- Title: Using Synchronous Tree Adjoining Grammar to Model the Typology of Bound Variable Pronouns

- Authors and Affiliations: Dennis Ryan Storoshenko (Yale University), Chung-hye Han (Simon Fraser University)

- Address for Dennis Ryan Storoshenko:
  Department of Linguistics, Yale University
  370 Temple St, Room 204
  New Haven, CT 06520-8366, USA

  Address for Chung-hye Han:
  Department of Linguistics, Simon Fraser University
  8888 University Drive
  Burnaby, British Columbia V5A1S6, Canada

- Email address:
  dennis.storoshenko@yale.edu
  chunghye@sfu.ca

- Telephone for Chung-hye Han: 778-782-5507

- Abstract:
  This paper presents a novel analysis of bound variable anaphora using Synchronous Tree Adjoining Grammar (STAG), a pairing of a Tree Adjoining Grammar (TAG) for syntax and a TAG for semantics. While a bound variable pronoun can occur at a distance from its binder, as in 'Every girl$_i$ believes that she$_i$ is intelligent,' languages vary, though in a limited way, as to how near or far from its binder a bound variable should be. As any dependency between two syntactic objects must be localized to a single predicate domain in TAG, modelling bound variable anaphora in syntax and semantics poses an interesting challenge for STAG. In our analysis, bound variable pronouns are represented as Multi-Component sets in both syntax and semantics, composing in delayed tree-local derivations. This allows us to not only account for variable binding at a distance, but it also allows us to define a single derivational parameter with which observed patterns of bound variable locality can be modelled, ruling out unobserved patterns, while capturing the range of interpretive possibilities for bound variable pronouns across languages.

- Keywords:
  Synchronous Tree Adjoining Grammar, Multi-Component Tree Adjoining Grammar, delayed tree-local derivation, bound variable pronoun, quantification

Using Synchronous Tree Adjoining Grammar to Model the Typology of Bound Variable Pronouns*

Dennis Ryan Storoshenko and Chung-hye Han

Abstract

This paper presents a novel analysis of bound variable anaphora using Synchronous Tree Adjoining Grammar (STAG), a pairing of a Tree Adjoining Grammar (TAG) for syntax and a TAG for semantics. While a bound variable pronoun can occur at a distance from its binder, as in ‘Every girl$_i$ believes that she$_i$ is intelligent,’ languages vary, though in a limited way, as to how near or far from its binder a bound variable should be. As any dependency between two syntactic objects must be localized to a single predicate domain in TAG, modelling bound variable anaphora in syntax and semantics poses an interesting challenge for STAG. In our analysis, bound variable pronouns are represented as Multi-Component sets in both syntax and semantics, composing in delayed tree-local derivations. This allows us to not only account for variable binding at a distance, but it also allows us to define a single derivational parameter from which observed patterns of bound variable locality can be derived, ruling out unobserved patterns, capturing the range of interpretive possibilities for bound variable pronouns across languages.

1 Introduction

In this paper, we present a novel analysis of bound variable anaphora within the Synchronous Tree Adjoining Grammar (STAG) formalism. STAG is a pairing of two Tree

*We are extremely indebted to the two anonymous reviewers for their insightful comments that were crucial in improving this paper. We are grateful to Robert Frank, Lisa Matthewson and Anoop Sarkar for helpful discussions on various aspects of the paper. We also thank the audiences at the 10th International Workshop on Tree Adjoining Grammars and Related Formalisms at Yale University, and the 29th West Coast Conference on Formal Linguistics at the University of Arizona for their helpful questions on previous versions of this paper. All remaining errors are ours. This work was partially supported by NSERC Discovery Grants RGPIN/341442-2008 to Han and SSHRC Postdoctoral Fellowship 756-2010-0677 to Storoshenko.

Adjoining Grammars (TAGs), a TAG for the syntax and a TAG for the semantics. Though the syntactic core of the STAG we develop is derived from the Chomskyan minimalist model, STAG presents a much different picture of the syntax-semantics interface than its Chomskyan counterpart. In the minimalist model, there is a spell-out point (or multiple spell-out points) at which a single syntactic derivational structure is transferred to distinct phonological and logical interfaces (7; 8). The input representation to the logical interface, called LF, can undergo further movement covertly to make the LF semantically interpretable. Indeed, one major approach to quantification relies on such covert movement, namely Quantifier Raising (25; 26; 23; 10). For instance, the LF of (1a) is generated by raising the quantified phrase *every course*, leaving a trace that functions as a variable, as in (1b). Interpreting this LF yields the semantic form in (1c).

(1) a. John took every course.

b. every course$_i$ [John took t$_i$]

c. $\forall x$ [course(x)] [took(John$, x)$]

Quantifier Raising (QR) is also used in accounting for quantifier scope ambiguity. For instance, the sentence with two quantified phrases in (2a) is ambiguous between the reading in which there is a particular student that took all the courses, and the reading in which for each course, there is some student or other that took it. This ambiguity is said to be derived via different orderings of the application of QR on the two quantified phrases, generating either (2b) or (2c).

(2) a. A student took every course.

b. a student$_i$ [every course$_j$ [t$_i$ took t$_j$]]

$\exists y$ [student(y)] [$\forall x$ [course(x)] [took(y, x)]]

c. every course$_j$ [a student$_i$ [t$_i$ took t$_j$]]

$\forall x$ [course(x)] [$\exists y$ [student(y)] [took(y, x)]]

QR however is not only important in accounting for quantification and scope ambiguities, but it also plays an important role in the interpretation of pronominal variable binding, following the work of Heim and Kratzer (16) and Büring (4). For instance, in interpreting sentences such as (3a), QR is the first step in initiating the interpretive mechanism by which a quantifier can bind a pronominal variable. Under this approach, (3a) generates an LF as in (3b), which then yields the semantic form in (3c).

(3) a. Every girl$_i$ believes that she$_i$ is intelligent.

b. every girl$_i$ [t$_i$ believes that she$_i$ is intelligent]

c. $\forall x$ $[\text{girl}(x)]$ $[\text{believes}(x, \text{intelligent}(x))]$

In the STAG model, on the other hand, the syntactic form and the semantic form are derived in parallel with a strict synchronicity of derivational steps. As a result, one distinguishing feature of this model is that there is no notion of LF or covert movement. The STAG model, therefore, has an account of quantification and quantifier scope ambiguity that is radically different from accounts which rely on QR. Shieber and Schabes (35), Nesson and Shieber (28) and Han et al. (14), for instance, each present an analysis that uses Multi-Component sets of elementary trees to represent the semantics of quantified phrases and multiple adjoining to derive scope ambiguity. We discuss this analysis in more detail in Section 4.

In addition, lacking QR, the STAG formalism will need a distinct implementation for the interpretive mechanism of pronominal variable binding, which turns out to pose an interesting challenge for the formalism. In TAG, any dependency between two syntactic objects is localized to a single predicate domain if all recursions are factored away. But in examples such as (3a), the binder in the matrix clause, *every girl*, and the bound variable in the embedded clause, *she*, belong to two different predicate domains. In addition, languages vary, though in a limited way, as to how near or far from its binder a bound variable should be. In this paper, we propose an analysis of pronominal variable binding that makes use of delayed tree-local derivation, as defined in Chiang and Scheffler (6). In the course of presenting an STAG account of variable binding, we show that our STAG analysis is more readily able to capture the range of interpretive possibilities for bound variable pronouns across languages than the QR-based account. Put simply, apart from the syntactic structure on which aspects of surface constituency are defined, TAG makes available additional formal mechanisms of derivation, the derivation structure, on which dependencies between elementary objects are defined. We show that an account based on the TAG derivation structure allows for a more principled explanation of the parameterization of locality constraints between the binder and the bound variable pronoun than a system which relies on the syntactic structure alone.

The paper is organized as follows. In Section 2 we paint the picture of the empirical ground of pronominal variable binding that this paper will cover. We discuss cross-linguistic variation in bound variable locality, exemplified by English, Korean, Norwegian and Shona. This is followed by an introduction to the syntactic underpinnings of STAG in Section 3, in which we introduce the fundamental concepts of a TAG derivation, and our analysis of the syntactic relationship between the binder and the bound variable pronoun. Section 4 moves on into the semantic side of the derivation, introducing the restrictions of a synchronous derivation. Here, we discuss the STAG analysis of quantification proposed in Han et al. (14), and extend this analysis to account for the interpretive mechanism of pronominal variable binding in English. From here we return to the cross-linguistic data in Section 5, explicating the single derivational parameter with which we model observed (and rule out unobserved) patterns of bound variable locality. Finally, Section 6 re-iterates the typological predictions of our analysis, and closes with some final remarks on the complexity of the STAG derivations proposed.

2 Three Types of Bound Variable Pronoun

When used as bound variables, English pronouns such as *her* must have a certain minimum syntactic distance from their c-commanding antecedents. As was shown in (3a), a bound pronoun and its antecedent can be separated by clausal boundaries. But when they appear within the same clause, they cannot be co-arguments, as illustrated in (4).

(4) a. * Every girl$_i$ loves her$_i$.

b. Every girl$_i$ loves her$_i$ father.

Turning away from English, the Korean long distance anaphor *caki* has been argued by Han and Storoshenko (15) to be best analyzed as a semantically bound variable. As such, its interpretation would follow from the same binding mechanisms as the English bound pronouns. While Korean *caki* shows the same pattern as English bound pronouns with regard to variable binding across clause boundaries, as in (5), it does not have the same restriction in local domains, as in (6).[1]

(5) Motun sonye$_i$-nun [caki$_i$-ka ttokttokha-tako] sayngkakha-n-ta.
every girl-TOP self-NOM intelligent-COMP think-PRES-DECL
'Every girl thinks that she is intelligent.'

(6) a. Motwu$_i$-ka caki$_i$-lul salangha-n-ta.
everyone-NOM self-ACC love-PRES-DECL
'Everyone loves himself.'

b. Motwu$_i$-ka caki$_i$-uy apeci-lul salangha-n-ta.
everyone-NOM self-GEN father-ACC love-PRES-DECL
'Everyone loves his father.'

Taking the data in (5) and (6) together, it appears that for the Korean *caki*, there is no constraint at all on the syntactic distance between the bound variable and its antecedent.

[1] We use the following abbreviations in glossing the examples in (5)-(6): TOP: topic, NOM: nominative, COMP: complementizer, PRES: present tense, DECL: declarative, ACC: accusative, and GEN: genitive.

A third possibility for pronominal variable binding is raised in the Déchaine and Wiltschko's (9) discussion of various types of reflexive elements across languages. Specifically, they describe a class of reflexives which are interpreted in the semantics through operator-variable binding, and claim that some of these are limited to local environments only. This is illustrated in (7) with examples from Norwegian.

(7) a. Jon$_i$ forakted seg selv$_i$.
Jon despised SEG self
'Jon despised himself.'

b. * Jon$_i$ bad oss [forakte seg selv$_i$].
Jon asked us despise SEG self
'Jon asked us to despise him.'

Under their account, the reflexive *seg selv* is a bound variable which must be bound locally by a co-argument. They go on to make similar claims about *se* reflexives in French and Spanish, which are obligatorily used in local contexts. While the conception of English reflexivity (*-self* pronouns) as a type of strictly local variable binding is not held universally, it does surface repeatedly in the literature. In Reinhart and Reuland (31) for example, variable binding is one of the possible implementations of English reflexivity discussed, and is ultimately the one used throughout their analysis. More recently, Storoshenko (36) argues that the reflexivity in the Bantu language Shona is derived through a bound variable, manifested as the morpheme *zvi*, which is restricted to co-argument binding only, as illustrated in (8a). In (8b), an equivalent bound variable reading across a clause boundary must use a distinct pronoun.[2]

(8) a. Imbwa y-oga-yoga ya-ka-zvi-rum-a.
dog.9 CL9-every-REDUP SUBJ.9-PST-REFL-bite-FV
'Every dog$_i$ bit self$_i$.'

[2]We use the following abbreviations in glossing the examples in (8): CL: noun class, REDUP: reduplication, SUBJ: subject agreement, PST: past tense, REFL: reflexive, and FV: final vowel.

b.	Mu-rume	w-oga-woga	a-ka-t-i	[Shingi
	CL1-man	CL1-each-REDUP	SUBJ.1-PST-say-FV	Shingi

a-ka-zvi-won-a].
SUBJ.1-PST-REFL-see-FV
'Each man$_i$ said that Shingi$_j$ saw self$_{*i/j}$.'

Thus, there is ample discussion in the literature to suggest that there are such things as strictly locally (co-argument) bound pronominal variables.

