

# Reasoning with Descriptions of Trees \*

James Rogers  
Dept. of Comp. & Info. Science  
University of Delaware  
Newark, DE 19716, USA

K. Vijay-Shanker  
Dept. of Comp. & Info. Science  
University of Delaware  
Newark, DE 19716, USA

## ABSTRACT

In this paper we introduce a logic for describing trees which allows us to reason about both the parent and domination relationships. The use of domination has found a number of applications, such as in deterministic parsers based on Description theory (Marcus, Hindle & Fleck, 1983), in a compact organization of the basic structures of Tree-Adjoining Grammars (Vijay-Shanker & Schabes, 1992), and in a new characterization of the adjoining operation that allows a clean integration of TAGs into the unification-based framework (Vijay-Shanker, 1992) Our logic serves to formalize the reasoning on which these applications are based.

## 1 Motivation

Marcus, Hindle, and Fleck (1983) have introduced *Description Theory* (D-theory) which considers the structure of trees in terms of the domination relation rather than the parent relation. This forms the basis of a class of deterministic parsers which build partial descriptions of trees rather than the trees themselves. As noted in (Marcus, Hindle & Fleck, 1983; Marcus, 1987), this approach is capable of maintaining Marcus' deterministic hypothesis (Marcus, 1980) in a number of cases where the original deterministic parsers fail.

A motivating example is the sentence: *I drove my aunt from Peoria's car.* The difficulty is that a deterministic parser must attach the NP "my aunt" to the tree it is constructing before evaluating the PP. If this can only be done in terms of the parent relation, the NP will be attached to the VP as its object. It is not until the genitive marker on "Peoria's" is detected that the correct attachment is clear. The D-theory parser avoids the trap by making only the judgment that the VP dominates the NP by a path of length *at least* one. Subsequent refinement can either add intervening components or not. Thus in this case, when "my aunt" ends up as part of the determiner of the object rather than the object itself, it is not inconsistent with its original placement. It is still dominated by the VP, just not immediately. When the analysis is complete, a tree, the *standard referent*, can be extracted from the description by taking immediate domination as the parent relation.

In other examples given in (Marcus, Hindle & Fleck, 1983) the left-of (linear precedence) relation is partially specified during parsing, with individuals related by "left-of or equals" or "left-of or dominates". The important point is that once a relationship is asserted, it is never subsequently rescinded. The D-theory parser builds structures which are always a partial description of its final product. These structures are made more specific, as parsing proceeds, by adding additional relationships.

Our understanding of the difficulty ordinary deterministic parsers have with these constructions is that they are required to build a structure covering an initial segment of the input at a time when there are multiple distinct trees that are consistent with that segment. The D-theory parsers succeed by building structures that contain only those relationships that are common to all the consistent trees. Thus the choice between alternatives for the relationships on which the trees differ is deferred until they are distinguished by the input, possibly after semantic analysis.

A similar situation occurs when Tree-Adjoining Grammars are integrated into the unification-based framework. In TAGs, syntactic structures are built up from sets of elementary trees by the *adjunction* operation, where one tree is inserted into another tree in place of one of its nodes. Here the difficulty is that adjunction is non-monotonic in the sense that there are relationships that hold in the trees being combined that do not hold in the resulting tree. In (Vijay-Shanker, 1992), building on some of the ideas from D-theory, a version of TAG is introduced which resolves this by manipulating partial descriptions of trees, termed *quasi-trees*. Thus an elementary structure for a transitive verb might be the quasi-tree  $\alpha'$  rather than the tree  $\alpha$  (Figure 1). In  $\alpha'$  the separation represented by the dotted line between nodes referred to by  $vp_1$  and  $vp_2$  denotes a path of length greater than *or equal to* zero. Thus  $\alpha'$  captures just those relationships which are true in  $\alpha$  and in all trees derived from  $\alpha$  by adjunction at VP. In this setting trees are extracted from quasi-trees by taking what is termed a *circumscriptive reading*, where each pair of nodes in which one dominates the other by a path that is possibly zero is identified.

This mechanism can be interpreted in a manner similar to our interpretation of the use of partial

\*This work is supported by NSF grant IRI-9016591

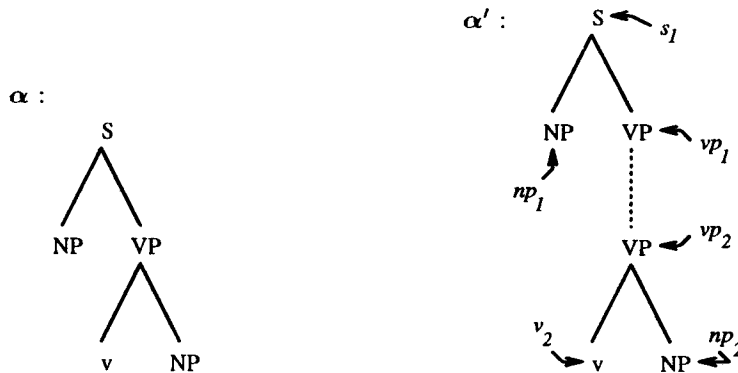


Figure 1. Quasi-trees

descriptions in D-theory parsers. We view a tree in which adjunction is permitted as the set of all trees which can be derived from it by adjunction. That set is represented by the quasi-tree as the set of all relationships that are common to all of its members.

The connection between partial descriptions of trees and the sets of trees they describe is made explicit in (Vijay-Shanker & Schabes, 1992). Here quasi-trees are used in developing a compact representation of a Lexicalized TAG grammar. The lexicon is organized hierarchically. Each class of the hierarchy is associated with that set of relationships between individuals which are common to all trees associated with the lexical items in the class but not (necessarily) common to all trees associated with items in any super-class. Thus the set of trees associated with items in a class is characterized by the conjunction of the relationships associated with the class and those inherited from its super-classes. In the case of transitive verbs, figure 2, the relationships in  $\alpha_1$  can be inherited from the class of all verbs, while the relationships in  $\alpha_2$  are associated only with the class of transitive verbs and its sub-classes.

