FG 2006:
The 11th conference on Formal Grammar
Malaga, Spain
July 29-30, 2006

Organizing Committee:

Paola Monachesi	Gerald Penn
Giorgio Satta	Shuly Wintner

CENTER FOR THE STUDY
OF LANGUAGE
AND INFORMATION

Contents

Simpler TAG Semantics through Synchronization

Rebecca Nesson and Stuart Shieber

Keywords Synchronous Tree-Adjoining Grammar, STAG semantics, quantifier scope, long-distance WH-movement, raising verbs, attitude verbs, adverbs, relative clauses, prepositional phrases.

Abstract

In recent years Laura Kallmeyer, Maribel Romero, and their collaborators have led research on TAG semantics through a series of papers refining a system of TAG semantics computation. Kallmeyer and Romero bring together the lessons of these attempts with a set of desirable properties that such a system should have. First, computation of the semantics of a sentence should rely only on the relationships expressed in the TAG derivation tree. Second, the generated semantics should compactly represent all valid interpretations of the input sentence, in particular with respect to quantifier scope. Third, the formalism should not, if possible, increase the expressivity of the TAG formalism. We revive the proposal of using synchronous TAG (STAG) to simultaneously generate syntactic and semantic representations for an input sentence. Although STAG meets the three requirements above, no serious attempt had previously been made to determine whether it can model the semantic constructions that have proved difficult for other approaches. In this paper we begin exploration of this question by proposing STAG analyses of many of the hard cases that have spurred the research in this area. We reframe the TAG semantics problem in the context of the STAG formalism and in the process present a simple, intuitive base for further exploration of TAG semantics. We provide analyses that demonstrate how STAG can handle quantifier scope, long-distance WH-movement, interaction of raising verbs and adverbs, attitude verbs and quantifiers, relative clauses, and quantifiers within prepositional phrases.

FG-2006.
Organizing Committee:, Paola Monachesi, Gerald Penn, Giorgio Satta, Shuly Wintner.

1.1 Introduction

In recent years Laura Kallmeyer, Maribel Romero, and their collaborators have led research on TAG semantics through a series of papers refining a system of TAG semantics computation using evolving techniques including enriched derivation tree structure (Kallmeyer, 2002a,b), flexible composition of feature-based TAG with a semantic representation associated with each elementary tree (Kallmeyer and Joshi, 2003, Joshi et al., 2003, Kallmeyer, 2003), semantic features in a more expressive extension of feature-based TAG (Gardent and Kallmeyer, 2003), and, most recently, semantic features on the derivation tree itself (Kallmeyer and Romero, 2004, Romero et al., 2004). Kallmeyer and Romero (2004) bring together the lessons of these attempts with a set of desirable properties that such a system should have. First, computation of the semantics of a sentence should rely only on the relationships expressed in the TAG derivation tree. Because TAG elementary trees represent minimal semantic units, the only information necessary for semantic computation should be the information encoded in the derivation tree: which elementary trees have combined and the address at which the combining operation took place. Second, the generated semantics should compactly represent all valid interpretations of the input sentence, in particular with respect to quantifier scope. Third, the formalism should not, if possible, increase the expressivity of the TAG formalism.

We revive the proposal of using synchronous TAG (STAG) to simultaneously generate syntactic and semantic representations for an input sentence (Shieber and Schabes, 1990). Although STAG meets the three requirements above, no serious attempt had previously been made to determine whether it can model the semantic constructions that have proved difficult for other approaches. In this paper we begin exploration of this question by proposing STAG analyses of many of the hard cases that have spurred the research in this area. We reframe the TAG semantics problem in the context of the STAG formalism and in the process present a simple, intuitive base for further exploration of TAG semantics.

After reviewing STAG in Section 1.2, we provide analyses in Sections 1.3.1 through 1.3.4 for sentences that exemplify several hard cases for TAG semantics that have been raised by Kallmeyer and others in recent papers: quantifier scope (as exemplified by sentences (1) and (5), presented below along with the desired semantic interpretations), long-distance WH-movement (2), interaction of raising verbs and adverbs, attitude verbs and quantifiers (3,4,5), relative clauses (6), and quantifiers within prepositional phrases (7) (Kallmeyer and Romero, 2004, Romero et al., 2004, Joshi et al., 2003, Kallmeyer, 2003, Kallmeyer and Joshi, 2003).[1]

