

- Title: An Analysis of Clausal Coordination using Synchronous Tree Adjoining Grammar
- Authors and Affiliations: Chung-hye Han (Department of Linguistics, Simon Fraser University), Sara Williamson (Department of Linguistics, Simon Fraser University), Logan Born (School of Computing Science, Simon Fraser University), Anoop Sarkar (School of Computing Science, Simon Fraser University)
- Address for Chung-hye Han:  
Department of Linguistics, Simon Fraser University  
8888 University Drive  
Burnaby, British Columbia V5A1S6, Canada
- Email address:  
chunghye@sfu.ca  
slw28@sfu.ca  
loborn@sfu.ca  
anoop@sfu.ca
- Telephone for Chung-hye Han: +1-778-782-5507
- Abstract:  
This paper presents a novel analysis of clausal coordination with shared arguments using Synchronous Tree Adjoining Grammar (STAG), a pairing of a Tree Adjoining Grammar (TAG) for syntax and a TAG for semantics. In clausal coordination, one or more arguments can be shared by the verbal predicates of the conjuncts, as in *Sue likes and Kim hates Pete*, where an object argument *Pete* is shared by *likes* and *hates*. As the predicate-argument structure must be represented within each predicative elementary tree in TAG, modeling argument sharing across clauses poses an interesting challenge for TAG. A widely adopted approach within the TAG literature at present is to employ the Conjoin Operation ([Sarkar and Joshi, 1996](#)), a non-standard tree-composing operation in TAG. This operation applies across elementary trees to identify and merge the arguments from each clause, yielding a derivation structure in which the shared arguments are combined with multiple elementary trees, and a derived tree in which the shared arguments are dominated by multiple verbal projections. In contrast, our STAG analysis pairs a syntactic elementary tree that participates in the derivation of clausal coordination with a semantic elementary tree that includes a  $\lambda$ -term to abstract over the shared argument. This allows the sharing of arguments in coordination to be instantiated in semantics without being represented in syntax in the form of multiple dominance, utilizing only the standard TAG operations, substitution and adjoining.
- Keywords:  
Synchronous Tree Adjoining Grammar, Multi-Component Tree Adjoining Grammar, delayed tree-local derivation, clausal coordination, argument sharing

# An Analysis of Clausal Coordination using Synchronous Tree Adjoining Grammar

Chung-hye Han, Sara Williamson, Logan Born, and Anoop Sarkar

## Abstract

This paper presents a novel analysis of clausal coordination with shared arguments using Synchronous Tree Adjoining Grammar (STAG), a pairing of a Tree Adjoining Grammar (TAG) for syntax and a TAG for semantics. In clausal coordination, one or more arguments can be shared by the verbal predicates of the conjuncts, as in *Sue likes and Kim hates Pete*, where an object argument *Pete* is shared by *likes* and *hates*. As the predicate-argument structure must be represented within each predicative elementary tree in TAG, modeling argument sharing across clauses poses an interesting challenge for TAG. A widely adopted approach within the TAG literature at present is to employ the Conjoin Operation (Sarkar and Joshi, 1996), a non-standard tree-composing operation in TAG. This operation applies across elementary trees to identify and merge the arguments from each clause, yielding a derivation structure in which the shared arguments are combined with multiple elementary trees, and a derived tree in which the shared arguments are dominated by multiple verbal projections. In contrast, our STAG analysis pairs a syntactic elementary tree that participates in the derivation of clausal coordination with a semantic elementary tree that includes a  $\lambda$ -term to abstract over the shared argument. This allows the sharing of arguments in coordination to be instantiated in semantics, without being represented in syntax in the form of multiple dominance, utilizing only the standard TAG operations, substitution and adjoining.

## 1 Introduction

Tree Adjoining Grammar (TAG) (Joshi et al., 1975) is an appealing vehicle for linguistic analysis because it has various useful formal properties: TAGs can, for example, model crossing dependencies in language while still affording computationally efficient polynomial time algorithms for parsing. Clausal coordination, however, is a syntactic construction

that has been hard to handle elegantly using TAGs due to the sharing of arguments between conjuncts (see [Sarkar and Joshi \(1996\)](#) for a survey of prior work in TAG on clausal coordination). In this work, we aim to address the issue of clausal coordination and sharing of arguments without adding to the computational power and without sacrificing the appealing linguistic properties of TAGs. We provide an analysis for clausal coordination using a Synchronous Tree Adjoining Grammar (STAG), a pairing of a TAG for syntax and a TAG for semantics. As a consequence, our analysis does not add any new non-standard tree-composing operations to TAG, enabling us to retain its useful formal and linguistic insights and at the same time provide a comprehensive analysis for clausal coordination.

In clausal coordination, one or more arguments can be shared by the verbal predicates of the conjuncts. For example, in (1), an object argument, *Pete*, is shared by *likes* and *hates*, and in (2), a subject argument, *Sue*, is shared by the two verbs.

(1) Sue likes and Kim hates Pete.

a.  $\text{likes}(\text{Sue}, \text{Pete}) \wedge \text{hates}(\text{Kim}, \text{Pete})$

(2) Sue hates Pete and likes Kim.

a.  $\text{hates}(\text{Sue}, \text{Pete}) \wedge \text{likes}(\text{Sue}, \text{Kim})$

A widely adopted analysis to such coordination, since [Ross \(1967\)](#), is to postulate an across-the-board (ATB) movement of the shared argument, in which multiple underlying copies of the shared material are identified during movement, yielding a single overt copy located outside of the coordinate structure. So, (1) and (2) would be derived from movement of the shared argument from both conjuncts to a position outside of the coordinate structure, as in (3) and (4).

(3) [Sue likes  $t_i$ ] and [Kim hates  $t_i$ ] **Pete** $_i$ .

(4) **Sue** $_i$  [ $t_i$  hates Pete] and [ $t_i$  likes Kim].

Ross's (1967) analysis implies that movement identifies two syntactically distinct objects due to the coordinate structure. This issue was partially addressed by derived syntactic types in Gazdar (1981) within a context-free grammar formalism. However, the ATB movement analysis incorrectly predicts that shared arguments be barred from islands, given that movement dependency is subject to island constraints (Wexler and Culicover, 1980). (5) illustrates that a *wh*-movement dependency cannot be formed across a relative clause, an island. In contrast, in (6), a shared argument can form a putative ATB movement dependency across relative clauses, violating the island constraints.

(5) \* What did Max denounce [the senator [who wrote  $t_i$ ]]? (Sabbagh, 2014, 14)

(6) Max publicly denounced [the senator [who wrote  $t_i$ ]], and Pauline outwardly criticized [the magazine editor [who published  $t_i$ ]], [**the speech that encouraged the riot**] $_i$ . (Sabbagh, 2014, 15)

Such ATB movement would also violate the right-raft constraint, which limits rightward movement (extraposition) to a landing site one bounding node above the source (Sabbagh, 2007). If the relevant bounding nodes are Verb Phrase (VP) and Tense Phrase (TP) and the shared argument originates in VP, movement outside of the coordinated TP structure would violate this constraint.

Combinatory Categorical Grammar (CCG) (Steedman, 1996) places the shared argument outside the coordinating conjuncts without postulating movement. It uses a syntactic combinator  $(X \backslash X) / X$  for conjunctions, which combines two constituents of any type (one on the left and the other on the right, as represented by the direction of the slash). In semantics, the coordinated constituents provide a predicate  $\lambda$ -term which is then reduced using the shared argument. CCG combines type-raising and function composition to handle coordination, which leads to a view of constituency that is quite different from traditional phrase structure. Further, function composition is typically restricted in order to derive is-

land effects. Thus, it would be difficult to account for examples such as (6) under the CCG analysis.

Another prominent analysis, starting with Wexler and Culicover (1980), is to postulate that the appearance of a shared argument is a result of ellipsis of corresponding arguments from other conjuncts. Under the ellipsis analysis, (1) and (2) would be a result of eliding the object argument from the first conjunct (7) and the subject argument from the second conjunct (8), respectively.

(7) [Sue likes Pete] and [Kim hates **Pete**].

(8) [**Sue** hates Pete] and [~~Sue~~ likes Kim].

The ellipsis analysis predicts that a clausal coordination with a shared argument and the corresponding non-elided version should have the same meaning. But this is not always the case (Sabbagh, 2007). For instance, while (9) means that the same student read every paper and summarized every book, (10) can mean different students read every paper and summarized every book.

(9) **A student** read every paper and summarized every book.

(10) A student read every paper and a student summarized every book.

This takes us to the multiple dominance analysis, first proposed by McCawley (1982), that postulates that a shared argument is multiply dominated by elements from multiple conjuncts. A version of this approach has been developed in Sarkar and Joshi (1996) within the TAG literature. Sarkar and Joshi (1996) posit that the shared argument is located in the canonical position within each conjunct, and introduce a new operation to the TAG formalism called the Conjoin Operation that applies across elementary trees. This operation identifies and merges the shared argument when two elementary trees combine via coordination, yielding a derived tree in which an argument is multiply dominated by

two verbal projections. The Conjoin Operation analysis has been used and extended often in TAG-based linguistic research, including the semantics of clausal coordination and scope (Banik, 2004; Han et al., 2008; Storoshenko and Frank, 2012) and the syntax of Right-Node-Raising (Han et al., 2010).

According to the multiple dominance analysis, because the shared argument is in a dominance relation within each conjunct, it must be syntactically licensed in each conjunct. However, as observed in Cann et al. (2005), all syntactic requirements of the shared argument must be met by the elements within the conjunct it occurs with, and not by elements in other conjuncts. For instance, a negative polarity item (NPI) in the shared object, which occurs in the second conjunct on the surface, is licensed by negation in the second conjunct (11), but not by negation in the first conjunct (12).<sup>1</sup>

(11) John has read, but he hasn't understood **any of my books**. (Cann et al., 2005, 1e)

(12) \* John hasn't understood, but has read **any of my books**.

Cann et al.'s (2005) analysis of right node raising uses the Dynamic Syntax (DS) formalism (Kempson et al., 2000). DS models the syntax-semantics interface as a monotonic tree growth process defined over the left–right sequence of words in the syntax which builds a semantic interpretation incrementally from left to right as well. The main distinction from our approach is that TAG is compatible with left to right incremental parsing but does not insist on a left to right algorithm to build the syntactic and semantic structures. TAG also does not eschew constituency unlike DS. Since TAG is a formal grammar rather than a linguistic theory, it may even be possible to generate the semantics produced by a DS analysis using an incremental left-to-right TAG parser. However, exploring such a different type of

---

<sup>1</sup>A reviewer questions how an NPI in a shared argument (a right-node-raised NPI) need only be licensed in the second conjunct, under the assumption that the licensing of NPIs is a matter of semantics. Even if semantics plays a role in NPI licensing, which we do not doubt, when the licenser is negation, there is a syntactic requirement that the NPI be c-commanded by the negation, and this syntactic requirement would be met if the NPI is in an object position of a clause that contains negation.

analysis is beyond the scope of this paper.

In this paper, using Synchronous Tree Adjoining Grammar, we propose an alternative to the TAG analysis of coordination that does not rely on the Conjoin Operation. In our proposal, the shared argument is syntactically present only in one conjunct, and syntactically null in other conjuncts. The syntactic elementary trees representing the conjuncts with missing arguments are paired with semantic elementary trees with unsaturated arguments, and the syntactic elementary trees with shared arguments are paired with semantic elementary trees that use  $\lambda$ -terms to abstract over the shared arguments. Composition of these trees via adjoining allows sharing of arguments to be instantiated in semantics, without being represented in syntax in the form of ATB movement, ellipsis or multiple dominance.

The remainder of this paper is organized as follows. In section 2, we introduce the basics of TAG for doing natural language syntax, and illustrate in more detail how TAG's Conjoin Operation identifies a shared argument in clausal coordination. In section 3, we introduce STAG and show how compositional semantics is done using STAG, and present our analysis where sharing of arguments takes place in semantics, not in syntax. This analysis is extended in section 4 to account for ATB *wh*-movement, the interaction of coordination and quantification, and argument sharing across islands.

## 2 TAG and the Conjoin Operation

### 2.1 Introduction to TAG syntax

Tree Adjoining Grammar (TAG) is a grammar formalism first defined in Joshi, Levy and Takahashi (1975). Similarly to a context-free grammar (Sipser, 1996), it uses the concepts of non-terminals (abstract symbols that appear only in the grammar) and terminal symbols (symbols that appear in the input). In a TAG, however, the non-terminals and terminals are used to create trees which are the elementary objects in the formal grammar. These

*elementary trees* are composed using recursive operations of *substitution*, which expands non-terminal leaf nodes in a tree, and *adjoining*, which expands non-terminals that are non-leaf nodes in the tree. These operations will become clear when we show examples of how they are used to compose elementary trees later in this section.

When the TAG formalism is used for the linguistic analysis of natural language, the elementary trees are lexicalized to represent extended projections of a lexical anchor (a terminal symbol). Furthermore, in linguistic applications of TAG, the lexicalized elementary trees are minimal in that all and only the syntactic/semantic arguments of the lexical anchor are encapsulated and all recursion is factored away. The elementary trees in TAG are therefore said to possess an Extended Domain of Locality (Kroch and Joshi, 1985). We follow Frank’s (2002) formulation of the extended projection property of elementary trees, the Condition on Elementary Tree Minimality (CETM), which states that “the syntactic heads in an elementary tree and their projections must form an extended projection of a single lexical head” (p. 54). According to Frank, the extended projection of a lexical head includes the projections of all functional heads that embed it, as in Grimshaw (1991). This means that an elementary tree anchoring a verb can project to Verb Phrase (VP) but also to Tense Phrase (TP) and Complementizer Phrase (CP), and an elementary tree anchoring a noun can project to Noun Phrase (NP) but also to Determiner Phrase (DP) and Prepositional Phrase (PP). Further, the fundamental thesis in TAG for natural language is that “every syntactic dependency is expressed locally within a single elementary tree” (Frank 2002, p. 22). This allows for a syntactic dependency created by movement to occur within an elementary tree, but not across elementary trees.

The elementary trees in Figure 1 and Figure 2 can be used to derive the example in (13).<sup>2</sup>

---

<sup>2</sup>Here and throughout, we label subject DPs as  $DP_A$  and object DPs as  $DP_B$ . In the following sections, this convention helps to disambiguate whether a tree contains a shared subject or object argument. We use subscripts such as  $i$  and  $j$  to indicate movement dependencies.



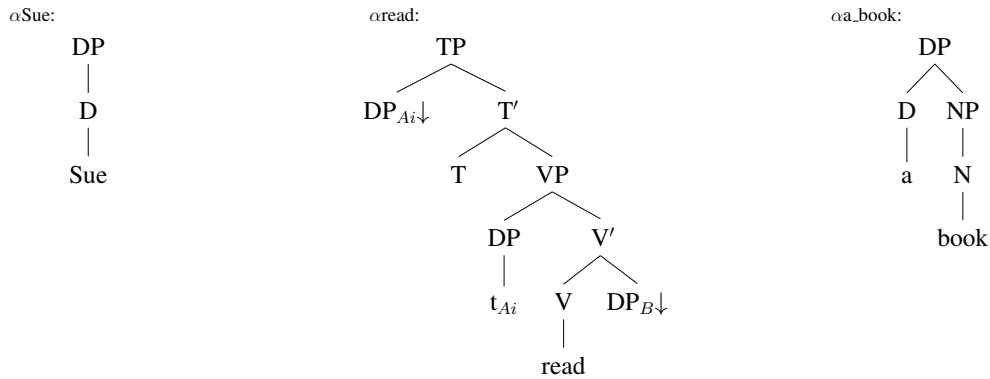


Figure 1: Initial trees in TAG for sentence (13).

(13) Sue read a fantastic book enthusiastically.

( $\alpha_{\text{read}}$ ) is an elementary tree because it is an extended projection of the lexical predicate *read* and has argument slots for the subject and the object, marked by downward arrows ( $\downarrow$ ). Moreover, the movement of the subject DP from [Spec,VP] to [Spec,TP], following the VP-internal Subject Hypothesis (Koopman and Sportiche, 1991), is an operation internal to the elementary tree, and therefore represents a syntactic dependency localized to the elementary tree. ( $\alpha_{\text{Sue}}$ ) and ( $\alpha_{\text{a\_book}}$ ) are valid elementary trees because these DP trees each contain a single lexical head, *Sue* for ( $\alpha_{\text{Sue}}$ ) and *book* for ( $\alpha_{\text{a\_book}}$ ), that can form an extended projection with a DP, in line with the DP Hypothesis (Abney, 1987).

