Exploring the Underspecified World of Lexicalized Tree Adjoining Grammars

K. Vijay-Shanker
Department of Computer & Information Science
University of Delaware
Newark, DE 19716
vijay@udel.edu

David Weir
School of Cognitive & Computing Sciences
University of Sussex
Brighton, BN1 9HQ, UK.
david.weir@cogs.susx.ac.uk

Abstract

This paper presents a precise characterization of the underspecification found in Lexicalized Tree Adjoining Grammars, and shows that, in a sense, the same degree of underspecification is found in Lexicalized D-Tree Substitution Grammars. Rather than describing directly the nature of the elementary objects of the grammar, we achieve our objective by formalizing the way in which underspecification in the *derived* objects is interpreted: i.e., how trees are read off from derived tree descriptions. Valid tree descriptions for LTAG turn out to be those that have a single acceptable interpretation, whereas those for LDSG may have multiple interpretations. In other respects, there is no difference in the way in which LTAG and LDSG tree descriptions are interpreted.

1 Introduction

The main component of a Lexicalized Tree Adjoining Grammar (LTAG) (Joshi and Schabes, 1991) are its elementary trees. These provide possible syntactic contexts for the lexical items that anchor them. One of the basic tenets of the formalism is that each elementary tree should include direct references to those components of derived trees that are in some way dependent on the anchor. For example, the elementary tree for a transitive verb would include references to nodes that, in a derived tree, will be at the root of the subtrees for the verb's subject and object.

In order to achieve this, the formalism is being asked to perform a delicate balancing act: the elementary trees should say everything that can be said about the anchor's context (given the identify of the anchor); however, this should be done in such a way that there are no unwanted side-effects. In other words, in describing the relationship between the anchor and one of its dependents, we do not want the notation we are using to force us to make commitments that, otherwise, need not be made.

For example, it would not be desirable if, in referring to a node for a verb's extracted object (the node marked with a † in Figure 1), we were forced to say that, in *every* derived tree it must remain a sibling of the grandparent of the verb (the node marked with $\diamond$ in Figure 1): this would leave no room for intervening structure.

If substitution were the only composition operation in LTAG, this would be precisely the kind of problem that could arise. The LTAG formalism addresses this by including an second operation (adjunction) with which tree structure can be inserted (adjoined) inside other tree structure. Adjunction makes it possible for the relationship between nodes to change during a derivation: for example, when a tree for a raising verb is adjoined, the extracted object will become the sibling of the verb's great-grandparent. Hence, the potential for adjunction at nodes in a tree means that the domination relationships among nodes in elementary trees are, in effect, underspecified.

Underspecification can be exploited more explicitly by adopting an approach suggested by work on D-Theory parsing (Marcus et al., 1983). The basis of D-Theory is that, by manipulating *descriptions* of trees rather than the trees themselves, it is possible to use underspecification in the descriptions to avoid the need to make unwanted commitments about tree structure. Vijay-Shanker (1992) considered how this methodology could be applied to LTAG, and in related work (Rogers and Vijay-Shanker, 1992), a logic for tree description was developed that could be used to formalize these underspecified tree descriptions. Viewed from this perspective, the elementary objects of LTAG can be taken to be *partial* descriptions of derived trees that the grammar could potentially generate for a sentence containing the anchor of the tree. The elementary tree in Figure 1 can be viewed as describing a number domination relationships, that could be further specified during the derivation process.

When LTAG is viewed in this way, certain aspects of the formalism appear somewhat arbitrary (for example, the fact that the root and foot of auxiliary trees must be labelled by the same nonterminal symbol). This has led to the development of a variant of LTAG called Lexicalized D-Tree Substitution Grammars (LDSG) (Rambow et al., 1995). LDSG employs

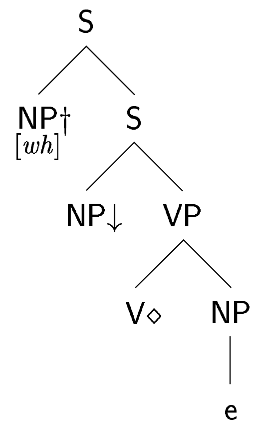

Figure 1: An elementary tree

only the substitution operation, since, by using descriptions of trees as the elementary objects of the grammar, the required underspecification is encapsulated directly in the description, making adjunction apparently unnecessary.