(1) Everyone likes someone.
$every(x, person(x), some(z, person(z), like(x, z)))$
$some(z, person(z), every(x, person(x), like(x, z)))$

(2) Who does Bill think Paul said John likes?
$who(y, think(bill, say(paul, like(john, y))))$

(3) Bill thinks John apparently likes Mary.
$think(bill, apparently(like(john, mary)))$

(4) John sometimes likes everyone.
$every(x, person(x), sometimes(like(john, x)))$
$sometimes(every(x, person(x), like(john, x)))$

(5) Bill thinks everyone likes someone.
$think(bill, every(x, person(x), some(z, person(z), likes(x, z))))$
$think(bill, some(z, person(z), every(x, person(x), likes(x, z))))$

(6) A problem whose solution is difficult stumped Bill.
$$\begin{array}{l} a(x,\ \ and(\ problem(x), \\ \qquad\qquad the(y, and(solution(y), poss(x, y)), isDifficult(y))), \\ \qquad stumped(bill, x)) \end{array}$$

(7) Two politicians spy on someone from every city.
$$\begin{array}{l} two(x,\ \ politician(x), \\ \qquad\quad every(z,\ \ city(z), \\ \qquad\qquad\qquad some(y, person(y) \wedge from(z, y), \\ \qquad\qquad\qquad spyOn(x, y)))) \\ every(z,\ \ city(z), \\ \qquad\quad some(y,\ \ person(y) \wedge from(z, y), \\ \qquad\qquad\qquad two(x, politcian(x), spyOn(x, y)))) \\ two(x,\ \ politician(x), \\ \qquad\quad some(y,\ \ every(z, city(z), person(y) \wedge from(z, y)) \\ \qquad\qquad\qquad spyOn(x, y))) \\ some(y,\ \ every(z, city(z), person(y) \wedge from(z, y)) \\ \qquad\quad two(x, politician(x), spyOn(x, y))) \end{array}$$

1.2 Introduction to Synchronous TAG

A tree-adjoining grammar (TAG) consists of a set of elementary tree structures and two operations, substitution and adjunction, used

[1] We notate curried two-place relations $P(x)(y)$ as $P(y, x)$ for readability.

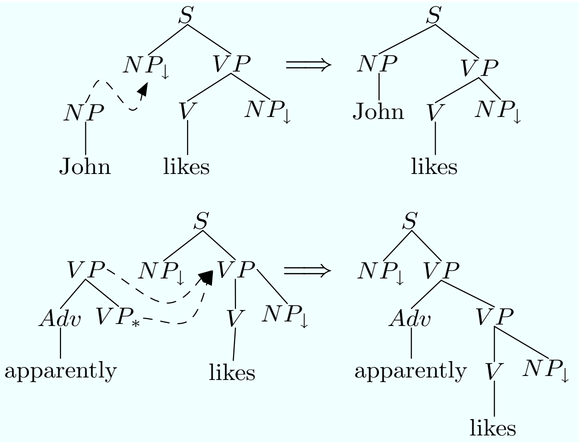

FIGURE 1 Example TAG substitution and adjuction operations.

to combine these structures. The elementary trees can be of arbitrary depth. Each internal node is labeled with a nonterminal symbol. Frontier nodes may be labeled with either terminal symbols or nonterminal symbols and one of the diacritics ↓ or $*$. Use of the diacritic ↓ on a frontier node indicates that it is a *substitution node*. The *substitution* operation occurs when an elementary tree rooted in the nonterminal symbol A is substituted for a substitution node labeled with the nonterminal symbol A. Auxiliary trees are elementary trees in which the root and a frontier node, called the *foot node* and distinguished by the diacritic $*$, are labeled with the same nonterminal. The *adjunction* operation involves splicing an auxiliary tree with root and designated foot node labeled with a nonterminal A at a node in an elementary tree also labeled with nonterminal A. Examples of the substitution and adjunction operations on sample elementary trees are shown in Figure 1.

Synchronous TAG (STAG) extends TAG by taking the elementary structures to be pairs of TAG trees with links between particular nodes in those trees. An STAG is a set of triples, $\langle t_L, t_R, \frown \rangle$ where t_L and t_R are elementary TAG trees and $\frown$ is a linking relation between nodes in t_L and nodes in t_R (Shieber, 1994, Shieber and Schabes, 1990). Derivation proceeds as in TAG except that all operations must be paired. That is, a tree can only be substituted or adjoined at a node if its pair is simultaneously substituted or adjoined at a linked node. We notate the links by using boxed indices $\boxed{i}$ marking linked nodes.