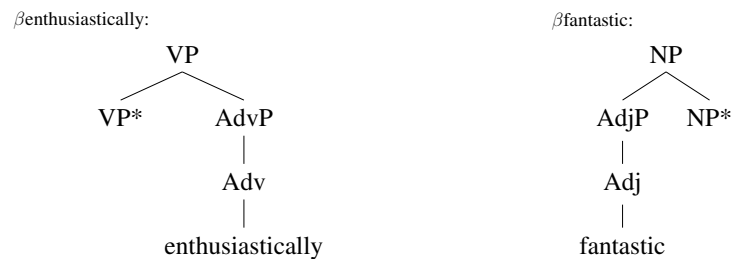


Figure 2: Auxiliary trees in TAG for sentence (13).

Elementary trees are of two types: initial trees and auxiliary trees. By convention, names of initial trees are prefixed with  $\alpha$ , and names of auxiliary trees are prefixed with  $\beta$ .

A derivation in TAG starts with initial trees, such as trees for simple clauses and nominal phrases. Some examples of initial trees are shown in Figure 1. Auxiliary trees are used to introduce recursive structures, for example, adjuncts or other recursive portions of the grammar. Auxiliary trees have a special non-terminal node called the foot node (marked with an asterisk) among the leaf nodes, which has the same label as the root node of the tree. The elementary trees in Figure 2 are examples of auxiliary trees. These auxiliary trees are well-formed because CETM requires only that each syntactic head and its projection form an extended projection, rendering the presence of the VP root node in ( $\beta$ enthusiastically) and the NP root node in ( $\beta$ fantastic) consistent with CETM. Following Frank (2002), we can count VP\* in ( $\beta$ enthusiastically) and NP\* in ( $\beta$ fantastic) as arguments of their lexical anchors, as the process of theta-identification (Higginbotham, 1985) obtains between the lexical anchors and these foot nodes.

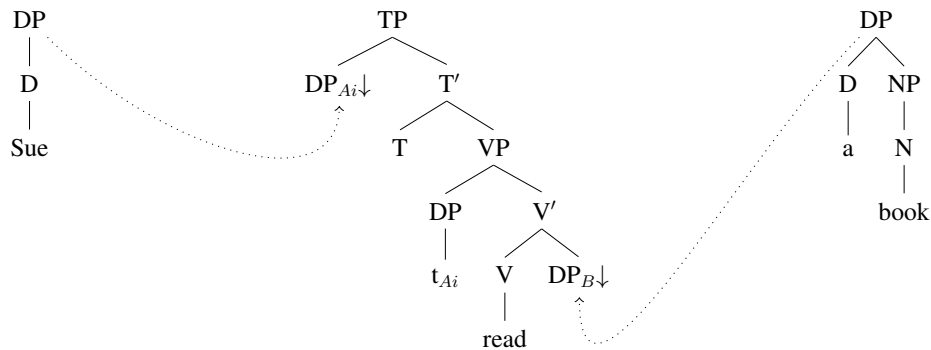


Figure 3: Substitution in TAG

Elementary trees are combined through two derivational operations: substitution and adjoining. In substitution, the root node on an initial tree is merged into a matching non-terminal leaf node marked for substitution ( $\downarrow$ ) in another tree. This is illustrated in Figure 3. In adjoining, an auxiliary tree is grafted onto a non-terminal node in another elementary tree that matches the root and foot nodes of the auxiliary tree. For example, Figure 4 illustrates ( $\beta$ enthusiastically) adjoining to the VP node in ( $\alpha$ read), and ( $\beta$ fantastic) adjoining to the

NP node in ( $\alpha a\_book$ ), which in turn substitutes into ( $\alpha read$ ).

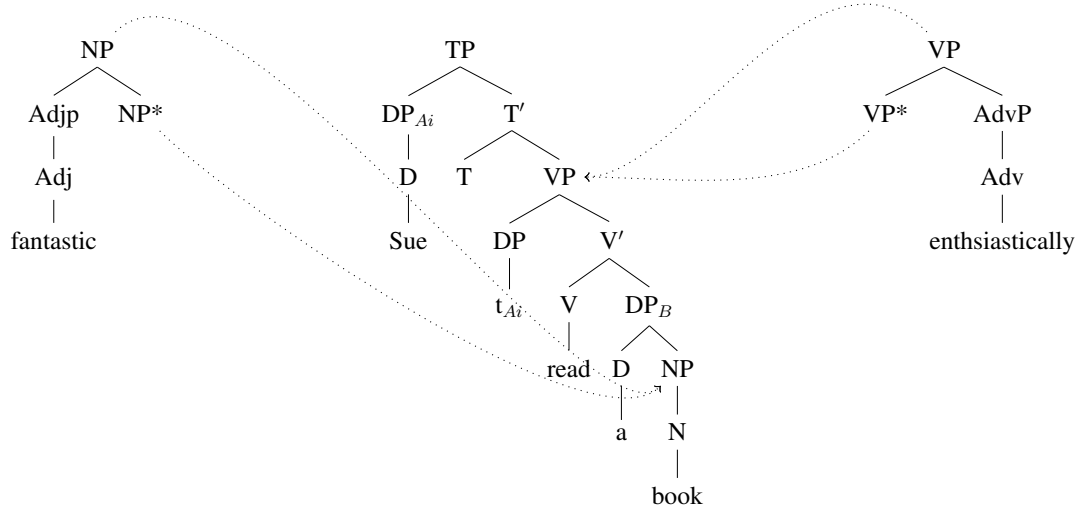


Figure 4: Adjoining in TAG

The process of composing the elementary trees of a TAG produces two structures: a TAG derived tree and a TAG derivation structure. The derived tree is the conventional phrase structure tree and represents surface constituency. For instance, combining the elementary trees in Figures 1 and 2 through substitution and adjoining as in Figures 3 and 4 generates the derived tree in Figure 5 (left). The derivation structure represents the history of composition of the elementary trees and the dependencies between the elementary trees. In a derivation structure, each node is an elementary tree, and the children of a node  $N$  represent the trees which are adjoined or substituted into the elementary tree represented by  $N$ . The link connecting a pair of nodes is annotated with the location in the parent elementary tree where adjoining or substitution has taken place.<sup>3</sup> An example of a derivation structure is given in Figure 5 (right). Figure 5 (right) records the history of composition of the elementary trees to produce the derived tree in Figure 5 (left): ( $\beta fantastic$ ) adjoins

<sup>3</sup>The location in the parent elementary tree is usually denoted by its Gorn tree address. Here, we use node labels such as DP or VP for the sake of simplicity.

to  $(\alpha a\_book)$  at NP,  $(\alpha Sue)$  and  $(\alpha a\_book)$  substitute into  $(\alpha read)$  at  $DP_A$  and  $DP_B$  respectively, and  $(\beta enthusiastically)$  adjoins to  $(\alpha read)$  at VP.<sup>4</sup>

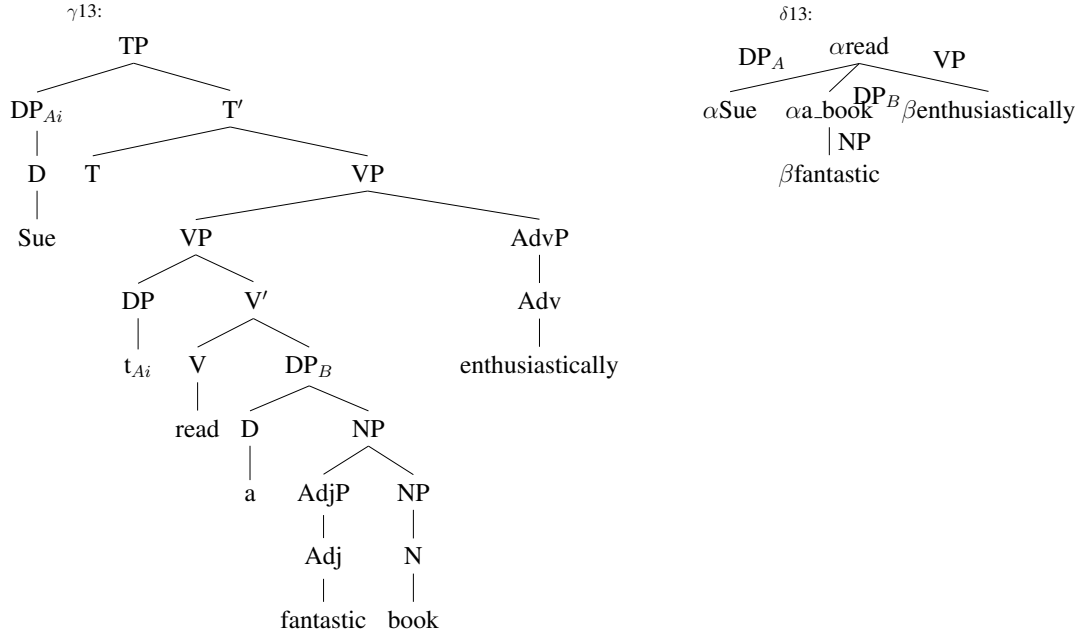


Figure 5: Derived tree and derivation structure in TAG for sentence (13).

The TAG formalism has been used to provide analyses for various complex syntactic phenomena, including unbounded dependency in *wh*-constructions and island effects (Frank, 2002, 2006), subject-to-subject raising (Frank, 2002), clitic-climbing in Romance (Bleam, 2000), West-Germanic verb raising (Kroch and Santorini, 1991), extraposition (Kroch and Joshi, 1987), tough-movement (Kulick, 2000) and *it*-clefts (Han and Hedberg, 2008).

## 2.2 Argument Sharing via the Conjoin Operation

In order to account for clausal coordination with shared arguments, Sarkar and Joshi (1996) utilize elementary trees with contraction sets and coordinating auxiliary trees. A contrac-

<sup>4</sup>By convention, names of derivation structures are prefixed with  $\delta$ , and names of derived trees are prefixed with  $\gamma$ .

tion set of an elementary tree contains nodes in the tree that undergo argument sharing in the derivation of a clausal coordination. Two elementary trees with matching contraction sets compose via the Conjoin operation, an operation that was introduced into the TAG formalism and cannot be reduced to the standard operations of substitution or adjoining. This allows the nodes in the contraction sets from the two trees to identify and merge into one. Substitution of an argument DP into the identified nodes will result in argument sharing.

The elementary trees necessary to derive (1) are given in Figure 6. Elementary trees such as  $(\beta\text{and\_hates}_{\{DP_B\}})$  are in accordance with CETM if we assume that the coordinator, which is a functional head, is part of the extended projection of the verb, and that it induces the requirement that there be a TP argument to participate in the clausal conjunction. In each of  $(\alpha\text{likes}_{\{DP_B\}})$  and  $(\beta\text{and\_hates}_{\{DP_B\}})$ , the object DP node is in the contraction set, notated as a subscript ( $\{DP_B\}$ ) in the tree name and marked in the tree with a circle around it, and represents a shared argument. When  $(\beta\text{and\_hates}_{\{DP_B\}})$  adjoins to  $(\alpha\text{likes}_{\{DP_B\}})$  via the Conjoin operation, the two trees undergo contraction, identifying the nodes in the contraction sets. This allows the DP tree  $(\alpha\text{Pete})$  to simultaneously substitute into both contraction nodes, and in the derived tree, the two nodes are identified, merging into one. The substitution of  $(\alpha\text{Sue})$  and  $(\alpha\text{Kim})$  into the subject DP nodes of  $(\alpha\text{likes}_{\{DP_B\}})$  and  $(\beta\text{and\_hates}_{\{DP_B\}})$ , in addition to the simultaneous substitution of  $(\alpha\text{Pete})$  into the object DP nodes of the two elementary trees, as recorded in the derivation structure  $(\delta 1)$  in Figure 7, generates the derived tree  $(\gamma 1)$  in Figure 7. The resulting derived tree is a directed graph as a single node is dominated by multiple nodes. The shared argument, *Pete*, is thus represented as a syntactic argument of both the verbs, *likes* and *hates*, capturing the meaning of the sentence that the person that Sue likes and the person that Kim hates are the same individual.

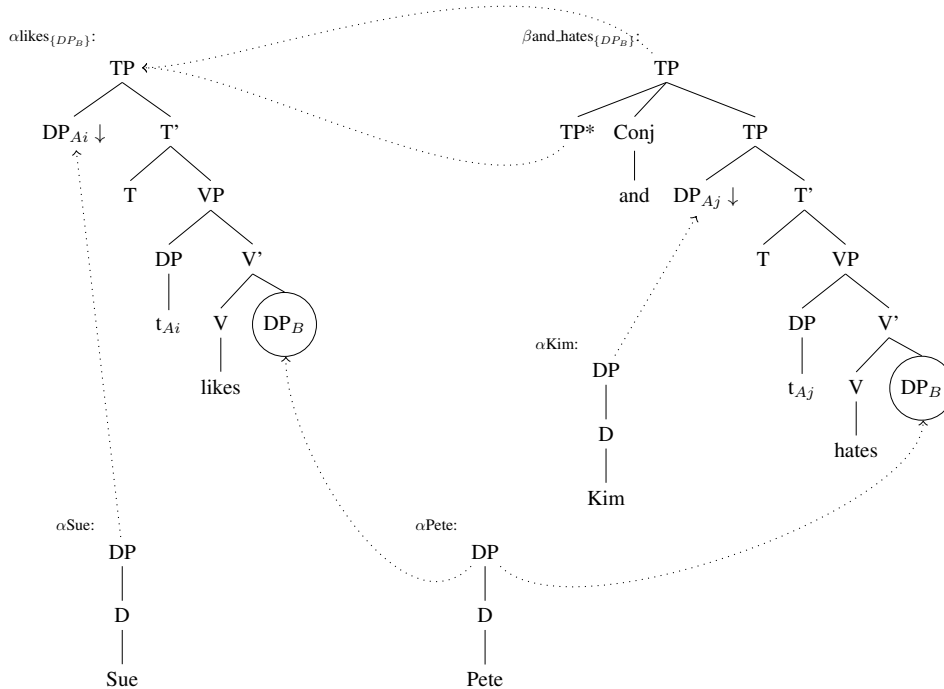


Figure 6: Elementary trees for *Sue likes and Kim hates Pete*

### 3 Argument Sharing via Semantics using STAG: Our Proposal

According to the NPI examples (11)-(12) discussed in section 1, the shared argument seems to form syntactic dependencies only with elements in the conjunct in which it appears on the surface, but not with elements in other conjuncts. We capture this intuition with the proposal that the shared argument is syntactically present only in one of the conjuncts, and missing in other conjuncts, resulting in predicates with unsaturated arguments in semantics. We implement our analysis using Synchronous Tree Adjoining Grammar (STAG). An introduction to doing compositional semantics of natural language using STAG is provided in subsection 3.1. Details of our analysis are given in subsection 3.2 using an object argument sharing example and in subsection 3.3 using a subject argument sharing exam-

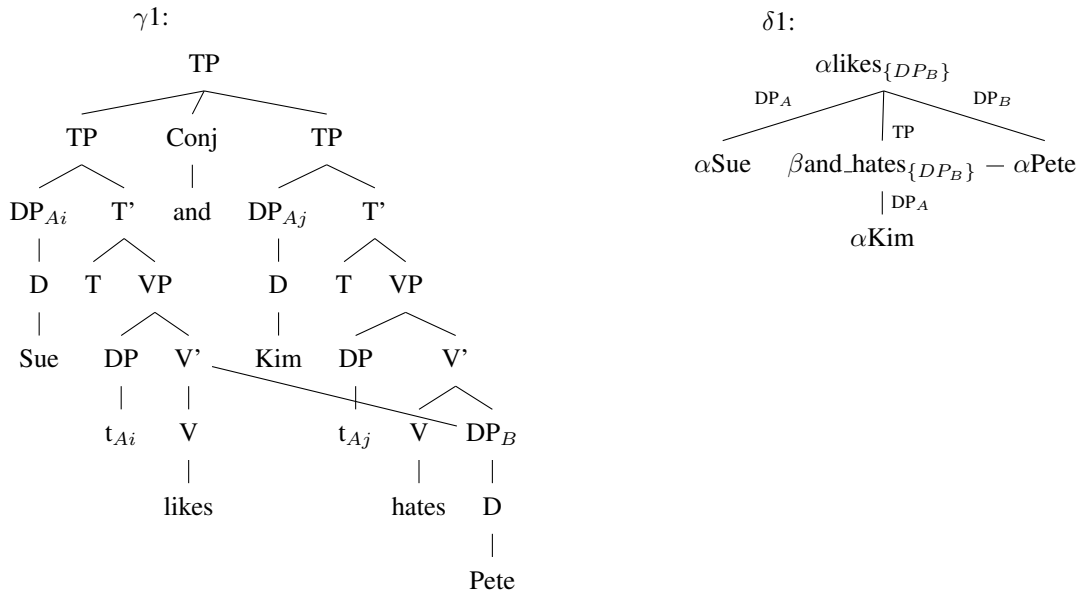


Figure 7: Derived tree and derivation structure for *Sue likes and Kim hates Pete* with the Conjoin Operation

ple. Subsection 3.4 contains our analysis of argument sharing across verbal complement clauses.