However, the added flexibility of this approach gives rise to difficult design issues. By choosing to express the elementary structures with a general purpose tree description logic, the potential for underspecification becomes enormous. Designing a formalism (as opposed to writing an individual grammar) involves deciding on the degree of underspecification to be allowed in the general case, and this can call for what can seems like rather arbitrary decisions. The goal of this paper is to explain the choices made in LTAG and LDSG and to characterize the degree of underspecification that they can tolerate.

Our approach revolves around the rather obvious observation that, if fully specified trees are to be produced at the end of a derivation, any underspecification in the elementary structures must be resolved at some stage in the derivation process. There are two places where this can happen: when the elementary descriptions are being combined by means of the composition operations, or after all the elementary objects have been assembled into a derived description. In the latter case, the formalism must provide some way of interpreting any underspecification that remains in the derived descriptions. This is not simply a matter of making valid logical deductions in order to find all fully specified trees compatible with the description.

The formalism embodies a set of *assumptions* regarding how to interpret the underspecification found in derived descriptions. These assumptions are implemented in what we will refer to as the formalism's **derived description reader** (ddr): a process that reads off trees from derived descriptions. In this paper, we specify the assumptions that determine how the LTAG ddr resolves underspecification. We find that valid LTAG tree descriptions have the property that the LTAG ddr gives them a single reading. We find that the same set of assumptions can be used to resolve a wider variety of underspecification than is found in derived LTAG tree descriptions. In fact, it is just this form of underspecification that is to be found in the derived descriptions produced by LDSG. Hence, by removing the requirement that descriptions have a unique reading, we obtain a characterization of the valid LDSG tree descriptions.

Earlier work has addressed the problem of characterizing the tree descriptions of LTAG (Rogers and Vijay-Shanker, 1992; Rogers and Vijay-Shanker, 1994). This work involved stating syntactic conditions that apply to valid descriptions, but did not manage to accurately capture the full range of tree descriptions used in LTAG. The approach we take here us more directly related to that of Rogers and Vijay-Shanker (1993). Their goal was to describe the nature of the nondeterminism resulting from underspecification that D-theory parsers can accommodate. To do this they used the notion of the standard referent of a description. Their claim was that those tree descriptions that have unique standard referents are exactly the descriptions that satisfy the determinism hypothesis of D-theory parsing. This resembles the approach taken here, since it concerns how underspecified tree descriptions are interpreted as trees, however, no single notion of a standard referent was specified.

2 Tree Adjunction and Underspecification

Consider the elementary tree γ shown in Figure 2. We can express this in a language of tree description (Rogers and Vijay-Shanker, 1992) with the description d_γ below, where the expression $x \triangle y$ should be read as x is the parent of (immediately

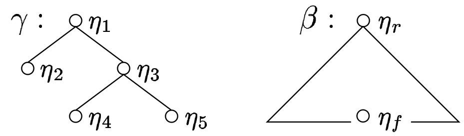

Figure 2: A Pair of Elementary Trees

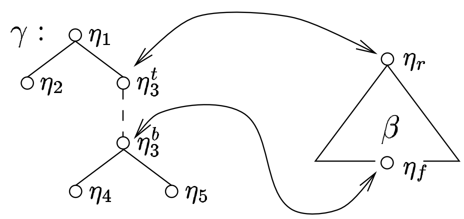

Figure 3: Adjunction of a Quasi Tree

dominates) y, and $x \prec y$ should be read as the node x precedes (is to the left of) the node y.

$$d_\gamma \quad : \quad x \vartriangle y \wedge x \vartriangle z \wedge y \prec z \wedge z \vartriangle u \wedge z \vartriangle v \wedge u \prec v \qquad (1)$$

Let s_γ be a mapping from vars(d_γ) (the variables in d_γ) to nodes in γ such that $s_\gamma(x) = \eta_1$, $s_\gamma(y) = \eta_2$, $s_\gamma(z) = \eta_3$, $s_\gamma(u) = \eta_4$ and $s_\gamma(v) = \eta_5$. We can express the fact that d_γ describes the tree γ by saying that γ, s_γ satisfies d_γ, denoted $\gamma, s_\gamma \models d_\gamma$.

Suppose that the auxiliary tree β shown in Figure 2 can be adjoined into γ at η_3 resulting in the tree γ'. Let d_β be a description and s_β be a variable mapping such that $\beta, s_\beta \models d_\beta$. Can we construct a description $d_{\gamma'}$ and a variable mapping $s_{\gamma'}$ such that $\gamma', s_{\gamma'} \models d_{\gamma'}$?

