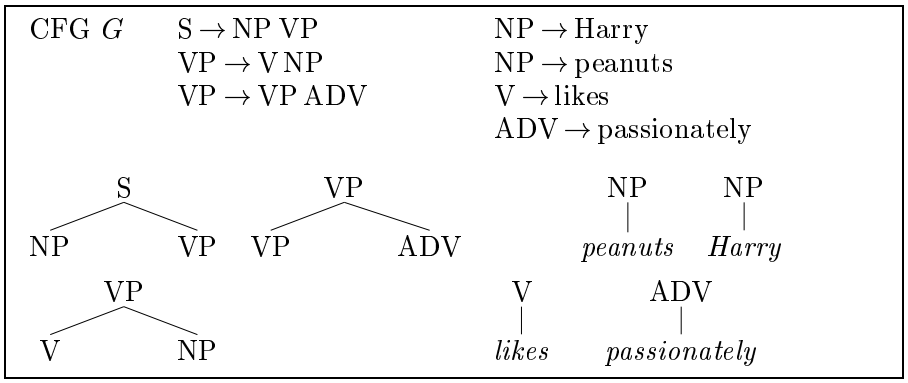

Figure 1: Domain of locality of a context-free grammar

(i.e., the tree language of G) is the same as the tree language $T(G')$ of G'? It can be shown that this is not the case (Joshi and Schabes (1997)). Of course, if we require that only the string languages of G and G' be the same (i.e., they are weakly equivalent) then any CFG can be lexicalized. This follows from the fact that any CFG can be put in the Greibach normal form where each rule is of the form $A \rightarrow w\ B1\ B2\ ...\ Bn$ where w is a lexical item and the $B's$ are nonterminals. The lexicalization we are interested in requires the tree languages (i.e., the set of structural descriptions) be the same, i.e., we are interested in the 'strong' lexicalization. To summarize, a CFG cannot be strongly lexicalized by a CFG. This follows from the fact that the domain of locality of CFG is a one level tree corresponding to a rule in the grammar. Note that there are two issues we are concerned with here– lexicalization of each elementary domain and the encapsulation of the arguments of the lexical anchor in the elementary domain of locality. The second issue is independent of the first issue. From the mathematical point of view the first issue, i.e., the lexicalization of the elementary domains of locality is the crucial one. We can obtain strong lexicalization without satisfying the requirement specified in the second issue (encapsulation of the arguments of the lexical anchor). Of course, from the linguistic point of view the second issue is very crucial. What this means is that among all possible strong lexicalizations we should choose only those that meet the requirements of the second issue. For our discussions in this paper we will assume that we always make such a choice.

1.2 Lexicalization of CFGs

Now we can ask the following question. Can we strongly lexicalize a CFG by a grammar with a larger domain of locality? Fig. 2 and Fig. 3 show a tree substitution grammar where the elementary objects (building blocks) are the three trees in Fig. 3 and the combining operation is the tree substitution operation shown in Fig. 2. Note that each tree in the tree substitution grammar (TSG), G' is lexicalized, i.e., it has a lexical anchor. It is easily seen that G' indeed strongly lexicalizes G. However, TSG's fail to strongly lexicalize CFG's in general. We show this by an example. Consider the CFG, G, in Fig. 4 and a proposed TSG, G'. It is easily seen that although G and G' are weakly equivalent they are not strongly equivalent. In G', suppose we start with the tree α_1 then by repeated substitutions of trees in G' (a node marked with a vertical arrow denotes a substitution site) we can grow the right side of α_1 as much as we want but we cannot grow the left side. Similarly for α_2 we can grow the left side as much as we want but not the right side. However, trees in G can grow on both sides. Hence, the TSG, G', cannot strongly lexicalize the CFG, G (Joshi and Schabes (1997)).

We now introduce a new operation called 'adjoining' as shown in Fig. 5. Adjoining involves splicing (inserting) one tree into another. More specifically, a tree β as shown in Fig. 5 is inserted (adjoined) into the tree α at the node X resulting in the tree γ. The tree β, called an auxiliary tree, has a special form. The root node is labeled with a nonterminal, say X and on the frontier there is also a node labeled X called