In sum, in the reflexive cases, we observe a binding relation that is constrained to obtain only within a very strict local domain. Conversely, English bound variable pronouns are infelicitous in a roughly congruent domain, displaying a pattern of anti-local binding. Finally, Korean *caki* does not show any locality restrictions whatsoever. What is interesting is that these three patterns represent the sum total of bound variable pronouns. As far as we know, we do not find, for example, some sort of hyper-anti-locality in which an antecedent must be two, three, or four clauses from the pronominal variable it binds, nor do we see cases where a bound variable is acceptable across one clause boundary, but no more.

From this, we now have our desiderata for moving forward. Based on available data, our analysis should be able to account for strict locality, anti-locality, and a lack of constraint on binding, ideally all as distinct values of a single parameter. Further, these should be the only possible values of that parameter, as we do not see any instances of hyper-anti-locality, or other highly-specified constraints. Before we can present our analysis though, we must first step back and illustrate how STAG handles quantification. Our discussion begins in the next section with the presentation of the syntactic underpinnings of STAG.

3 The Syntax of Quantification and Variables in TAG

In this section, we open with the basics of syntax in a Tree Adjoining Grammar, introducing well-formedness conditions on elementary trees, the combinatory operations of substitution and adjoining, and the notion of derivation in TAG. We then move on to present the syntax of quantification and bound variable pronouns.

3.1 Core Concepts of TAG Syntax

A tree adjoining grammar (TAG) is a system for the combination of elementary tree structures, first formalized in Joshi et al. (19). The first significant application of this mathematical model to syntactic theory is generally regarded to be found in Kroch and Joshi (24), which introduces the idea that apparent long-distance dependencies can be reduced to local dependencies which are stretched through the adjoining of recursive tree structures.

In what follows, we illustrate the core concepts of TAG that we are assuming, using the example in (9). We restrict the discussion to the properties of TAG that are most relevant for appreciating the analysis proposed in this paper.

(9) A smart student took every course enthusiastically.

Following Joshi and Schabes (20), we assume a lexicalized TAG. That is, every elementary tree will be anchored by a unique lexical item which carries predicate-argument information. Other overt functional items (determiners, auxiliary verbs), which do not take their own arguments, will be present in elementary trees in addition to the lexical anchor which projects the appropriate argument structure. Thus, for the example in (9), there will be five elementary trees, one each for *take*, *student*, *course*, *enthusiastically* and *smart*.

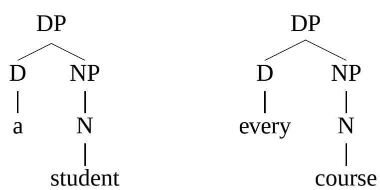
To determine the shape of elementary trees, we appeal to two well-formedness conditions on elementary trees defined in Frank (11). These well-formedness conditions stem from what Frank calls the "Fundamental TAG Hypothesis" that every syntactic dependency be expressed locally within a single elementary tree. The first is the Condition on Elementary Tree Minimality (CETM), stated in (10).

(10) Condition on Elementary Tree Minimality:

The syntactic heads in an elementary tree and their projections must form the extended projection of a single lexical head.

This provides an upper bound on the size of syntactic elementary trees. For example, for the nouns *course* and *student*, we can assume that the upper bound on their trees will be the Determiner Phrase (DP), the maximal functional projection for nominals, in line with the DP Hypothesis (1), as in (11). Under this approach, quantifiers are functional elements which are optional in the noun's extended projection, preserving the notion that each elementary tree has a unique lexical anchor. Furthermore, with this appeal to the extended projection, we are able to exploit elementary trees which contain (functional) heads in addition to lexical anchors.

(11) DP trees for (9)



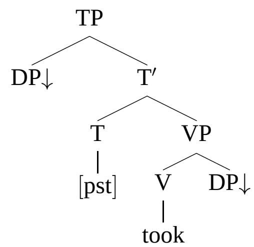
For verbs, a parallel verbal functional structure can be projected. That is, an elementary tree anchored by a verb will project a Tense (T) head, and even possibly further. The exact structure of that tree will depend on the second well-formedness condition which governs argument positions, as stated in (12).

(12) TAG θ Criterion:

If H is the lexical head of the elementary tree T, H assigns all of its θ-roles within T. If A is a frontier nonterminal node of the elementary tree T, A must be assigned a θ-role in T.

So, the elementary tree anchoring the transitive verb *took* in (9) will have two frontier nonterminals, as in (13).

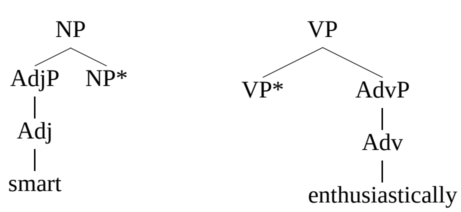
(13) Transitive verb *took* in (9)



The two arguments of *took*, the subject and the object, are necessarily going to be distinct lexical items, therefore cannot be a part of the verb's own elementary tree. The positions for these arguments are left open as substitution sites, indicated by the down arrow. Substitution, one of the two tree-combination operations available, replaces a frontier nonterminal node with an elementary tree with a root of the same category. With the subject and object argument positions generated within the transitive verb elementary tree, elementary trees become a natural definition for the domain of locality. The T head, a part of the predicate's extended projection, has no overt phonological form in this case, but carries the tense information for the clause.

Finally, we come to the modifiers *smart* and *enthusiastically*. As lexical items, these will anchor their own elementary trees, and are represented as recursive trees which modify an NP or a VP, respectively, as in (14).

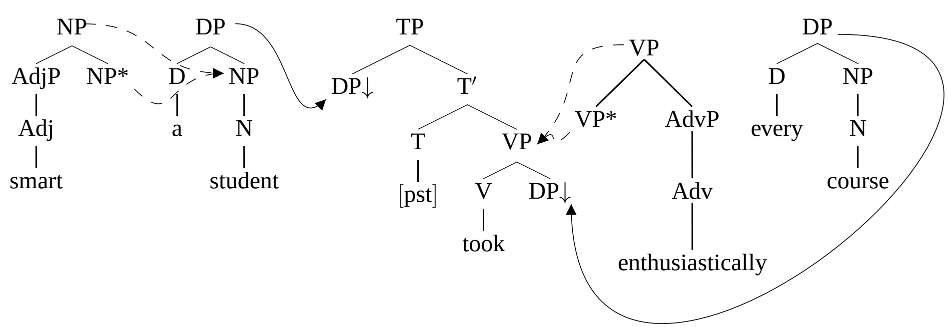
(14) Modifiers for (9)



Crucially, these trees contain one frontier node which is identical in category to the root node of the tree. This node, marked with an asterisk, is called the foot node. Such trees are thus recursive structures and are known as auxiliary trees. Following Frank (11), we can count VP* and NP* in these auxiliary trees as arguments of the lexical anchor, as the process of theta-identification (17) obtains between them and the lexical anchor. These auxiliary trees can combine with other elementary trees by way of the second tree-combination operation: adjoining. Adjoining is best visualized as a splicing operation: a node in the destination elementary tree which has the same category as the root and foot nodes of the auxiliary tree is targeted. This node is split into two parts, a top and a bottom; the auxiliary tree is inserted beneath the top part of the targeted node, and the bottom part (along with the nodes it dominates) is inserted at the foot of the auxiliary tree. Thus, the derivation of (9) will have two instances of adjoining.

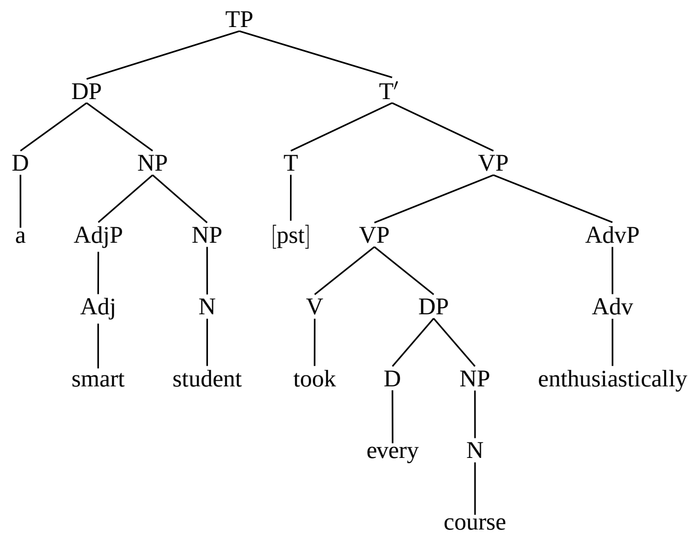
All of the tree combinations to derive (9) are shown in (15). Solid arrows represent substitutions, while dashed arrows indicate adjoining.

(15) Deriving (9)



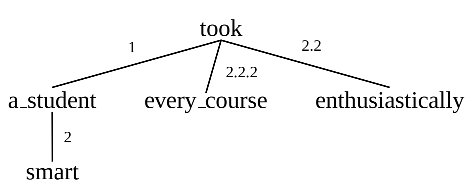
The various combinations in (15) yield the derived tree in (16).

(16) Derived tree for (9)



In addition to the derived tree, which represents the surface constituency, TAG derivation produces a derivation structure (or a derivation tree), a record of the history of composition of elementary trees. This can be seen as a derivational dependency structure in which the children of a given node are either substituted or adjoined into the parent tree. The derivation tree for (9) is given in (17). No distinction is made between substitution and adjoining in a derivation tree, as the relevant combinatory operation can be deduced from the shape of the child elementary trees: auxiliary trees adjoin, non-auxiliary trees substitute. Each edge connecting a pair of nodes is annotated with the location in the parent elementary tree where the TAG operation took place. Here, we use Gorn addresses to denote locations in parent elementary trees, following the convention in the TAG literature.

(17) Derivation Tree for (9)



It has been noted in the literature that the TAG derivation tree does not conform exactly to a semantic dependancy tree (5; 30). For example, while (17) correctly encodes the two arguments of the verb *took* as dependants of that verb, it may appear to be somewhat counter-intuitive that the modifiers are dependants of the predicates they modify, and not vice versa. We will not present an analysis that addresses this issue here, but it does serve as a simple example of why a distinct semantic representation will ultimately be called for. Further, there is nothing in the syntactic derivation to account for the quantifier scope ambiguity in (9). In Section 4, we show how the quantifier scope ambiguity is derived in the semantic part of the STAG. But before moving on to the semantics, we illustrate how we will represent the syntax of bound variable pronouns in English in the next section.

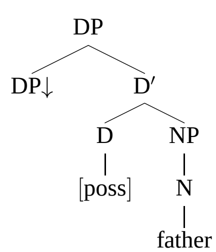
3.2 The Bound Variable Pronoun as a Multi-Component Set

To begin our discussion of English bound variable pronouns, we present the syntactic analysis of (4b), repeated below as (18).

(18) Every girl$_i$ loves her$_i$ father.

The elementary trees for *every girl* and for *loves* are merely slight variations on the trees for *every course* and *took* from above. The two new structures we need are the possessed noun *father* and the bound variable pronoun *her*. Recalling that each elementary tree will have exactly one lexical item with predicate-argument information, the elementary tree anchored by *father* will have a position for one argument, the possessor.

(19) *father_of*



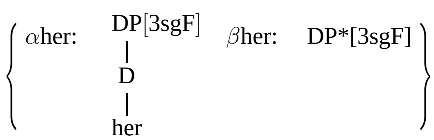
The tree in (19) contains a null D head, which we consider to be a silent variant of the genitive *'s* normally employed in possessive structures.

We now turn to the bound variable pronoun. At first glance, it seems clear enough that *her* will be a DP tree, but this alone will not guarantee the ungrammaticality of a sentence such as (20).

(20) * Every boy$_i$ loves her$_i$ father.

In addition to serving as an argument of the possessed *father*, our bound variable *her* stands in a syntactic agreement relation with its antecedent. To capture this, we adopt an extension of TAG known as Multi-Component (MC) TAG (37). In MC-TAG, lexical items do not anchor single elementary trees, but rather sets of elementary trees. While these sets may be singletons, it is possible for a lexical item to be represented using two distinct pieces of syntactic structure. In the case of the bound variable *her*, we propose the Multi-Component set (MCS) in (21). This bipartite construction of a bound element has its roots in treatments of English reflexive pronouns including those of Ryant & Scheffler (32) and Kallmeyer & Romero (22). Both make use of a degenerate node forming a relation with the antecedent.