To do this we need to clarify which nodes in γ' correspond to $\eta_1, \ldots, \eta_5$. For η_1, η_2, η_4 and η_5 this is unproblematic, but the node η_3 has in effect been equated with both the root and foot nodes of β and therefore "corresponds" to two distinct nodes in γ'.

If we want to think of elementary trees as tree descriptions, and define adjunction as a monotonic operation on tree descriptions, we need to anticipate the possibility of adjunction at a node by using two names for that node. In the event that an adjunction takes place, these will refer to the root and foot of the adjoined tree, respectively. When adjunction does not occur, these two names are taken to refer to the same node. Vijay-Shanker (1992) called these **quasi nodes**, and descriptions involving such pairs **quasi trees**.

In expressing the elementary tree γ in the language of tree descriptions, we are faced with the problem of finding a single description that is compatible with both the adjunction and no adjunction scenario. This is achieved by under specifying domination. Instead of statement 1, the following description can be used, where $x \vartriangle^* y$ should be read as: the node x dominates the node y.

$$d_\gamma \quad : \quad x \vartriangle y \wedge x \vartriangle z \wedge y \prec z \wedge z \vartriangle^* z' \wedge z' \vartriangle u \wedge z' \vartriangle v \wedge u \prec v \qquad (2)$$

This description is shown graphically in Figure 3, where η_3^t and η_3^b are the two names used instead of η_3 and a broken line is used to indicate domination.

Specifying that η_3^t dominates η_3^b leaves open the possibility that they refer to the same node, while being compatible with the presence of intervening nodes brought about by adjunction. Figure 3 illustrates adjunction graphically. Every node in an elementary tree at which adjunction can take place is split in two in this way.

We now return to the problem of how to build a description of γ', the result of adjoining β at η_3 in γ. Let d'_γ and d'_β be copies of d_γ and d_β such that vars(d'_γ) $\cap$ vars(d'_β) $= \emptyset$. Also, let s_γ and s_β be such that

$$d'_\gamma, s_\gamma \models \gamma \text{ and } d'_\beta, s_\beta \models \beta$$

and let $x, y \in vars(d'_\gamma)$ and $u, v \in vars(d'_\beta)$ be such that $s_\gamma(x)$ and $s_\gamma(y)$ are η_3^t and η_3^b, respectively, and $s_\beta(u)$ and $s_\beta(v)$ are the η_r and η_f, the root and foot of β, respectively (see Figure 3).

The description $d_{\gamma'}$ describing γ' is produced by taking the conjunction of d'_γ with the result of substituting x for u and y for v in d'_β. This description is satisfied by the tree γ' with the variable mapping that is the union of s_γ and s_β.

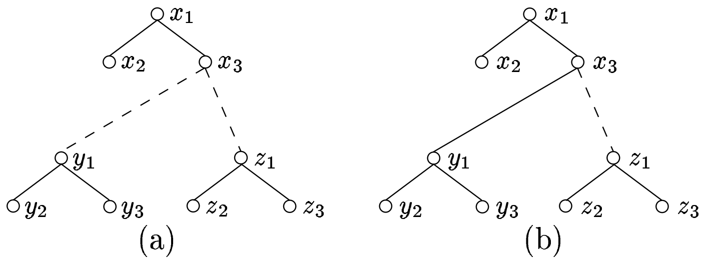

Figure 4: Unreadable underspecification

The operation of substituting one tree description into another can be defined in a similar way, again involving taking the conjunction of two descriptions. The only difference is that in this case, only one pair of nodes are unified. Given a LTAG grammar, the set of derived descriptions is produced by closing the set of elementary descriptions under adjunction and substitution.

How is the underspecification found in the elementary tree descriptions resolved: is it removed during the derivation process? Although adjoining a tree between two quasi nodes further refines the domination relationship that holds between them, the possibility that adjunction can take place in unboundedly embedded contexts means that, the derivation process will not be able to complete the domination relations once and for all. In other words, the result of a derivation is not a fully specified tree, since they can contain underspecified domination relations that could get further specified by additional adjunctions. That is, they exhibit exactly the same kind of underspecification that was introduced in the elementary descriptions.