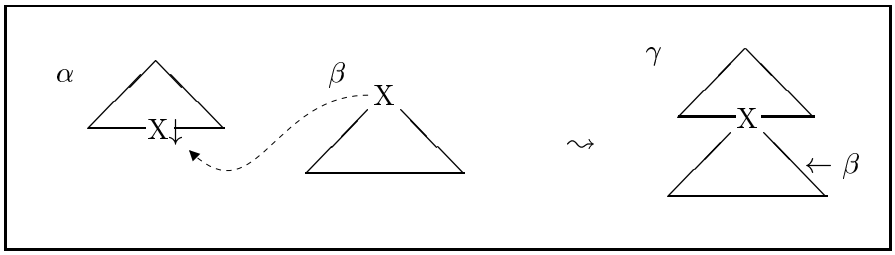

Figure 2: Substitution

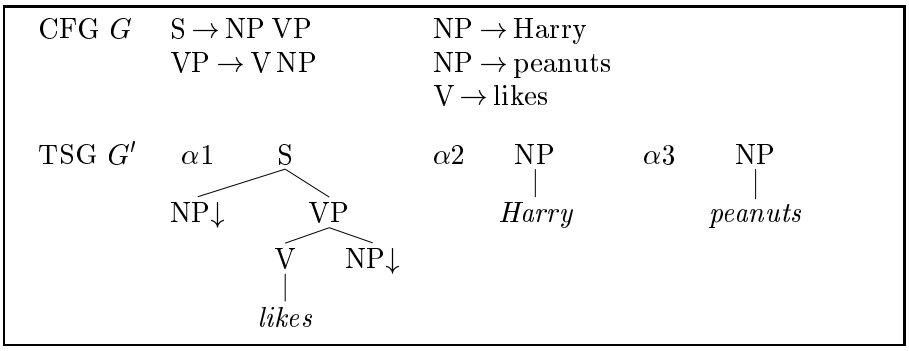

Figure 3: Tree substitution grammar

the foot node (marked with *). There could be other nodes (terminal or nonterminal) nodes on the frontier of β, the nonterminal nodes will be marked as substitution sites (with a vertical arrow). Thus if there is another occurrence of X (other than the foot node marked with *) on the frontier of β it will be marked with the vertical arrow and that will be a substitution site. Given this specification, adjoining β to α at the node X in α is uniquely defined. Adjoining can also be seen as a pair of substitutions as follows: The subtree at X in α is detached, β is substituted at X and the detached subtree is then substituted at the foot node of β. A tree substitution grammar when augmented with the adjoining operation is called a tree-adjoining grammar (lexicalized tree-adjoining grammar because each elementary tree is lexically anchored). In short, LTAG consists of a finite set of elementary trees, each lexicalized with at least one lexical anchor. The elementary trees are either initial or auxiliary trees. Auxiliary trees have been defined already. Initial trees are those for which all nonterminal nodes on the frontier are substitution nodes. It can be shown that any CFG can be strongly lexicalized by an LTAG (Joshi and Schabes (1997)).

In Fig. 6 we show a TSG, G', augmented by the operation of adjoining, which strongly lexicalizes the CFG, G. Note that the LTAG looks the same as the TSG considered in Fig. 4. However, now trees α_1 and α_2 are auxiliary trees (marked with *) that can participate in adjoining. Since adjoining can insert a tree in the interior of another tree it is possible to grow both sides of the tree α_1 and tree α_2, which was not possible earlier with substitution alone. In summary, we have shown that by increasing the domain of locality we have achieved the following: (1) lexicalized each elementary domain, (2) introduced an operation of adjoining, which would not be possible without the increased domain of locality (note that with one level trees as elementary domains adjoining becomes the same as substitution since there are no interior nodes to be operated upon), and (3) achieved strong lexicalization of CFGs.

1.3 Lexicalized tree-adjoining grammars

Rather than giving formal definitions for LTAG and derivations in LTAG we will give a sim-

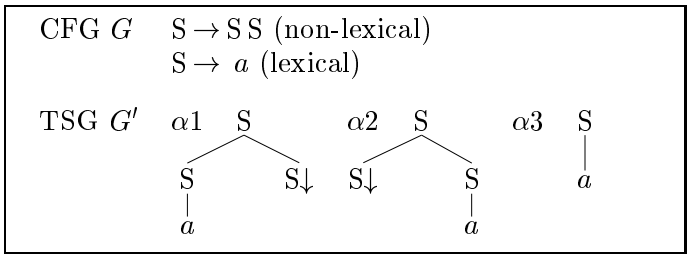

Figure 4: A tree substitution grammar

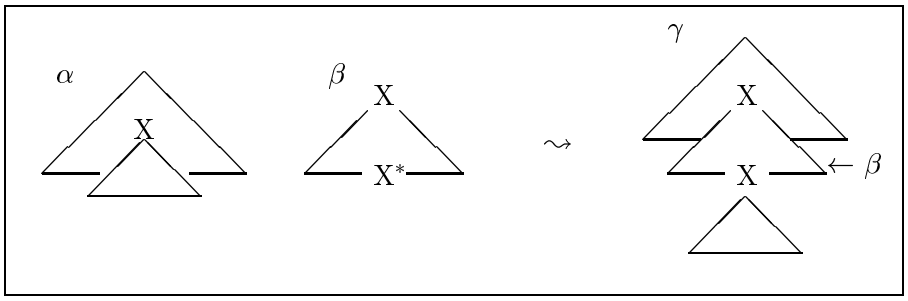

Figure 5: Adjoining

ple example to illustrate some key aspects of LTAG. We show some elementary trees of a toy LTAG grammar of English. Fig. 7 shows two elementary trees for a verb such as *likes*. The tree α_1 is anchored on *likes* and encapsulates the two arguments of the verb. The tree α_2 corresponds to the object extraction construction. Since we need to encapsulate all the arguments of the verb in each elementary tree for *likes*, for the object extraction construction, for example, we need to make the elementary tree associated with *likes* large enough so that the extracted argument is in the same elementary domain. Thus, in principle, for each 'minimal' construction in which *likes* can appear (for example, subject extraction, topicalization, subject relative, object relative, passive, etc.) there will be an elementary tree associated with that construction. By 'minimal' we mean when all recursion has been factored away. This factoring of recursion away from the domain over which the dependencies have to be specified is a crucial aspect of LTAGs as they are used in linguistic descriptions. This factoring allows all dependencies to be localized in the elementary domains. In this sense, there will, therefore, be no long distance dependencies as such. They will all be local and will become long distance on account of the composition operations, especially adjoining.

Fig. 8 shows some additional trees. Trees α_3, α_4, and α_5 are initial trees and trees β_1 and β_2 are auxiliary trees with foot nodes marked with *. A derivation using the trees in Fig. 7 and Fig. 8 is shown in Fig. 9. The trees for *who* and *Harry* are substituted in the tree for *likes* at the respective NP nodes, the tree for *Bill* is substituted in the tree for *think* at the NP node, the tree for *does* is adjoined to the root node of the tree for *think* tree (adjoining at the root node is a special case of adjoining), and finally the derived auxiliary tree (after adjoining β_2 to β_1) is adjoined to the indicated interior S node of the tree α_2. This derivation results in the derived tree for *who does Bill think Harry likes* as shown in Fig. 10. Note that the dependency between *who* and the complement NP in α_2 (local to that tree) has been stretched in the derived tree in Fig. 10. This tree is the conventional tree associated with the sentence.