### 3.1 Introduction to STAG and compositional semantics

We illustrate the framework of Synchronous Tree Adjoining Grammar (STAG) and STAG-based compositional semantics and clarify our assumptions using (14), a simple sentence that contains an existential quantifier and an attributive adjective. A similar example was used in subsection 2.1 to illustrate a syntactic derivation in TAG.

(14) Sue read a fantastic book.

The STAG formalism used in this paper is based on the re-definition of STAG in Shieber (1994), which is a more computationally constrained version of the STAG defined in Shieber and Schabes (1990). In STAG, each syntactic elementary tree is paired with one or more semantic trees that represent its meaning with links between matching

nodes. A synchronous derivation proceeds by mapping a derivation structure from the syntax side to an isomorphic derivation structure on the semantics side, and is synchronized by the links specified in the elementary tree pairs. In the tree pairs given in Figure 8, the trees on the left side are syntactic elementary trees and the ones on the right side are semantic trees. In the semantic trees, all nodes are labelled according to their semantic types, following the notation in [Nesson and Shieber \(2006\)](#), and the lexical anchor is represented as an unreduced  $\lambda$ -expression, following the notation in [Han \(2007\)](#). Making use of unreduced  $\lambda$ -expressions in semantic trees allows the reduction of semantic derived trees to logical forms through the application of  $\lambda$ -conversion and other operations defined on  $\lambda$ -expressions. The linked nodes are shown with boxed numbers. For the sake of simplicity, in the following elementary tree pairs, we only include links that are relevant for the derivation of the given examples.<sup>5</sup>

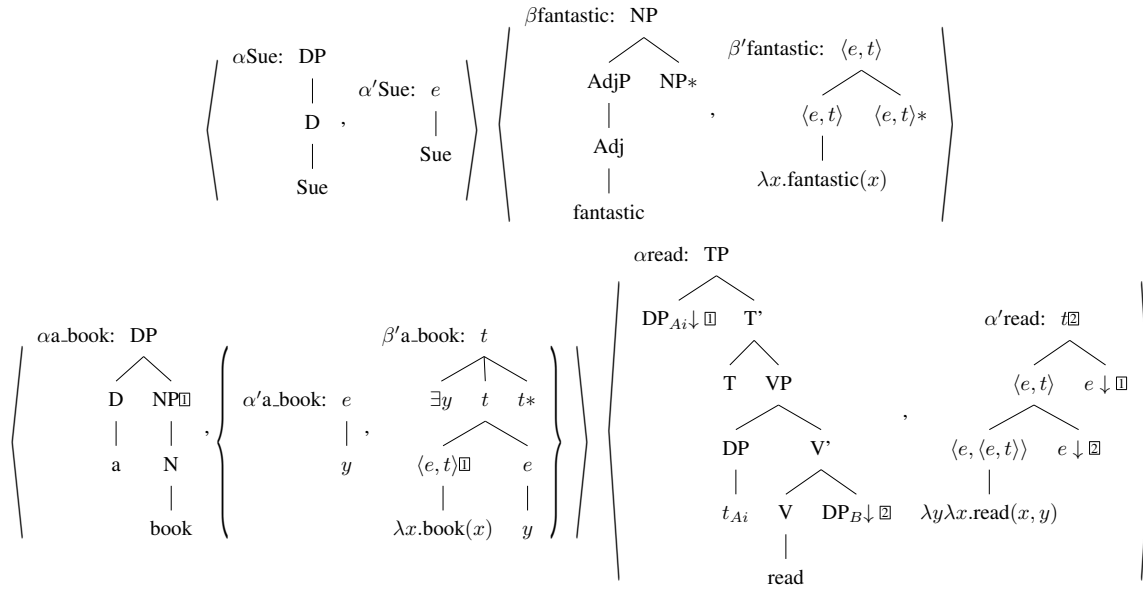


Figure 8: Syntactic and semantic elementary trees for *Sue read a fantastic book*.

Figure 8 contains the elementary trees required to generate the syntactic structure and

<sup>5</sup>By convention, names of semantic elementary trees are prefixed with  $\alpha'$  or  $\beta'$ , names of semantic derivation structures are prefixed with  $\delta'$ , and names of semantic derived trees are prefixed with  $\gamma'$ .



the logical form of (14). The proper name tree in  $(\alpha\text{Sue})$  is paired with a tree representing an entity on the semantics side, and the attributive adjective tree in  $(\beta\text{fantastic})$  is paired with an auxiliary tree on the semantics side that represents a one-place predicate to be adjoined to another one-place predicate. For quantified DPs, we follow Shieber and Schabes (1990) and use tree-local multi-component TAG (MC-TAG) on the semantics side. In tree-local MC-TAG, the basic object of a derivation is a set of elementary trees, called a multi-component (MC) set. At each step in a derivation, all the trees in an MC set are restricted to adjoin or substitute simultaneously into a single elementary tree. With this restriction, MC-TAG is shown to have the same weak and strong generative capacity as basic TAG (Weir, 1988). Thus, the DP in  $(\alpha\text{a\_book})$  is paired with an MC set  $\{(\alpha'\text{a\_book}), (\beta'\text{a\_book})\}$  on the semantics side:  $(\alpha'\text{a\_book})$  provides an argument variable, and  $(\beta'\text{a\_book})$  provides an existential quantifier with the restriction and scope. The transitive tree in  $(\alpha\text{read})$  is paired with a semantic tree representing a formula that consists of a two-place predicate and two  $e$  nodes for the arguments of the predicate. The links, shown as boxed numbers in the figure, guarantee that if a syntactic tree substitutes into  $\text{DP}_A$ , its corresponding semantic tree will substitute into the  $e$  node marked with  $\boxed{1}$ . Likewise, the tree which substitutes into  $\text{DP}_B$  must be paired with a semantic MC set, of which one component will substitute into the  $e$  node marked with  $\boxed{2}$  and the other will adjoin to the  $t$  node marked with  $\boxed{2}$ . The syntactic and semantic derivation structures are given in Figure 9, and the derived trees are given in Figure 10. Following Nesson and Shieber (2006; 2007), instead of Gorn addresses, we label STAG derivation structures with the link numbers where child trees adjoin, to highlight the synchronization in the derivation.

The semantic derived trees can be reduced by applying  $\lambda$ -conversion, as the nodes dominate typed  $\lambda$ -expressions and terms. When reducing the semantic derived trees, in addition to  $\lambda$ -conversion, we use Predicate Modification, as defined in Heim and Kratzer (1998) in (15).

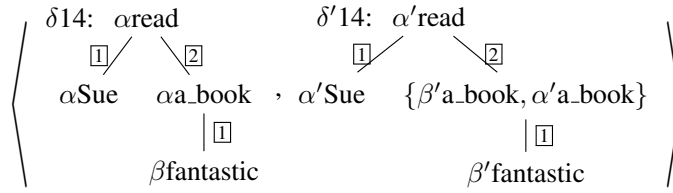


Figure 9: Syntactic and semantic derivation structures for *Sue read a fantastic book*

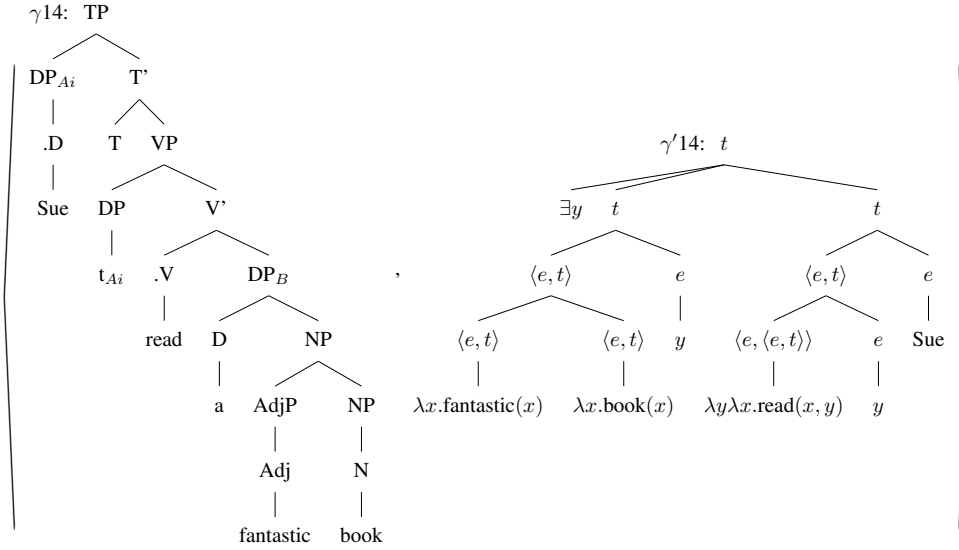


Figure 10: Syntactic and semantic derived trees for *Sue read a fantastic book*

(15) Predicate Modification

If  $\alpha$  has the form  $\alpha$ , and  $\llbracket\beta\rrbracket$  and  $\llbracket\gamma\rrbracket$  are both of type  $\langle e, t \rangle$ ,



then  $\llbracket\alpha\rrbracket = \lambda x[\llbracket\beta\rrbracket(x) \wedge \llbracket\gamma\rrbracket(x)]$ .

The application of Predicate Modification and  $\lambda$ -conversion reduces  $(\gamma'14)$  to the formula in (16).

$$(16) \quad \exists y[\text{fantastic}(y) \wedge \text{book}(y)][\text{read}(\text{Sue}, y)]$$

## 3.2 Object argument sharing

Consider the example of object argument sharing we introduced earlier in (1), repeated here as (17). We illustrate our approach by providing an analysis for this example.

(17) Sue likes and Kim hates Pete.

a.  $\text{likes}(\text{Sue}, \text{Pete}) \wedge \text{hates}(\text{Kim}, \text{Pete})$

For the analysis of (17), an example of clausal coordination with a shared object argument, we use elementary tree pairs in  $(\beta\text{likes\_and}_{\{DP_B\}})$  and  $(\beta'\text{likes\_and}_{\{DP_B\}})$  in Figure 11.  $(\beta\text{likes\_and}_{\{DP_B\}})$  is an auxiliary TP tree that introduces a coordinator and adjoins to another TP, with which it coordinates. The object argument of this auxiliary tree is null, directly reflecting the fact that it is absent in the first conjunct. The content of the null object argument, however, must be resolved in semantics. This requirement is implemented by the semantic elementary tree  $(\beta'\text{likes\_and}_{\{DP_B\}})$ , in which the variable corresponding to the object argument ( $x$ ) has been  $\lambda$ -abstracted over, turning the conjunct into a predicate  $\langle\langle e, t \rangle\rangle$ . This predicate must adjoin to another predicate whose object argument has been similarly  $\lambda$ -abstracted over. This adjunction requirement is represented in the elementary tree by the *selective adjunction* or *sa* constraint (Vijay-Shanker, 1992) on the TP node of  $(\alpha\text{hates}_{\{DP_B\}})$ . The *sa* constraint should also provide a list of auxiliary trees, compatible with this elementary tree, or feature specification of compatible auxiliary trees, in order to satisfy the object sharing requirement in the semantic structure. Imposing this *sa* constraint at the TP node ensures parallelism across the conjuncts, only allowing sharing of arguments that have the same grammatical function. To save space in the figures, we show the *sa* constraint but we do not explicitly provide a list of trees or feature specification. In all the subsequent trees we will also provide such an *sa* constraint, and since it serves the same purpose in all of them, we do not comment on it further. The boxed numeral  $\boxed{1}$  in  $(\beta\text{likes\_and}_{\{DP_B\}})$  and  $(\beta'\text{likes\_and}_{\{DP_B\}})$  indicates a link between the syntactic and se-

semantic tree pairs to ensure a synchronous derivation between the syntax and the semantics: a DP tree substitutes into the subject position marked with  $\boxed{1}$  in  $(\beta\text{likes\_and}_{\{DP_B\}})$ , and the semantic tree paired with this DP must substitute into the position marked with  $\boxed{1}$  in  $(\beta'\text{likes\_and}_{\{DP_B\}})$ .

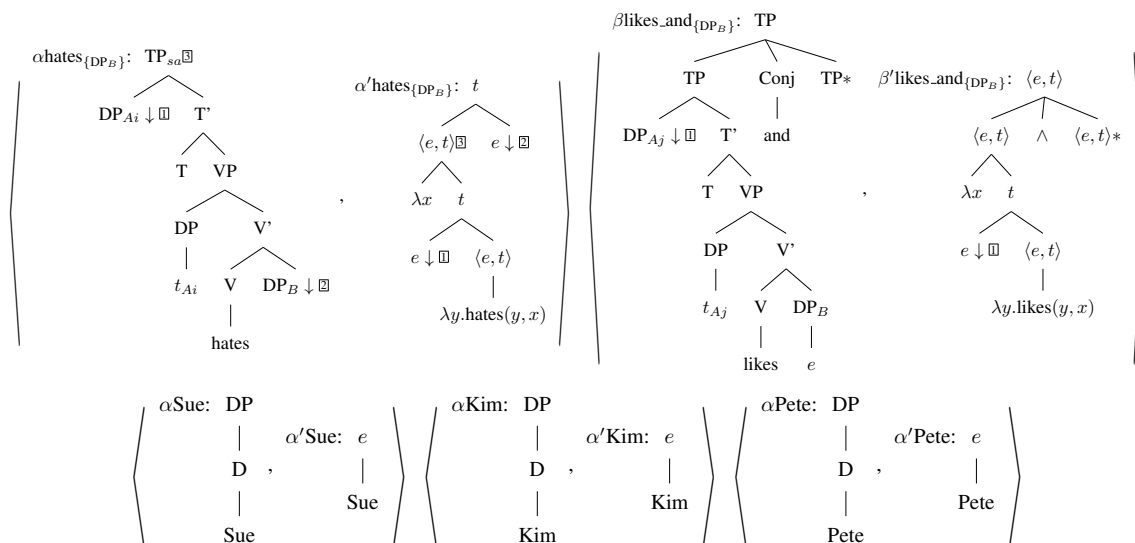


Figure 11: Elementary trees for *Sue likes and Kim hates Pete*

The TP and the predicate that  $(\beta\text{likes\_and}_{\{DP_B\}})$  and  $(\beta'\text{likes\_and}_{\{DP_B\}})$  adjoin to are provided by elementary tree pairs in  $(\alpha\text{hates}_{\{DP_B\}})$  and  $(\alpha'\text{hates}_{\{DP_B\}})$  in Figure 11.  $(\alpha\text{hates}_{\{DP_B\}})$  is a typical transitive initial tree in syntax with subject and object substitution sites.  $(\alpha'\text{hates}_{\{DP_B\}})$ , however, is an atypical transitive elementary tree in semantics in which the object argument has been  $\lambda$ -abstracted over: here, the variable corresponding to the object argument ( $x$ ) is  $\lambda$ -abstracted over to yield a predicate  $(\langle e, t \rangle)$ .<sup>6</sup> This predicate

<sup>6</sup>Semantic elementary trees in which  $\lambda$ -operators abstract over argument variables have been proposed and utilized in Frank and Storoshenko (2012) to handle many difficult cases of quantifier scope within tree-local MC-TAG. In Frank and Storoshenko, each predicative tree is an MC-set with a scope tree containing  $\lambda$ -binders for all arguments, and an argument tree containing the argument variables. The analysis we propose could be recast using a variant of  $\lambda$ -abstracted scope trees of Frank and Storoshenko with the correct hierarchical ordering of  $\lambda$ -binders for object sharing and subject sharing. Here, we opted not to use Frank and Storoshenko's MC-set predicative trees to streamline our exposition and analysis. We would also like to utilize the idea that  $\lambda$ -abstracted arguments generally have wide scope. Since in Frank and Storoshenko, all arguments are  $\lambda$ -abstracted, appeal to such a mechanism would not be available.

will combine with the meaning of the object argument to yield a formula ( $t$ ). Note that the TP node in  $(\alpha\text{hates}_{\{DP_B\}})$  and the highest  $\langle e, t \rangle$  node in  $(\alpha'\text{hates}_{\{DP_B\}})$  are marked with the link [3]. These are the positions onto which  $(\beta\text{likes\_and}_{\{DP_B\}})$  and  $(\beta'\text{likes\_and}_{\{DP_B\}})$  adjoin in syntax and semantics respectively.