So how should fully specified trees be read off from these underspecified descriptions? We call the process that achieves this the LTAG derived description reader (ddr). Obviously, since we are modelling LTAG, we know that if no adjunction takes place at a pair of quasi nodes, the LTAG ddr should take them to refer to the same node. However, derived descriptions describe an *infinite* number of trees. One possibility is that a parenthood relation be assumed between all pairs of quasi nodes; this is just as plausible, from a logical point of view, as the assumption that the two quasi nodes are the same node. In order that the LTAG ddr produce only the desired readings of derived descriptions, it must embody various assumptions as to what kind of underspecification will be encountered. We formulate this as a set of constraints applying to any description that determine which readings are permitted.

3 The Derived Description Reader

In this section we specify exactly those trees that LTAG ddr will read off from a derived description. The language used for tree descriptions involves the use of predicate symbols, $\triangle$, $\triangle^*$, and $\prec$. Clearly, not all structures appropriate for this language correspond to trees. We refer to Backofen et al. (1995) for a characterization of those structures which correspond to trees. The descriptions under consideration are conjunctions of positive literals involving the predicates $\triangle$, $\triangle^*$ and $\prec$. Suppose that d is such a description. A tree γ with associated variable mapping s will obtained from a derived description d by the LTAG ddr if and only if the following conditions hold. In what follows, $d \Rightarrow d'$ is used to indicate that the description d' logically follows from d, in other words, that d' is known in d.[1]

1. $\gamma, s \models d$.

   This condition guarantees that γ is consistent with the description d.

2. $s(x) = s(y)$ implies $d \Rightarrow (x \triangle^* y \vee y \triangle^* x)$ for all $x, y \in \text{vars}(d)$

   This condition states that all equated nodes are known to be in a domination relationship in d.

3. $\gamma, s \models x \triangle y$ iff $s(x') = s(x)$, $s(y') = s(y)$ and $d \Rightarrow x' \triangle y'$ for some $x', y' \in \text{vars}(d)$

   This condition states that all parenthood relation in t are known d. Note that this condition prevents the inclusion of nodes in t that have no counterparts in d.

4. for all $x, y, x', y' \in \text{vars}(d)$,
   if $s(x) = s(x')$, $s(y) = s(y')$ and $d \Rightarrow (x \triangle y \wedge x' \triangle y')$
   then
   $d \Rightarrow (x = x' \wedge y = y')$
   where $x = y$ is syntactic sugar for
   $x \triangle^* y \wedge y \triangle^* x$.

[1] $d \Rightarrow d'$ iff $d \wedge \neg d'$ is not satisfied by any tree model.

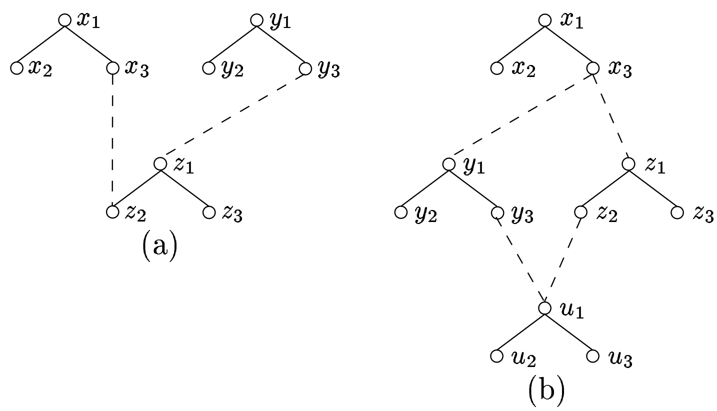

Figure 5: Positioning of D-edges

This condition, in conjunction with condition 3, states that there is a one-to-one correspondence between the parenthood relationships found in t and parenthood relationships known in d.

5. $\gamma, s \models (z \bigtriangleup x \wedge z \bigtriangleup y \wedge x \prec y)$
   iff $s(x) = s(x')$,
   $s(y) = s(y')$,
   $s(z) = s(z')$ and
   $d \Rightarrow (z' \bigtriangleup x' \wedge z' \bigtriangleup y' \wedge x' \prec y')$
   for some $x', y', z' \in \text{vars}(d)$

   This condition states that all siblinghood relations and precedence relations among siblings found in t are known in d.

In the remainder of this section we consider why these conditions are needed in order to properly interpret derived quasi trees. Condition 2 is a direct way of indicating that the ddr should only consider equating two quasi nodes that are related by domination in some way. This captures the fact that the potential to equate quasi nodes is licensed by the fact that they are put in a domination relation in an elementary tree.