The structure  $\alpha'$  of figure 1 can be derived by combining  $\alpha_2$  with  $\alpha_1$  along with the assertion that  $v_2$  and  $v_1$  name the same object. In any tree described by these relationships either the node named  $vp_1$  must dominate  $vp_2$  or vice versa. Now in  $\alpha_1$ , the relationship “ $vp_1$  dominates  $v_1$ ” does not itself preclude  $vp_1$  and  $v_1$  from naming the same object. We can infer, however, from the fact that they are labeled incompatibly that this is not the case. Thus the path between them is at least one. From  $\alpha_2$  we have that the path between  $vp_2$  and  $v_2$  is precisely one. Thus in all cases  $vp_1$  must dominate  $vp_2$  by a path of length greater than or equal to zero. Hence the dashed line in  $\alpha'$ .

The common element in these three applications is the need to manipulate structures that partially describe trees. In each case, we can understand this as a need to manipulate sets of trees. The structures, which we can take to be quasi-trees in each case, represent these sets of trees by capturing

the set of relationships that are common to all trees in the set. Thus we are interested in quasi-trees not just as partial descriptions of individual trees, but as a mechanism for manipulating sets of trees.

Reasoning, as in the LTAG example, about the structures described by combinations of quasi-trees requires some mechanism for manipulating the quasi-trees formally. Such a mechanism requires, in turn, a definition of quasi-trees as formal structures. While quasi-trees were introduced in (Vijay-Shanker, 1992), they have not been given a precise definition. The focus of the work described here is a formal definition of quasi-trees and the development of a mechanism for manipulating them.

In the next section we develop an intuitive understanding of the structure of quasi-trees based on the applications we have discussed. Following that, we define the syntax of a language capable of expressing descriptions of trees as formulae and introduce quasi-trees as formal structures that define the semantics of that language. In section 4 we establish the correspondence between these formal models and our intuitive idea of quasi-trees. We then turn to a proof system, based on semantic tableau, which serves not only as a mechanism for reasoning about tree structures and checking the consistency of their descriptions, but also serves to produce models of a given consistent description. Finally, in section 7 we consider mechanisms for deriving a representative tree from a quasi-tree. We develop one such mechanism, for which we show that the tree produced is the circumscriptive reading in the context of TAG, and the standard referent in the context of D-theory. Due to space limitations we can only sketch many of our proofs and have omitted some details. The omitted material can be found in (Rogers & Vijay-Shanker, 1992).

## 2 Quasi-Trees

In this section, we use the term *relationship* to informally refer to any positive relationship between individuals which can occur in a tree, “ $a$  is the parent of  $b$ ” for example. We will say that a tree *satisfies* a relationship if that relationship is true of the individuals it names in that tree.

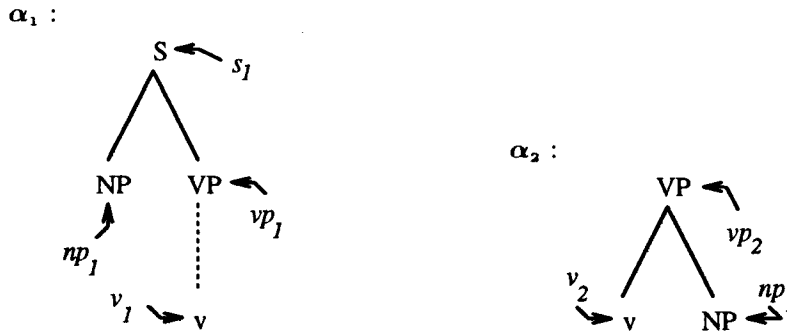


Figure 2. Structure Sharing in a Representation of Elementary Structures

It's clear, from our discussion of their applications, that quasi-trees have a dual nature — as a set of trees and as a set of relationships. In formalizing them, our fundamental idea is to identify those natures. We will say that a tree is (partially) *described* by a set of relationships if every relationship in the set is true in the tree. A set of trees is then described by a set of relationships if each tree in the set is described by the set of relationships. On the other hand, a set of trees is *characterized* by a set of relationships if it is described by that set and if every relationship that is common to all of the trees is included in the set of relationships. This is the identity we seek; the quasi-tree viewed as a set of relationships characterizes the same quasi-tree when viewed as a set of trees.

Clearly we cannot easily characterize arbitrary sets of trees. As an example, our sets of trees will be upward-closed in the sense that, it will contain every tree that extends some tree in the set, i.e. that contains one of the trees as an initial sub-tree. Similarly quasi-trees viewed as sets of relationships are not arbitrary either. Since the sets they characterize consist of trees, some of the structural properties of trees will be reflected in the quasi-trees. For instance, if the quasi-tree contains both the relationships “a dominates b” and “b dominates c” then every tree it describes will satisfy “a dominates c” and therefore it must contain that relationship as well. Thus many inferences that can be made on the basis of the structure of trees will carry over to quasi-trees. On the other hand, we cannot make all of these inferences and maintain any distinction between quasi-trees and trees. Further, for some inferences we will have the choice of making the inference or not. The choices we make in defining the structure of the quasi-trees as a set of relationships will determine the structure of the sets of trees we can characterize with a single quasi-tree. Thus these choices will be driven by how much expressive power the application needs in describing these sets. Our guiding principle is to make the quasi-trees as tree-like as possible consistent with the needs of our applications. We discuss these considerations more fully in (Rogers & Vijay-Shanker, 1992).

One inference we will not make is as follows: from “a dominates b” infer either “a equals b” or, for

some  $a'$  and  $b'$ , “a dominates  $a'$ ,  $a'$  is the parent of  $b'$ , and  $b'$  dominates  $b$ ”. In structures that enforce this condition path lengths cannot be left partially specified. As a result, the set of quasi-trees required to characterize  $\alpha'$  viewed as a set of trees, for instance, would be infinite.