(21) Bound variable *her*

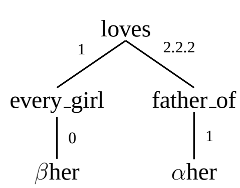


To distinguish MCS members, we will use the prefix α for non-auxiliary trees, and the prefix β for auxiliary trees. αher is exactly the expected form for the bound variable pronoun, a DP which can fill a substitution node. βher consists of a single "defective" node, DP, which crucially carries the ϕ features of the bound variable. This tree has a root node and a foot node of the same category, in that both are the same node; as such, trees of this type meet the definition of an auxiliary tree, and are marked with an asterisk accordingly.

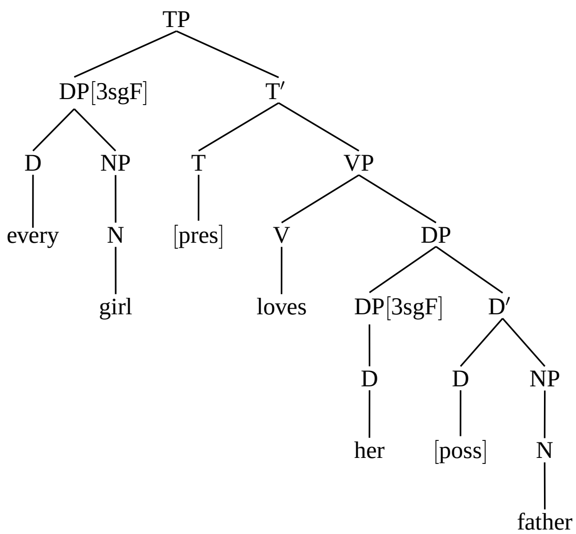
To avoid a feature clash then, this single-node auxiliary tree will need to adjoin to a DP with matching features. While this seems to be a natural way to capture agreement, it is problematic in that the necessary features are carried by the *every_girl* tree, the binder of *her*, not the *father_of* tree where αher must substitute.

In the first formulation of MC-TAG, all derivations were constrained to be tree-local in that all members of an MCS would have to combine with the same elementary tree. However, Chiang and Scheffler (6) argue that it is possible to delay the tree-local combination of MCS elements, allowing those elements to combine with different elementary trees. Locality is "delayed" in the sense that MCS elements combine locally relative to a destination node, which they define as the lowest derivation tree node which dominates all MCS members. In terms of a felicity condition on the combination of MCS elements, this is reducible to a constraint merely that all MCS members must be present in the derivation, as the derivation tree root node trivially dominates all nodes in the derivation tree, no matter which elementary tree(s) immediately dominate the MCS members. This delayed tree-locality is precisely the mechanism which we need in order to capture the fact that the bound variable pronoun can act as the argument of *father of* while still agreeing with *every girl*. Thus, (18) can be derived according to the derivation tree in (22), yielding the derived tree in (23).

(22) Derivation tree for (18)



(23) Derived tree for (18)



In examining (22), this derivation falls under the definition of delayed tree-locality as outlined by Chiang and Scheffler in that while both members of the bound variable MCS initially compose with different elementary trees, they are both dominated by the root (destination) node, *loves*.

Although merely identifying a derivation as employing delayed tree-locality is a trivial matter, as described above, it will be necessary for our purposes to more distinctly characterize the nature of the delay. Chiang and Scheffler define the delay of a derivation for a particular MCS as the set of derivation tree nodes along a path from one member of the MCS to the other, including the MCS members, but excluding the destination node. Thus, for any given MCS, where n is the cardinality of the MCS, d, the cardinality of the delay, will always be at least n. The delay for *her* in (18) is given in (24). We will return to the issue of delays of MCSs and their cardinalities in Section 5.

(24) $\{\alpha$her, βher, father_of, every_girl$\}$

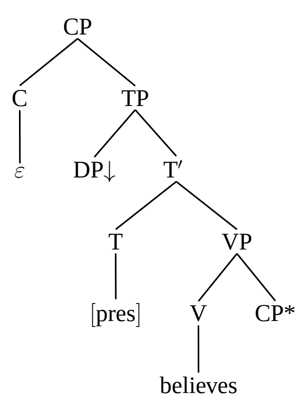
These then are the basic ingredients of our syntactic account for the bound variable pronoun: the bound variable pronoun is instantiated as a two-member syntactic MCS containing the DP tree for the pronominal variable itself, and a defective DP node auxiliary tree which is valued for the relevant ϕ features. Including this second piece of structure is necessary to rule out those cases such as (20), wherein a clash of ϕ features (masculine vs. feminine) would block the adjoining operation. The first component substitutes in as a regular argument while the second component exploits delayed tree-locality to combine directly with the antecedent, ensuring agreement.

The same analysis can be used to establish the syntactic relationship between a bound variable pronoun and its antecedent across clauses, as in (3a), repeated below as (25).

(25) Every girl$_i$ believes that she$_i$ is intelligent.

Much as with the modifiers, TAG syntax models clausal embedding by way of an adjoining structure. This is accomplished through a projection of the matrix clause tree *believes* up to the Complementizer Phrase (CP) node, and the standard assumption that the complement of the verb is also a CP. Thus, the matrix clause becomes an auxiliary tree, as in (26).[3]

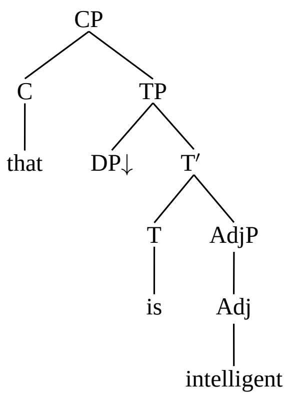
(26) βbelieves



This can combine with the embedded clause, also extended to CP in accordance with the CETM, as in (27).

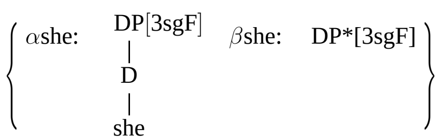
[3]From this point on in the paper, we standardly use the α/β notation for all elementary trees, even if they are singleton sets.

(27) αintelligent



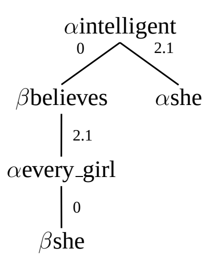
The MCS for the bound variable *she* will be identical to that in (21), aside from the different case form, as in (28).

(28) Bound variable *she*

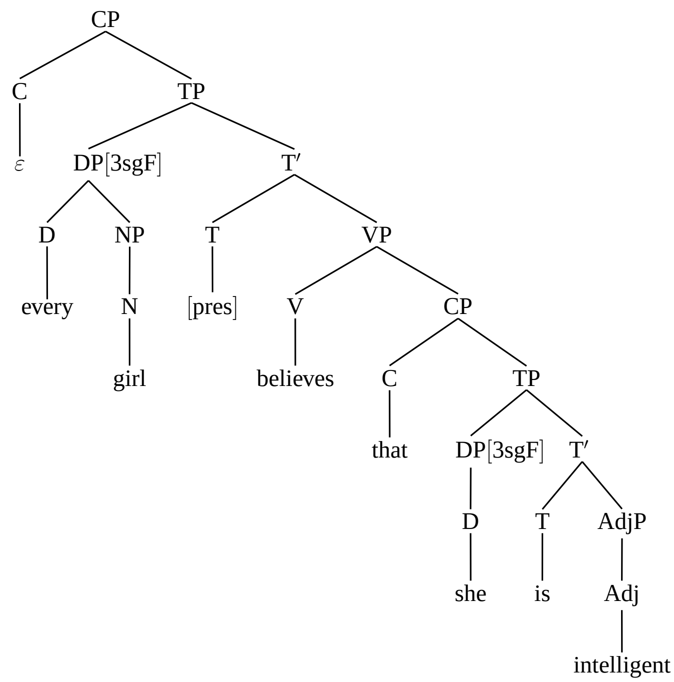


The substitution sites of βbelieves and αintelligent will be filled by αevery_girl and αshe, respectively. βshe will adjoin at the root of αevery_girl, as a check for agreement. (25) is thus derived according to the derivation tree in (29), yielding the derived tree in (30).

(29) Derivation tree for (25)



(30) Derived tree for (25)



As before, the root node of the derivation tree is the destination, the common dominating node, though the delay now spans both predicates. With these tools, quantifier-variable dependencies of any arbitrary length can be constructed.

While this account readily allows us to generate sentences containing bound variable pronouns, we still have not addressed the actual interpretive mechanics of pronominal variable binding. For this, we will need to move beyond a syntax-oriented TAG and into STAG.

4 Semantics and Synchronous TAG

In this section, we introduce the STAG model of the syntax-semantics interface. We begin by presenting the STAG analysis of quantification and deriving quantifier scope ambiguity within a single clause. In so doing, we show how scope ambiguity is captured in a system which lacks QR. From here, we move on to augment our account of the syntax of bound variable pronouns with a semantic representation which is also implemented without recourse to QR.

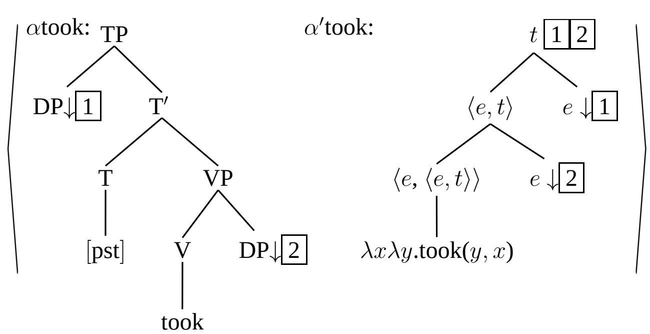
4.1 Quantification in STAG

Synchronous Tree Adjoining Grammar, or STAG, builds upon TAG syntax by deriving both syntactic and semantic trees in parallel (35; 34; 28). That is, the lexicalized syntax is augmented to include a semantics tree as well as the syntax one. In STAG, each lexical anchor is associated with a pair of elementary trees, one syntactic and the other semantic. The syntactic elementary trees will be identical to those developed in the previous section (with one minor modification to be discussed below). In contrast to the feature-based TAG semantics of Gardent and Kallmeyer (12) and the unification-based TAG semantics of Kallmeyer and Romero (21), STAG is able to make reference to two distinct derived forms, one syntactic and the other semantic, in addition to the derivation tree.

The semantic elementary trees represent the lexical anchor as an unreduced λ-expression, essentially following the model of Han (13), making minor changes in the semantic notation such that all nodes are labelled according to their semantic types. Thus, the transitive verb *took* from (2a), repeated below as (31), now becomes a part of the pair in (32).[4]

(31) A student took every course.

(32)



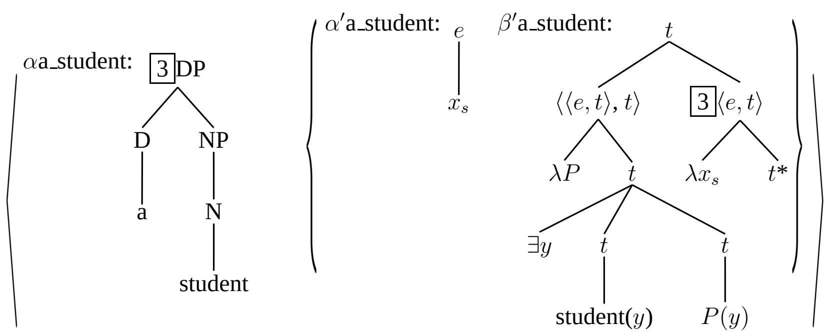
The one modification to the syntax side of the derivation comes in the form of the boxed

[4] We prefix the labels of semantic elementary trees with α' for non-auxiliary trees and β' for auxiliary trees.

numerals added as extra annotations to the substitution sites. These numerals represent links between the specified nodes in the members of the elementary tree pair, and are a key part of synchronizing the STAG derivation. Whenever a TAG operation takes place at a linked node, parallel operations are carried out between the same two elementary tree sets on both the syntax and the semantics sides of the derivation. For example, in (32), if a DP substitutes as the subject of *took*, the semantic counterpart of that DP must combine with all linked nodes in the semantics.[5] Derivation of the syntax and semantics occur in parallel, being synchronized via these links, resulting in isomorphic derivation trees for the syntax and the semantics.