However, in LTAG there is also a derivation tree, the tree that records the history of composition of the elementary trees associated with the lexical items in the sentence. This deriva-

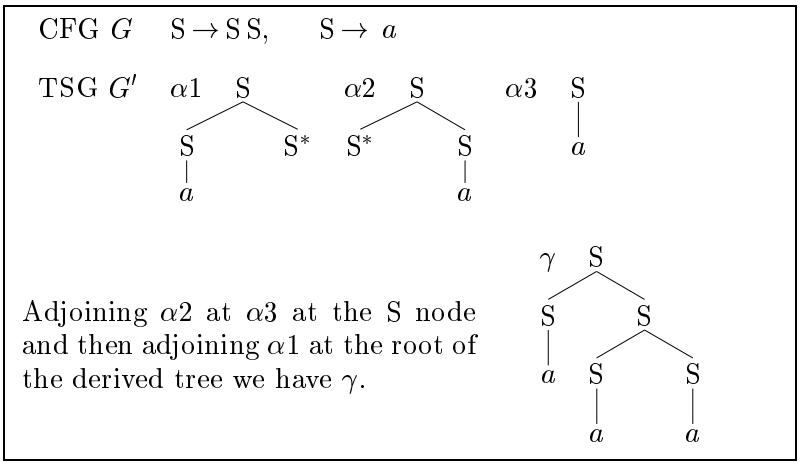

Figure 6: Adjoining arises out of lexicalization

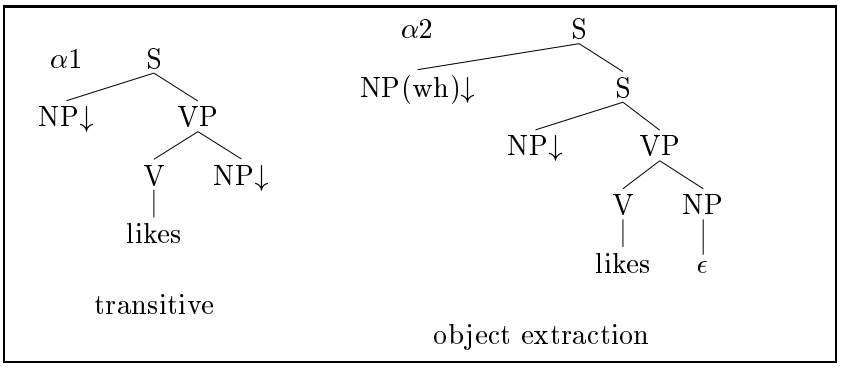

Figure 7: LTAG: Elementary trees for *likes*

tion tree is shown in Fig. 11. The nodes of the tree are labeled by the tree labels such as α_2 together with the lexical anchor.[1] The derivation tree is the crucial derivation structure for LTAG. We can obviously build the derived tree from the derivation tree. For semantic computation the derivation tree (and not the derived tree) is the crucial object. Compositional semantics is defined on the derivation tree. The idea is that for each elementary tree there is a semantic representation associated with it and these representations are composed using the derivation tree. Since the semantic representation for each elementary tree is directly associated with the tree there is no need to reproduce necessarily the internal hierarchy in the elementary tree in the semantic representation(Joshi and Vijay-Shanker (1999)). This allows the so-called 'flat' semantic representation as well as helps in dealing with some non-compositional aspects as in the case of rigid and flexible idioms.

2 Some important properties of LTAG

The two key properties of LTAG are (1) extended domain of locality (EDL) (for example, as compared to CFG), which allows (2) factoring recursion from the domain of dependencies (FRD), thus making all dependencies local. All other properties of LTAG (mathematical, linguistic, and even psycholinguistic) follow from EDL and FRD. TAGs (LTAGs) belong to the so-called class of mildly context-

[1] The derivation trees of LTAG have a close relationship to the dependency trees, although there are some crucial differences; however, the semantic dependencies are the same.