Similarly, we will not make the inference: for all  $a, b$ , either “a is left-of b”, “b is left-of a”, “a dominates b”, or “b dominates a”. In these structures the left-of relation is no longer partial, i.e. for all pairs  $a, b$  either every tree described by the quasi-tree satisfies “a is left-of b” or none of them do. This is not acceptable for D-theory, where both the analyses of “pseudo-passives” and coordinate structures require single structures describing sets including both trees in which some  $a$  is left-of  $b$  and others in which the same  $a$  is either equal to or properly dominates that same  $b$  (Marcus, Hindle & Fleck, 1983).

Finally, we consider the issue of negation. If a tree does not satisfy some relationship then it satisfies the negation of that relationship, and vice versa. For quasi-trees the situation is more subtle. Viewing the quasi-tree as a set of trees, if every tree in that set fails to satisfy some relationship, then they all satisfy the negation of that relationship. Hence the quasi-tree must satisfy the negated relationship as well. On the other hand, viewing the quasi-tree as a set of relationships, if a particular relationship is not included in the quasi-tree it does not imply that none of the trees it describes satisfies that relationship, only that some of those trees do not. Thus it may be the case that a quasi-tree neither satisfies a relationship nor satisfies its negation.

Since trees are completed objects, when a tree satisfies the negation of a relationship it will always be the case that the tree satisfies some (positive) relationship that is incompatible with the first. For example, in a tree “a does not dominate b” iff “a is left-of b”, “b is left-of a”, or “b properly dominates a”. Thus there are inferences that can be drawn from negated relationships in trees that may be incorporated into the structure of quasi-trees. In making these inferences, we dispense with the need to include negative relationships explicitly in the quasi-trees. They can be defined in terms of the

positive relationships. The price we pay is that to characterize the set of all trees in which “ $a$  does not dominate  $b$ ”, for instance, we will need three quasi-trees, one characterizing each of the sets in which “ $a$  is left-of  $b$ ”, “ $b$  is left-of  $a$ ”, and “ $b$  properly dominates  $a$ ”.

### 3 Language

Our language is built up from the symbols:

- $K$  — non-empty countable set of names,<sup>1</sup>  
 $r$  — a distinguished element of  $K$ , the root  
 $\triangleleft, \triangleleft^+, \triangleleft^*, \prec$   
— two place predicates, *parent*,  
*proper domination*, *domination*,  
and *left-of* respectively,  
 $\approx$  — equality predicate,  
 $\wedge, \vee, \neg$  — usual logical connectives  
 $(, ), [ , ]$  — usual grouping symbols

Our *atomic formulae* are  $t \triangleleft u, t \triangleleft^+ u, t \triangleleft^* u, t \prec u$ , and  $t \approx u$ , where  $t, u \in K$  are terms. *Literals* are atomic formulae or their negations. *Well-formed formulae* are generated from atoms and the logical connectives in the usual fashion.

We use  $t, u, v$  to denote terms and  $\phi, \psi$  to denote wffs.  $R$  denotes any of the five predicates.

#### 3.1 Models

Quasi-trees as formal structures are in a sense a reduced form of the quasi-trees viewed as sets of relationships. They incorporate a canonical subset of those relationships from which the remaining relationships can be deduced.

**Definition 1** A model is a tuple  $\langle \mathcal{U}, \mathcal{I}, \mathcal{P}, \mathcal{D}, \mathcal{A}, \mathcal{L} \rangle$ , where:

- $\mathcal{U}$  is a non-empty universe,  
 $\mathcal{I}$  is a partial function from  $K$  to  $\mathcal{U}$   
(specifying the node referred to by each name),  
 $\mathcal{P}, \mathcal{A}, \mathcal{D}$ , and  $\mathcal{L}$  are binary relations over  $\mathcal{U}$   
(assigned to  $\triangleleft, \triangleleft^+, \triangleleft^*$ , and  $\prec$  respectively).

Let  $\mathcal{R}$  denote  $\mathcal{I}(r)$ .

**Definition 2** A quasi-tree is a model satisfying the conditions  $C_1$ :

- For all  $w, x, y, z \in \mathcal{U}$ ,
- $C_1$   $\langle \mathcal{R}, x \rangle \in \mathcal{D}$ ,  
 $C_2$   $\langle x, x \rangle \in \mathcal{D}$ ,  
 $C_3$   $\langle x, y \rangle, \langle y, z \rangle \in \mathcal{D} \Rightarrow \langle x, z \rangle \in \mathcal{D}$ ,  
 $C_4$   $\langle x, z \rangle, \langle y, z \rangle \in \mathcal{D} \Rightarrow$   
 $\langle x, y \rangle \in \mathcal{D}$  or  $\langle y, x \rangle \in \mathcal{D}$ ,  
 $C_5$   $\langle x, y \rangle \in \mathcal{A} \Rightarrow \langle x, y \rangle \in \mathcal{D}$ ,  
 $C_6$   $\langle x, y \rangle \in \mathcal{A}$  and  $\langle w, x \rangle, \langle y, z \rangle \in \mathcal{D} \Rightarrow$   
 $\langle w, z \rangle \in \mathcal{A}$ ,  
 $C_7$   $\langle x, y \rangle \in \mathcal{P} \Rightarrow \langle x, y \rangle \in \mathcal{A}$   
 $C_8$   $\langle x, z \rangle \in \mathcal{P} \Rightarrow$

<sup>1</sup>We use *names* rather than *constants* to clarify the link to description theory.