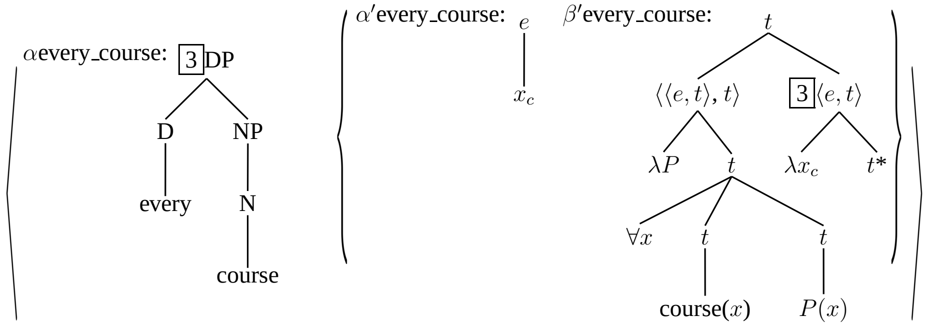
The general account of quantifier scope ambiguity in STAG dates back to Shieber and Schabes (35), where it is shown that by leaving unspecified the order of the combination of each argument with the predicate, either scope reading is possible. We adopt this analysis, but with representations for the semantics of quantification following the model of Han et al. (14), which implements a form of generalized quantifier analysis along the lines of Barwise and Cooper (2). In (33), we give the elementary tree pairs for *a student* and *every course*.[6]

(33)



[5]For sake of simplicity, we include only the links that are relevant for the current discussion. In (32), in addition to the two links already specified, there can be another link on the VP node in the syntax and the $\langle e, t\rangle$ node in the semantics for adverb modification.

[6]The form of the quantifiers could be further simplified: a reviewer correctly points out that there is nothing in the present analysis specifically requiring the type $\langle\langle e, t\rangle, t\rangle$ node. However, this form has been shown to be necessary for analyses of DP-coordination (14), as in *Every boy and every girl jumped*, and so it is retained here to remain compatible with a larger body of work.

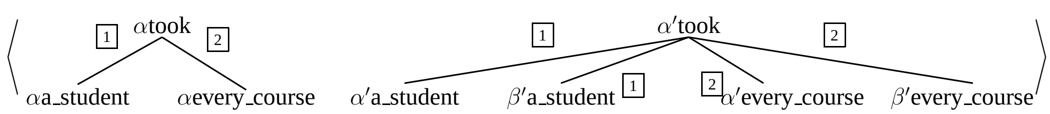


Again, the syntactic form of the quantifier remains unchanged, save for the addition of a link. On the semantics side, the quantifier is represented as an MCS. This treats the quantifier as having two distinct parts: a variable part, α'a_student and α'every_course, which substitutes into the predicate elementary tree at the argument position, and a scope part, β'a_student and β'every_course, which combines with the predicate by way of adjoining.

Scope ambiguity is accounted for by incorporating multiple adjoining in the derivation on the semantics side. Multiple adjoining allows multiple auxiliary trees to be adjoined at the same node in an elementary tree, as defined in Schabes and Shieber (33). Thus, in the derivation of (31), the two quantifiers will each have its own scope part, β'a_student and β'every_course, and they will multiply adjoin to the root node of the predicate elementary tree, α'took. The links $\boxed{1}$ and $\boxed{2}$ on the root node in the α'took tree guarantee this multiple adjoining.

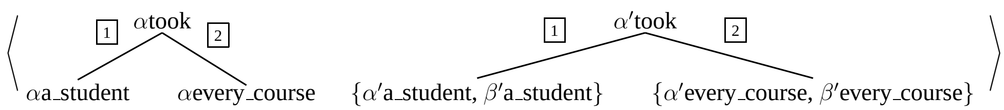
The derivation trees for the syntax and the semantics of (31) are given in (34). For STAG derivation trees, instead of the Gorn addresses, we use boxed numerals for links to denote locations in parent elementary trees to highlight the synchronization in the derivation.

(34) Derivation trees for (31)



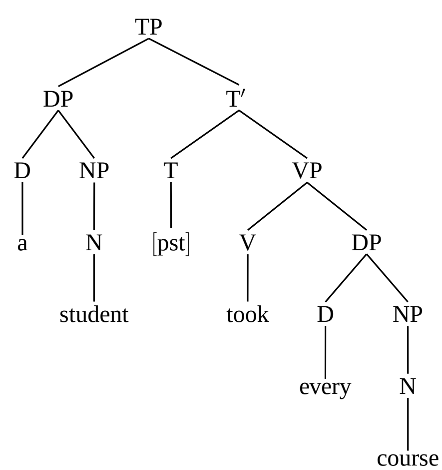
At first glance, the derivation trees in (34) are not isomorphic. However, the forms observed are a result of the fact that each member of the semantic MCS is represented as a single node, as is necessary in the discussion of delayed tree-locality. Isomorphism would result if the semantic derivation tree treated each MCS as a single node using a set notation, as in (35). This is possible here as all members of the MCSs are sisters, indicating that they are combining with the same elementary tree. This, we take as a sufficient condition for describing the derivation trees in (34) as maximally isomorphic and therefore synchronous, and will continue to represent each member of an MCS as a single node in derivation trees.

(35) Alternative derivation trees for (31)



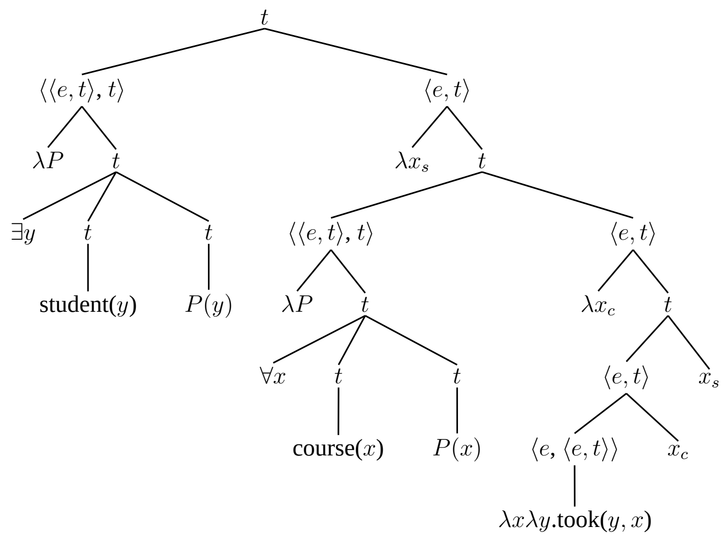
While there is one possible way in which the syntax could be derived, resulting in a single derived syntax tree in (36), in semantics, as the order in which the two scope trees, β'a_student and β'every_course, adjoin to the root node of α'took is underspecified, two possible semantics derived trees can be produced, given in (37).

(36) Derived syntax tree for (31)

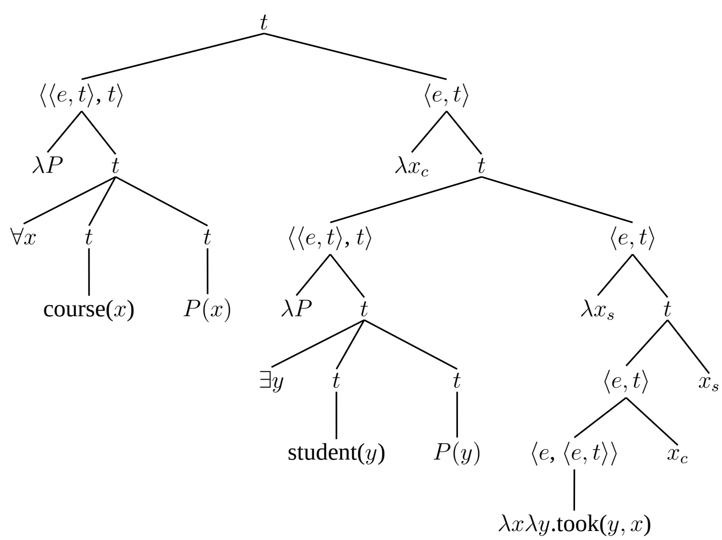


(37) Derived semantics trees for (31)

a. Surface Scope $\exists > \forall$



b. Inverse Scope $\forall > \exists$



In the final stage of the derivation, the semantics trees are simplified through bottom-up computation via instances of function application or λ-abstraction. This yields the forms in (38), matching the expected readings initially schematized in (2).

(38) Final semantic forms for (31)

a. $\exists y$ [student(y)] [$\forall x$ [course(x)] [took(y, x)]]

b. $\forall x$ [course(x)] [$\exists y$ [student(y)] [took(y, x)]]

Thus, representing semantics of quantifiers as MCSs that are composed of the argument variable elementary tree and the scope elementary tree, and allowing the scope tree to participate in unordered multiple adjoining, STAG is able to model quantifier scope ambiguity. Having laid out the mechanics of the STAG system for parallel derivation of syntax and semantics, and the way STAG handles quantification and scope ambiguity, we take the remaining step of updating our syntactic account of the bound variable pronoun with an appropriate semantic form in the next section.

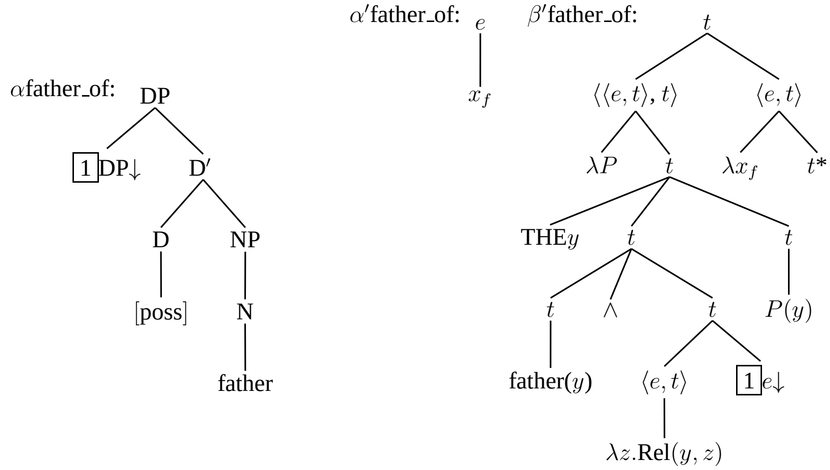
4.2 Binding Pronominal Variables in STAG

Armed with the details of semantic representations in the STAG model, we now return to the main focus of this paper: bound variable anaphora. We begin with the grammatical monoclausal example from English (4b), repeated again as (39).

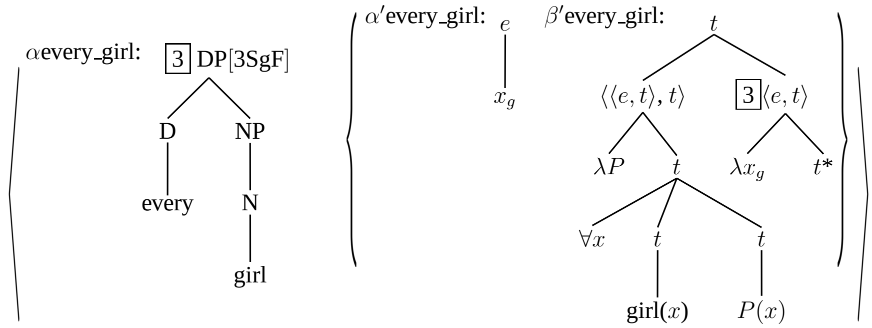
(39) Every girl$_i$ loves her$_i$ father.

The syntactic analysis of this example will not change drastically from that presented in Section 3. In semantics, the *loves* predicate will be essentially identical to the transitive *took* in (32), and *every girl* will again be a minor variant of the generalized quantifiers in (33). New here will be the semantics of *father of*, and the bound variable pronoun itself. First, we present the semantics for the possessed nominal in (40), alongside the now-familiar semantic form for the antecedent *every girl* in (41).

(40)

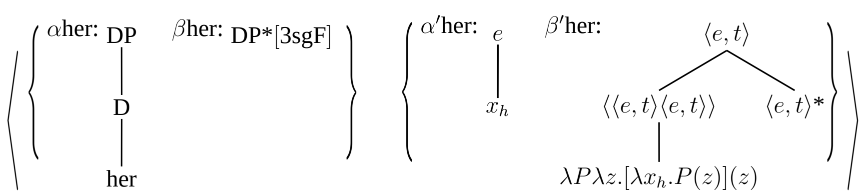


(41)



Following Han (13), the STAG representation of possession is itself embedded within a generalized quantifier structure. Here, we identify the unique individual who is a father and who stands in a generically defined relation with some other entity, open as a substitution site. This substitution site will be the destination for the bound variable, which we treat as an MCS in the semantics as well as in the already-shown syntax, as presented in (42).