Figure 2 contains a sample English syntax/semantics grammar frag-

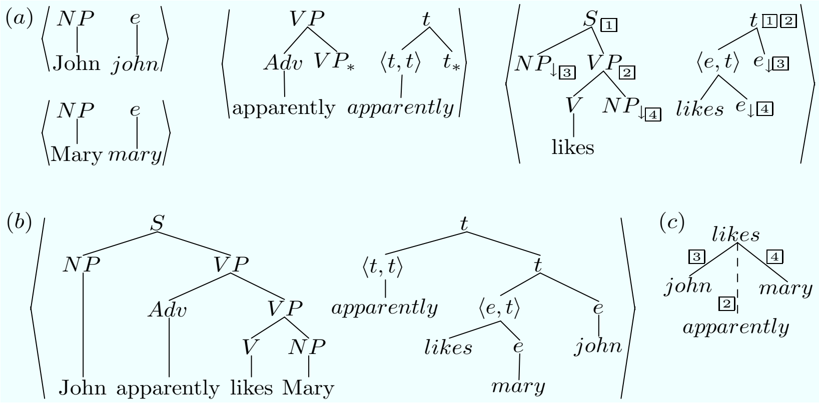

FIGURE 2 An English syntax/semantics STAG fragment (a), derived tree pair (b), and derivation tree (c) for the sentence "John apparently likes Mary."

ment that can be used to parse the sentence "John apparently likes Mary". The node labels we use in the semantics correspond to the semantic types of the phrases they dominate.[2] Variables such as x in the semantic tree in Figure 3 are taken to be bound in the obvious way, so that in multiple uses of the tree they can be presumed to be renamed apart.

Figure 2(c) shows the derivation tree for the sentence. Substitutions are notated with a solid line and adjunctions are notated with a dashed line. Note that each link in the derivation tree specifies a link number in the elementary tree pair. The links provide the location of the operations in the syntax tree and in the semantics tree. These operations must occur at linked nodes in the target elementary tree pair. In this case, the noun phrases $John$ and $Mary$ substitute into $likes$ at links $\boxed{3}$ and $\boxed{4}$ respectively. The word *apparently* adjoins at link $\boxed{2}$. The resulting semantic representation can be read off the derived tree by treating the leftmost child of a node as a functor and its siblings as its arguments. Our sample sentence thus results in the semantic representation $apparently(likes(john, mary))$.

[2] This representation is for the sake of readability. The labels could be replaced using any well-chosen finite set of nonterminal symbols.

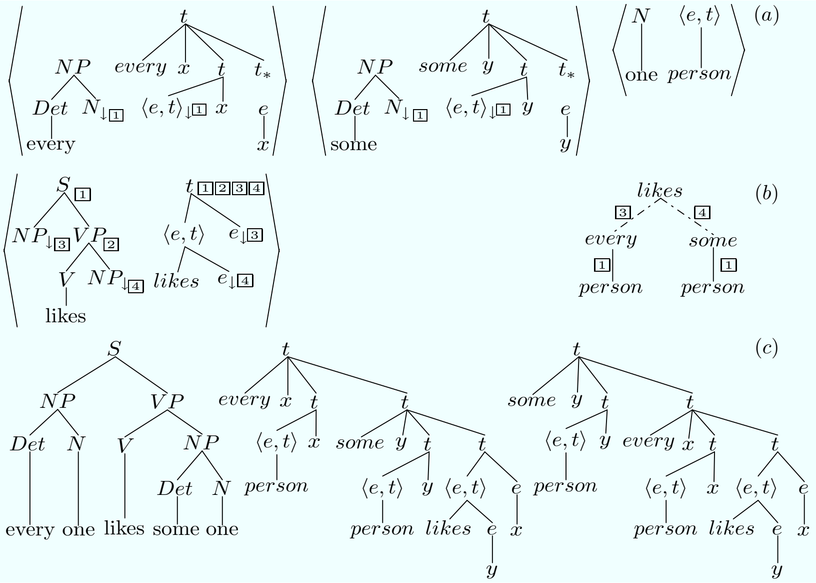

FIGURE 3 The elementary tree pairs (a), derivation tree (b), and derived syntactic and semantic trees (c) for the sentence "Everyone likes someone". Note that the derivation tree is a scope neutral representation: depending on whether *every* or *some* adjoins higher, different semantic derived trees and scope orderings are obtained.