- $\langle z, y \rangle \in \mathcal{L}$  or  $\langle y, z \rangle \in \mathcal{L}$   
or  $\langle y, x \rangle \in \mathcal{D}$  or  $\langle z, y \rangle \in \mathcal{D}$ ,  
 $C_9$   $\langle x, y \rangle \in \mathcal{L}$  and  $\langle x, w \rangle, \langle y, z \rangle \in \mathcal{D} \Rightarrow$   
 $\langle w, z \rangle \in \mathcal{L}$ ,  
 $C_{10}$   $\langle x, y \rangle \in \mathcal{L}$  and  $\langle w, x \rangle \in \mathcal{D} \Rightarrow$   
 $\langle w, y \rangle \in \mathcal{L}$  or  $\langle w, x \rangle, \langle w, y \rangle \in \mathcal{A}$ ,  
 $C_{11}$   $\langle x, y \rangle \in \mathcal{L}$  and  $\langle w, y \rangle \in \mathcal{D} \Rightarrow$   
 $\langle x, w \rangle \in \mathcal{L}$  or  $\langle w, x \rangle, \langle w, y \rangle \in \mathcal{A}$ ,  
 $C_{12}$   $\langle x, y \rangle \in \mathcal{L}$  and  $\langle y, z \rangle \in \mathcal{L} \Rightarrow \langle x, z \rangle \in \mathcal{L}$ ,

And meeting the additional condition: for every  $x, z \in \mathcal{U}$  the set  $B_{x,z} = \{y \mid \langle x, y \rangle, \langle y, z \rangle \in \mathcal{D}\}$  is finite, ie: the length of path from any node to any other is finite.<sup>2</sup>

A quasi-tree is consistent iff

- $CC_1$   $\langle x, y \rangle \in \mathcal{A} \Rightarrow \langle y, x \rangle \notin \mathcal{D}$ ,  
 $CC_2$   $\langle x, y \rangle \in \mathcal{L} \Rightarrow$   
 $\langle x, y \rangle \notin \mathcal{D}$ ,  $\langle y, x \rangle \notin \mathcal{D}$ , and  $\langle y, x \rangle \notin \mathcal{L}$ .

It is normal iff

- $RC_1$  for all  $x \neq y \in \mathcal{U}$ , either  
 $\langle x, y \rangle \notin \mathcal{D}$  or  $\langle y, x \rangle \notin \mathcal{D}$ .

At least one normal, consistent quasi-tree (that consisting of only a root node) satisfies all of these conditions simultaneously. Thus they are consistent. It is not hard to exhibit a model for each condition in which that condition fails while all of the others hold. Thus the conditions are *independent* of each other.

Trees are distinguished from (ordinary) quasi-trees by the fact that  $\mathcal{D}$  is the reflexive, transitive closure of  $\mathcal{P}$ , and the fact that the relations  $\mathcal{P}, \mathcal{D}, \mathcal{A}, \mathcal{L}$  are maximal in the sense that they cannot be consistently extended.

**Definition 3** A consistent, normal quasi-tree  $M$  is a tree iff

- $TC_1$   $\mathcal{D}^M = (\mathcal{P}^M)^*$ ,  
 $TC_2$  for all pairs  $\langle x, y \rangle \in \mathcal{U}^M \times \mathcal{U}^M$ ,  
exactly one of the following is true:  
 $\langle x, y \rangle, \langle y, x \rangle \in \mathcal{D}^M; \langle x, y \rangle \in \mathcal{A}^M;$   
 $\langle y, x \rangle \in \mathcal{A}^M; \langle x, y \rangle \in \mathcal{L}^M; \text{ or } \langle y, x \rangle \in \mathcal{L}^M.$

Note that  $TC_1$  implies that  $\mathcal{A}^M = (\mathcal{P}^M)^+$  as well. It is easy to verify that a quasi-tree meets these conditions iff  $\langle \mathcal{U}^M, \mathcal{P}^M \rangle$  is the graph of a tree as commonly defined (Aho, Hopcroft & Ullman, 1974).

#### 3.2 Satisfaction

The semantics of the language in terms of the models is defined by the satisfaction relation between models and formulae.

**Definition 4** A model  $M$  satisfies a formula  $\phi$  ( $M \models \phi$ ) as follows:

<sup>2</sup>The additional condition excludes “non-standard” models which include components not connected to the root by a finite sequence of immediate domination links.

$$\begin{aligned}
M \models t \triangleleft^* u & \text{ iff } \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{D}^M; \\
M \models \neg t \triangleleft^* u & \text{ iff } \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{L}^M, \\
& \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{L}^M, \\
& \text{ or } \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{A}^M; \\
M \models t \approx u & \text{ iff } \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{D}^M \\
& \text{ and } \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{D}^M; \\
M \models t \not\approx u & \text{ iff } \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{A}^M, \\
& \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{A}^M, \\
& \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{L}^M, \\
& \text{ or } \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{L}^M; \\
M \models t \triangleleft^+ u & \text{ iff } \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{A}^M; \\
M \models \neg t \triangleleft^+ u & \text{ iff } \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{D}^M, \\
& \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{L}^M, \\
& \text{ or } \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{L}^M; \\
M \models t \triangleleft u & \text{ iff } \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{P}^M; \\
M \models \neg t \triangleleft u & \text{ iff } \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{D}^M, \\
& \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{L}^M, \\
& \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{L}^M, \text{ or} \\
& \langle \mathcal{I}^M(t), x \rangle, \langle x, \mathcal{I}^M(u) \rangle \in \mathcal{A}^M, \\
& \text{ for some } x \in \mathcal{U}^M; \\
M \models t < u & \text{ iff } \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{L}; \\
M \models \neg t < u & \text{ iff } \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{L}, \\
& \langle \mathcal{I}^M(t), \mathcal{I}^M(u) \rangle \in \mathcal{D}, \\
& \text{ or } \langle \mathcal{I}^M(u), \mathcal{I}^M(t) \rangle \in \mathcal{D}; \\
M \models \neg\neg\phi & \text{ iff } M \models \phi; \\
M \models \phi \wedge \psi & \text{ iff } M \models \phi \text{ and } M \models \psi; \\
M \models \neg(\phi \wedge \psi) & \text{ iff } M \models \neg\phi \text{ or } M \models \neg\psi; \\
M \models \phi \vee \psi & \text{ iff } M \models \phi \text{ or } M \models \psi; \\
M \models \neg(\phi \vee \psi) & \text{ iff } M \models \neg\phi \text{ and } M \models \neg\psi.
\end{aligned}$$

In addition we require that  $\mathcal{I}^M(k)$  be defined for all  $k$  occurring in the formula.