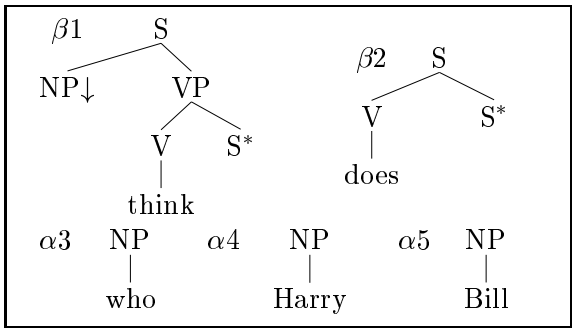

Figure 8: LTAG: Sample elementary trees

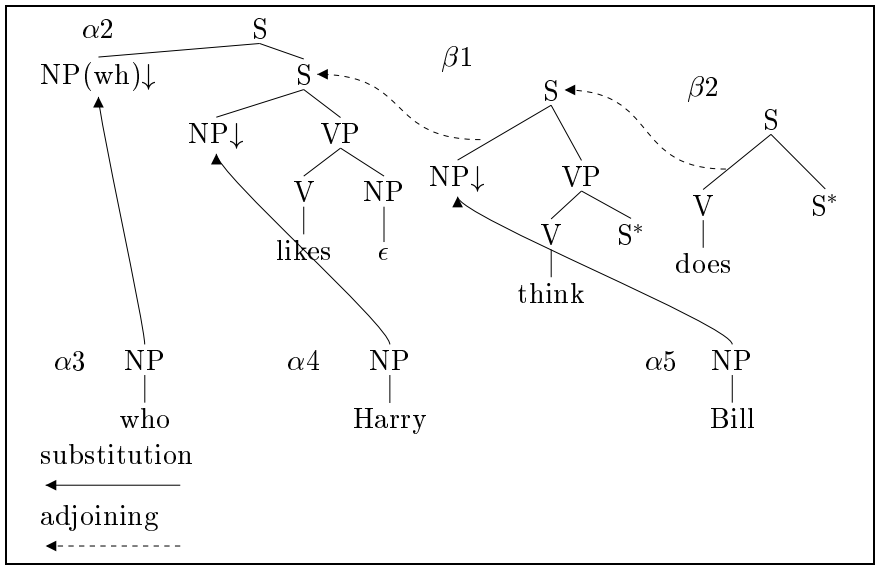

Figure 9: LTAG derivation for *who does Bill think Harry likes*

sensitive grammars (Joshi (1985)). Context-free languages (CFL) are properly contained in the class of languages of LTAG, which in turn are properly contained in the class of context-sensitive languages. There is a machine characterization of TAG (LTAG), called embedded pushdown automaton (EPDA) (Vijay-Shanker (1987)), i.e., for every TAG language there is an EPDA which corresponds to this (and only this) language and the language accepted by any EPDA is a TAG language. EPDAs have been used to model some psycholinguistic phenomena, for example, processing crossed dependencies and nested dependencies have been discussed in (Joshi (1990)). With respect to formal properties, the class of TAG languages enjoys all the important properties of CFLs, including polynomial parsing (with complexity $O(n^6)$).

Large scale wide coverage grammars have been built using LTAG, the XTAG system (LTAG grammar and lexicon for English and a parser) being the largest so far (for further details see (The XTAG Research Group (2000)). In the XTAG system, each node in each LTAG tree is decorated with two feature structures (top and bottom feature structures), in contrast to the CFG based feature structure grammars. This is necessary because adjoining can augment a tree internally, while in a CFG based grammar a tree can be augmented only at the frontier. It is possible to define adjoining and substitution (as it is done in the XTAG system) in terms of appropriate unifications of the top and bottom feature structures. Because of

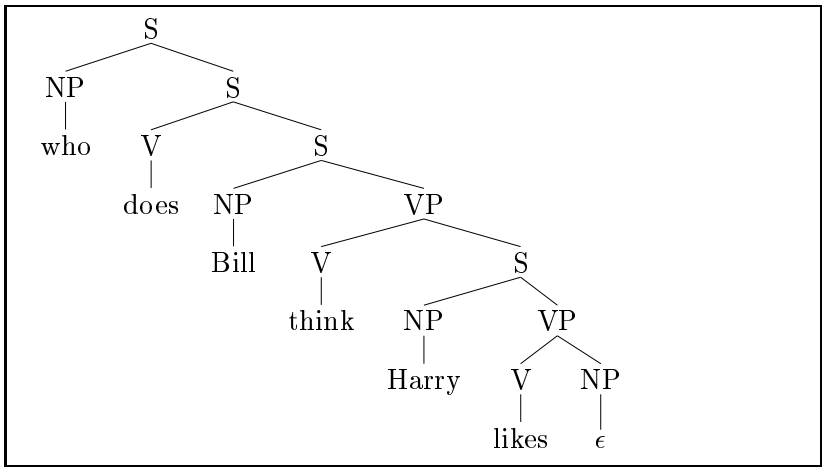

Figure 10: LTAG derived tree for *who does Bill think Harry likes*

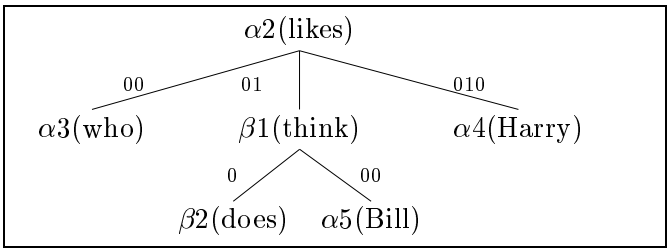

Figure 11: LTAG derivation tree

FRD (factoring recursion from the domain of dependencies), there is no recursion in the feature structures. Therefore, in principle, feature structures can be eliminated. However, they are crucial for linguistic descriptions. Constraints on substitution and adjoining are modeled via these feature structures (Vijay-Shanker (1987)). This method of manipulating feature structures is a direct consequence of the extended domain of locality of LTAG.

3 Implications for statistical computations

It is certainly possible to construct stochastic LTAGs in the same way as stochastic CFGs (Schabes (1992); Resnik (1992)). However, in the context of LTAG there is an interesting way in which statistical information can be used. In an LTAG, for each lexical item, a finite set of elementary trees is associated with that item. The combining operations–substitution and adjoining are universal, i.e., that they are language independent. The lexicon (i.e., the lexical items together with associated elementary trees) is the grammar, in a sense. The elementary trees associated with a lexical item can be regarded as more complex part-of-speech as compared to the standard part-of speech, such as Noun, Verb, Adjective, Preposition, etc. We will call these complex part-of-speech (i.e., the elementary trees) as supertags. Suppose we have information about the unigram, bigram, trigram etc. statistics for the supertags (estimated, for example, from a corpus parsed with an LTAG parser). These statistics combine the lexical statistics as well as the statistics of the constructions in which the lexical items appear. Now we can consider the possibility of using the statistical techniques for disambiguation of standard part-of-speech, which have been very successful. We call this supertagging, which is illustrated in Fig. 12 and Fig. 13.