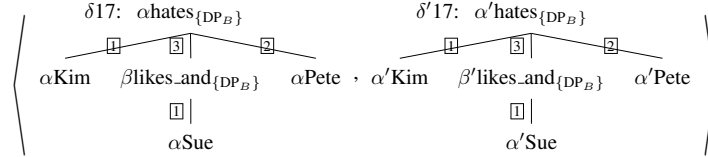


Figure 12: Derivation structures for *Sue likes and Kim hates Pete*

Figure 12 depicts the isomorphic syntactic and semantic derivation structures for (17). The syntactic and the semantic derived trees are given in Figure 13. In contrast to the Conjoin Operation approach, in our analysis,  $(\alpha\text{Pete})$ , the syntactic elementary tree representing the shared argument, composes only with a single predicative elementary tree,  $(\alpha\text{hates}_{\{DP_B\}})$ . In the syntactic derived tree  $(\gamma 17)$ , therefore, *Pete* is represented as the object DP of *hates*, but not *likes*. Similarly, in semantics,  $(\alpha'\text{Pete})$  composes only with  $(\alpha'\text{hates}_{\{DP_B\}})$ . However, because the object abstracted predicate of  $(\beta'\text{likes\_and}_{\{DP_B\}})$  is adjoining onto the predicate node in the object abstracted  $(\alpha'\text{hates}_{\{DP_B\}})$ , the correct meaning of (5) is derived, in which the person Sue likes and Kim hates is Pete, as stated in the logical form in (17a).  $(\gamma'17)$  can be reduced to (17a) via  $\lambda$ -conversion following the application of the Generalized Conjunction (GC) Rule (Barwise and Cooper, 1981) defined in (18).

(18) Generalized Conjunction (GC) Rule:

$$[\text{Pred1} \wedge \text{Pred2}] = \lambda z[\text{Pred1}(z) \wedge \text{Pred2}(z)]$$

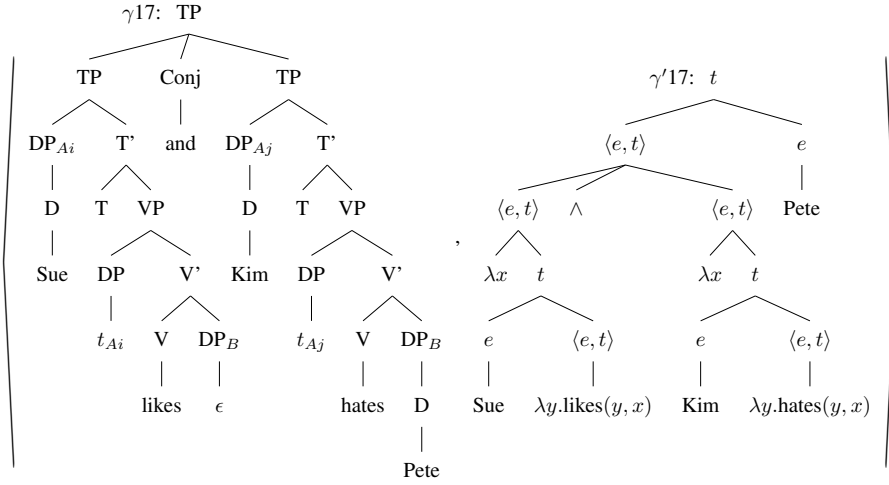


Figure 13: Derived trees for *Sue likes and Kim hates Pete*

### 3.3 Subject argument sharing

Consider the example of subject argument sharing we introduced earlier in (2), repeated here as (19). We illustrate our approach by providing an analysis for this example.

(19) Sue hates Pete and likes Kim.

a.  $\text{hates}(\text{Sue}, \text{Pete}) \wedge \text{likes}(\text{Sue}, \text{Kim})$

Figure 14 contains our proposed elementary trees to derive (19), an example of clausal coordination with a shared subject argument.  $(\beta_{\text{and\_likes}}\{DP_A\})$  introduces a coordinator and its subject argument is null, reflecting the fact that it is absent in the second conjunct. In  $(\beta'_{\text{and\_likes}}\{DP_A\})$ , the variable corresponding to the subject argument ( $x$ ) has been  $\lambda$ -abstracted over, turning the conjunct into a predicate ( $\langle e, t \rangle$ ). This implements the requirement that the subject argument still needs to be saturated.  $(\alpha_{\text{hates}}\{DP_A\})$  is a typical transitive initial tree in syntax. In  $(\alpha'_{\text{hates}}\{DP_A\})$ , however, the variable corresponding to the subject argument ( $x$ ) has been  $\lambda$ -abstracted over to yield a predicate ( $\langle e, t \rangle$ ) which will combine with the meaning of the subject argument to yield a formula ( $t$ ).

The isomorphic syntactic and semantic derivation structures for (19) are provided in

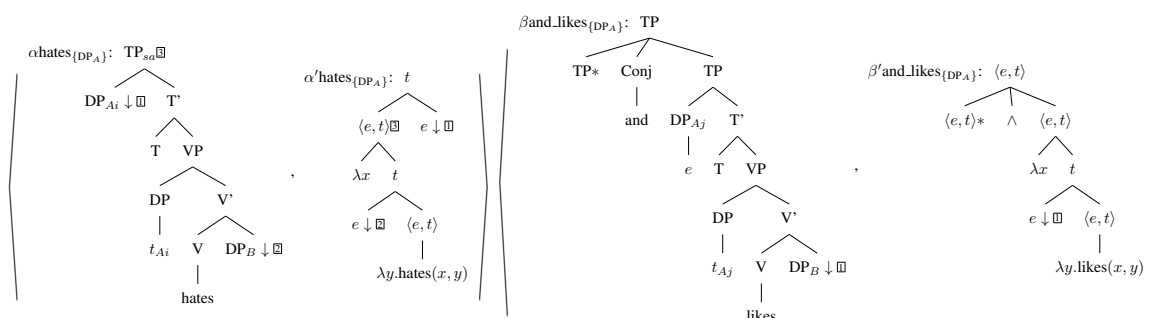


Figure 14: Elementary trees for *Sue hates Pete and likes Kim*

Figure 15 and the derived trees are given in Figure 16. The shared subject argument represented by the elementary tree pair  $\langle \alpha\text{Sue}, \alpha'\text{Sue} \rangle$  composes only with the predicative elementary tree pair  $\langle \alpha\text{hates}_{\{DP_A\}}, \alpha'\text{hates}_{\{DP_A\}} \rangle$ . Therefore, in the syntactic derived tree, *Sue* is represented as the subject DP of *hates*, but not *likes*. In the semantic derived tree, however, the subject abstracted predicate of  $(\beta'\text{and.likes}_{\{DP_A\}})$  adjoins onto the predicate node in the subject abstracted  $(\alpha'\text{hates}_{\{DP_A\}})$ , and so the correct meaning of (19) is derived via the application of  $\lambda$ -conversion and the GC Rule to  $(\gamma'19)$ , in which the person that hates Pete and likes Kim is Sue, as stated in the logical form in (19a).

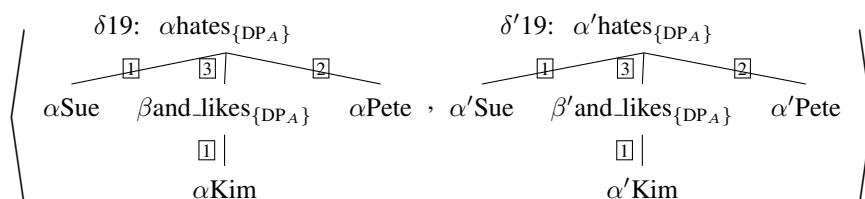


Figure 15: Derivation structures for *Sue hates Pete and likes Kim*

### 3.4 Argument sharing across verbal complement clauses

Argument sharing can take place across verbal complement clauses, as in (20), where an object argument, *Pete*, is shared by two embedded verbal predicates, *remembered* and *forgot*. A standard approach to syntactically deriving a complex sentence such as *Sue thought*

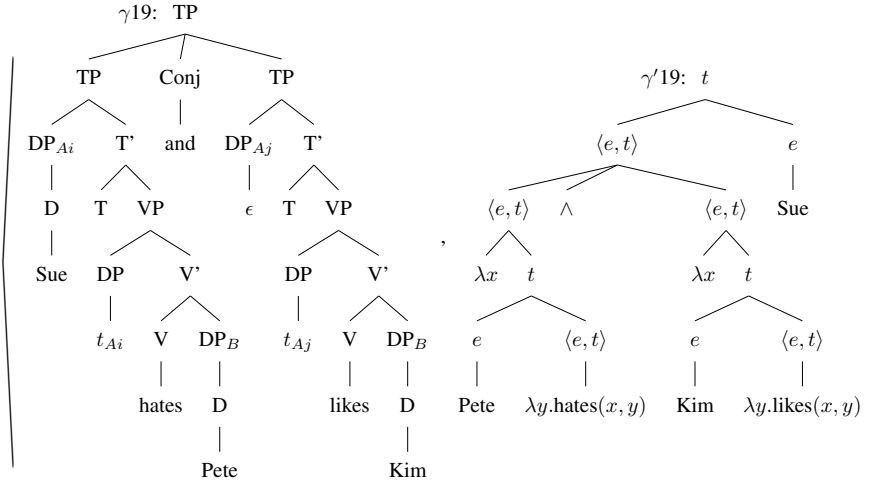


Figure 16: Derived trees for *Sue hates Pete and likes Kim*

that you forgot Pete in TAG is to adjoin an auxiliary tree anchored by the matrix verb *thought* as in  $(\beta\text{thought})$  onto the  $C'$  node of the initial tree anchored by the embedded verb *forgot* as in  $(\alpha\text{forgot}_{\{DP\}})$  in Figure 17 (Kroch and Joshi, 1985; Frank, 2002).<sup>7</sup> We utilize this approach in our analysis of (20).

(20) Sue thought that you remembered and Kim insisted that you forgot Pete.

a.  $\text{thought}(\text{Sue}, \text{remembered}(\text{you}, \text{Pete})) \wedge \text{insisted}(\text{Kim}, \text{forgot}(\text{you}, \text{Pete}))$

The elementary trees required to derive (20) are given in Figure 17. The coordinator is introduced in  $(\beta\text{remembered\_and}_{\{DP_B\}})$ , which contains a null object argument. In  $(\beta'\text{remembered\_and}_{\{DP_B\}})$ , the object argument ( $z_1$ ) has been  $\lambda$ -abstracted over, turning the conjunct into a predicate  $(\langle e, t \rangle)$ .  $(\beta\text{remembered\_and}_{\{DP_B\}})$  adjoins to  $(\alpha\text{forgot}_{\{DP_B\}})$ , and  $(\beta'\text{remembered\_and}_{\{DP_B\}})$  adjoins to  $(\alpha'\text{forgot}_{\{DP_B\}})$ .  $(\alpha\text{forgot}_{\{DP_B\}})$  is an embedded transitive initial tree projected up to CP, and  $(\alpha'\text{forgot}_{\{DP_B\}})$  contains a  $\lambda$ -abstraction of the object argument ( $z_2$ ).  $(\beta\text{insisted})$  is a  $C'$ -rooted auxiliary tree which adjoins to the  $C'$  in

<sup>7</sup>This adjoining approach allows for unbounded dependency, such as in *Who<sub>i</sub> did Sue think that you forgot  $t_i$ ?*, without a long-distance movement of the *wh*-phrase. In TAG, *who* undergoes local movement within the elementary tree anchored by *forgot*, and the local dependency between *who* and its trace is extended by the adjoining of the auxiliary tree anchored by *think*.



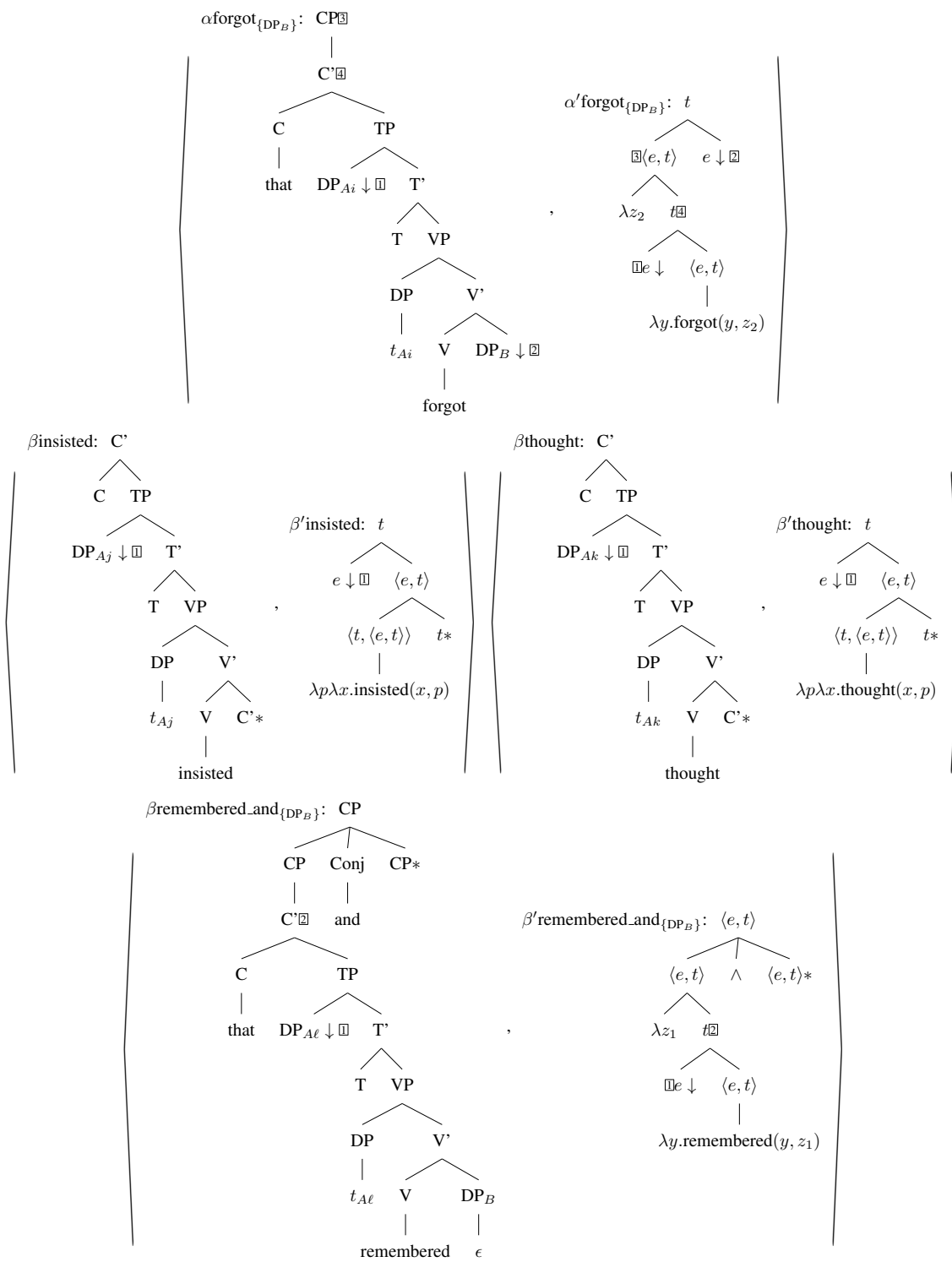


Figure 17: Elementary trees for *Sue thought that you remembered and Kim insisted that you forgot Pete*.

$(\alpha\text{forgot}_{\{DP_B\}})$ , and the corresponding semantic tree,  $(\beta'\text{insisted})$ , is a  $t$ -rooted auxiliary tree which adjoins to the inner  $t$  node of  $(\alpha'\text{forgot}_{\{DP_B\}})$  at link [4]. Similarly,  $(\beta\text{thought})$  is a  $C'$ -rooted auxiliary tree which adjoins to the  $C'$  in  $(\beta\text{remembered\_and}_{\{DP_B\}})$ , and the corresponding semantic tree,  $(\beta'\text{thought})$ , is a  $t$ -rooted auxiliary tree and adjoins to the inner  $t$  node of  $(\beta'\text{remembered\_and}_{\{DP_B\}})$  at link [2].