(42)



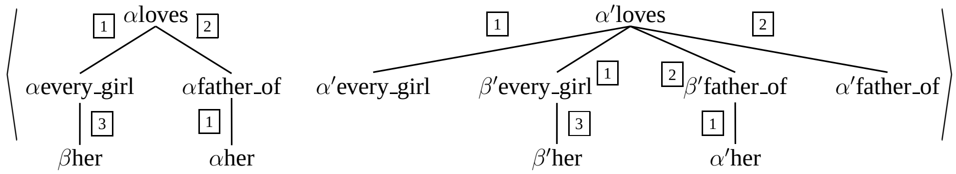
As with the possessed nominal, there has been no change to the syntax. The semantic MCS consists of two members: α'her which is a type e variable, and β'her, which is a function recursive on type $\langle e, t \rangle$. Just as the MCS $\{\alpha$her, βher$\}$ participates in delayed tree-local derivation in the syntax, $\{\alpha'$her, β'her$\}$ does the same. α'her will combine with β'father_of of the MCS $\{\alpha'$father_of, β'father_of$\}$, and β'her will combine with β'every_girl of the MCS $\{\alpha'$every_girl, β'every_girl$\}$.

Because links cannot be formed between elementary trees with different lexical anchors, it will be impossible for components of the bound variable to exploit a single set of derivational links while maintaining delayed tree-locality. This is a distinct situation from the type of tree-local MCS combination seen with the quantifiers; there, links were required between two semantic positions in the semantic elementary tree anchoring a predicate, and the single syntactic argument position in the corresponding syntactic elementary tree. These links are required to preserve the notion that tree-local combination can be implemented as a single derivational step. As this is by definition not possible under delayed tree-locality, we adopt the more relaxed constraint that for each component of an MCS combining via delayed tree-locality, that component's syntactic and semantic correlates must exploit a well-formed derivational link in the target elementary trees. Thus, the derivation remains synchronized in that each of the bound variable's components exploits a single well-formed pair of links. The α components in syntax and semantics combine with linked nodes corresponding to the argument position of the bound variable inside the larger nominal, link $\boxed{1}$ in (40), while the β components exploit a distinct pair of well-formed links within the syntactic and semantic trees of the antecedent, link $\boxed{3}$ in (41).

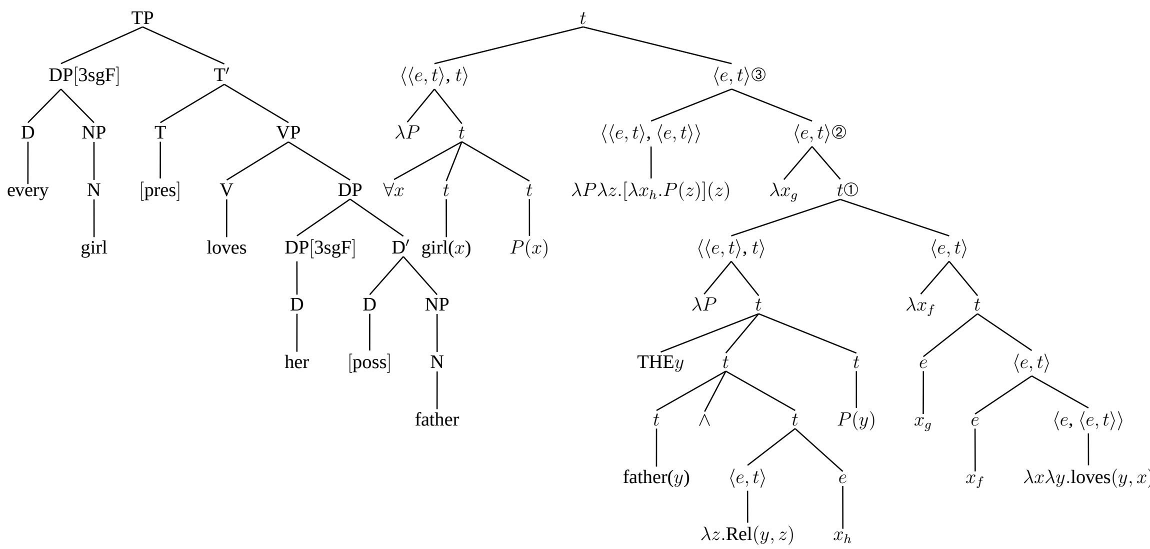
In examining the form and function of β'her, it is worth pausing to consider how pronominal variable binding has previously been treated. A sentence such as (39) would be base-generated with no inherent connection between the quantifier and the bound variable pronoun; co-indexation is treated as nothing more than a coincidence at the stage where arguments are merged. Binding would take place through a series of derivational steps triggered at the root of the syntactic tree after QR takes place. QR leaves behind a variable which the quantifier also binds, but a separate mechanism is required to formally connect the quantifier and the additional bound variable pronoun. Büring (4) accomplishes this through a two-step process. First, he posits an operation of Index Transfer, by which the binder index of a raised quantifier is lowered into a newly-created node which is sister to the node that dominates all materials after QR. Then, he proposes a binder index evaluation rule (called "BIER"), which effects a change in the assignment function while simultaneously λ-abstracting over the variable left behind by the quantifier. The result is a type $\langle e, t \rangle$ predicate with a single λ term binding two instances of the same variable, which can then serve as input to the generalized quantifier. For a more precise picture of how this works, we refer readers to Büring (4), as well as to Heim and Kratzer (16), who have a slightly different approach to the same problem, winding up at exactly the same result. Working in STAG, however, with no QR operation, existing accounts which are parasitic on that operation cannot simply be adapted. Rather, we implement the variable binding with a λ-expression which modifies the type $\langle e, t \rangle$ function created by the scope part of the antecedent (in this case binding the quantifier's variable x_g), abstracting over the bound variable x_h, yielding a new λ-expression of type $\langle e, t \rangle$ in which both x_g and x_h have been λ-converted with z, bound under a single λz operator. This has the same effect as the earlier-discussed approaches, but with no reference to QR, or binder indices.

The derivation trees for (39) in (43) and the derived trees in (44) illustrate how everything is put together. The maximally isomorphic derivation trees show that the requirement of synchronicity holds here with delayed tree-locality, as before. Likewise, the derived syntactic tree is unchanged from (23).

(43) Derivation trees for (39)



(44) Derived trees for (39)



In examining the semantic derived tree, we specifically call attention to those nodes indicated with circled numerals. Semantic calculations at these nodes are provided below.

① THEy [father(y) $\wedge$ Rel(y, x_h)] [loves(x_g, y)]
② λx_g.THEy [father(y) $\wedge$ Rel(y, x_h)] [loves(x_g, y)]
③ $\lambda P \lambda z.[[\lambda x_h.P(z)](z)](\lambda x_g$.THE$y$ [father(y) $\wedge$ Rel(y, x_h)] [loves(x_g, y)])
= $\lambda z[[\lambda x_h.\lambda x_g$.THE$y$ [father(y) $\wedge$ Rel(y, x_h)] [loves(x_g, y)](z)](z)]
= $\lambda z[\lambda x_g$.THEy [father(y) $\wedge$ Rel(y, z)] [loves(x_g, y)](z)]
= λz.THEy [father(y) $\wedge$ Rel(y, z)] [loves(z, y)]

The node ① corresponds to the root of β'father_of after it has been adjoined to the root of α'loves. This contains the bound variable x_h. Moving up to node ②, this is the λ-abstraction over x_g which comes with β'every_girl. β'her adjoins to this node, which places our binding function as sister to ②. As such, at node ③, the function in β'her takes ② as its argument,

λ-converting for the P variable. The end result of simplifying this expression is shown on the final line of the semantic calculation, and variable binding is completed. The resulting $\langle e, t\rangle$ function passes on as the argument of the higher generalized quantifier, and the final semantic form is given in (45).

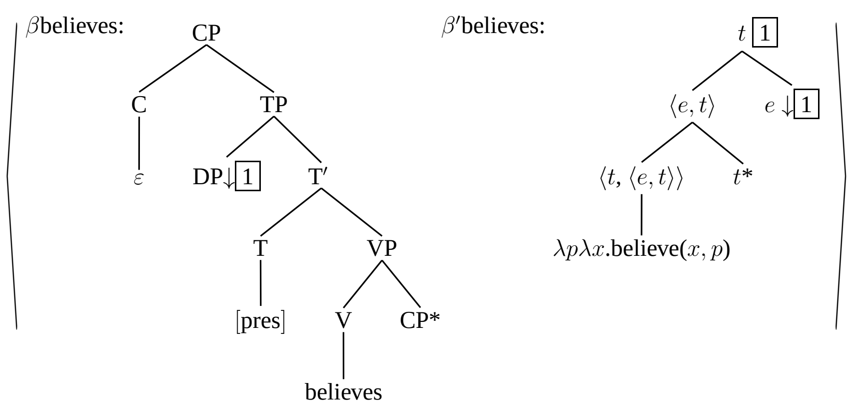
(45) $\forall x$ [girl(x)] [THEy [father(y) $\wedge$ Rel(y, x)] [loves(x, y)]]

At this point, close readers may note that we have only presented one of two possible derived semantic trees for (39). An alternate derivation in which β'father_of adjoins above β'every_girl is also possible, but because α'her substituted into β'father_of, the semantic computation would end with x_h as an unbound variable, β'her having applied vacuously lower in the derived semantic tree. We assume that this derivation is blocked as it results in an unbound variable.

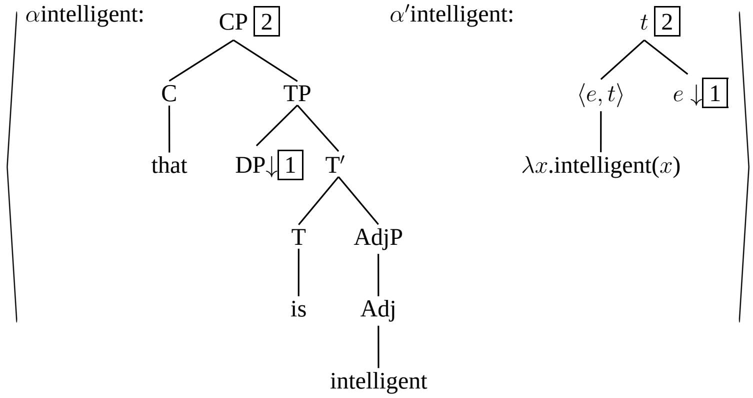
As was the case in the syntax, the semantic derivation is also unimpeded by extending the binding relation across clauses. In (47) and (48), we present the STAG tree pairs for *believes* and *intelligent*, to be used in the derivation of (46), repeated from (3a).

(46) Every girl$_i$ believes that she$_i$ is intelligent.

(47)

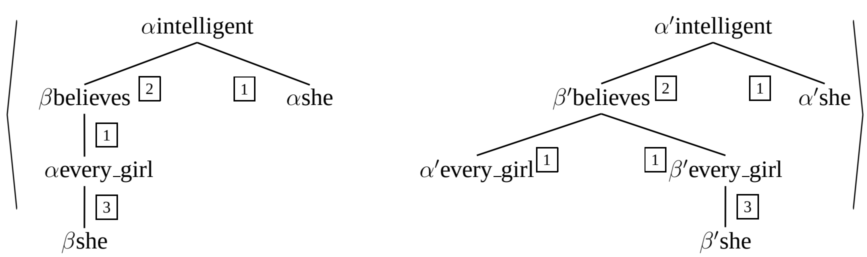


(48)