In Fig. 12 for the example sentence we show the supertags (elementary trees) associated

	α1	α2	α3		α4	α5
	β2	α6	α7		β4	α8
β1	α9	α10	α11	β3	α12	α13
	.	.	.		.	.
	.	.	.		.	.
	.	.	.		.	.
the	purchase	price	includes	two	ancillary	companies

Figure 12: Supertagging: 1

with each word, as specified by LTAG. Some words have only one supertag while others have more than one.[2] Supertagging now applies the disambiguation techniques for standard part-of-speech to the supertagged sequence, in the current experiments using trigrams. The job of the supertagger is to select the correct supertag for each word, where by a correct supertag for a word we mean the supertag that corresponds to that word in the correct parse of the sentence. In Fig. 13 we have shown the correct supertag for each word in a box. The results reported in (Srinivas and Joshi (1998)) show that with a training corpus of 1 million words and a test corpus of 47,000 words the supertagging accuracy is 92%. Note that the base line performance is only 75%, which is due to the very high degree of local ambiguity. Thus the improvement from 75% to 92% is quite significant. Once a sentence is correctly supertagged we have an 'almost' parse, almost in the sense that we have identified for each word the construction in which it appears; however, we have not yet done the attachments–by substitution and adjoining. For some applications, as in some IR systems, the almost parse is quite adequate. The supertagger can also be used to speed up LTAG parsing by first supertagging a sentence before sending it to the parser.

From a theoretical perspective the significance of supertagging is as follows. The supertags are complex descriptions of the primitives (lexical anchors). This is somewhat contrary to the conventional wisdom that the descriptions of the primitives are to be kept simple and complex descriptions are to be made from simple descriptions. In LTAG we pack all the information associated with a lexical anchor into the elementary trees associated with the anchor. Making the descriptions of the primitives more complex increases local ambiguity as there are more descriptions for each primitive. However, these richer descriptions locally constrain each other quite efficiently. A jigsaw puzzle is a good analogy—the richer the description of each piece of the puzzle the better, i.e., it is easier to determine where it goes or which other pieces go with it. This provides an explanation of the results of supertagging experiments mentioned above.

4 An alternate perspective on adjoining

In adjoining we insert an auxiliary tree, say with root and foot nodes labeled with X in a tree at a node with label X. In Fig. 14 and Fig. 15 we present an alternate perspective on adjoining. The tree α which receives adjunction at X can be viewed as made up of two trees, the supertree at X and the subtree at X as shown in Fig.14. Now, instead of the auxiliary tree β adjoined to the tree α at X we can view this composition as a wrapping operation–the supertree of α and the subtree of α are wrapped around the auxiliary tree β as shown in Fig. 15. The resulting tree γ is the same as before. Wrapping of the supertree at the root node of β is like adjoining at the root (a special case of adjoining) and the wrapping of the subtree at the foot note of β is like substitution. Hence, this wrapping operation can be described in terms of substitution and adjoining. This is clearly seen in the linguistic example in Fig. 16 and Fig. 17. The

[2] In the current LTAG for English, as in the XTAG system, there can be as many as 30 supertags per word on the average, a number much larger than the average number of standard part-of-speech per word, which is about 1.5.

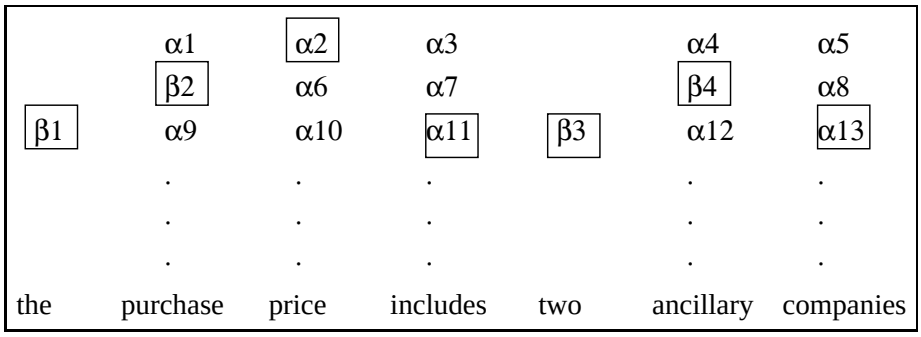

Figure 13: Supertagging: 2

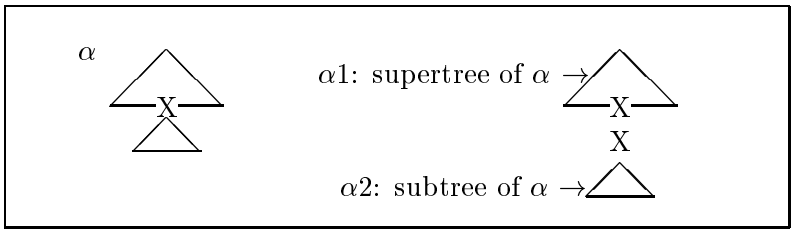

Figure 14: Adjoining as Wrapping 1

auxiliary tree β can be adjoined to the tree α at the indicated node in α as shown in Fig. 16. Alternatively, we can view this composition as adjoining the supertree α_1 (the *wh* tree) at the root node of β and substitution of the subtree α_2 (the *likes* tree) at the foot node of β as shown in Fig. 17. The two ways of composing α and β are semantically coherent.