1.3 STAG Analyses of the Phenomena

1.3.1 Quantifier Scope and Wh-Words

For sentence (1), we would like to generate a scope-neutral semantic representation that allows both the reading where *some* takes scope over *every* and the reading where *every* takes scope over *some*. We propose a solution in which a derivation tree with multiple adjunction nondeterministically determines multiple derived trees each manifesting explicit scope (Schabes and Shieber, 1993); the derivation tree *itself* is therefore the scope neutral representation.

The multi-component quantifier approach followed by Joshi et al. (2003) suggests a natural implementation of quantifiers in STAG.[3] In this approach the syntactic tree for quantifiers has two parts, one that

[3] The multi-component approach to quantifiers in STAG was first suggested by Shieber and Schabes (1990) under the rewriting definition of STAG derivation where the order of rewriting produced the scope ambiguity. Williford (1993) explored the use of multiple adjunction to achieve scope ambiguity.

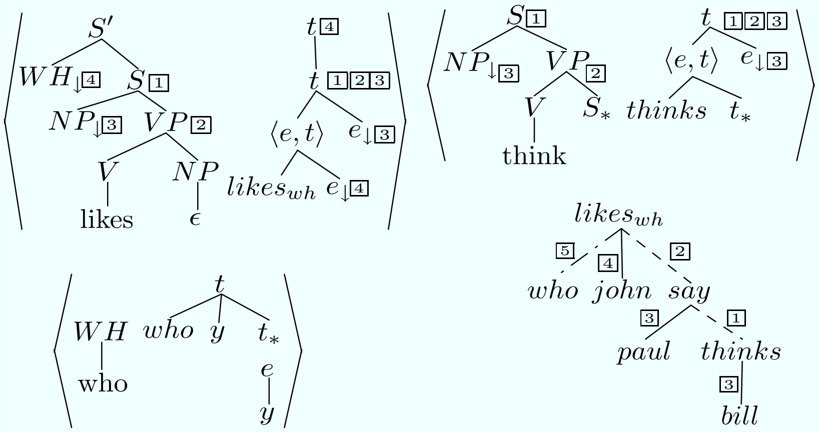

FIGURE 4 Selection of elementary trees and full derivation tree for the sentence "Who does Bill think Paul said John likes?".

corresponds to the scope of the quantifier and attaches at the point where the quantifier takes scope, and the other that contains the quantifier itself and its restriction and attaches where syntactically expected at a noun phrase. In their work, a single-node auxiliary tree is used for the scope part of the syntax in order to get the desired relationship between the quantifier and the quantified expression in features threaded through the derivation tree and hence in the semantics. Using STAG, we do not need the single-node auxiliary tree in the syntax because we can pair the usual syntactic representation for quantified NPs with a multi-component semantic representation that expresses the same idea (Figure 3). In order to use these quantifiers, we change the links in the elementary trees for verbs to allow a single link to indicate two positions in the semantics where a tree pair can adjoin, as shown in Figure 3.[4]

Given this representation of quantifiers we get the derivation tree shown in Figure 3 for sentence (1).[5] Note that the resulting derivation tree necessarily incorporates *multiple adjunction* (Schabes and Shieber, 1993), that is, multiple auxiliary trees are adjoined at the same node

[4] We have chosen here to add the three-way links in addition to the existing links in the tree for unquantified noun phrases such as proper nouns (though we suppress the two-way NP links in the figures for readability). Another possibility would be to remove the two-way links. In this case, all noun phrases would be "lifted" à la Montague. That is, even unquantified noun phrases would have a scope part, which could be a single-node auxiliary tree.

[5] We notate multi-component insertions that involve both a substitution and an adjunction with a combination dashed and dotted line.

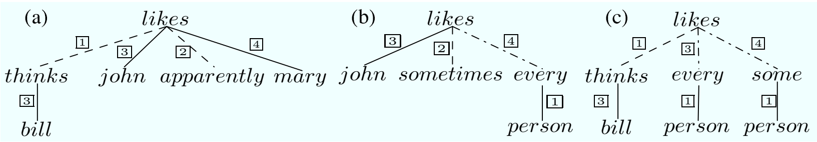

FIGURE 5 Derivation trees for (a) “Bill thinks John apparently likes Mary”, (b) “John sometimes likes everyone”, and (c) “Bill thinks everyone likes someone.”

in an auxiliary tree. In particular, the scope parts of both *every* and *some* attach at the root of the semantic tree of *likes*. Such cases of multiple adjunction induce ambiguity; the derivation tree represents multiple derived trees. In the case at hand, the derivation is ambiguous as to which quantifier scopes higher than the other. This ambiguity in the derivation tree thus models the set of valid scopings for the sentence. In essence, this method uses multiple adjunction to model scope-neutrality.