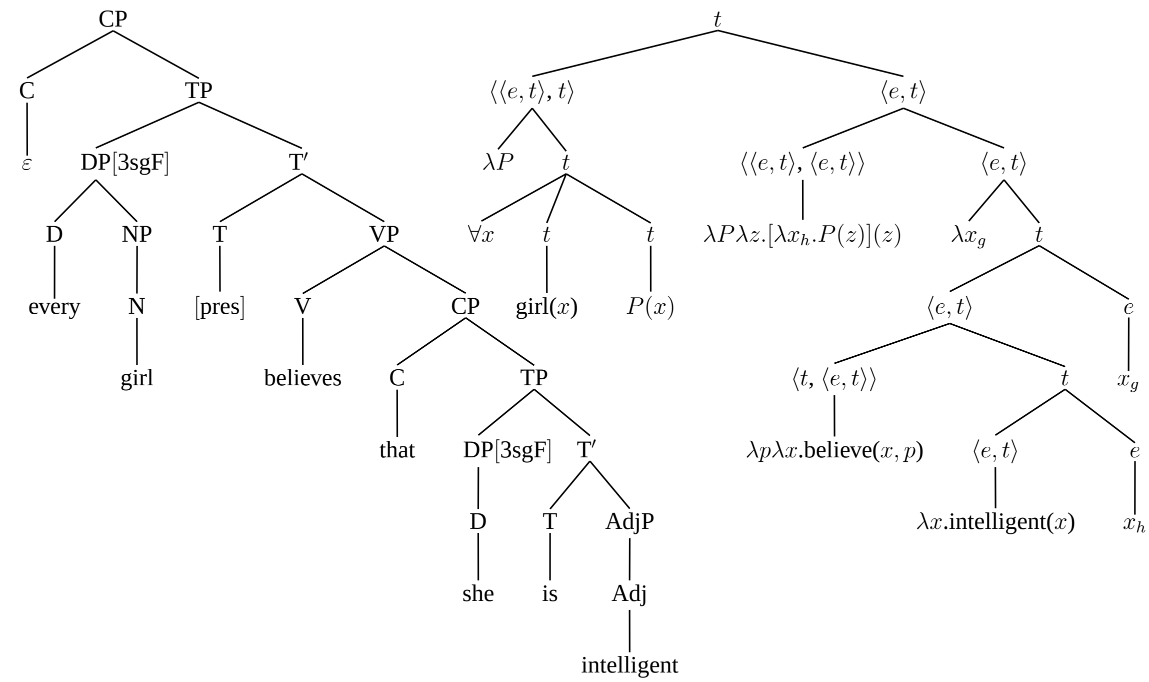
The matrix predicate *believes* is an auxiliary tree in both the syntax and semantics, and the familiar syntax elementary tree has been updated to include the necessary derivational links. Likewise, links are added to the syntax tree of *intelligent*. A link between the syntactic CP root and the semantic t root is added to reflect the possibility for a higher clause to adjoin. No link is included between the e and t nodes of α'intelligent, as there will be no generalized quantifier attaching to this elementary tree.[7] With these trees, along with the already established trees for the generalized quantifier and the bound variable pronoun, the derivation proceeds in a predictable fashion. Derivation trees are given in (49), followed by the derived trees in (50).

(49) Derivation trees for (46)



[7]This is not to say such a link could not be posited, but for the sake of clarity we are only indicating links that are vital for a given derivation.

(50) Derived trees for (46)



Following the derivation, β'she is adjoined into β'every_girl, which is carried into α'intelligent via adjoining of β'believes. Unlike the previous example, there is no potential for a second possible derived semantic interpretation, as there is no quantifier scope ambiguity in this example. Thus, the single possible derived form, after semantic calculation, is as in (51).

(51) $\forall x$ [girl(x)] [believe(x, intelligent(x))]

With this, we can now show how it is the case that pronominal variable binding in both the syntax and semantics can be applied across arbitrarily long distances. The key is in the exploitation of delayed tree-locality; as further clauses are embedded, delays increase, but derivations remain well-formed. Furthermore, we have shown that this can be accomplished within the derivational constraints imposed by the STAG formalism. In all cases considered, the derivation trees for the syntax and semantics are maximally isomorphic.

Before we move on to consider further consequences of our analysis, we briefly discuss an alternative STAG-based analysis of bound pronouns given in Nesson (27) in the next section, highlighting the differences between our approach and hers.

4.3 Binding in Nesson

In Nesson (27), a number of different TAG variants are discussed alongside their potential applications to various linguistic phenomena. Indeed, her discussions of Vector TAG, delayed tree locality, and STAG have much in common with the STAG developed here, and her discussion of derivational constraints in Vector TAG foreshadows the constraints to be discussed in Section 5. However, with all of this commonality between the formal methods employed, our analysis of bound variable pronouns differs sharply from that proposed by Nesson.

Nesson's analysis is predicated on capturing the well known locality contrast between English reflexive and non-reflexive pronouns, as in (52).

(52) a. John$_i$ hurt himself$_i$/*him$_i$.

b. John$_i$ said [that Mary hurt *himself$_i$/him$_i$].

In essence, the reflexive pronouns can only be bound locally, whereas *him* must have a minimal distance between itself and its antecedent. Nesson captures this using an alternative representation of λ-calculus known as De Bruijn notation. Rather than using alphabetic variables and corresponding binders, variables are represented as numeric indices which count the levels of λ-embedding within or beyond which the variable can be bound. To illustrate, we reproduce Nesson's STAG tree sets for *himself* and *him*.

(53)

$$\left\langle \left\{ \begin{array}{cc} \text{S*} & \text{NP} \\ & | \\ & \text{himself} \end{array} \right\} \left\{ \begin{array}{cc} t* & e \\ & | \\ & 1 \end{array} \right\} \right\rangle$$

(54)

$$\left\langle \left\{ \begin{array}{cc} \text{S*} & \text{NP} \\ & | \\ & \text{him} \end{array} \right\} \left\{ \begin{array}{cc} t* & e \\ & | \\ & >1 \end{array} \right\} \right\rangle$$

Both *himself* and *him* are represented as type e in the semantic forms. *Himself* is given as index 1, meaning that it must be bound by the closest available binder. Conversely, *him* is given as > 1, indicating that at least one binder (the closest) must be overlooked as the variable searches upward for an antecedent. In contrast to our analysis, the λ-binders themselves are not part of the semantic representations of the pronouns and instead are encoded in the predicate elementary trees of which the pronouns are arguments. These pronoun tree sets therefore make no explicit connection between the pronominal and its antecedent, and thus there is no need here to resort to delayed tree-locality.

Another difference between Nesson's analysis and ours is in the treatment of unbound pronouns: while Nesson's proposed lexical entry for pronouns in (54) subsumes both bound and unbound pronouns, our proposed lexical entry for bound pronouns in (42) crucially does not. For Nesson, the bound reading of *him* in (55) is generated if *him* takes on the value 2, but the unbound reading is generated if it takes on the value 3 or greater.

(55) John$_i$ thinks Mary$_k$ likes him$_{i/j}$. (Nesson 2009, ex 5.46)

For us, unbound *him* is simply a free variable, having a distinct lexical entry from (42), whose reference is determined by the discourse context.

Similarly, our approach does not subsume cases of cross-sentential anaphora, as in (56).

(56) Bill$_i$ entered the room. John greeted him$_i$.

Crucially, this is not a case of semantic binding, but a case of discourse co-reference. Unlike the quantificational antecedents we have discussed thus far, there is no semantic requirement that the given co-indexation obtain. Even replacing the proper name with a quantifier, the fact would remain that binding does not span clauses, and that co-indexation is not required. Semantic binding as we have formulated it here requires that binding be captured within a single derived semantic tree. Cross-sentential dependencies are a different mechanism entirely and should be handled by a discourse processing component. We thus treat *him* in (56) as a free variable whose reference is again determined by the discourse context, in this case co-referring with an expression in a previous sentence. For Nesson, on the other hand, *him* has the semantics in (54) and will end up being unbound as there is only one λ-binder in the elementary tree anchoring *greeted*.

In sum, while Nesson provides a unified analysis for bound and unbound pronouns, assigning the same semantic representation to both, we only claim to cover bound pronouns and make no further comment on the use of unbound pronouns. We have, however, shown how semantic considerations can block certain possible derivations in cases of unbound variables, and in the next section we move on to consider some outright ungrammatical examples, and discuss how they may be ruled out in our system.

4.4 Eliminating Spurious Derivations

In this section, we consider two well-known cases of ungrammatical instances of bound variable pronouns in English, as illustrated in (57).

(57) a. * Her$_i$ father loves every girl$_i$.

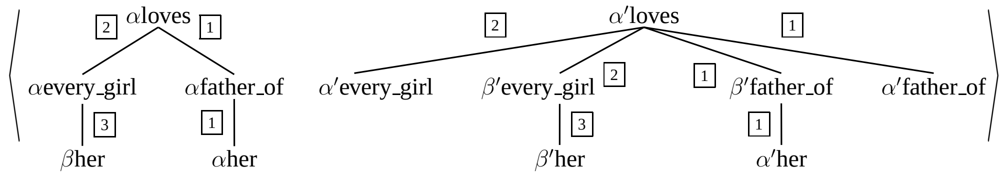
b. * She$_i$ loves every girl$_i$.

In QR-based accounts, the examples in (57) result in crossover violations, triggered by QR. In (57a), as *every girl* QRs across *her* embedded in the subject *her father*, a weak crossover violation is incurred, and in (57b), as *every girl* QRs across the c-commanding *she*, a strong crossover violation is incurred. In STAG, as QR is not active, we must set out a new analysis for how examples such as those in (57) are ruled out. As it is the more challenging of the two, we first deal with weak crossover, and show how the account can be extended to cover strong crossover.

Recalling the discussion of (39), there are two possible derivations where there are two generalized quantifiers, one of which leaves the variable contributed by α'her unbound.

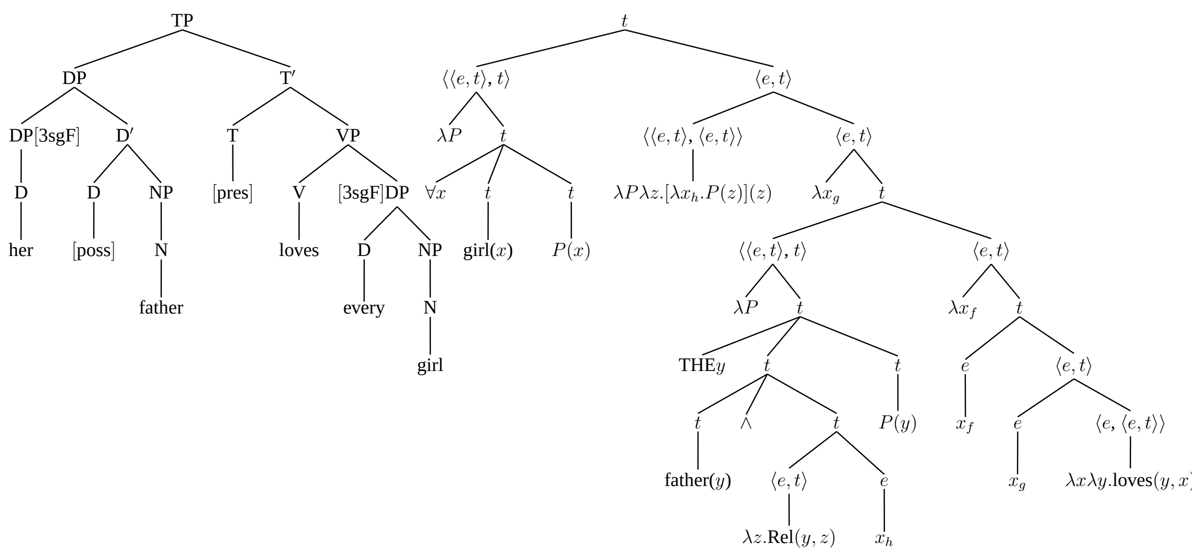
However, a perfectly legitimate derivation for (57a) is possible, with the derivation trees shown in (58).

(58) Derivation Trees for (57a)



This example cannot be blocked on the basis of any derivational constraints based on isomorphism or any constraint against unbound variables in the semantic derived tree. Indeed, derivation trees in (58) are structurally identical to those in (43). Semantic composition from the semantic derived tree in (59) results in the semantic form in (60) with the variable bound, and the intended meaning intact.