The wrapping perspective can be formalized in terms of the so-called multi-component LTAGs (MC-LTAGs). They are called multi-component because the elementary objects can be sets of trees, in our examples, we have two components (in which α was split). When we deal with multi-components we can violate the locality of the composition very quickly because the different components may be 'attached' (by adjoining or substitution) to different nodes of a tree and these nodes may or not be part of an elementary tree depending on whether the tree receiving the multi-component attachments is an elementary or a derived tree. We obtain what are known as tree-local MC-LTAGs if we put the constraint that the tree receiving multi-component attachments must be an elementary tree. It is known that tree-local MC-TAGs are weakly equivalent to LTAGs, however they can give rise to structural descriptions not obtainable by LTAGs, i.e., they are more powerful than LTAG in the sense of strong generative capacity (Weir (1988)). Thus the alternate perspective leads to greater strong generative capacity without increasing the weak generative capacity.

We will now present an example illustrating the use of this alternate perspective in characterizing the scope ambiguity in *some student hates every course* as shown in Fig. 18, Fig. 19, and Fig. 20 (Kallmeyer and Joshi (1999)). In Fig. 18, we show a tree-local MC-LTAG for our example. The trees for *hates*, *student*, and *course* are standard LTAG trees. The trees for *some* and *every* are multi-component trees. For example, the tree α_1 for *some* has two components, α_{11} and α_{12}, one of the components α_{11} is a degenerate tree in this special case. The multi-component tree α_1 is lexically anchored by *some*. Similarly, for the tree α_2 for *every*. The main idea here is that the α_{12} component corresponds to the contribution of *some* to the predicate-argument structure of the tree for *hates* and the α_{11} component contributes to the scope structure. Similarly for the two components of α_2.

Fig. 19 shows the derivation. The main point to note here is that the two components of α_1

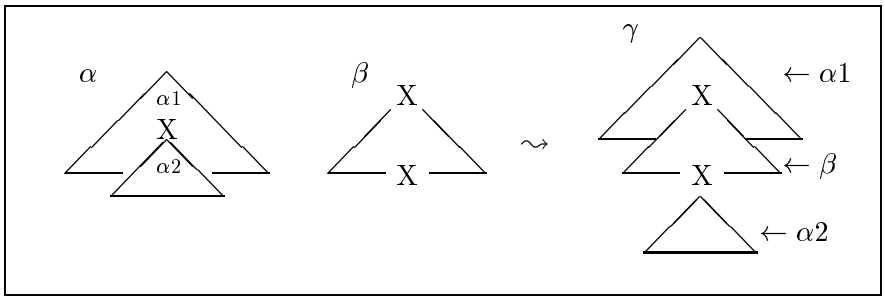

Figure 15: Adjoining as Wrapping 2

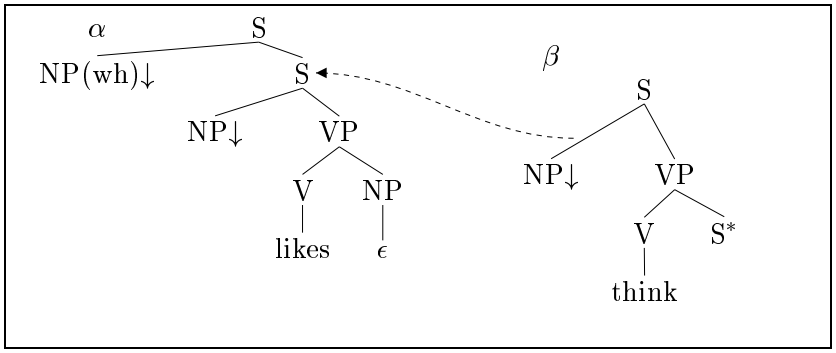

Figure 16: Wrapping as substitution and adjunction 1

are attached (by substitution or adjoining) to α_3 at the appropriate nodes simultaneously. This composition is tree local as $\alpha 3$ is an elementary tree. Similarly for the tree α_2. In this example, the two top components α_{11} and α_{21} are attached to the same node (the root node) of α_3.[3] This may give the impression that the composition is non-local because once α_1 is attached to α_3 we have a derived tree to which α_2 is attached. However, the two components, α_{11} and α_{21} are degenerate and it can be shown that in this case the composition of α_2 with α_3 (after α_1 has been composed with α_3) is still effectively tree-local (Kallmeyer and Joshi (1999)).

It is clear in this example that α_2 could have been attached to α_3 first and then α_1 attached to α_3. Fig. 20 shows the derivation tree for the derivation in Fig. 19. The numbers on the edges of the tree refer to the addresses of attachments, which are shown here for convenience. Note that both α_{11} and α_{21}, the scope information carrying components, are attached to $\alpha 3$ at the same node. Thus they could be attached in any order (strictly speaking, α_1 and α_2 could be attached to α_3 in any order). Hence α_{11} will outscope α_{21} if α_{21} is attached first and then α_{11} and vice versa. The scope ambiguity is thus directly reflected in the derivation tree for *some student hates every course*. This is in contrast to all other approaches (which are essentially CFG based) where the scope ambiguity is represented at another level of representation. It is possible to represent in LTAG the scope ambiguity at the level of the derivation tree itself because of the alternate perspective on adjoining, which in turn is due to the extended domain of locality discussed in this paper.

More recently, similar ideas have been explored in the context of other linguistic phenomena such as scrambling and clitic climbing, both with respect to linguistic coverage and certain psycholinguistic implications. A particularly interesting result is that all word order variations up to two levels of embedding (i.e.,

[3] In general in a multi-component LTAG multiple adjunctions to the same node are not allowed as this violates the tree-locality and also takes the system beyond LTAG. However, we are dealing here with a special case.