It is easy to verify that for all quasi-trees  $M$

$$(\exists t, u, R)[M \models t R u, \neg t R u] \Rightarrow M \text{ inconsistent.}$$

If  $\mathcal{I}^M$  is surjective then the converse holds as well.

It is also not hard to see that if  $T$  is a tree

$$(\forall t, u, R)[T \models \neg t R u \Leftrightarrow T \not\models t R u].$$

## 4 Characterization

We now show that this formalization is complete in the sense that a consistent quasi-tree as defined characterizes the set of trees it describes. Recall that the quasi-tree describes the set of all trees which satisfy every literal formula which is satisfied by the quasi-tree. It characterizes that set if every literal formula which is satisfied by every tree in the set is also satisfied by the quasi-tree. The property of satisfying every formula which is satisfied by the quasi-tree is captured formally by the notion of subsumption, which we define initially as a relationship between quasi-trees.

**Definition 5** Subsumption.

Suppose  $M = \langle \mathcal{U}^M, \mathcal{I}^M, \mathcal{P}^M, \mathcal{D}^M, \mathcal{A}^M, \mathcal{L}^M \rangle$  and  $M' = \langle \mathcal{U}^{M'}, \mathcal{I}^{M'}, \mathcal{P}^{M'}, \mathcal{D}^{M'}, \mathcal{A}^{M'}, \mathcal{L}^{M'} \rangle$  are consistent quasi-trees, then  $M$  subsumes  $M'$  ( $M \sqsubseteq M'$ ) iff there is a function  $h : \mathcal{U}^M \rightarrow \mathcal{U}^{M'}$  such that:

$$\begin{aligned}
\mathcal{I}^{M'}(t) &= h(\mathcal{I}^M(t)), \\
\langle x, y \rangle \in \mathcal{P}^M &\Rightarrow \langle h(x), h(y) \rangle \in \mathcal{P}^{M'}, \\
\langle x, y \rangle \in \mathcal{D}^M &\Rightarrow \langle h(x), h(y) \rangle \in \mathcal{D}^{M'}, \\
\langle x, y \rangle \in \mathcal{A}^M &\Rightarrow \langle h(x), h(y) \rangle \in \mathcal{A}^{M'}, \\
\langle x, y \rangle \in \mathcal{L}^M &\Rightarrow \langle h(x), h(y) \rangle \in \mathcal{L}^{M'}.
\end{aligned}$$

We now claim that any quasi-tree  $Q$  is subsumed by a quasi-tree  $M$  iff it is described by  $M$ .

**Lemma 1** If  $M$  and  $Q$  are normal, consistent quasi-trees and  $\mathcal{I}^M$  is surjective, then  $M \sqsubseteq Q$  iff for all formulae  $\phi$ ,  $M \models \phi \Rightarrow Q \models \phi$ .

The proof in the forward direction is an easy induction on the structure of  $\phi$  and does not depend either on normality or surjectiveness of  $\mathcal{I}^M$ . The opposite direction follows from the fact that, since  $\mathcal{I}^M$  is surjective, there is a model  $M'$  in which  $\mathcal{U}^{M'}$  is the set of equivalence classes wrt  $\approx$  in the domain of  $\mathcal{I}^M$ , such that  $M \sqsubseteq M' \sqsubseteq Q$ .

The next lemma allows us, in many cases, to assume that a given quasi-tree is normal.

**Lemma 2** For every consistent quasi-tree  $M$ , there is a normal, consistent quasi-tree  $M'$  such that  $M \sqsubseteq M'$ , and for all normal, consistent quasi-tree  $M''$ ,  $M \sqsubseteq M'' \Rightarrow M' \sqsubseteq M''$ .

The lemma is witnessed by the quotient of  $M$  with respect to  $\mathcal{E}^M$ , where  $\mathcal{E}^M = \{ \langle x, y \rangle \mid \langle x, y \rangle, \langle y, x \rangle \in \mathcal{D}^M \}$ .

We can now state the central claim of this section, that every consistent quasi-tree characterizes the set of trees which it subsumes.

**Proposition 1** Suppose  $M$  is a consistent quasi-tree. For all literals  $\phi$

$$M \models \phi \Leftrightarrow (\forall T, \text{ tree})[M \sqsubseteq T \Rightarrow T \models \phi]$$

The proof follows from two lemmas. The first establishes that the set of quasi-trees subsumed by one quasi-tree  $M$  is in fact characterized by it. The second extends the result to trees. Their proofs are in (Rogers & Vijay-Shanker, 1992).

**Lemma 3** If  $M$  is a consistent quasi-tree and  $\phi$  a literal then

$$M \not\models \phi \Leftrightarrow (\exists Q, \text{ consistent quasi-tree})[M \sqsubseteq Q \text{ and } Q \models \neg\phi]$$

**Lemma 4** If  $M$  is a consistent quasi-tree, then there exists a tree  $T$  such that  $M \sqsubseteq T$ .

**Proof**(of proposition 1)

$$\begin{aligned}
(\forall T)[M \sqsubseteq T \Rightarrow T \models \phi] \\
&\Leftrightarrow \neg(\exists T)[M \sqsubseteq T \text{ and } T \models \neg\phi] \\
&\Rightarrow \text{by consistency, } \Leftarrow \text{by completeness of trees} \\
&\Leftrightarrow \neg(\exists Q, \text{ consistent q-t})[M \sqsubseteq Q \text{ and } Q \models \neg\phi] \\
&\Rightarrow \text{by lemma 4, } \Leftarrow \text{since } T \text{ is a quasi-tree} \\
&\Leftrightarrow M \models \phi \\
&\Rightarrow \text{by lemma 3, } \Leftarrow \text{by lemma 1} \quad \square
\end{aligned}$$

## 5 Semantic Tableau

Semantic tableau as introduced by Beth (Beth, 1959; Fitting, 1990) are used to prove validity by means of refutation. We are interested in satisfiability rather than validity. Given  $\Sigma$  we wish to build a model of  $\Sigma$  if one exists. Thus we are interested in the cases where the tableau succeeds in constructing a model.