This same method can be used to obtain the correct scope relations for sentences with long-distance WH-movement such as sentence (2) using the multi-component elementary tree pair for *who* and the elementary tree pairs for *thinks* (the tree pair for *says* is similar) and *likes* in the WH context given in Figure 4. Kallmeyer and Romero (2004) highlight this case as difficult because in the usual syntactic analysis there is no link in the derivation tree between *who* and *thinks* or between *thinks* and *likes*, but in the desired semantics *who* takes scope over the *thinks* proposition and the *likes* proposition is an argument to *thinks*.

In our analysis, by contrast, the semantics follows quite naturally from the standard syntactic analysis of the structure of the *likes* elementary tree in the WH context and the elementary tree pair for *thinks* given in Figure 4. The derivation of this sentence is also given in Figure 4. Note that it is required by the structure of the trees that *who* take scope over *thinks*.

1.3.2 The Interaction Between Attitude Verbs, Raising Verbs, Adverbs and Quantifiers

The interaction between attitude verbs and raising verbs or adverbs as in sentences (3), (4), and (5) has been problematic for TAG semantics (Kallmeyer and Romero, 2004). A successful analysis must be flexible enough to produce the correct semantics for sentence (3) even though there is no link between *thinks* and *apparently* in the derivation tree. It must also be flexible enough to allow all scope orderings between

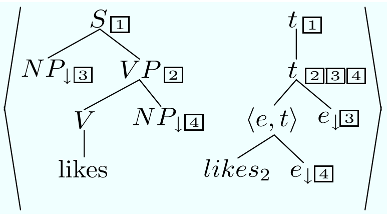
$S\boxed{1}$
$NP_{\downarrow}\boxed{3}$
$VP\boxed{2}$
V
$NP_{\downarrow}\boxed{4}$
likes
$t\boxed{1}$
$t\boxed{2}\boxed{3}\boxed{4}$
$\langle e,t\rangle$
$e_{\downarrow}\boxed{3}$
$likes_2$
$e_{\downarrow}\boxed{4}$

FIGURE 6 Modified tree for *likes* that enforces a restriction on quantifiers scoping outside of the finite clause.

VP modifiers and quantifiers as in sentence (4). In fact, given the elementary trees we have already presented and the ones for attitude verbs demonstrated by Figure 4, our analysis already allows for scope interactions among all these elements. Indeed, because the semantic components of attitude verbs, VP modifiers, and quantifiers all adjoin at the same node in the semantic tree of the verb, our analysis allows all scope orderings among them. This is clearly too permissive, because it allows quantifiers to scope out of the finite clause in which they appear and would allow a reading of sentence (3) in which *apparently* scopes over *thinks*. To prevent quantifiers from scoping out of the finite clause in which they appear, as in sentences (3) and (5), we can add an additional adjunction site to the semantic trees for verbs above the current root node. This is shown in Figure 6 in the $likes_2$ tree pair. The link configuration ensures that attitude verbs (adjoining at link $\boxed{1}$) will now scope higher than all VP modifiers (adjoining at $\boxed{2}$) and quantifiers (adjoining at links $\boxed{3}$ and $\boxed{4}$). VP modifiers and quantifiers will still be able to take all scope orderings relative to each other. Using the modified verb trees, STAG produces the correct semantics for sentences (3), (4), and (5) with the derivations given in Figure 5.

1.3.3 Relative Clauses

Relative clauses provide another putatively difficult case for TAG semantics because both the main verb and the relative clause need access to the variable introduced by the determiner as in sentence (6) (Kallmeyer, 2003). We overcome this difficulty and compute the desired semantics by introducing higher-order functions into the semantic trees using lambda-calculus notation. This modification allows us to maintain tree-locality. The syntactic analysis we use is similar to that of Kallmeyer (2003) in that it maintains the *Condition on Elementary Tree Minimality* (Frank, 1992) and uses the relative pronoun to introduce the relative clause. However, it treats