The full derivation structures and derived trees with all the arguments substituted into appropriate places are given in Figures 18 and 19. In the syntactic derived tree ( $\gamma 20$ ), *Pete* is represented as the object DP of the embedded verb *forgot*, but not *remembered*. In the semantic derived tree, through the adjoining of the semantics of *thought*, the distance between the  $\lambda$ -binder ( $\lambda z_1$ ) and the bindee ( $z_1$ ) has been extended, and similarly, through the adjoining of *insisted*, the distance between  $\lambda z_2$  and  $z_2$  has been extended. The two conjoined  $\langle e, t \rangle$  predicates combine with the single argument *Pete*, via the GC Rule and  $\lambda$ -conversion, to produce the meaning that the person Sue thought that you remembered and the person Kim insisted that you forgot is Pete, as represented in the logical form in (20a).

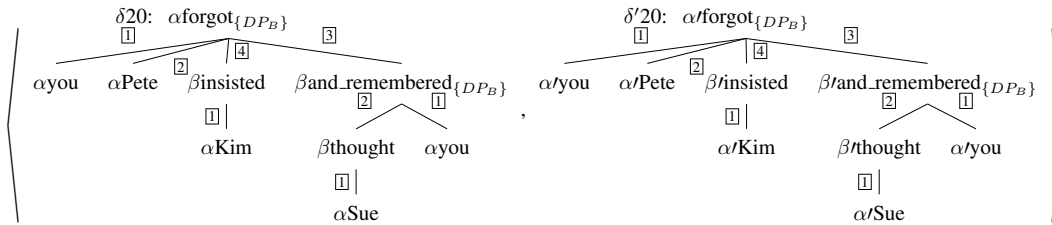


Figure 18: Derivation structures for *Sue thought that you remembered and Kim insisted that you forgot Pete*

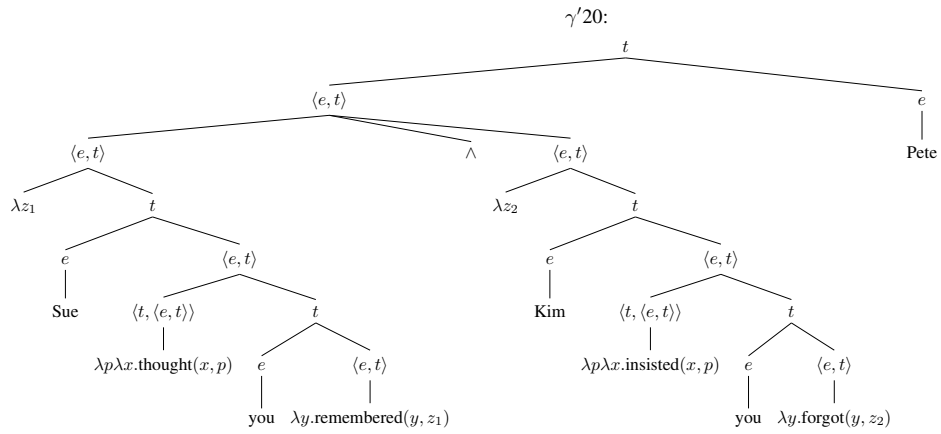
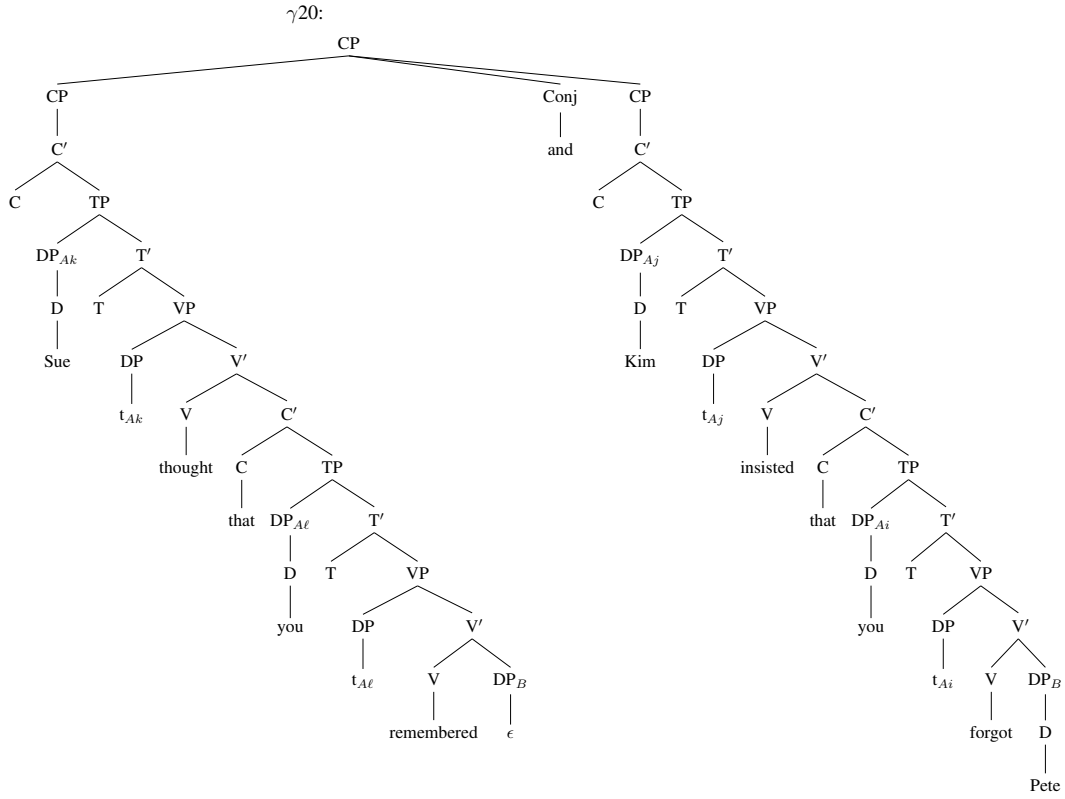


Figure 19: Derived trees for *Sue thought that you remembered and Kim insisted that you forgot Pete*

## 4 Extensions

### 4.1 ATB *wh*-movement

According to the Conjoin Operation analysis, instances of ATB *wh*-movement, as in (21), involve a *wh*-movement in each clausal conjunct followed by identification and merging of the *wh*-phrases as the two clauses compose. In our analysis, a *wh*-movement takes place only in one conjunct in syntax, while the function of the *wh*-phrase as a shared argument is captured in semantics.

(21) Who does Sue like and Kim hate?

a.  $\text{WH}z[\text{person}(z)][\text{likes}(\text{Sue}, z) \wedge \text{hates}(\text{Kim}, z)]$

Additional elementary trees required to derive (21) are given in Figure 20.  $(\alpha\text{wh\_hates}_{\{DP_B\}})$  is a typical transitive initial tree with a *wh*-movement of the object argument.  $(\alpha'\text{wh\_hates}_{\{DP_B\}})$  is a corresponding semantics tree with a  $\lambda$ -abstracted object argument. Here, we abstract away from the full semantics of *wh*-questions and simply represent the predicate-argument structure. In representing the semantics of *who*, we follow the tree-local multi-component treatment of quantification (Shieber and Schabes, 1990; Nesson and Shieber, 2006). We thus propose that the semantics of *who* has two components:  $(\alpha'\text{who})$  is a variable and substitutes into the argument position  $e$  linked with  $\boxed{2}$  in  $(\alpha'\text{wh\_hates}_{\{DP_B\}})$ , and  $(\beta'\text{who})$  represents the scope and adjoins onto  $t$ , again linked with  $\boxed{2}$ , in  $(\alpha'\text{wh\_hates}_{\{DP_B\}})$ . The coordinating auxiliary trees  $(\beta\text{likes\_and}_{\{DP_B\}})$  and  $(\beta'\text{likes\_and}_{\{DP_B\}})$  depicted in Figure 11 will adjoin to the TP node in  $(\alpha\text{wh\_hates}_{\{DP_B\}})$  and the  $\langle e, t \rangle$  node in  $(\alpha'\text{wh\_hates}_{\{DP_B\}})$ , both linked with  $\boxed{3}$ . The full derivation structures and derived trees are given in Figure 21 and Figure 22.

In our analysis, the *wh*-movement of the object argument takes place within the predicative initial tree representing the second conjunct. The coordinating auxiliary tree repre-

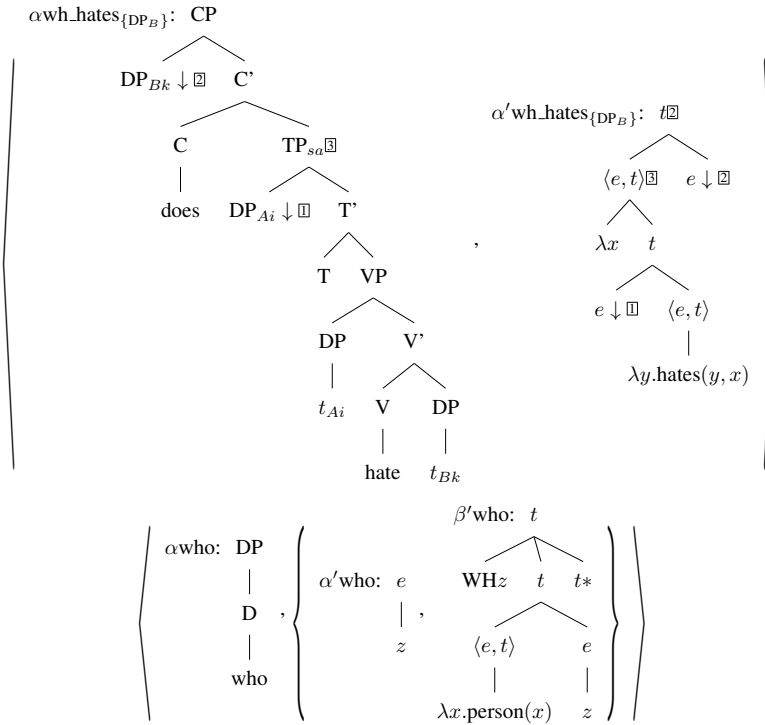


Figure 20: Elementary trees for *Who does Sue like and Kim hate?*

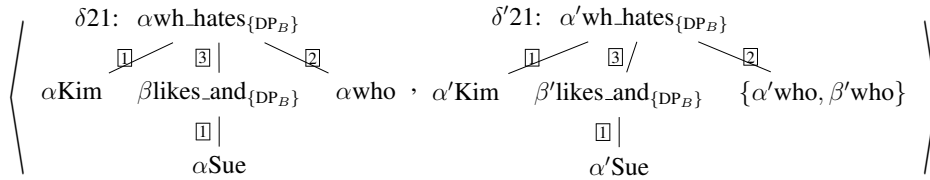


Figure 21: Derivation structures for *Who does Sue like and Kim hate?*

senting the first conjunct adjoins to this tree, stretching the distance between the *wh*-moved DP in [Spec,CP] and the trace position within the VP.<sup>8</sup> Application of  $\lambda$ -conversion and the

<sup>8</sup>A reviewer observes that our analysis predicts that in cases where there is a morphosyntactic cue for *wh*-movement, for example, complementizer agreement in Irish, it would be present only in the second conjunct. Irish has a special *wh*-complementizer,  $a^L$ , that is found in clauses in which *wh*-movement has taken place (McCloskey, 1991). In our analysis, in examples such as (21), we would expect to see  $a^L$  in the C of the derived tree (which comes from the C of the  $(\alpha\text{wh\_hates}_{\{DP_B\}})$  tree), because  $(\beta\text{likes\_and}_{\{DP_B\}})$  adjoins to TP. What would be more telling would be to test this prediction on cases where the cue for *wh*-movement occurs within TPs. We leave this for future research.

The same reviewer also asks how agreement is enforced between the fronted auxiliary verb and the subject in the first conjunct clause, which is adjoined in. In TAG syntax, agreement can be modelled using feature structures (Frank, 2002). In examples such as (21), we must make sure that the subjects in both conjuncts agree with the initial auxiliary verb. This can be done by postulating an Agr feature on the TP

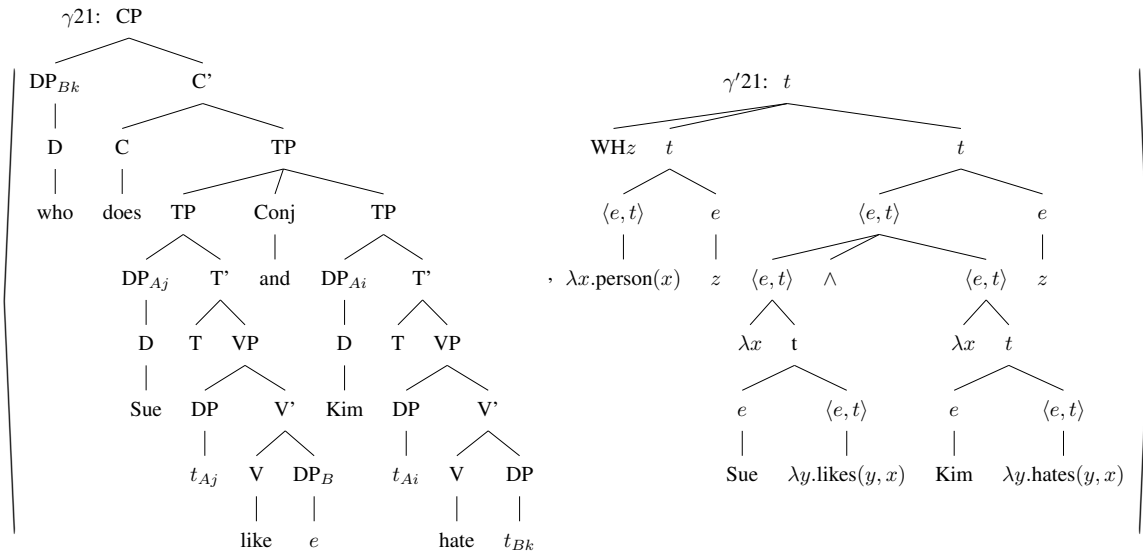


Figure 22: Derived trees for *Who does Sue like and Kim hate?*

GC Rule to ( $\gamma'21$ ) reduces it to the logical form in (21a), which correctly states that the person that Sue likes and Kim hates is the same individual and the question is asking for the identity of this individual.<sup>9</sup>

of  $\alpha_{wh.hates}_{\{DP_B\}}$  that inherits its value from the Agr feature in [Spec,TP] and unifies with the Agr feature in C, and an Agr feature on TP\* of  $\beta_{likes.and}_{\{DP_B\}}$  that inherits its value from the Agr feature in its [Spec,TP]. Agreement between the initial auxiliary verb and the two subjects will take place by enforcing feature unification among these Agr features.

<sup>9</sup>A reviewer asks if our analysis of ATB *wh*-movement has any implications for acceptable cases of Coordinate Structure Constraint (CSC) violations, as in (i) and (ii).

- (i) The advice, the committee decided to follow and proceeded to set up a new subcommittee. (Heycock and Kroch, 1994)
- (ii) How much can you drink and still stay sober? (Kehler, 1996)

These examples have extractions from the first conjuncts only and sharing of subject arguments. They are thus different from our ATB *wh*-movement examples, in which the extracted *wh*-phrase is also the shared argument. A possible analysis of these apparent CSC violations within the general approach presented here is that the first conjunct is represented by an elementary tree with topicalization or *wh*-movement in syntax and an elementary tree with  $\lambda$ -abstraction of the shared argument in semantics, and the second conjunct is represented by a coordinate auxiliary tree with null subject argument in syntax and a parallel  $\lambda$ -abstracted auxiliary tree in semantics.

## 4.2 Quantification and coordination

In clausal coordination with shared arguments, in general, the shared arguments scope over the coordinator, and the non-shared arguments scope under the coordinator (Banik, 2004; Han et al., 2008). This is illustrated in (22) (repeated from (9)) for a subject shared argument, and (23) for an object shared argument. In addition, clausal coordination with multiple shared arguments, as in (24), exhibits scope ambiguity. All three examples are taken from Han et al. (2008).