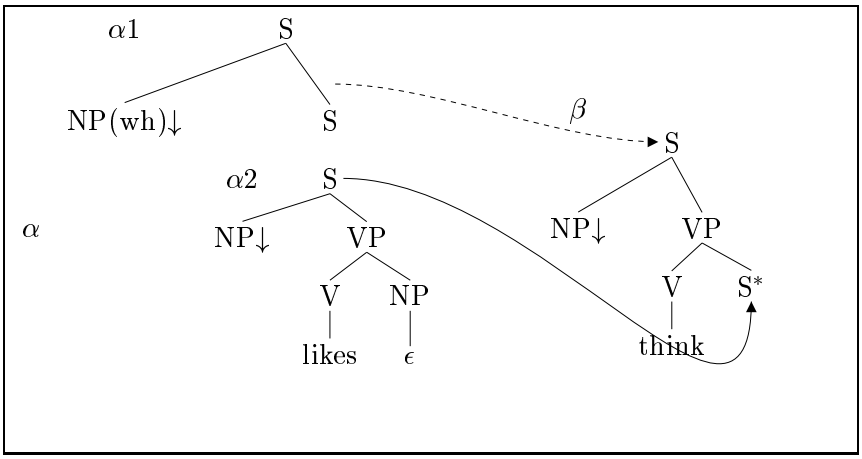

Figure 17: Wrapping as substitution and adjunction 2

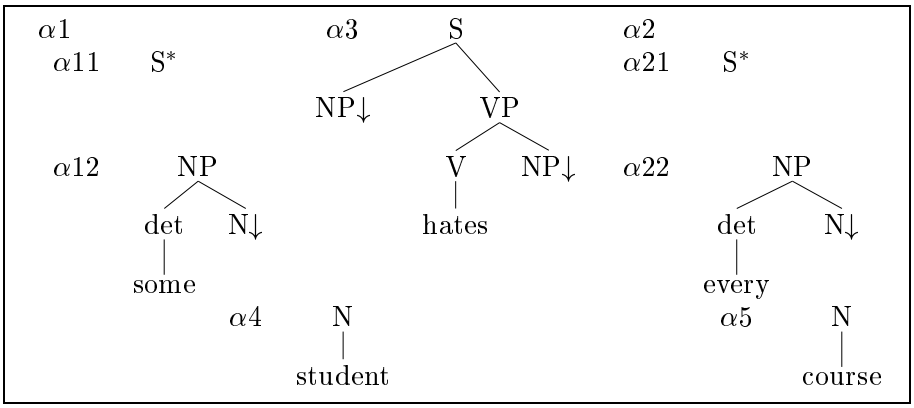

Figure 18: Scope ambiguity: An example

three clauses in all) can be correctly described by tree-local MC-LTAGs, correctly in the sense of providing the appropriate structural descriptions. Beyond two levels of embedding not all patterns of word order variation will be correctly described (Joshi, Becker, and Rambow (2000), Kulick (2000)).

5 Some related systems

Categorial Grammars: Categories assigned to lexical items in a categorial grammar framework do encapsulate the arguments of the lexical anchor. It is of interest to see how the basic ideas in LTAG could be incorporated in a categorial framework. The idea is not to translate LTAG into a categorial grammar but rather construct a categorial system with properties similar to LTAG. This is achieved by associating partial proof trees with lexical items. These partial proof trees are obtained by starting with the type assignment for a lexical item as specified by a categorial grammar and then 'unfolding' it by using certain categorial inference rules such as function application. This unfolding is done until the slots for the arguments of the lexical anchor are exposed. We thus have a finite collection of partial proof trees (lexicalized, of course) which serve as the building blocks of our system (analogous to the finite set of elementary trees in LTAG). These proof trees are then combined by universal categorial inference rules in terms of cuts. Informally speaking, the proof trees are hooked up by linking the conclusion nodes of one tree to the assumption nodes of another tree. Further discussion of such systems and their relationship to LTAG can be found

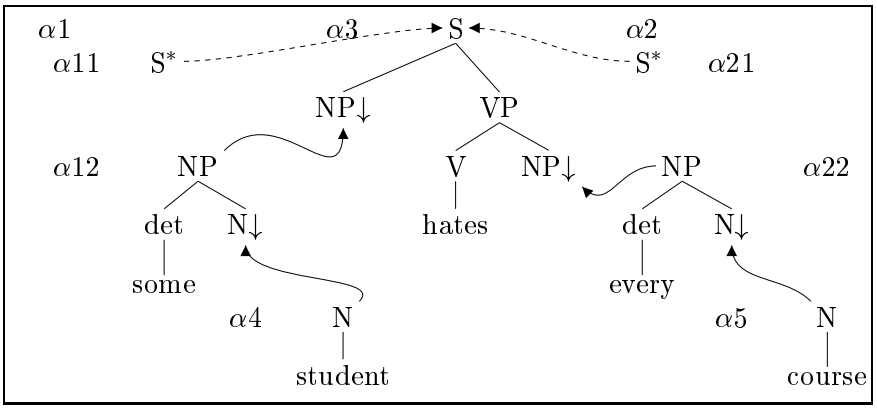

Figure 19: Derivation with scope information

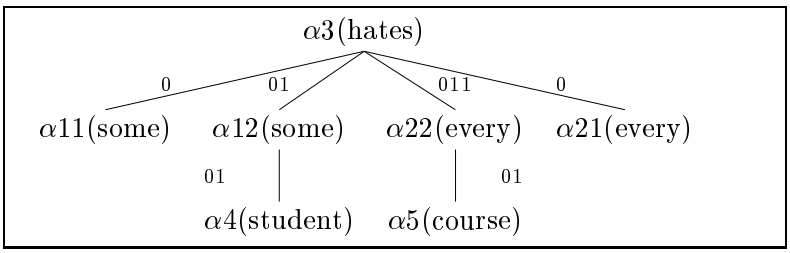

Figure 20: Derivation tree with scope underspecification

in (Joshi and Kulick (1997); Joshi, Kulick, and Kurtonina (1999)).