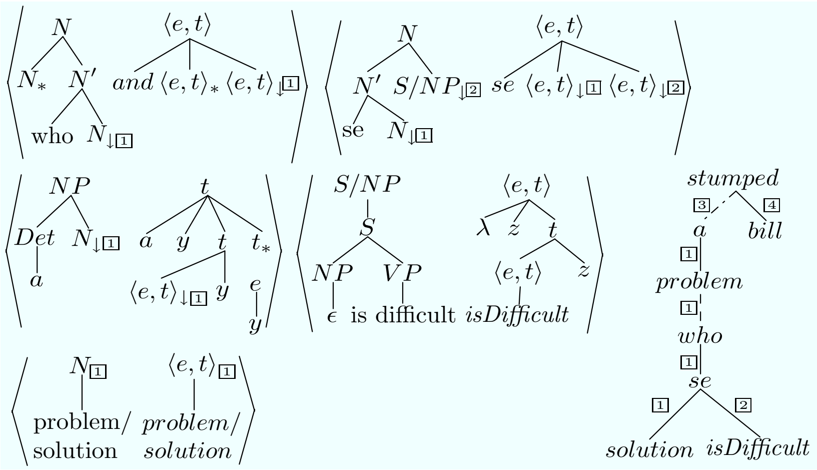

FIGURE 7 Key elementary trees and derivation for "A problem whose solution is difficult stumped Bill."

the relative pronoun as a noun modifier rather than a noun phrase modifier. We also posit the existence of "lifted" versions of the elementary trees for verbs in which their argument positions have been abstracted over. We use a higher-order conjunction *and* that relates two properties: $\lambda PQx.P(x) \wedge Q(x)$, and a higher-order *se* function that relates two properties and makes use of the higher-order conjunction: $\lambda PQx.the(y, and(P, \lambda z.poss(x,z))(y), Q(y))$. The elementary tree pairs and resulting derivation tree for sentence (6) are given in Figure 7. The derived tree is given in Figure 8. When reduced, the resulting semantics is $a(z, \lambda x.(problem(x) \wedge the(y, solution(y) \wedge poss(x,y), isDifficult(y))), stumped(bill, z))$.

1.3.4 Nested Quantifiers and Inverse Linking

Quantifiers in prepositional phrases such as in sentence (7) pose another challenge for TAG semantics (Joshi et al., 2003). Although a nested quantifier may take scope over the quantifier within which it is nested (so-called "inverse linking") not all permutations of scope orderings of the quantifiers are available (Joshi et al., 2003). In particular, readings in which a quantifier intervenes between a nesting quantifier and its nested quantifier are not valid. In our example sentence (7), this predicts that the readings *some* > *two* > *every* and *every* > *two* > *some* should not be valid. Joshi et al. (2003) introduce a special device allowing nesting and nested quantifiers to form an indivisible quantifier

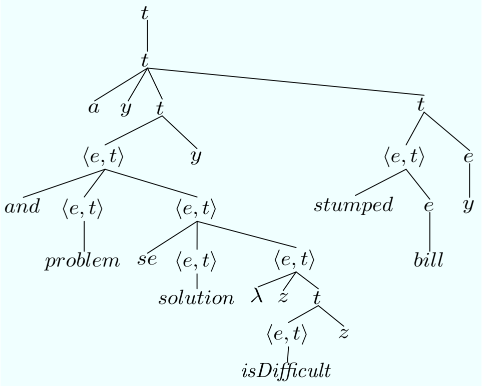

FIGURE 8 Derived tree for "A problem whose solution is difficult stumped Bill."

set during the derivation, which prevents other quantifiers from intervening between them. In our solution, because the nested quantifier is introduced through the prepositional phrase, which in turn modifies the noun phrase containing the nesting quantifier, the two quantifiers already naturally form a set that operates as a unit with respect to the rest of the derivation.[6] The elementary tree pairs and derivation trees for our analysis of (7) are shown in Figure 9.

One notable feature of this analysis is that the four different scope readings that result are not the product of a single derivation tree. The alternate scope orderings for the nested and nesting quantifier exist because there are two available adjunction sites for the scope of quantifiers in the prepositional phrase to attach. This results in two distinct derivation trees. The alternate scope orderings for this quantifier set and the remaining quantifier are obtained by multiple adjunction at the root of the verb tree. The set of valid derivation trees for a sentence thus constitutes the scope neutral representation. This set of trees may be compactly represented, for instance as a shared forest.[7]

[6] We make use of tree-set-local TAG in the semantics where the tree set for *every* adjoins into the tree set for *from*. Although tree-set-local TAG is more powerful than TAG, this particular use is benign because it cannot be iterated. More concretely, we could conventionally make the grammar tree-local by including all combinations of prepositions with quantifiers as elementary trees in the grammar.