(59) Derived Trees for (57a)

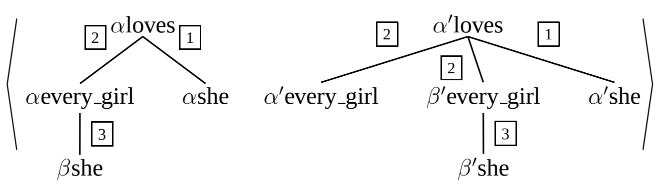


(60) $\forall x$ [girl(x)] [THEy [father(y) $\wedge$ Rel(y, x)] [loves(y, x)]]

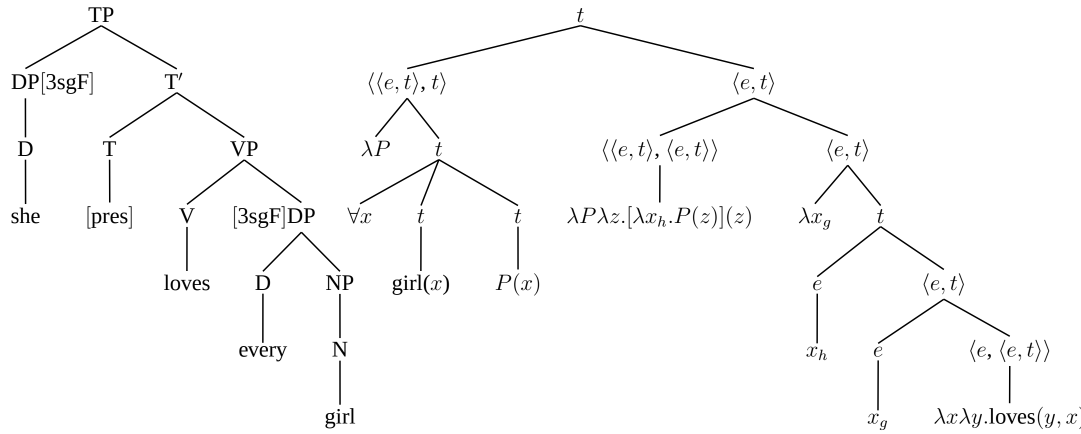
With no alternatives, we must check for well-formedness in the derived syntactic tree. While agreement has been checked, we observe that the bound variable pronoun *her* is not c-commanded by its antecedent *every girl*, and therefore that αher is not c-commanded by the node at which βher adjoins. Thus, we impose a c-command constraint between the elementary trees of the bound variable MCS: in the derived syntactic tree, the defective DP* elementary tree must c-command the argument DP tree. The notion of a c-command constraint between syntactic MCS members is often used; Frank (11) imposes exactly this sort of c-command constraint in a proposed MCS analysis of *wh*-binding, and Ryant and Scheffler (32) make use of the same c-command constraint in their MCS analysis of the binding conditions of *self*-anaphors. Returning to the discussion of (57a), in the syntax derived tree in (59), βher has adjoined at the root of αevery_girl, while αher substitutes at a higher position in αfather_of: the necessary c-command relation does not hold, ruling out this sentence.

The same c-command constraint will rule out strong crossover as in (57b). The elementary trees already established can be used to derive this example following the derivation in (61) to derive the trees in (62).

(61) Derivation Trees for (57b)



(62) Derived Trees for (57b)



As was the case in the weak crossover example, there is nothing inherent in this derivation or in the semantics to rule out this sentence. Again, we turn to the derived syntax tree and find that the final position of βshe does not c-command the final position of αshe. Based on this, we can rule out (57b).[8]

Our analysis thus far does not make any distinction between the weak and the strong crossover cases. One distinction can be found though if the delays for the bound variable MCS in each case are examined. Here, we show the delays for the semantics side only, but the results would be equivalent if we looked to the syntax.

(63) a. Delay for *her* in (57a)

$\{\alpha'$her, β'father_of, β'every_girl, β'her$\}$

b. Delay for *she* in (57b)

$\{\alpha'$she, β'every_girl, β'she$\}$

[8]The proposed c-command constraint on the bound variable MCS does not cover examples such as in (i), from Büring (2004).

(i) a. Every boy's mother likes him.

b. Somebody from every city hates its climate.

This does not pose a problem for our analysis, as examples such as these have been analyzed as involving E-type pronouns and not bound variables (3; 4).

We define the delay length of an MCS as the cardinality of the delay minus the cardinality of the MCS. Under this definition, while the delay length for the bound variable in (63a) is 4-2, two, it is 3-2, one, in (63b), which is the bare minimum delay length in the definition of derivation with delayed tree-locality. We will return to this issue in Section 5 when we formulate a derivational constraint based on the length of the delay.

Having established that there are well-formedness constraints on both the syntax and the semantics of bound variable pronouns in English, we have still not addressed the typological problem laid out in Section 2. Furthermore, we still have not provided an account to rule out the English example (4a), repeated below as (64).

(64) * Every girl$_i$ loves her$_i$.

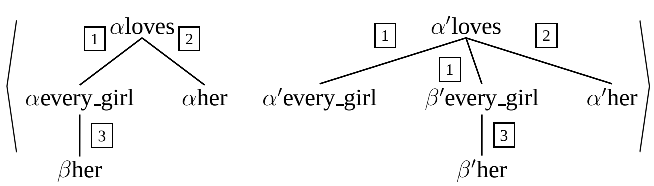
The account of this example will lead us into the next section of the paper where we present our analysis of the cross-linguistic variation in the locality constraints between the bound variable pronoun and its binder.

5 Modelling the Typology

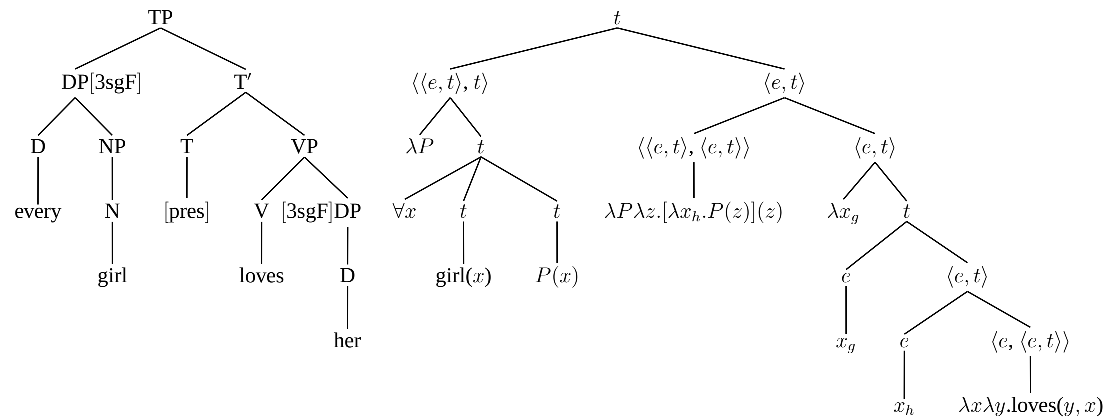
In this section, we propose a derivational constraint on the length of the bound variable delay, which will account for the remaining case of ungrammaticality in English, as was shown in (64), and discuss how this constraint can be parameterized to capture the typology of bound variable pronouns presented in Section 2.

Using the elementary trees already defined, the derivation trees and derived trees for (64) can be generated as in (65) and (66).

(65) Derivation Trees for (64)



(66) Derived Trees for (64)



The derivation trees for (64) are in fact identical to those for (57b), the case of strong crossover violation. However, unlike the strong crossover case, the c-command constraint on the bound variable MCS defined at the end of Section 4.4 is not violated. Thus, based on what we have said so far, (64) should have the reading in (67).

(67) $\forall x$ [girl(x)] [loves(x, x)]

With a well-formed syntax and semantics, we are left to consider the derivation as being the source of ungrammaticality. Recalling our discussion from Section 2, different classes of bound variable pronoun can be defined based on the syntactic distance between the pronominal variable and its antecedent. That is, there are permutations on locality constraints on bound variable pronouns across languages. We note that while the derived syntax tree may provide one means of expressing this locality constraint, the STAG derivation tree provides another. Specifically, we propose that the length of the delays for the bound variable pronouns can be interpreted as a measure of locality. In (68), we present the delays for the bound variables in three of the examples presented thus far in English.

(68) a. * Every girl$_i$ loves her$_i$.

$\{\alpha'$her, β'every_girl, β'her$\}$

b. Every girl$_i$ loves her$_i$ father.

$\{\alpha'$her, β'father_of, β'every_girl, β'her$\}$

c. Every girl$_i$ believes that she$_i$ is intelligent.

$\{\alpha'$she, β'believes, β'every_girl, β'she$\}$

Let d be the cardinality of a delay of an MCS and let n be the cardinality of the MCS. We define the length of the delay, l, to be $d - n$. Then, l for the ungrammatical example is one, and the l values of the grammatical examples are two. Further, we know that sentences in which the dependency between the bound variable pronoun and its binder is longer than (68c), with more layers of clausal embedding, are also fine. We thus propose that the length of delay for a bound variable pronoun in English is constrained to be greater than one.

A constraint based upon the delay length, essentially a restriction on derivational locality, is not without precedent. Working within the framework of Vector TAG, which also allows for the non-local combination of MCS members, Nesson and Shieber (29) propose that MCS derivations can be constrained by imposing a maximum derivational distance permitted between MCS members. In the case at hand, we are proposing that for bound variable pronouns in English, the l value must be greater than one. At first glance, this seems like a highly arbitrary move. However, we argue that not only is this value, one, far from arbitrary, it is in fact the only value on which any such constraint on bound variable locality should be based.

The reasoning for this follows from the fact that a delayed tree-local derivation can in fact be defined in terms of the d and n values for any given MCS, as noted earlier in Section 3. For any MCS with n members, $d \geq n$, as the delay for any MCS will always contain at least the members of the MCS itself. Under this definition, even the generalized quantifiers we are using will have their own trivial delays. Thus, we can update the list of delays in (68) to include the delays for the generalized quantifiers as well, as in (69).

(69) a. * Every girl$_i$ loves her$_i$.

{α'her, β'every_girl, β'her}

{α'every_girl, β'every_girl}

b. Every girl$_i$ loves her$_i$ father.

{α'her, β'father_of, β'every_girl, β'her}

{α'every_girl, β'every_girl}

{α'father_of, β'father_of}

c. Every girl$_i$ believes that she$_i$ is intelligent.

{α'she, β'believes β'every_girl, β'she}

{α'every_girl, β'every_girl}

In all cases, the generalized quantifier MCS members combine tree-locally with a single elementary tree. This type of tree-local combination can be defined in terms of d and n as well, specifically $d = n$. In this case, the delay length, $l = d - n$, is zero. A derivation moves into delayed tree-locality as soon as d is greater than n, in which case l is at least one. Crucially, we have already stated that the bound variable MCS obligatorily combines via delayed tree-locality; a tree-local derivation is not possible for the bound variable MCS. As such, *one* becomes the threshold value, the minimal possible delay length for an MCS that must exploit delayed tree-locality. For English bound variable pronouns, we can therefore recast our constraint in terms of l and the threshold value, requiring that $l > 1$ hold true. This constraint can also explain the distinction between the earlier weak and strong crossover cases, as weak crossover incurred only a violation in the syntax, while the strong crossover example violates this derivational constraint as well.

We can now take this idea to model the typology of bound variable locality. We have already argued that all bound variable pronouns must combine via delayed tree-locality; this is mandated by the forms of the trees in the MCS of bound variable pronouns, and it is at the core of deriving the semantic binding relationship. Once a derivation is forced into delayed tree-locality, there are three logical possible relations between l and the threshold value, listed in (70).

(70) a. $l = 1$

b. $l > 1$

c. $l \geq 1$

The case in (70a) would be that of a pronominal variable which must combine via delayed tree-locality, but can only do so at the minimal threshold value. Looking back to the ungrammatical (64), this is exactly the characterization of co-argument binding. For those languages in which reflexivity is expressed in terms of bound variable anaphora, as in Norwegian and Shona, this would be exactly the right constraint. A bound variable pronoun respecting (70a) would exploit just enough delayed tree-locality in order to be felicitously used, but then go no further. (70b) corresponds to what we have seen for English bound variable pronouns: there is no upper limit on how far the variable may be from its antecedent, but it must at least be further than the threshold value for delayed tree-locality. Finally, we are left with (70c), the most lenient constraint, allowing any possible delayed tree-local derivation; this would correspond to Korean *caki*, as we have shown, in (5) and (6), there to be no locality constraints on its use.

In sum, these are the only three logical possibilities available, translating to only three possible types of bound variable pronoun: one restricted to co-arguments, one viable for anything but co-arguments, and one which is unconstrained. A pronominal variable binding constraint solely based on the derived syntax tree has no equivalently principled way of keeping down the number of possible constraints. Under such a system, a binding constraint for English type pronominal variables, for example, would have to be formulated as a statement such as ‘a bound variable pronoun must be at least one DP or one TP away from the binder.' However, once a binding constraint makes reference to at least one TP/DP node, for example, there is no reason why further counting constraints (at least two, at least three, etc...) could not be formulated. As such, we might expect to find bound variable pronouns which must be at least two or three clauses removed from their antecedents, or even possibly cases where a bound variable should be at most one (but not two or three) clauses away from its antecedent.

The formalization of a threshold value which acts as a pivot point for derivational locality once again sets the present analysis apart from that of Nesson (27). While she uses the De Bruijn notation > 1 as in (54), there is nothing inherent in the De Bruijn notation restricting the number of possible (anti-)locality conditions. Indeed, in Nesson's system, it would be trivially easy to propose a pronoun whose antecedent must be any arbitrary number of λ-binders higher in the derived semantic tree, simply by using a different numeric index in the lexical entry for a given pronoun. By restricting our locality constraint to the permutations in (70), however, we reduce the number of possible bound variable pronouns in natural language from a theoretically infinite number of possibilities to the observed three, all of which pivot around the threshold value.