(22) A student read every paper and summarized every book.  $(\exists > \wedge > \forall)$

a.  $\exists x_1[\text{student}(x_1)][\forall x_2[\text{paper}(x_2)][\text{read}(x_1, x_2)] \wedge \forall x_2[\text{book}(x_2)][\text{summarized}(x_1, x_2)]]]$

(23) A student takes and a professor teaches every course.  $(\forall > \wedge > \exists)$

a.  $\forall x_2[\text{course}(x_2)][\exists x_1[\text{student}(x_1)][\text{takes}(x_1, x_2)] \wedge \exists x_1[\text{professor}(x_1)][\text{teaches}(x_1, x_2)]]]$

(24) A student likes and takes every course.  $(\exists > \forall > \wedge, \forall > \exists > \wedge)$

a.  $\exists x_1[\text{student}(x_1)][\forall x_2[\text{course}(x_2)][\text{likes}(x_1, x_2) \wedge \text{takes}(x_1, x_2)]]]$

b.  $\forall x_2[\text{course}(x_2)][\exists x_1[\text{student}(x_1)][\text{likes}(x_1, x_2) \wedge \text{takes}(x_1, x_2)]]]$

In Han et al. (2008), semantic derivation of examples such as (22)-(24) requires a composition of an initial predicative tree and a coordinating auxiliary tree, each with a contraction node representing the shared argument. These elementary trees both project to  $t$ . The wide scope of the shared argument is enforced by stipulating that the scope component of the contraction node is active only in the coordinating auxiliary tree, which adjoins onto the highest  $t$  above the coordinator. The scope information of the contraction node in the initial predicative tree is inherited from the scope component of the contraction node in the coordinating auxiliary tree. In our analysis, the shared argument is present only in one of the conjuncts, and so a single scope component straightforwardly interacts with the coordinator as the coordinating auxiliary tree adjoins below the scope of the shared argument.

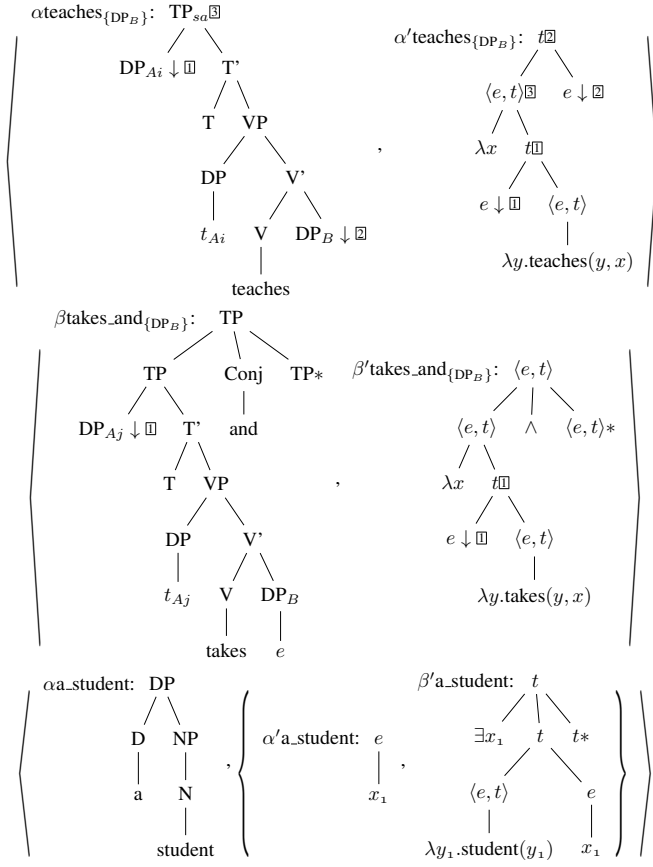


Figure 23: Elementary trees for *A student takes and a professor teaches every course*

We use (23) to illustrate our analysis with a single shared argument and discuss (24) to illustrate how our analysis can be extended to multiple shared arguments. Additional elementary trees needed to derive (23) are given in Figure 23. We represent the semantics of quantified nominal phrases as MC sets, as we did for the semantics of *who*. For example, for the semantics of *a student*, ( $\alpha'$ a.student) provides the argument variable, and ( $\beta'$ a.student) introduces the existential quantifier and provides the scope of the quantifica-

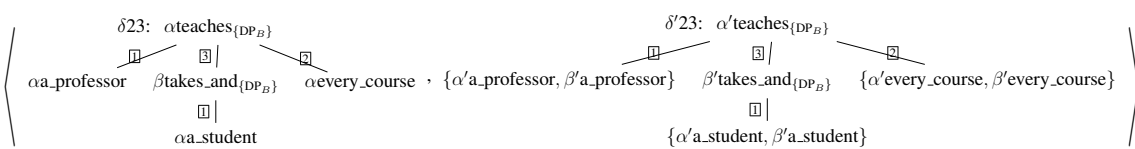


Figure 24: Derivation structures for *A student takes and a professor teaches every course*



tion. We will utilize similar elementary tree pairs for *a professor* and *every course*, with the difference that the elementary tree representing the scope component of *every course* will contain a universal quantifier in place of an existential quantifier. The elementary tree pairs  $\langle \alpha \text{teaches}_{\{DP_B\}}, \alpha' \text{teaches}_{\{DP_B\}} \rangle$  and  $\langle \beta \text{takes\_and}_{\{DP_B\}}, \beta' \text{takes\_and}_{\{DP_B\}} \rangle$  are similar to the predicative initial tree and the coordinating auxiliary tree we have seen before in Figure 11. The only difference is that  $(\alpha' \text{teaches}_{\{DP_B\}})$  and  $(\beta' \text{takes\_and}_{\{DP_B\}})$  are now augmented with links to accommodate the scope components of the quantified noun phrases. In  $(\beta' \text{takes\_and}_{\{DP_B\}})$ , the link  $\boxed{1}$  for the scope component of the subject DP is on  $t$ , which is below the coordinator. In  $(\alpha' \text{teaches}_{\{DP_B\}})$ , the link  $\boxed{1}$  for the scope component of the subject DP is on the  $t$  below the  $\langle e, t \rangle$  node onto which the coordinating auxiliary tree adjoins. Together, the non-shared arguments in each conjunct are guaranteed to scope below the coordinator. Moreover, in  $(\alpha' \text{teaches}_{\{DP_B\}})$ , the scope component of the object DP, which is the shared argument, is linked to the highest  $t$  above the  $\langle e, t \rangle$  node onto which the coordinating auxiliary tree adjoins. This then ensures that the shared argument scopes over the coordinator.<sup>10</sup>

The isomorphic syntactic and semantic derivation structures are given in Figure 24 and the derived trees are given in Figure 25. To save space, we have reduced the restrictor  $t$  node of each quantifier in the semantic derived tree ( $\gamma'23$ ). Application of the GC Rule and  $\lambda$ -conversion to ( $\gamma'23$ ) further reduces it to the logical form in (23a).

To derive the clausal coordination with subject and object shared arguments in (24), elementary tree pairs for *likes* and *takes* consistent with our proposal are provided in Figure 26.  $(\alpha \text{takes}_{\{DP_A, DP_B\}})$  is an initial predicative tree with an empty DP position for the subject and a DP substitution site for the object, and is paired with  $(\alpha' \text{takes}_{\{DP_A, DP_B\}})$ , which has

---

<sup>10</sup>A reviewer asks what forces parallelism in scope across the clauses in examples such as (23). We are assuming that the  $\lambda$ -abstracted arguments are constrained to have wide scope over the arguments that have not been  $\lambda$ -abstracted. With this assumption, adjoining  $(\beta' \text{takes\_and}_{\{DP_B\}})$  lower than the scope link for the  $\lambda$ -abstracted argument in  $(\alpha' \text{teaches}_{\{DP_B\}})$  will guarantee that the shared argument has scope over the non-shared arguments in both conjuncts.

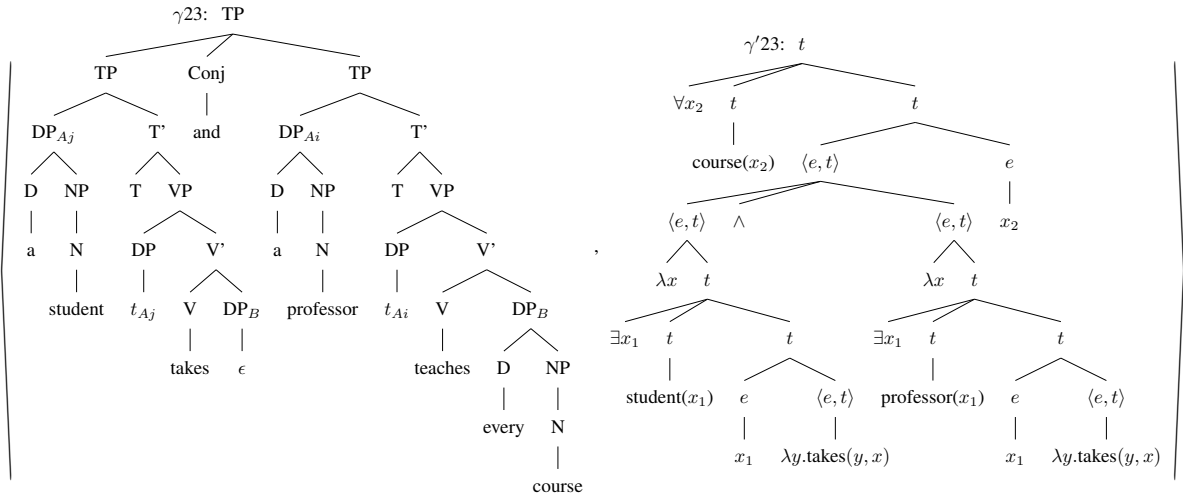


Figure 25: Derived trees for *A student takes and a professor teaches every course*.

$e$  substitution sites for the subject and the object argument variables. Note that the  $t$  node of  $(\alpha'takes_{\{DP_A, DP_B\}})$  has multiple links,  $\boxed{1}$  and  $\boxed{2}$ , for the scope components of the subject and the object DPs. This indicates that the two scope component trees will multiply-adjoin to the  $t$  node, as defined in Schabes and Shieber (1994). This predicts scope ambiguity, as the order in which the two trees adjoin is not specified.  $(\beta'likes\_and_{\{DP_A, DP_B\}})$  is a coordinating auxiliary tree with an empty DP position for the object and a DP substitution site for the subject, and is paired with an MC set in semantics that includes an auxiliary tree recursive on  $\langle e, \langle e, t \rangle \rangle$ , a type  $e$  tree introducing a variable to be substituted into an object argument variable position in  $(\alpha'takes_{\{DP_A, DP_B\}})$ , and an auxiliary tree recursive on  $t$  binding the variable and providing a substitution site for the subject argument variable and a link  $\boxed{1}$  on  $t$  for the scope component of the subject argument.

The isomorphic derivation structures given in Figure 27 yield the syntactic derived tree in Figure 28. In semantics,  $(\alpha'a\_student)$  substitutes into  $(\alpha'likes\_and_{\{DP_A, DP_B\}_2})$ , providing the subject argument variable, and  $(\beta'a\_student)$ , the scope component of the subject DP, adjoins onto the  $t$  node of  $(\beta'likes\_and_{\{DP_A, DP_B\}_2})$ .  $(\beta'likes\_and_{\{DP_A, DP_B\}_1})$  adjoins to  $(\alpha'takes_{\{DP_A, DP_B\}})$  at  $\langle e, \langle e, t \rangle \rangle$  with link  $\boxed{1}$ ,  $(\beta'likes\_and_{\{DP_A, DP_B\}_3})$  substitutes into

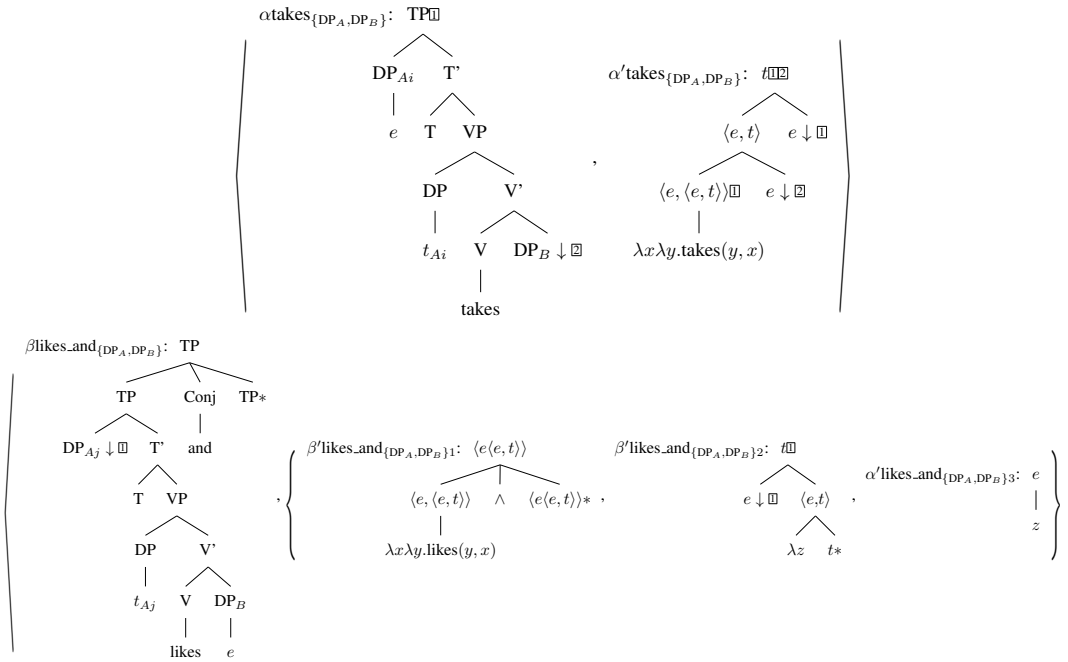


Figure 26: Elementary trees for *A student likes and takes every course*

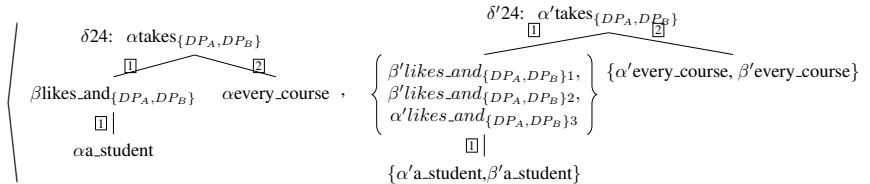


Figure 27: Derivation structures for *A student likes and takes every course*

$(\alpha' \text{takes}_{\{DP_A, DP_B\}})$  at  $e$  with  $\boxed{1}$ , and  $(\beta' \text{likes\_and}_{\{DP_A, DP_B\}_3})$  adjoins to  $(\alpha' \text{takes}_{\{DP_A, DP_B\}})$  at  $t$  with link  $\boxed{1}$ . Note that this is a tree-local derivation as all elementary trees in an MC set are combining with a single elementary tree.  $(\beta' \text{likes\_and}_{\{DP_A, DP_B\}_2})$  that carries the scope component of the subject DP, and  $(\beta' \text{every\_course})$ , the scope component of the object DP, multiply adjoin onto the  $t$  node of  $(\alpha' \text{takes}_{\{DP_A, DP_B\}})$ . The two possible orderings of adjoining the two scope components allows for scope ambiguity between the subject and the object quantifiers. Regardless of the scope between the subject and the object quantifiers, the coordinator scopes below the two quantifiers, as  $(\beta' \text{likes\_and}_{\{DP_A, DP_B\}_1})$ , which

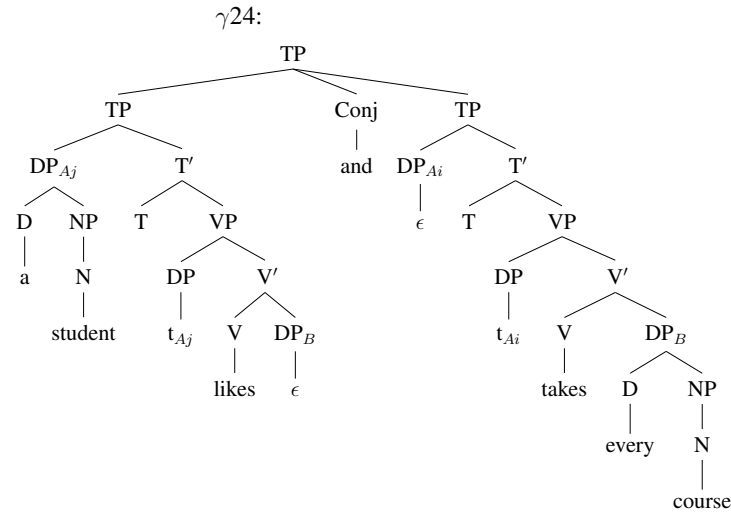


Figure 28: Syntactic derived tree for *A student likes and takes every course*

introduces the coordinator, adjoins to  $(\alpha' \text{takes}_{\{DP_A, DP_B\}})$  at  $\langle e, \langle e, t \rangle \rangle$ , below the  $t$  node to which the scope components of the two quantifiers adjoin. The two semantic derived trees representing the two scopal relation between the subject and the object quantifiers are provided in Figure 29. Application of the GC Rule and  $\lambda$ -conversion to ( $\gamma'$ 24a) and ( $\gamma'$ 24b) further reduces them to the logical forms in (24a) and (24b), respectively.<sup>11</sup>

<sup>11</sup>A reviewer points out that in clausal coordinate sentences with quantifiers, the interpretation is available in which the coordinator has the widest scope, and a universal, as a shared or a non-shared argument, scopes over an existential, as in (i) and (ii).