The distinction between these uses of semantic tableau is important, since our mechanism is not suitable for refutational proofs. In particular, it cannot express “some model fails to satisfy  $\phi$ ” except as “some model satisfies  $\neg\phi$ ”. Since our logic is non-classical the first is a strictly weaker condition than the second.

**Definition 6** Semantic Tableau:

A branch is a set,  $S$ , of formulae.

A configuration is a collection,  $\{S_1, \dots, S_n\}$ , of branches.

A tableau is a sequence,  $\langle C_1, \dots, C_n \rangle$ , of configurations where each  $C_{i+1}$  is a result of the application of an inference rule to  $C_i$ .

If  $\frac{S}{S_1 \dots S_n}$  is an inference rule,  $(C_i \setminus \{S\}) \cup \{S_1, \dots, S_n\}$  is the result of applying the rule to  $C_i$  iff  $S \in C_i$ .

A tableau for  $\Sigma$ , where  $\Sigma$  is a set of formulae, is a tableau in which  $C_1 = \{\Sigma\}$ .

A branch is closed iff  $(\exists\phi)[\{\phi, \neg\phi\} \subseteq S]$ . A configuration is closed iff each of its branches is closed, and a tableau is closed iff it contains some closed configuration. A branch, configuration, or tableau that is not closed is open.

### 5.1 Inference Rules

Our inference rules fall into three groups. The first two, figures 3 and 4, are standard rules for propositional semantic tableau extended with equality (Fitting, 1990). The third group, figure 5, embody the properties of quasi-trees.

The  $\neg\triangleleft$  rule requires the introduction of a new name into the tableau. To simplify this, tableau are carried out in a language augmented with a countably infinite set of new names from which these are drawn in a systematic way.

The following two lemmas establish the correctness of the inference rules in the sense that no rule increases the set of models of any branch nor eliminates all of the models of a satisfiable branch.

**Lemma 5** Suppose  $S'$  is derived from  $S$  in some tableau by some sequence of rule applications. Suppose  $M$  is a model, then:

$$M \models S' \Rightarrow M \models S.$$

This follows nearly directly from the fact that all of our rules are non-strict, ie: the branch to which an inference rule is applied is a subset of every branch introduced by its application.

**Lemma 6** If  $S$  is a branch of some configuration of a tableau and  $S'$  is the set of branches resulting from applying some rule to  $S$ , then if there is a

consistent quasi-tree  $M$  such that  $M \models S$ , then for some  $S'_i \in S'$  there is a consistent quasi-tree  $M'$  such that  $M' \models S'_i$ .

We sketch the proof. Suppose  $M \models S$ . For all but  $\neg\triangleleft$  it is straightforward to verify  $M$  also satisfies at least one of the  $S'_i$ . For  $\neg\triangleleft$ , suppose  $M$  fails to satisfy either  $u \triangleleft^* t$  or  $\neg t \triangleleft^* u$ . Then we claim some quasi-tree satisfies the third branch of the conclusion. This must map the new constant  $k$  to the witness for the rule.  $M$  has no such requirement, but since  $k$  does not occur in  $S$ , the value of  $\mathcal{I}^M(k)$  does not affect satisfaction of  $S$ . Thus we get an appropriate  $M'$  by modifying  $\mathcal{I}^M$  to map  $k$  correctly.

**Corollary 1** If there is a closed tableau for  $\phi$  then no consistent quasi-tree satisfies  $\phi$ .

No consistent quasi-tree satisfies a closed set of formulae. The result then follows by induction on the length of the tableau.

## 6 Constructing Models

We now turn to the conditions for a branch to be sufficiently complete to fully specify a quasi-tree. In essence these just require that all formulae have been expanded to atoms, that all substitutions have been made and that the conditions in the definition of quasi-trees are met.

### 6.1 Saturated Branches

**Definition 7** A set of sentences  $S$  is downward saturated iff for all formulae  $\phi, \psi$ , and terms  $t, u, v$ :