Condition 3 states that the ddr should not be allowed to invent parenthood relationships that are not determined in the derived quasi tree. This captures the fact that all parent relationships in derived trees must be licensed by some parent relationship in an elementary quasi tree.

Condition 4 indicates that all parent relations should be treated as referring to distinct pairs of nodes. The ddr should not be allowed to overlap different different bits of tree structure.

Finally, condition 5 indicates that precedence relationships among siblings must be licensed by precedence relationships among siblings in the derived quasi tree, which in turn means that these precedence relationships must have been licensed in elementary quasi trees.

Given a description d, we say that $\langle \gamma, s \rangle \in \mathsf{ddr}(d)$ iff γ, s is obtained from the description d as stated above.

4 Quasi Trees

Having specified the LTAG ddr we take the definition of quasi trees to be those descriptions for which the LTAG ddr gives exactly one tree.

In characterizing quasi trees in terms of unique readability, we do not wish to distinguish between structures that are isomorphic. We write $\gamma_1 \approx \gamma_2$ when the two structures are isomorphic. Given two structures, γ_1 and γ_2 let h be an isomorphism from the domain of γ_1 to the domain of γ_2 witnessing $\gamma_1 \approx \gamma_2$. Then, given a description d, we say $\langle \gamma_1, s_1 \rangle \approx_d \langle \gamma_2, s_2 \rangle$ iff for all $x \in \text{vars}(d)$, $h(s_1(x)) = s_2(x)$. $\approx_d$ is an equivalence relation, and let $[\langle \gamma, s \rangle]$ be the equivalence class containing $\langle \gamma, s \rangle$. Finally, let $\mathsf{ddr}_{\approx}(d) = \{[\langle \gamma, s \rangle] \mid \langle \gamma, s \rangle \in \mathsf{ddr}(d)\}$.

We can now state that a quasi tree d is a description such that the cardinality of $\mathsf{ddr}_{\approx}(d)$ is one.

There are two important characteristics of quasi trees given in Vijay-Shanker (1992): (1) top quasi nodes (which dominate their bottom quasi node) cannot, in addition, immediately dominate another node; and (2) top quasi nodes cannot dominate more than one node. Examples of this are illustrated in the two descriptions shown in Figure 4.[2] What would the LTAG ddr make of these?

When presented with the description in Figure 4a the LTAG ddr cannot add new parent relationships between x_3 and either y_1 or z_1 (see condition 3). The only alternative would be to equate x_3 with one or both of y_1 or z_1. Suppose x_3 and y_1 are equated, in which case, z_1 must be equated with x_3, y_2 or y_3. The first possibility (equating z_1 and x_3) is

[2] Note that when we display a description graphically, precedence among siblings is assumed to be fully specified, and will correspond to the left-to-right order shown.

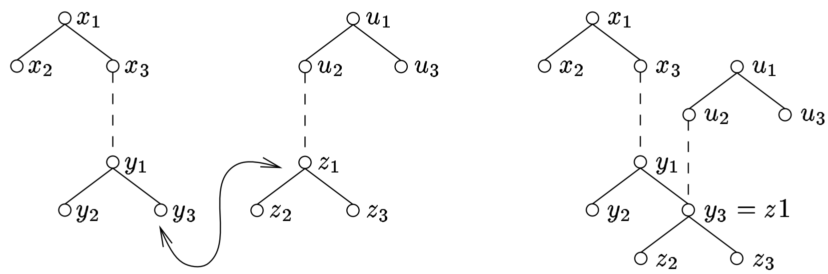

Figure 6: Substitution of D-Trees

ruled out by the fact that either condition 5 would be violated because siblinghood relations would be asserted that are not licensed in the description, or the children of y_1 and z_1 would be merged violating condition 4. Equating z_1 with either y_2 or y_3 is ruled out by condition 2, since there is no known domination relationship between these nodes. Thus, there is no way that the LTAG ddr can interpret this description. Similar reasoning can be used to show that the description in Figure 4b cannot be interpreted.