[7] This analysis, like that of Joshi et al. (2003), makes several predictions about quantifier scope that might be disputed. First, some argue that more than four scope orderings should be available for sentences like sentence (7) (VanLehn, 1978, Hobbs and Shieber, 1987). This analysis cannot generate additional scope orderings without breaking tree set locality. Second, the scope readings in which the nesting

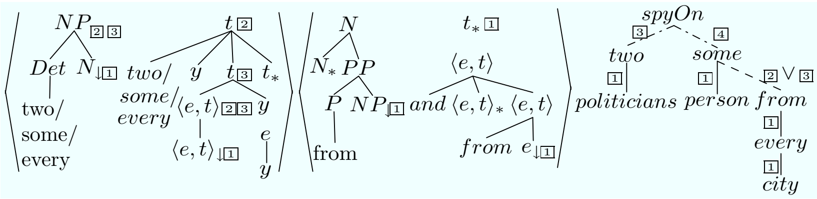

FIGURE 9 Key elementary trees and derivations for "Two politicians spy on someone from every city."

1.4 Comparison to the Kallmeyer and Romero Approach

As mentioned above, research on TAG semantics has been led by Laura Kallmeyer, Maribel Romero, and their collaborators through a series of papers refining a system of TAG semantics computation using feature unification and other formal devices (Kallmeyer and Romero, 2004, Romero et al., 2004, Kallmeyer, 2003, Kallmeyer and Joshi, 2003, Joshi et al., 2003, Gardent and Kallmeyer, 2003). Although their approach has evolved over time, the underlying principles of using the relationships expressed in the derivation tree as the basis for the computation and generating underspecified semantic representations have been constant. In its current formulation, they perform semantic computation by attaching semantic feature structures directly to the nodes in the derivation tree. When carefully chosen, these features unify to produce an underspecified representation of the semantics of a sentence that, when further disambiguated, generates the set of valid interpretations. In one or another of their recent papers they have provided successful analyses of each of the hard cases that we have addressed here, though some of their analyses might have to be restated to bring them up to date with the newest formulation of their method.

Our work owes much to theirs both for the clear formulation of the problems and the progress in formulating analyses for some of the hard cases. The primary advantage of our approach is its conceptual simplicity. The clear separation of syntax and semantics, the directness of the link interface, and the familiarity of the TAG operations used in our approach make it very simple. The semantic-feature-unification-based approach has become cleaner and easier to understand as Kallmeyer

quantifier takes scope over the nested quantifier result in the nested quantifier having scope over the restriction of the nesting quantifier but not over its scope. Donkey sentence constructions such as "Every man with two books loves them" call this prediction into question.

and others have refined it over the years. Nonetheless, it is safe to say that the amount of formal machinery—including propositional labels, separate individual and propositional variables, semantic representations consisting of a set of formulas and a set of scope constraints, features on the derived tree and the derivation tree, each semantic feature structure containing a nested feature structure for each address in the elementary syntax tree, each of these feature structures containing features to handle binding of propositional and individual variables, feature unification, flexible composition, and quantifier sets—necessary to solve the range of problems that we have addressed here, is qualitatively more complex. In fact, we use no formal machinery that had not been introduced by 1994 in the TAG literature.

An additional advantage of our approach is that it does not increase the expressivity of the TAG formalism. One might think that the inclusion of multiple adjunction would lead to an increase in expressivity (Dras, 1999). However, because links can only be used once in an STAG derivation, only a finite number of multiple adjunctions may occur at a single adjunction site. This rules out problematic uses of multiple adjunction. Kallmeyer and Romero maintain the semantic features on the derivation tree rather than in the feature structures already used in the feature-based TAGs (FTAG) of their syntax in part because the set of semantic feature structures is not finite, potentially increasing the expressivity of the FTAG formalism (Kallmeyer and Romero, 2004). Although moving the features to the derivation tree avoids increasing the expressivity of the formalism used for syntax when taken alone, the additional expressivity in the features of the semantics could be used to block operations in the syntax thereby filtering the syntax to produce non-tree-adjoining languages. It remains to be seen whether this additional expressivity will be required for TAG semantics.