- H1**  $\neg\neg\phi \in S \Rightarrow \phi \in S$
- H2**  $\phi \vee \psi \in S \Rightarrow \phi \in S$  or  $\psi \in S$
- H3**  $\neg(\phi \vee \psi) \in S \Rightarrow \neg\phi \in S$  and  $\neg\psi \in S$
- H4**  $\phi \wedge \psi \in S \Rightarrow \phi \in S$  and  $\psi \in S$
- H5**  $\neg(\phi \wedge \psi) \in S \Rightarrow \neg\phi \in S$  or  $\neg\psi \in S$
- H6**  $t \approx t \in S$  for all terms  $t$  occurring in  $S$
- H7**  $t_1 \approx u_1, t_2 \approx u_2 \in S \Rightarrow$   
 $t_1 \triangleleft^* t_2 \in S \Rightarrow u_1 \triangleleft^* u_2 \in S,$   
 $t_1 \triangleleft^+ t_2 \in S \Rightarrow u_1 \triangleleft^+ u_2 \in S,$   
 $t_1 \triangleleft t_2 \in S \Rightarrow u_1 \triangleleft u_2 \in S,$   
 $t_1 \prec t_2 \in S \Rightarrow u_1 \prec u_2 \in S,$   
 $t_1 \approx t_2 \in S \Rightarrow u_1 \approx u_2 \in S.$
- H8**  $r \triangleleft^* t \in S$  for all terms  $t$  occurring in  $S$
- H9**  $t \approx u \in S \Rightarrow t \triangleleft^* u \in S$
- H10**  $t \not\approx u \in S \Rightarrow \neg t \triangleleft^* u \in S$  or  $\neg u \triangleleft^* t \in S$
- H11**  $t \triangleleft^* u, u \triangleleft^* t \in S \Rightarrow t \approx u \in S$
- H12**  $t \triangleleft^* u, u \triangleleft^* v \in S \Rightarrow t \triangleleft^* v \in S$
- H13**  $t \triangleleft^* v, u \triangleleft^* v \in S \Rightarrow$   
 $t \triangleleft^* u \in S$  or  $u \triangleleft^* t \in S$
- H14**  $\neg t \triangleleft^* u \in S \Rightarrow$   
 $t \prec u \in S$  or  $u \prec t \in S$  or  $u \triangleleft^+ t \in S$
- H15**  $t \triangleleft^+ u \in S \Rightarrow t \triangleleft^* u, \neg u \triangleleft^* t \in S$
- H16**  $t \triangleleft^+ u, s \triangleleft^* t, u \triangleleft^* v \in S \Rightarrow s \triangleleft^+ v \in S$
- H17**  $\neg t \triangleleft^+ u \in S \Rightarrow \neg t \triangleleft^* u \in S$  or  $u \triangleleft^* t \in S$
- H18**  $t \triangleleft u \in S \Rightarrow t \triangleleft^+ u \in S$

$$\begin{array}{c}
\frac{S, \phi \vee \psi}{S, \phi \vee \psi, \phi \mid S, \phi \vee \psi, \psi} \vee \\
\frac{S, \phi \wedge \psi}{S, \phi \wedge \psi, \phi, \psi} \wedge \\
\frac{S, \neg\neg\phi}{S, \neg\neg\phi, \phi} \neg\neg \\
\frac{S, \neg(\phi \vee \psi)}{S, \neg(\phi \vee \psi), \neg\phi, \neg\psi} \neg\vee \\
\frac{S, \neg(\phi \wedge \psi)}{S, \neg(\phi \wedge \psi), \neg\phi \mid S, \neg(\phi \wedge \psi), \neg\psi} \neg\wedge
\end{array}$$

Figure 3. Elementary Rules

- H19**  $t \triangleleft v \in S \Rightarrow u \triangleleft v \in S$  or  $v \triangleleft u \in S$   
or  $u \triangleleft^* t \in S$  or  $v \triangleleft^* u \in S$
- H20**  $\neg t \triangleleft u \in S \Rightarrow u \triangleleft^* t \in S$  or  $\neg t \triangleleft^* u \in S$   
or  $t \triangleleft^* w, w \triangleleft^* u \in S$ , for some term  $w$
- H21**  $t \triangleleft u \in S \Rightarrow \neg t \triangleleft^* u, \neg u \triangleleft^* t, \neg u \triangleleft t \in S$
- H22**  $t \triangleleft u, t \triangleleft^* s, u \triangleleft^* v \in S \Rightarrow s \triangleleft v \in S$
- H23**  $t \triangleleft u, v \triangleleft^* t \in S \Rightarrow$   
 $v \triangleleft u \in S$  or  $v \triangleleft^* t, v \triangleleft^* u \in S$
- H24**  $t \triangleleft u, v \triangleleft^* u \in S \Rightarrow$   
 $t \triangleleft v \in S$  or  $v \triangleleft^* t, v \triangleleft^* u \in S$
- H25**  $t \triangleleft u, u \triangleleft v \in S \Rightarrow t \triangleleft v \in S$
- H26**  $\neg t \triangleleft u \in S \Rightarrow$   
 $u \triangleleft t \in S$  or  $t \triangleleft^* u \in S$  or  $u \triangleleft^* t \in S$ .

The next lemma (essentially Hintikka's lemma) establishes the correspondence between saturated branches and quasi-trees.

**Lemma 7** For every consistent downward saturated set of formulae  $S$  there is a consistent quasi-tree  $M$  such that  $M \models S$ . For every finite consistent downward saturated set of formulae, there is a such a quasi-tree which is finite.

Again, we sketch the proof. Consider the set  $\mathcal{T}(S)$  of terms occurring in a downward saturated set  $S$ . **H6** and **H7** assure that  $\approx$  is reflexive and substitutive. Since  $t \approx u, u \approx v \in S \Rightarrow t \approx v \in S$ , and  $u \approx u, u \approx v \in S \Rightarrow v \approx u \in S$  by substitution of  $v$  for (the first occurrence of)  $u$ , it is transitive and symmetric as well. Thus  $\approx$  partitions  $\mathcal{T}(S)$  into equivalence classes.

Define the model  $H$  as follows:

$$\begin{aligned}
\mathcal{U}^H &= \mathcal{T}(S)/\approx, \\
\mathcal{I}^H(k) &= [k]_{\approx}, \\
\mathcal{P}^H &= \{([t]_{\approx}, [u]_{\approx}) \mid t \triangleleft u \in S\}, \\
\mathcal{D}^H &= \{([t]_{\approx}, [u]_{\approx}) \mid t \triangleleft^* u \in S\}, \\
\mathcal{A}^H &= \{([t]_{\approx}, [u]_{\approx}) \mid t \triangleleft^+ u \in S\}, \\
\mathcal{L}^H &= \{([t]_{\approx}, [u]_{\approx}) \mid t \triangleleft u \in S\}.
\end{aligned}$$

Since each of the conditions **C1** through **C12** corresponds directly to one of the saturation conditions, it is easy to verify that  $H$  satisfies  $C_g$ . It is equally easy to confirm that  $H$  is both consistent and normal.