In closing this section, we would like to re-iterate that the constraints defined in (70) are defined with respect to individual lexical items within a given language, rather than as a global constraint across all derivations involving an MCS within a given language. Indeed, we have already seen for English that while bound variable pronouns are governed by the constraint in (70b), generalized quantifiers are not. The constraints in (70) cover only those lexical items which *must* be derived via delayed tree-locality, presumably as a result of an inherent non-local dependency. Because d is always at least n, the missing case $l < 1$ redundantly defines the situation where $d = n$, describing a tree-local derivation for the MCS. We consider this to also be available as a possible constraint, and indeed maintain that generalized quantifiers in English are constrained in exactly this way. The last possibility,

$l \leq 1$ describes a case wherein an MCS could either combine tree-locally, or at the minimal threshold value for delayed tree-locality. We are not presently aware of any phenomenon which fits this description of varying between a local and a strictly-constrained non-local dependency. Thus, we maintain that the constraints defined in (70) exhaust the sum total of possible definitions of delayed tree-locality, which stand in opposition to the tree-local constraint $d = n$, and that any or all of these may be active within a given language.

6 Conclusion

We have proposed an analysis of the syntax and semantics of pronominal variable binding using STAG, in which the parallel syntactic and semantic derivations are required to be isomorphic. In our analysis, a bound variable pronoun is represented as an MCS in both syntax and semantics, and participates in delayed tree-local derivation. We also make use of constraints in the semantics, syntax, and derivation to rule out unattested cases of variable binding: a constraint against unbound variables in semantics, a constraint that one component of the bound variable MCS c-command the other component in syntax, and a constraint on the length of the delay of the bound variable MCS.

We have also argued that our STAG model of the semantics of bound variable pronouns affords a restricted range of parametric variations which is congruent with the observed range of cross-linguistic data. Splitting the entire set of bound variable pronouns across languages into natural classes based on the derivational delay relative to the minimal possible delay length that must exploit delayed tree-locality yields exactly the three classes of bound variable pronouns described in Section 2. This is a welcome finding, as it not only provides a more principled account of the observed locality parameters than Chomskyan generative syntax, but it serves as a compelling argument for the soundness of a derivation tree based syntax-semantics interface for LTAG, regardless of the type of semantic representation chosen. With the adoption of delayed tree-locality and the derivational constraint on the delay length, the parameters in (70) emerge naturally, rather than by stipulation.

This work stands as a point in favour of the adoption of delayed tree-locality within the STAG framework, and in LTAG generally, although we have strayed somewhat from the original conception of delayed tree-locality as defined by Chiang and Scheffler (6). In their original proposal, they defined constraints based not on the length of delays, but on the number of simultaneous delays in a given derivation. Simultaneity of delays is determined by examining the entire set of delays in a given derivation, and looking for nodes which participate in more than one delay. For example, looking back to the examples in (69), they each define derivations with two simultaneous delays. (69b), for example, spells out a scenario in which both β'father_of and β'every_girl participate in two delays each.

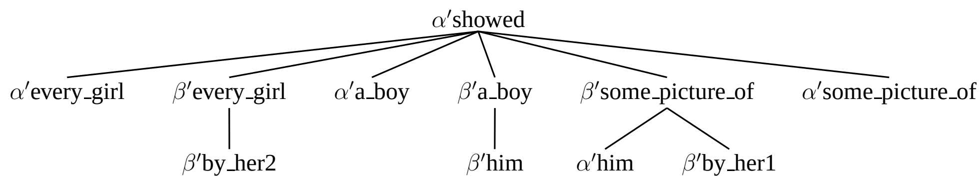
For Chiang and Scheffler, it was important to show that a derivation involving two simultaneous delays generated structures weakly equivalent to those derived from a proposed combinatory operation known as flexible composition, or reverse adjoining. As such, they do not foresee the need to propose derivations with more than two simultaneous delays, nor do they place any constraints on the length of a delay. We however observe that natural language syntax/semantics may require derivations that go beyond two simultaneous delays. Specifically, we draw attention to cases where more than one bound variable pronoun is embedded in a DP, as in (71). The semantic derivation tree for (71a) is given in (72).[9]

(71) a. Every girl$_i$ showed a boy$_j$ some picture of him$_j$ by her$_i$.

b. Every girl$_i$ told a boy$_j$ that some professor$_k$ liked some picture of him$_j$ that she$_i$ gave him$_k$.

[9] For sake of readability, we leave out the boxed numerals that denote locations of parent elementary trees in the derivation tree in (72).

(72) Semantic derivation tree for (71a)



As can be seen from the semantic derivation tree in (72), β'some_picture_of occurs in three delays, those of *some picture of*, *him* and *by her*. And in (71b), it occurs in four delays, those of *some picture of*, *him*$_j$, *she*, and *him*$_k$. So, as the number of bound variables embedded in a DP increases, so does the number of simultaneous delays in the derivation. As embedding is in principle unbounded, this raises the question as to whether the grammar should allow an unbounded number of simultaneous delays. We speculate that as the number of simultaneous delays increases, so does the processing load in deriving the sentence. Speakers encountering an example with four simultaneous delays, such as (71b), may have difficulty in reaching (or even fail to reach) the desired interpretation. It is possible that this decreased comprehensibility with bound variables is indicative of a formal bound on the number of simultaneous delays, in which case, it would follow from a competence property of the grammar. This is analogous to Joshi et al.'s (18) proposal that processing difficulty in German scrambling that requires more than two levels of embedding is due to the formal restriction imposed by the grammar, and not due to a performance limitation. Applying this idea to the case at hand, we can postulate that there is a fixed number k, though it must be at least two under our treatment of bound variables, such that a derivation may have no more than k simultaneous delays. Restricted this way, MC-TAG with delayed tree-locality is weakly equivalent to standard TAG, as shown by Chiang and Scheffler. We leave the connection between comprehensibility, competence and the number of simultaneous delays for future work.

Finally, a question arises as to how the introduction of constraints based upon delay length, a contribution of this paper, interacts with constraints on the number of simultaneous delays. Fundamentally, the two constraints operate on different aspects of the derivation: our delay length constraint is a locality constraint and is concerned only with the delay of a single MCS; it governs the use of a single lexical anchor. On the other hand, constraints on the number of simultaneous delays speak to the overall complexity of the derivation as a whole. The exploration of the interplay between derivational locality and derivational complexity remains as future work as well.

References

[1] S. Abney. *The English noun phrase in its sentential aspect*. PhD thesis, MIT, 1988.

[2] J. Barwise and R. Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219, 1981.

[3] D. Büring. Crossover situations. *Natural Language Semantics*, 12:23–62, 2004.

[4] D. Büring. *Binding Theory*. Cambridge University Press, Cambridge, U.K., 2005.

[5] M.-H. Candito and S. Kahane. Can the TAG derivation tree represent a semantic graph? An answer in the light of Meaning-Text Theory. In *Proceedings of the 4th International Workshop on Tree Adjoining Grammars and Related Formalisms*, Technical report IRCS 98-12, pages 25–28, Philadelphia, 1998. Institute for Research in Cognitive Science, University of Pennsylvania.

[6] D. Chiang and T. Scheffler. Flexible Composition and Delayed Tree-Locality. In C. Gardent and A. Sarkar, editors, *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 17–24, Tübingen, Germany, 2008.

[7] N. Chomsky. *The Minimalist Program*. The MIT Press, Cambridge, MA, 1995.

[8] N. Chomsky. Minimalist inquiries: The framework. In R. Martin, D. Michaels, and

J. Uriagereka, editors, *Step by Step*, pages 89–155. The MIT Press, Cambridge, MA, 2000.

[9] R.-M. Déchaine and M. Wiltschko. Deriving reflexives. In L. Mikkelsen and C. Potts, editors, *Proceedings of the 21st West Coast Conference on Formal Linguistics*, pages 71–84. Cascadilla Press, 2002.

[10] D. Fox. *Economy and Semantic Interpretation*. MIT Press, Cambridge, MA, 1999.

[11] R. Frank. *Phrase Structure Composition and Syntactic Dependencies*. MIT Press, Cambridge, MA, 2002.

[12] C. Gardent and L. Kallmeyer. Semantic Construction in F-TAG. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 123–130, Budapest, 2003.

[13] C.-h. Han. Pied-piping in relative clauses: Syntax and compositional semantics using Synchronous Tree Adjoining Grammar. *Research on Language and Computation*, 5(4):457–479, 2007.

[14] C.-h. Han, D. Potter, and D. R. Storoshenko. Compositional semantics of coordination using Synchronous Tree Adjoining Grammar. In C. Gardent and A. Sarkar, editors, *Proceedings of the 9th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 33–41, Tübingen, Germany, 2008.

[15] C.-h. Han and D. R. Storoshenko. Semantic binding of long-distance anaphor *caki* in Korean. *Language*, 88(4), 2012.

[16] I. Heim and A. Kratzer. *Semantics in Generative Grammar*. Blackwell Publishers, Oxford, 1998.

[17] J. Higginbotham. On semantics. *Linguistic Inquiry*, 16:547–594, 1985.

[18] A. K. Joshi, T. Becker, and O. Rambow. Complexity of scarmbling: A new twist to the competence-performance distinction. In A. Abeillé and O. Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing*, pages

167–181. CSLI Publications, Stanford, 2000.

[19] A. K. Joshi, L. Levy, and M. Takahashi. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10:136–163, 1975.

[20] A. K. Joshi and Y. Schabes. Tree-Adjoining Grammars. In *Handbook of Formal Languages*, pages 69–123. Springer, Berlin, 1997.

[21] L. Kallmeyer and M. Romero. LTAG semantics with semantic unification. In O. Rambow and M. Stone, editors, *Proceedings of the 7th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 155–162, Vancouver, Canada, 2004.

[22] L. Kallmeyer and M. Romero. Reflexive and reciprocals in LTAG. In J. Geertzen, E. Thijsse, H. Bunt, and A. Schiffrin, editors, *Proceedings of the 7th International Workshop on Computational Semantics*, pages 271–282, Tilburg, 2007.

[23] C. Kennedy. Antecedent-contained deletion and the syntax of quantification. *Linguistic Inquiry*, 28(4):662–688, 1997.

[24] A. Kroch and A. Joshi. The linguistic relevance of Tree Adjoining Grammar. Technical Report MS-CS-85-16, Department of Computer and Information Sciences, University of Pennsylvania, 1985.

[25] R. May. *The Grammar of Quantification*. PhD thesis, MIT, 1977.

[26] R. May. *Logical Form*. MIT Press, Cambridge, MA, 1985.

[27] R. Nesson. *Synchronous and Multicomponent Tree-Adjoining Grammars: Complexity, Algorithms, and Applications*. PhD thesis, Harvard University, 2009.

[28] R. Nesson and S. M. Shieber. Simpler TAG Semantics through Synchronization. In *Proceedings of the 11th Conference on Formal Grammar*, Malaga, Spain, 2006. CSLI.

[29] R. Nesson and S. M. Shieber. Efficiently parsable extensions to Tree-Local Multicomponent TAG. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Lin-

guistics, pages 92–100. Association for Computational Linguistics, 2009.

[30] O. Rambow, K. Vijay-Shanker, and D. Weir. D-Tree Grammars. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 151–158. Association for Computational Linguistics, 1995.

[31] T. Reinhart and E. Reuland. Reflexivity. *Linguistic Inquiry*, 24(4):657–720, 1993.

[32] N. Ryant and T. Scheffler. Binding of anaphors in LTAG. In L. Kallmeyer and T. Becker, editors, *Proceedings of the 8th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 65–72, Sydney, Australia, 2006. COLING-ACL workshop.

[33] Y. Schabes and S. M. Shieber. An Alternative Conception of Tree-Adjoining Derivation. *Computational Linguistics*, 20(1):91–124, 1994.

[34] S. Shieber. Restricting the weak-generative capacity of Synchronous Tree-Adjoining Grammars. *Computational Intelligence*, 10(4):371–385, 1994.

[35] S. M. Shieber and Y. Schabes. Synchronous Tree Adjoining Grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, volume 3, pages 253–258, Helsinki, Finland, 1990.

[36] D. R. Storoshenko. *A Cross-Linguistic Account of Reflexivity Using Synchronous Tree Adjoining Grammar*. PhD thesis, Simon Fraser University, 2010.

[37] D. Weir. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. PhD thesis, University of Pennsylvania, 1988.