Notice that in reasoning that these two descriptions are uninterpretable by the LTAG ddr, we did not appeal to the unique reading condition. In the next section we show that there are descriptions that can be read by the LTAG ddr, but which are not given unique readings? First however, we consider some descriptions that the LTAG ddr gives a single reading, but which are not normally considered to be quasi trees.

$$x \vartriangle^* y \wedge x \vartriangle^* z \quad (3)$$

$$x \vartriangle^* y \wedge z \vartriangle^* y \quad (4)$$

Both (3) and (4) are given a single reading by the LTAG ddr: a tree consisting of a single node. What anomalous descriptions such as these have in common is that they involve nodes for which neither a parent or child is known. This can be resolved by including an additional constraint on quasi trees disallowing descriptions involving nodes with neither a parent or child. This is justifiable on the grounds that each local tree in an LTAG elementary tree is intended to involve the combination of a head with a complement or modifier with underspecification of domination being used to allow for structure between these local trees. As a result, only a single quasi node would be permitted at the root or foot of LTAG elementary trees. This is consistent with the view that in producing elementary trees by projecting from the lexical anchor we have no information about such quasi nodes, since they are not really be part of the anchor's domain of locality.

5 Beyond Quasi Trees

Consider how the LTAG ddr would interpret the descriptions shown in Figure 5. In the case of the description in Figure 5a, the variables x_3 and z_2 cannot be equated since that would mean that the ddr would have to merge the parent relationships between x_1 and x_3 and between z_1 and z_2 which is not allowed by condition 4. However, x_3 and z_1 *can* be equated, in which case y_3 and x_1 can then be equated. This gives a valid reading of the tree description. However, there is a second possibility: the variables y_3 and z_1 can be equated, in which case x_3 and y_1 are also equated.

The description in Figure 5b can also be interpreted by the LTAG ddr despite the fact that x_3 dominates two distinct nodes. There are two valid interpretations of this description: one that equates x_3 with y_1 and y_3 with z_1 and z_2 with u_1; a second that equates x_3 with z_1 and z_2 with y_1 and y_3 with u_1. The crucial difference between this description and the one given in Figure 4a is that the domination statements involving y_3 and z_2 in Figure 5b have the effect of licensing the exact positioning of the nodes y_1, y_2 and y_3 under z_1 (by allowing y_1 and z_2 to be equated) or the positioning of the nodes z_1, z_2 and z_3 under y_1 (by allowing z_1 and y_3 to be equated).

Thus, in Figure 5 we see examples of tree descriptions that the LTAG ddr can interpret, but for which it gives multiple readings. This gives us a precise characterization of a set of tree descriptions that is larger than the set of quasi trees: i.e., the set of those descriptions for which the LTAG ddr gives at least one reading. We define this set of descriptions to be the valid LDSG tree descriptions.

The LDSG formalism is similar to LTAG except that the only composition operation employed is substitution. Substitution of descriptions is defined as follows. For a description d a node $x \in \text{vars}(d)$ is a **substitution node** of d if for all $y \in \text{vars}(d)$ such that $x \neq y$ it is not the case that $d \Rightarrow x \vartriangle^* y$. Also, $x \in \text{vars}(d)$ is a **component root** of d if for