Advantages and disadvantages of the different methods aside, in this still nascent area of research it is desirable to have several quite different approaches at our disposal as we explore the hard problems presented by generating natural language semantics in the TAG framework. Our approach revives an old idea with the aim of opening a new avenue for research into semantics in the TAG framework.

1.5 Conclusion

We have presented the synchronous TAG formalism as a method for computing semantics in the TAG framework, and have shown that it enables simple, natural analyses for all of the cases that have exercised recent attempts at formulating formal semantics for TAG. It satisfies each of the desiderata laid out at the beginning of this paper. First, it does not require any additional information other than that available in the derivation tree to generate the semantics. Because the syntax and semantic representations are built up synchronously, the derivation tree set is a complete specification of the relationship between them. Nothing other than the set of elementary tree pairs and the synchronous TAG operations are required to generate a semantic representation. Second, the derivation tree set provides a compact representation for all valid semantic interpretations of the given sentence. Using multiply-adjoined quantifiers we take advantage of the ambiguity in the interpretation of the derivation tree that is introduced by multiple adjunction. We take each possible ordering of multiply-adjoined trees to be valid. We leave open the possibility of using an additional method to prefer certain scope orders and disprefer or eliminate others. Third, the STAG system, as used, does not increase the expressivity of the TAG formalism (Shieber, 1994). Finally, our analysis is a straightforward expression of a simple idea: we use TAG for both syntax and semantics and use the derivation tree and the links between trees in elementary tree pairs as the interface between them.

1.6 Acknowledgements

This work was supported in part by grant IIS-0329089 from the National Science Foundation. We wish to thank Rani Nelken and the three anonymous reviewers for valuable comments on earlier drafts.

References

Dras, Mark. 1999. A meta-level grammar: Redefining synchronous TAG for translation and paraphrase. In *Proceedings of the Thirty-Seventh Annual Meeting of the Association for Computational Linguistics*, pages 80–87. Maryland, USA.

Frank, Robert. 1992. Syntactic locality and Tree Adjoining Grammar: Grammatical, acquisition and processing perspectives. Ph.D. Thesis, University of Pennsylvania.

Gardent, Claire and Laura Kallmeyer. 2003. Semantic construction in feature-based TAG. In *Proceedings of the 10th Meeting of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary.

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

Joshi, Aravind K., Laura Kallmeyer, and Maribel Romero. 2003. Flexible composition in LTAG: Quantifier scope and inverse linking. In I. v. d. S.

Harry Bunt and R. Morante, eds., *Proceedings of the Fifth International Workshop on Computational Semantics IWCS-5*, pages 179–194. Tilburg.

Kallmeyer, Laura. 2002a. Enriching the TAG derivation tree for semantics. In S. Busemann, ed., *KONVENS 2002. 6. Konferenz zur Verarbeitung natürlicher Sprache.*, pages 67–74. Saarbrücken.

Kallmeyer, Laura. 2002b. Using an enriched tag derivation structure as basis for semantics. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6)*, pages 127–136. Venice.

Kallmeyer, Laura. 2003. LTAG semantics for relative clauses. In I. v. d. S. Harry Bunt and R. Morante, eds., *Proceedings of the Fifth International Workshop on Computational Semantics IWCS-5*, pages 195–210. Tilburg.

Kallmeyer, Laura and Aravind K. Joshi. 2003. Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. *Research on Language and Computation* 1:3–58.

Kallmeyer, Laura and Maribel Romero. 2004. LTAG semantics with semantic unification. In *Proceedings of TAG+7*, pages 155–162. Vancouver.

Romero, Maribel, Laura Kallmeyer, and Olga Babko-Malaya. 2004. LTAG semantics for questions. In *Proceedings of TAG+7*, pages 186–193. Vancouver.

Schabes, Yves and Stuart M. Shieber. 1993. An alternative conception of tree-adjoining derivation. *Computational Linguistics* 20(1):91–124.

Shieber, Stuart M. 1994. Restricting the weak-generative capacity of synchronous tree-adjoining grammars. *Computational Intelligence* 10(4):371–385.

Shieber, Stuart M. and Yves Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, vol. 3, pages 253–258. Helsinki.

VanLehn, Kurt. 1978. Determining the scope of English quantifiers. Tech. Rep. 483, MIT Artificial Intelligence Laboratory, Cambridge, MA.

Williford, Sean. 1993. Application of synchronous tree-adjoining grammar to quantifier scoping phenomena in English. Undergraduate Thesis, Harvard College.