- (i) A different student read every paper and summarized every book.  $(\wedge > \forall > \exists)$
- (ii) A different professor teaches and a different graduate student grades for each course.  $(\wedge > \forall > \exists)$

The analysis we present here for shared arguments does not account for these scopal interpretations. We think that sentences such as (i) and (ii) do not actually contain shared arguments, but are rather instances of argument ellipsis (Ha, 2008; Barros and Vicente, 2011). For instance, in (i), the subject argument *a different student* is present in syntax and in semantics in the second conjunct, but has been elided for the purposes of pronouncing the sentence. Similarly in (ii), *each course* is present in syntax and in semantics in the first conjunct, but has been elided in pronouncing the sentence. Under this approach, the subject and the object quantifiers in each conjunct would be expected to interact within it. The view that some apparent cases of argument sharing should be given an ellipsis analysis have been proposed by Barros and Vicente (2011).

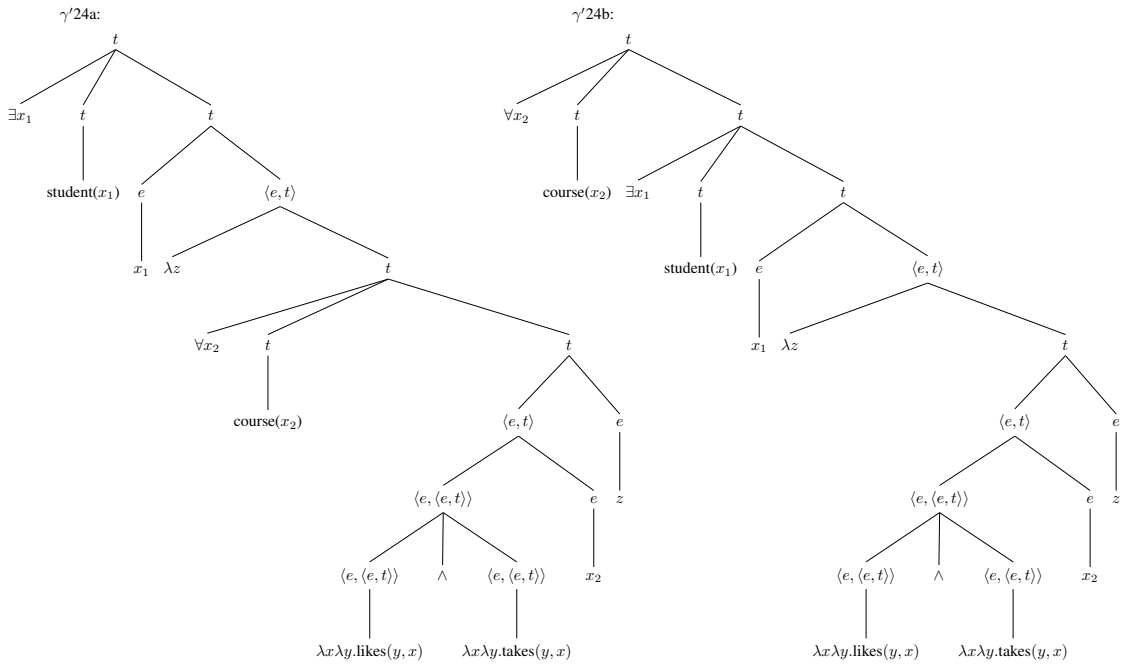


Figure 29: Semantic derived trees for *A student likes and takes every course*

### 4.3 Argument sharing across islands

In clausal coordination, arguments may be shared out of structures which are otherwise considered syntactic islands. This is illustrated in (25) (repeated from (6)) and (26), where the shared argument is embedded in a relative clause within one of the clausal conjuncts.

- (25) Max publicly denounced the senator who wrote and Pauline outwardly criticized the magazine editor who published [**the speech that encouraged the riot**]. (Sabbagh, 2014, 15)
- (26) Max denounced the senator who harassed and Paul criticized the editor who punched [**John**].

According to the ATB movement analysis reviewed in section 1, argument sharing involves the movement of identical arguments out of both conjuncts. Yet in (25) and (26), such movement would violate syntactic island constraints (Wexler and Culicover, 1980).

By contrast, our analysis of argument sharing does not posit any movement operations: rather, the shared argument is taken to be syntactically present in only one conjunct, and  $\lambda$ -abstraction is used to distribute it across both clauses in the semantics. As such, our analysis can be extended to account for argument sharing out of islands without violating movement constraints. We provide an analysis for (26) to demonstrate the method.

Figure 30 depicts the elementary trees needed to derive (26); these have been adapted from the STAG analysis of relative clauses due to Han (2007).<sup>12</sup>  $(\beta\text{harassed}_{\{DP_B\}})$  and  $(\beta\text{punched}_{\{DP_B\}})$  represent subject gap relative clauses anchored by the lexical heads *harassed* and *punched*, respectively. Subject relativization is represented by the raising of the subject DP substitution site, into which an instance of  $(\alpha\text{who})$  will substitute during the derivation. As the relative clausal trees are recursive on NP, they may adjoin at the NP nodes in  $(\alpha\text{the\_senator})$  and  $(\alpha\text{the\_editor})$ , respectively, to form the two complex DPs found in (26). Semantically, the relative clauses are represented by auxiliary trees anchored by the two-place predicates *harassed* and *punched*, depicted in  $(\beta'\text{harassed}_{\{DP_B\}_1})$  and  $(\beta'\text{punched}_{\{DP_B\}_1})$ . By adjoining the generalized quantifier  $(\beta'\text{who})$  at the  $\langle e, t \rangle$  nodes labeled by  $\boxed{1}$ , we force the relative clauses and their respective head nouns to predicate over the same variable. As the relative clausal trees are recursive on  $\langle e, t \rangle$ , they may adjoin to  $(\beta'\text{the\_senator})$  and  $(\beta'\text{the\_editor})$  at the nodes labeled by  $\boxed{1}$ , yielding a final reading where these nouns head their respective relative clauses.

We now account for the object argument *John* which is shared across both relative clauses. Following our treatment of basic argument sharing, we posit that  $(\alpha\text{John})$  substitutes into the relative clause  $(\beta\text{punched}_{\{DP_B\}})$  where it surfaces, while the syntactic object argument of  $(\beta\text{harassed}_{\{DP_B\}})$  is null to reflect the lack of an overt object argument in the first conjunct. The content of this null argument must be resolved in the semantics.

---

<sup>12</sup>To simplify the trees, we show subjects originating in [Spec,TP]; this should not be taken as a rejection of the VP-internal subject hypothesis.

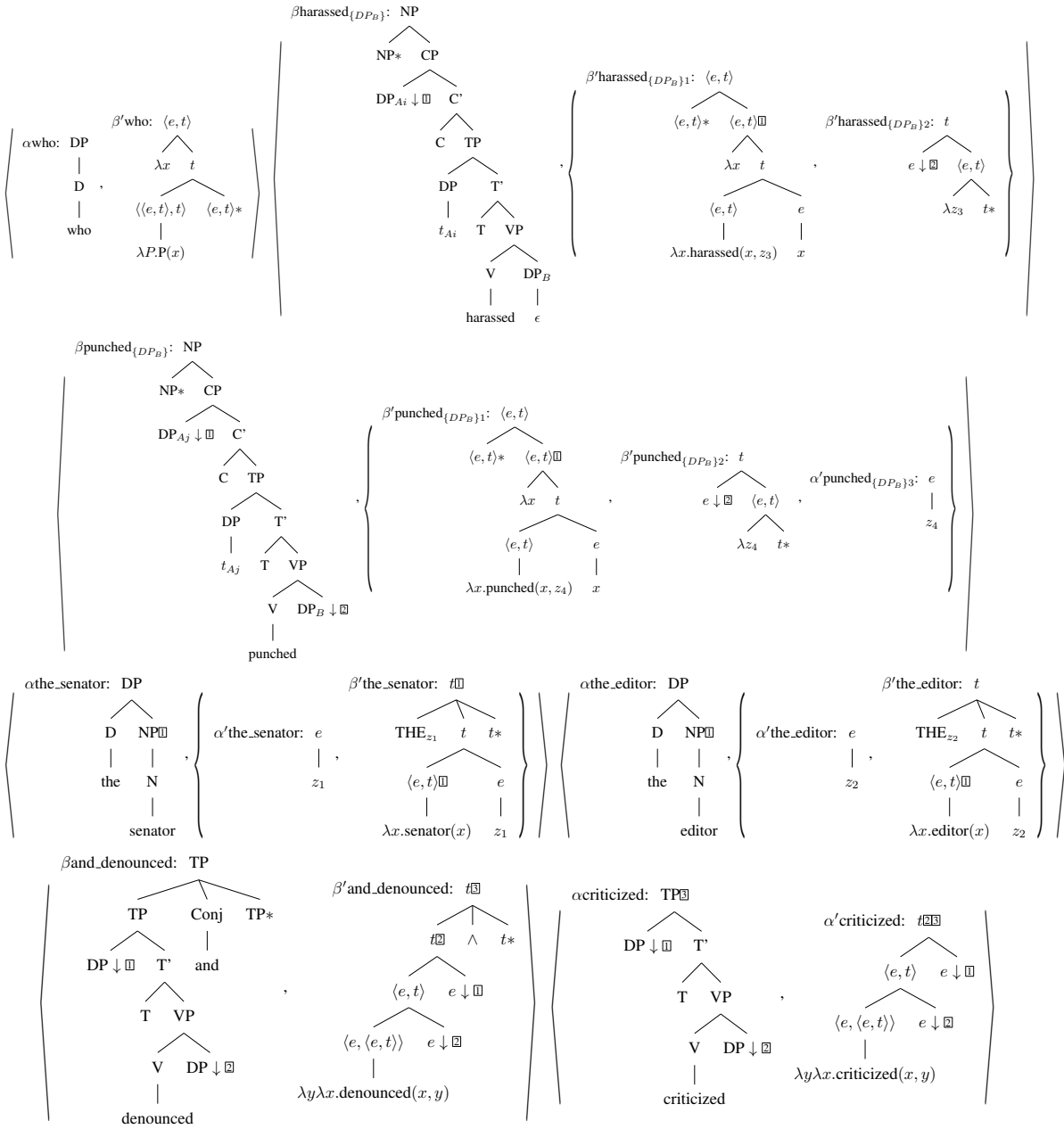


Figure 30: Selected elementary trees for *Max denounced the senator who harassed and Paul criticized the editor who punched John*

We achieve this by expanding the semantic representation of relative clauses in STAG. We posit that the semantic representation of a relative clause with argument sharing is an MC set which participates in a delayed tree-local derivation, as defined in [Chiang and Scheffler \(2008\)](#). According to Chiang and Scheffler, delayed tree-locality allows other elementary trees to intervene in between the elements of an MC set and the combination with a single elementary tree (as required by strict tree-locality). Locality is preserved by requiring the existence of a destination node, which they define as the lowest node in the derivation structure which dominates all members of the MC set.<sup>13</sup> The derivation is then local in the sense that all members of the MC set eventually combine with this destination tree, but this locality is ‘delayed’ as other trees may intervene in the derivation structure.

In our analysis, the MC set for *harassed* contains  $(\beta'harassed_{\{DP_B\}1})$  in which the object argument slot has been filled by the variable  $z_3$ . Similarly, the MC set for *punched* contains  $(\beta'punched_{\{DP_B\}1})$  in which the object argument slot has been filled by the variable  $z_4$ .  $z_3$  and  $z_4$  are bound by  $\lambda$ -terms in  $(\beta'harassed_{\{DP_B\}2})$  and  $(\beta'punched_{\{DP_B\}2})$  respectively, which must scope over these variables in the final derived tree in order for the result to be interpretable. To enforce the requirement that  $z_3$  and  $z_4$  resolve to the same shared argument entity, the tree  $(\alpha'punched_{\{DP_B\}3})$  contains a copy of  $z_4$  which substitutes into the node linked with  $\boxed{2}$  in  $(\beta'harassed_{\{DP_B\}2})$ . From this position, it identifies with  $z_3$  via  $\lambda$ -conversion, ensuring that  $z_3$  and  $z_4$  refer to the same individual. The  $\lambda$ -term which binds  $z_3$  is located in  $(\beta'harassed_{\{DP_B\}2})$ , which adjoins to the root  $t$  node of  $(\beta'the\_senator)$  where it correctly scopes over all instances of  $z_3$  in the tree. Similarly,  $(\beta'punched_{\{DP_B\}2})$  contains the  $\lambda$ -term which binds  $z_4$ ; this tree adjoins at the root of the coordinating auxiliary tree  $(\beta'denounced\_and)$ . In this way,  $\lambda z_4$  ends up near the root of the final derived tree, so that there are no unbound instances of  $z_4$  in the final tree. Finally, substitution of

---

<sup>13</sup>In effect, this constraint merely requires all members of the MC set to be present in the derivation: the root node of the derivation structure trivially dominates all other nodes in the derivation tree, and by extension dominates all members of the set in question.



( $\alpha'$ John) into the  $e$  node linked with  $\boxed{2}$  in ( $\beta'$ punched $_{\{DP_B\}2}$ ) saturates  $\lambda z_4$  with the shared object *John*, yielding the desired result that  $z_4$  (and by extension  $z_3$ ) resolves to *John*. The final interpretation is thus one in which John is both the individual being harassed and the individual being punched.

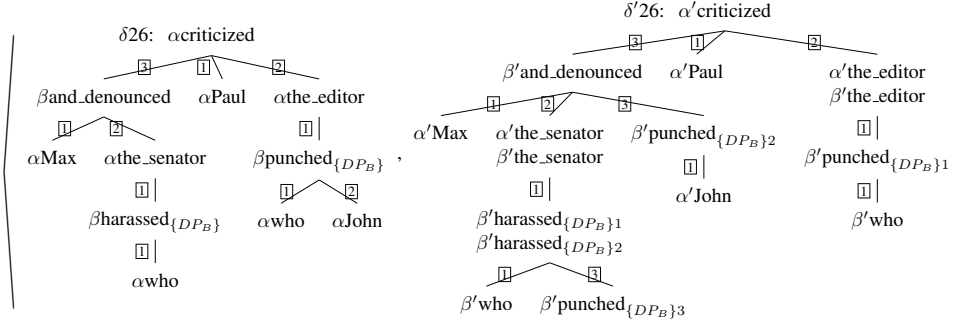


Figure 31: Derivation structures for *Max denounced the senator who harassed and Paul criticized the editor who punched John*

The full derivation structures and derived trees are given in Figure 31 and Figure 32, respectively, and the final  $\lambda$ -converted logical form is given in (27):

$$(27) \quad \text{THE}_{z_1}[\text{senator}(z_1) \wedge \text{harassed}(z_1, \text{John})][\text{denounced}(\text{Max}, z_1)] \wedge \\ \text{THE}_{z_2}[\text{editor}(z_2) \wedge \text{punched}(z_2, \text{John})][\text{criticized}(\text{Paul}, z_2)]$$

The semantic derivation in Figure 31 satisfies Chiang and Scheffler’s definition of delayed tree-locality, in that while all three members of *punched* initially compose with different elementary trees, they are dominated by the root node, ( $\alpha'$ criticized). According to Chiang and Scheffler, the delay of an MC set is defined as the union of all of the nodes along paths from the trees in that set to their destination node (excluding the destination itself). A derivation is  $k$ -delayed just in case no node is a member of more than  $k$  delays. The semantic derivation in Figure 31 is therefore 2-delayed, because ( $\alpha'$ the\_senator) is a member of the delay for *the\_senator* and the delay for *punched*, and also ( $\beta'$ harassed $_{\{DP_B\}2}$ ) is a member of the delay for *harassed* and the delay for *punched*. Chiang and Scheffler (2008)

prove that every  $k$ -delayed MC-TAG with finitely-bounded  $k$  is weakly equivalent to some TAG, so our use of delayed locality does not increase the weak generative capacity of the STAG formalism which we employ.