From sentence structure to discourse structure: Using the insights from LTAG a structural and presuppositional account of local discourse structure has been presented in (Webber, Joshi, Knott, and Stone (1999)). The idea is to start the analysis of discourse in the same way as one starts the analysis of a clause, looking at how its syntax and semantics project from the lexicon. This is complementary to the issue of discourse pragmatics –how these small syntactic units of discourse are used in achieving communicative intentions –and to the other discourse processes that provide additional organizational overlays on these units. A key feature of this approach is that semantic discourse relations are associated with syntactic structures and anaphoric links, and that the properties of the two are (not surprisingly) different. Together they allow more complex semantics to be conveyed through simpler structures.

Phrase structure composition and syntactic dependencies: Frank (2000) presents a comprehensive perspective on phrase structure composition and syntactic dependencies in a TAG-based grammatical architecture and compares it to the minimalist framework of Chomsky, showing that a number of stipulative and problematic aspects of that theory can be eliminated.

6 Summary

The domain of locality of a grammar formalism, i.e., the domain over which various dependencies can be specified determines to a large extent the syntactic, semantic, computational, and even psycholinguistic properties of the formalism. From this perspective the extended domain of Lexicalized Tree-Adjoining Grammars (LTAG) – extended as compared to the domain of locality of CFGs, for example – was explored. This extended domain is achieved by specifying the elementary objects of the grammar as structured objects instead of strings, together with two universal combining operations (substitution and adjoining). This perspective allows us to study directly many aspects of strong genera-

tive capacity which are more useful for linguistic description.

7 References

Abeille, A. and Candito, M. (2000). FTAG: A lexicalized tree-adjoining grammar for French. In A. Abeille and O. Rambow (editors), *Tree-Adjoining Grammar*, CSLI, Stanford.

Frank R. (2000). *Phrase Structure Composition and Syntactic Dependencies*. MIT Press, Cambridge, to appear in 2000.

Joshi A. K. (1985). Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Parsing*, 206-250, Cambridge University Press.

Joshi A. K. (1990). Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes*, 5, 1-27.

Joshi A. K., Becker T., and Rambow O. (2000). Complexity of scrambling: A new twist to the competence-performance distinction. In A. Abeille and O. Rambow, editors, *Tree-Adjoining Grammars*, CSLI, Stanford.

Joshi A. K. and Kulick S. (1997). Partial proof trees as building blocks for a categorial grammar. *Linguistics and Philosophy*, 20, 637-667.

Joshi A. K., Kulick, S., and Kurtonina N. (1999). Semantic composition for partial proof trees. In *Proc. of the Twelfth Amsterdam Colloquium*, 169-174, University of Amsterdam, Amsterdam.

Joshi A. K. and Schabes, Y. (1997). Tree-Adjoining Grammars. In G. Rosenberg and A. Salomaa, editors, *Handbook of Formal Languages*, 69-123. Springer, Berlin.

Joshi A. K. and Vijay-Shanker K. (1999). Compositional semantics with lexicalized tree-adjoining grammar (LTAG): How much underspecification is necessary? In H.C. Bunt and E.G.C. Thijsse, editors, *Proceedings of the Third International Workshop on Computational Semantics (IWCS-3)*, 131–145, Tilburg.

Kallmeyer L. and Joshi A. K. (1999). Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. In *Proc. of the Twelfth Amsterdam Colloquium*, 169-174, University of Amsterdam, Amsterdam.

Kulick S. (2000). *Constrained non-locality: Long-distance dependencies* in TAG. Ph.D. Dissertation, University of Pennsylvania, Philadelphia, PA, USA.

Resnik P. (1992). Probabilistic tree-adjoining grammars as a framework for statistical natural language processing. In *Proceedings of COLING '92*, 2, 418-424, Nantes.

Schabes Y. (1992). Stochastic lexicalized grammars. In *Proceedings of COLING '92*, 2, 426-432, Nantes.

Schabes Y. and Shieber S. M. (1994). An alternative conception of tree-adjoining derivation. *Computational Linguistics*, 20:1.

Srinivas B. and Joshi A. K. (1998). Supertagging: An approach to almost parsing. *Computational Linguistics*, *Computational Linguistics*, 22, 1-29.

Steedman M. (1996). *Surface Structure and Interpretation*. MIT Press, Cambridge, MA.

The XTAG Research Group (2000). A Lexicalized Tree-Adjoining Grammar for English. *Technical Report*, 98-18, Institute for Research in Cognitive Science, Philadelphia.

Vijay-Shanker K. (1987). *A Study of Tree-Adjoining Grammars*, Ph.D. Dissertation, University of Pennsylvania.

Webber B., Joshi A. K., Knott A., and Stone M. (1999). What are little texts made of? A structural and presuppositional account using lexicalized TAG. In *Proceedings of the International Workshop on Levels of Representation in Discourse (LORID '99)*, 151-156, Edinburgh.

Weir D. (1988). *Characterizing mildly context-sensitive grammar formalisms*. Ph.D. Dissertation, University of Pennsylvania, Philadelphia. PA, USA.