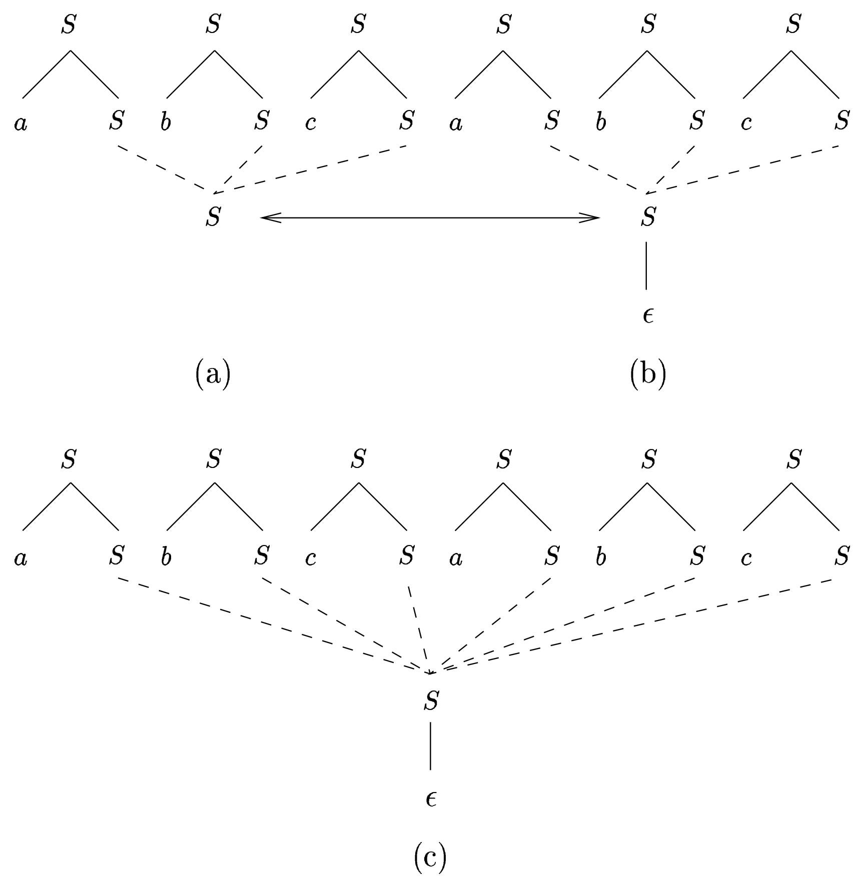

Figure 7: A grammar for Mix

$y \in \text{vars}(d)$ it is not the case that $d \Rightarrow y \triangle x$. Suppose we have two d-trees d_1 and d_2 where d_1 and d_2 involve distinct variables. Let x be a substitution node in d_1 and y be a component root of d_2. The result of substituting node y of d_2 at node x of d_1 is the d-tree that is the conjunction of d_2 with the result of replacing all occurrences of x in d_1 by y. An example of substitution is shown in Figure 6.

Note that, crucially, the set of valid LDSG trees, as defined above, is closed under this substitution operation. Hence, if the elementary descriptions of an LDSG are all valid LDSG trees, then the result of a derivation will necessarily be a valid LDSG tree, and the LTAG ddr will be able to read off at least one tree from it.

6 Underspecification of Linear Precedence

This paper has been concerned with discussing the kind of underspecification found in LTAG and LDSG, and this has involved the underspecification of domination. Another candidate for underspecification that is often considered is the underspecification of precedence. One place where this has been particularly useful is in the description of languages in which the order of symbols is very free.

While it would be relatively straightforward to extend the LTAG ddr so that it could interpret descriptions in which precedence among siblings was underspecified, we end this paper by showing that underspecification of domination can be used to achieve a similar effect.

Consider the LDSG elementary descriptions shown in Figure 7a and Figure 7b. These generate the language

$$\{w \in \{a, b, c\}^+ \mid w \text{ contains an equal number of } a\text{'s, } b\text{'s and } c\text{'s}\}$$

The description shown in Figure 7c has one reading for every possible ordering of the symbols.

7 Conclusions

The use of underspecified tree descriptions has been suggested in the context of Tree Adjoining Grammars and some of its extensions. However, there has not been a precise characterization of exactly what kind of underspecification is needed. In this paper, we have addressed this question, offering a new approach which we apply to both LTAG and LDSG. This brings out the relationship between the degree and type of underspecification in these two frameworks, and we believe that it could be applied more generally, and can be used to relate underspecification of precedence and domination to account for constructions involving flexible word order.

References

Rolf Backofen, James Rogers, and K. Vijay-Shanker. 1995. A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language, and Information*, 4(1):5–39.

Aravind K. Joshi and Yves Schabes. 1991. Tree-adjoining grammars and lexicalized grammars. In Maurice Nivat and Andreas Podelski, editors, *Definability and Recognizability of Sets of Trees*. Elsevier.

Mitchell Marcus, Donald Hindle, and Margaret Fleck. 1983. D-theory: Talking about talking about trees. In *Proceedings of the 21st Annual Meeting of the Association f or Computational Linguistics*, Cambridge, Mass.

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

James Rogers and K. Vijay-Shanker. 1992. Reasoning with descriptions of trees. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, pages 72–80.

James Rogers and K. Vijay-Shanker. 1993. Towards a formal understanding of the determinism hypothesis in d-theory. In *Proceedings of the 3rd International Workshop on Parsing Technologies.*

James Rogers and K. Vijay-Shanker. 1994. Obtaining trees from their descriptions: An application to Tree-Adjoining Grammars. *Computational Intelligence*, 10(4):401–422.

K. Vijay-Shanker. 1992. Using descriptions of trees in a Tree Adjoining grammar. *Computational Linguistics*, 18(4):481–517, December.