In this worked out example of an object argument from two relative clauses shared by coordination, we chose to use delayed locality only in the semantic derivation structure. The same delayed locality can be imposed on the syntactic derivation as well by creating multi-component syntax tree sets with the extra trees being degenerate (trees with a single node that is both a root node and a foot node) which are used as a placeholder in the derivation structure. This would ensure that the syntax and semantic derivation structures are isomorphic (Shieber, 1994). We chose not to use delayed locality in the syntax trees to keep the derivation simpler and easier to understand for the reader.

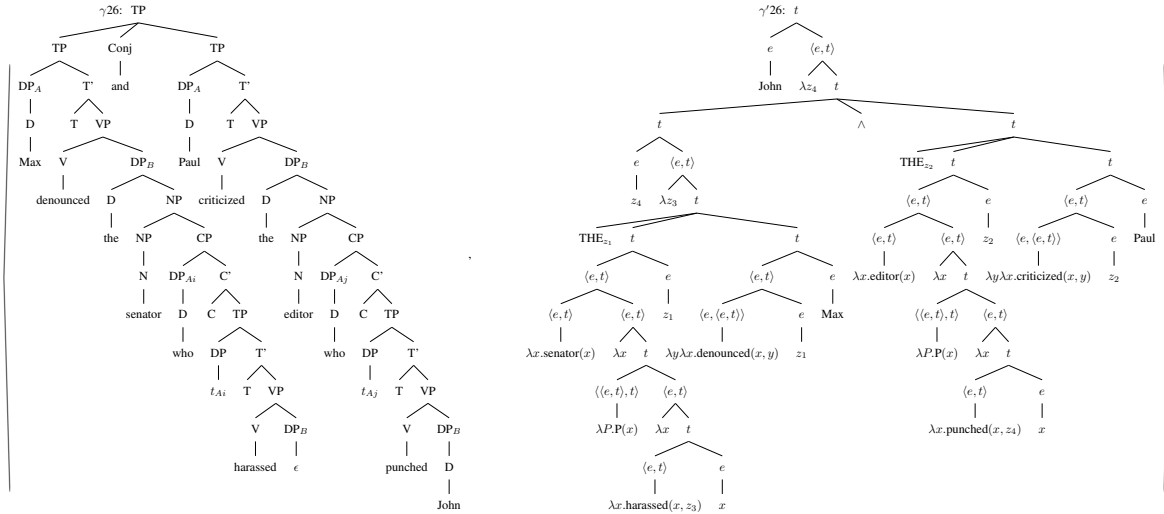


Figure 32: Derived trees for *Max denounced the senator who harassed and Paul criticized the editor who punched John*

While the use of delayed tree-locality in semantics handles argument sharing across islands, we would not want to allow delayed locality more generally in the grammar of English. As pointed out by a reviewer, this would then incorrectly predict that other semantic dependencies, such as quantifier scope, would be insensitive to islands. To regulate

the use of delayed locality in the grammar, we follow [Storoshenko and Han \(2015\)](#) in constraining each MC set with a *delay length*, which is defined to be the cardinality of the delay minus the cardinality of the MC set. Thus, they propose that a generalized quantifier MC sets are constrained to have a delay length of zero, meaning that they have to be in a tree-local derivation. This captures the fact that quantifier scope is generally clause-bound in English. Moreover, [Storoshenko and Han \(2015\)](#) postulate an MC set analysis for bound variable pronouns, and propose that this MC set for English is constrained to be greater than one to account for the fact that the dependency between a quantifier and its bound pronoun in English is non-local, as illustrated in (28). Along these lines, we can say that the delay length for MC sets for clauses with shared arguments is constrained to be equal or greater than zero, allowing for tree-local as well as delayed tree-local derivation in this case.

- (28) a. \* Every girl<sub>i</sub> loves her<sub>i</sub>.  
b. Every girl<sub>i</sub> believes that she<sub>i</sub> is intelligent.

Crucially, these delay length constraints are defined with respect to individual lexical items within a given language, rather than as a global constraint across all derivations involving an MC set within a given language. Hence, delay length constraints can be used to restrict the use of delayed locality in the given grammar.

## 5 Conclusion

We have proposed a Synchronous TAG analysis of clausal coordination with shared arguments which utilizes only the standard TAG operations, substitution and adjoining, without relying on the Conjoin Operation. Therefore, we do not require modified parsing algorithms to handle the Conjoin Operation, unrooted trees, or tree nodes with multiple parents as in [Sarkar and Joshi \(1996\)](#). In our analysis, the shared argument is present syntactically

only in the single conjunct in which it appears on the surface. In the semantics, the conjunct with a missing argument is represented as a predicate with an unsaturated argument, which adjoins onto a predicate node that has been  $\lambda$ -abstracted over by the shared argument. We take advantage of TAG's extended domain of locality in both the syntax and semantics, and use a synchronized formalism to simulate movement of the semantic shared argument via  $\lambda$ -abstraction, without any actual movement in the syntax. We also exploit the extended domain of locality to use unreduced  $\lambda$ -terms in the semantic trees; these are modified using substitution and adjunction during the derivation to handle complex dependencies in the final reduced logical form. When the scope of the shared argument extends beyond a syntactic island, such as a relative clause, we model this using the delayed tree-local extension to TAG. In our analysis, the shared argument therefore extends its scope in the semantics without participating in movement, ellipsis or multiple-dominance in the syntax, eschewing the incorrect predictions made by these previous approaches.

Challenge to our analysis would be data where the interpretation of the shared argument does not seem to be identical across both conjuncts, as observed by a reviewer. We thank the reviewer for pointing out the examples in (29)-(32) from the literature.

(29) Fred spent and Mia lost a total of \$10,000. (cumulative reading, [Chaves, 2012](#))

(30) Alice composed and Beatrix performed different songs.

(internal reading of *different*, [Barros and Vicente, 2011](#))

(31) Where did Mary vacation and Bill decide to live? (pair-list reading, [Munn, 1999](#))

(32) John likes but Bill hates his father. (sloppy reading of *his*, [Ha, 2008](#))

In (29), the shared argument *a total of \$10,000* is the sum of the money Fred spent and Mia lost. To generate this cumulative reading, our analysis would have to be augmented with Chaves' proposal to use plurality-forming cumulation operator in the semantics of *total*. (30) has an internal reading where Alice's songs are different from Beatrix's. [Barros](#)

and Vicente (2011) argue that the internal reading can be generated through a multiple dominance analysis with *different songs* taking scope over the conjunction. Although we do not make use of multiple dominance approach, in our analysis, *different songs* would take scope over the conjunction in semantics via  $\lambda$ -abstraction. So, perhaps with the addition of distributivity associated with the semantics of *different*, the internal reading should be possible with our general approach. (31) has a pair-list reading in which the answer can be *Mary vacationed in Paris and Bill decided to live in Toronto*. According to Munn, this is an instance of ATB-movement of a functional *wh*-phrase. The functional trace has an argument index subject to sloppy identity, represented as a superscript on each trace in (33). Here, the argument index of the functional *wh*-trace refers to *Mary* in the first conjunct and *Bill* in the second conjunct. In our analysis, this pair-list reading can be generated with the addition of semantics for functional *wh*-questions and corresponding functional variables.

(33) Where<sub>*i*</sub> did Mary<sub>*x*</sub> vacation  $t_i^x$  and Bill<sub>*y*</sub> decide to live  $t_i^y$ ?

Lastly, (32) has a sloppy identity reading of *his* and the sentence can be paraphrased as *John likes John's father and Bill hates Bill's father*. According to Ha (2008) and Barros and Vicente (2011), examples such as (32) are instances of argument ellipsis, not argument sharing. This means that *his father* is active in each conjunct both syntactically and semantically, but only pronounced in the second conjunct. Then, our analysis of argument sharing will not apply to such cases.

In sum, although details need to be worked out, many of these challenging cases can be addressed with an analysis compatible with our proposed approach. Specific semantic analysis of cumulativity, distributivity, functional *wh*-questions and ellipsis within the STAG framework will have to wait for future research.

## Funding

This work was supported by Social Sciences and Humanities Research Council of Canada [SSHRC 435-2014-0161] to Han and Natural Sciences and Engineering Research Council of Canada [NSERC RGPIN 262313, RGPAS 446348] to Sarkar

## Acknowledgments

We are extremely indebted to the two anonymous reviewers for their insightful comments that were crucial in improving this paper. We also thank the audience at the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms at Umeå University for their helpful questions on previous version of this paper. We would like to acknowledge the years of insight and guidance provided by Prof. Aravind Joshi without which this work would not exist. We wish to dedicate this paper to his memory.

## References

- Abney, Steven. 1987. The English noun phrase in its sentential aspect. Doctoral Dissertation, MIT.
- Banik, Eva. 2004. Semantics of VP coordination in LTAG. In *Proceedings of the 7th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+7)*, 118–125. Vancouver, Canada.
- Barros, Matthew, and Luis Vicente. 2011. Right node raising requires both ellipsis and multidomination. *University of Pennsylvania Working Papers in Linguistics* 17:Article 2.
- Barwise, Jon, and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4:159–219.
- Bleam, Tonia. 2000. Clitic climbing and the power of Tree Adjoining Grammar. In *Tree*

- Adjoining Grammars: Formalisms, linguistic analysis, and processing*, ed. Ann Abeillé and Owen Rambow, 193–220. Stanford, Calif.: CSLI.
- Cann, Ronnie, Ruth Kempson, Lutz Marten, and Masayuki Otsuka. 2005. Right node raising, coordination and the dynamics of language processing. *Lingua* 115:503–525.
- Chaves, Rui P. 2012. Conjunction, cumulation and respectively readings. *Journal of Linguistics* 48:297–344.
- Chiang, David, and Tatjana Scheffler. 2008. Flexible Composition and Delayed Tree-Locality. In *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 9)*. Tübingen, Germany.
- Frank, Robert. 2002. *Phrase structure composition and syntactic dependencies*. Cambridge, MA: MIT Press.
- Frank, Robert. 2006. Phase theory and tree adjoining grammar. *Lingua* 116:145–202.
- Frank, Robert, and Dennis Ryan Storoshenko. 2012. The shape of elementary trees and scope possibilities in STAG. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, 232–240. Paris.
- Gazdar, Gerald. 1981. Unbounded dependencies and coordinate structure. *Linguistic Inquiry* 12:155–184.
- Grimshaw, Jane. 1991. Extended projection. Manuscript, Brandeis University.
- Ha, Seungwan. 2008. Ellipsis, Right Node Raising, and Across-the-Board Constructions. Doctoral Dissertation, Boston University.
- Han, Chung-hye. 2007. Pied-piping in relative clauses: Syntax and compositional semantics using Synchronous Tree Adjoining Grammar. *Research on Language and Computation* 5:457–479.
- Han, Chung-hye, and Nancy Hedberg. 2008. Syntax and semantics of *it*-clefts: a Tree-Adjoining Grammar analysis. *Journal of Semantics* 25:345–380.
- Han, Chung-hye, David Potter, and Dennis Storoshenko. 2008. Compositional semantics of

- coordination using synchronous tree adjoining grammar. In *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*, 33–40. Tuebingen, Germany.
- Han, Chung-hye, David Potter, and Dennis Storoshenko. 2010. Non-local right node raising: an analysis using delayed tree-local MC-TAG. In *Proceedings of the 10th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+10)*. Yale University.
- Heim, Irene, and Angelika Kratzer. 1998. *Semantics in generative grammar*. Blackwell Publishers.
- Heycock, Caroline, and Anthony Kroch. 1994. Verb movement and coordination in a dynamic theory of licensing. *The Linguistic Review* 11:257–283.
- Higginbotham, James. 1985. On semantics. *Linguistic Inquiry* 16:547–594.
- Joshi, Aravind K., L. Levy, and M. Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences* .
- Kehler, Andrew. 1996. Coherence and the coordinate structure constraint. In *Proceedings of the Twenty-Second Annual Meeting of the Berkely Linguistics Society: General Session and Parasession on the Role of Learnability in Grammatical Theory*, 220–231.
- Kempson, R., W. Meyer-Viol, and D.M. Gabbay. 2000. *Dynamic syntax: The flow of language understanding*. Wiley.
- Koopman, Hilda, and Dominique Sportiche. 1991. The position of subjects. *Lingua* 85:211–258.
- Kroch, Anthony, and Aravind K. Joshi. 1985. The linguistic relevance of Tree Adjoining Grammar. Technical Report MC-CS-85-16, Department of Computer and Information Sciences, University of Pennsylvania.
- Kroch, Anthony, and Beatrice Santorini. 1991. The derived constituent structure of the West Germanic verb raising construction. In *Principles and parameters in comparative*



- grammar*, ed. Robert Freidin, 269–338. Cambridge, MA: MIT Press.
- Kroch, Anthony S., and Aravind K. Joshi. 1987. Analyzing extraposition in a tree adjoining grammar. In *Discontinuous constituents*, ed. G. Huck and A. Ojeda, volume 20 of *Syntax and Semantics*. Academic Press.
- Kulick, Seth. 2000. Constraining Non-Local Dependencies in Tree Adjoining Grammar: Computational and Linguistic Perspectives. Doctoral Dissertation, University of Pennsylvania.
- McCawley, James D. 1982. Parentheticals and discontinuous constituent structure. *Linguistic Inquiry* 13:91–106.
- McCloskey, James. 1991. Clause structure, ellipsis, and proper government in Irish. *Lingua* 85:259–302.
- Munn, Alan. 1999. On the identify requirement of atb extraction. *Natural Language Semantics* 7:421–425.
- Nesson, Rebecca, and Stuart Shieber. 2007. Extraction phenomena in Synchronous TAG syntax and semantics. In *Proceedings of the Workshop on Syntax and Structure in Statistical Translation*, ed. Dekai Wu and David Chiang. Rochester, New York.
- Nesson, Rebecca, and Stuart M. Shieber. 2006. Simpler TAG Semantics through Synchronization. In *Proceedings of the 11th Conference on Formal Grammar*. Malaga, Spain: CSLI.
- Ross, John Robert. 1967. Constraints on variables in syntax. Doctoral Dissertation, MIT, Cambridge, MA.
- Sabbagh, Joseph. 2007. Ordering and linearizing rightward movement. *Natural Language and Linguistic Theory* 25:349–401.
- Sabbagh, Joseph. 2014. Right node raising. *Language and Linguistic Compass* 8/1:24–35.
- Sarkar, Anoop, and Aravind Joshi. 1996. Coordination in Tree Adjoining Grammars: formalization and implementation. In *Proceedings of COLING'96*, 610–615. Copenhagen.

- Schabes, Yves, and Stuart M. Shieber. 1994. An Alternative Conception of Tree-Adjoining Derivation. *Computational Linguistics* 20:91–124.
- Shieber, Stuart. 1994. Restricting the weak-generative capacity of Synchronous Tree-Adjoining Grammars. *Computational Intelligence* 10.
- Shieber, Stuart, and Yves Schabes. 1990. Synchronous Tree Adjoining Grammars. In *Proceedings of COLING'90*. Helsinki, Finland.
- Sipser, Michael. 1996. *Introduction to the theory of computation*. Boston: PWS Publishing Company.
- Steedman, Mark. 1996. *Surface structure and interpretation*. Cambridge, MA: MIT Press.
- Storoshenko, Dennis Ryan, and Robert Frank. 2012. Deriving syntax-semantics mappings: node linking, type shifting and scope ambiguity. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, 10–18. Paris.
- Storoshenko, Dennis Ryan, and Chung-hye Han. 2015. Using synchronous tree adjoining grammar to model the typology of bound variable pronouns. *Journal of Logic and Computation, Language and Computation Corner* 25:371–403.
- Vijay-Shanker, K. 1992. Using descriptions of trees in a tree adjoining grammar. *Computational Linguistics* 18:481–517.
- Weir, David. 1988. Characterizing mildly context-sensitive grammar formalisms. Doctoral Dissertation, University of Pennsylvania.
- Wexler, Ken, and Peter W. Culicover. 1980. *Formal principles of language acquisition*. Cambridge, MA: MIT Press.