We claim that  $\phi \in S \Rightarrow H \models \phi$ . As is usual for versions of Hintikka's lemma, this is established by an induction on the structure of  $\phi$ . Space prevents us from giving the details here.

For the second part of the lemma, if the set of formulae is finite, then the set of terms (and hence the set of equivalence classes) is finite.

## 6.2 Saturated Tableau

Since all of our inference rules are non-strict, if a rule once applies to a branch it will always apply to a branch. Without some restriction on the application of rules, tableau for satisfiable sets of formulae will never terminate. What is required is a control strategy that guarantees that no rule applies to any tableau more than finitely often, but that will always find a rule to apply to any open branch that is not downward saturated.

**Definition 8** Let  $\text{EQ}_S$  be the reflexive, symmetric, transitive closure of  $\{(t, u) \mid t \approx u \in S\}$ .

An inference rule,  $\mathbf{I}$ , applies to some branch  $S$  of a configuration  $C$  iff

- $S$  is open
- $S \notin \{S_i \mid S_i \text{ results from application of } \mathbf{I} \text{ to } S\}$
- if  $\mathbf{I}$  introduces a new constant  $a$  occurring in formulae  $\phi_j(a) \in S_i$ , there is no term  $t$  and pairs  $\langle u_1, v_1 \rangle, \langle u_2, v_2 \rangle, \dots \in \text{EQ}_S$  such that for each of the  $\phi_j$ ,  $\phi\{t/a, u_1/v_1, u_2/v_2, \dots\} \in S$ . (Where  $\phi\{t/a, u_1/v_1, u_2/v_2, \dots\}$  denotes the result of uniformly substituting  $t$  for  $a$ ,  $u_1$  for  $v_1$ , etc., in  $\phi$ .)

The last condition in effect requires all equality rules to be applied before any new constant is introduced. It prevents the introduction of a formula involving a new constant if an equivalent formula already exists or if it is possible to derive one using only the equality rules.

We now argue that this definition of applies does not terminate any branch too soon.

**Lemma 8** If no inference rule applies to an open branch  $S$  of a configuration, then  $S$  is downward saturated.

This follows directly from the fact that for each of **H1** through **H26**, if the implication is false there is a corresponding inference rule which applies.

$$\frac{S}{S, t \approx t} \approx_r \text{ (reflexivity of } \approx) \quad \frac{S, t \approx u, \phi(t)}{S, t \approx u, \phi(t), \phi(\frac{u}{t})} \approx_s \text{ (substitution)}$$

any term  $t$   
occurring in  $S$

$\phi(\frac{u}{t})$  denotes the result of substituting  $u$  for any or all occurrences of  $t$  in  $\phi$ .

Figure 4. Equality Rules

$$\frac{S}{S, r \triangleleft^* t} \triangleleft_m^* \text{ (} r \text{ minimum wrt } \triangleleft^*)$$

$t$  any term occurring in  $S$   
or  $t = r$

$$\frac{S, t \approx u}{S, t \approx u, t \triangleleft^* u, u \triangleleft^* t} \triangleleft_r^* \text{ (reflexivity of } \triangleleft^*) \quad \frac{S, t \not\approx u}{S, t \not\approx u, \neg t \triangleleft^* u \mid S, t \not\approx u, \neg u \triangleleft^* t} \triangleleft_{r-}^*$$

$$\frac{S, t \triangleleft^* u, u \triangleleft^* t}{S, t \triangleleft^* u, u \triangleleft^* t, t \approx u} \triangleleft_a^* \text{ (anti-symmetry)} \quad \frac{S, t \triangleleft^* u, u \triangleleft^* v}{S, t \triangleleft^* u, u \triangleleft^* v, t \triangleleft^* v} \triangleleft_t^* \text{ (transitivity)}$$

$$\frac{S, t \triangleleft^* v, u \triangleleft^* v}{S, t \triangleleft^* v, u \triangleleft^* v, t \triangleleft^* u \mid S, t \triangleleft^* v, u \triangleleft^* v, u \triangleleft^* t} \triangleleft_l^* \text{ (branches linearly ordered)}$$

$$\frac{S, \neg t \triangleleft^* u}{S, \neg t \triangleleft^* u, t \prec u \mid S, \neg t \triangleleft^* u, u \prec t \mid S, \neg t \triangleleft^* u, u \triangleleft^+ t} \neg \triangleleft^*$$

$$\frac{S, t \triangleleft^+ u}{S, t \triangleleft^+ u, t \triangleleft^* u, \neg u \triangleleft^* t} \triangleleft^+{}_1 \quad \frac{S, t \triangleleft^+ u, s \triangleleft^* t, u \triangleleft^* v}{S, t \triangleleft^+ u, s \triangleleft^* t, u \triangleleft^* v, s \triangleleft^+ v} \triangleleft^+{}_2$$

$$\frac{S, \neg t \triangleleft^+ u}{S, \neg t \triangleleft^+ u, \neg t \triangleleft^* u \mid S, \neg t \triangleleft^+ u, u \triangleleft^* t} \neg \triangleleft^+ \quad \frac{S, t \triangleleft u}{S, t \triangleleft u, t \triangleleft^+ u} \triangleleft_1$$

$$\frac{S, t \triangleleft v}{S, t \triangleleft v, u \prec v \mid S, t \triangleleft v, v \prec u \mid S, t \triangleleft v, u \triangleleft^* t \mid S, t \triangleleft v, v \triangleleft^* u} \triangleleft_2$$

any term  $u$  occurring in  $S$ .

$$\frac{S, \neg t \triangleleft u}{S, \neg t \triangleleft u, u \triangleleft^* t \mid S, \neg t \triangleleft u, \neg t \triangleleft^* u \mid S, \neg t \triangleleft u, t \triangleleft^+ k, k \triangleleft^+ u} \neg \triangleleft$$

$k$  new name

$$\frac{S, t \prec u}{S, t \prec u, \neg t \triangleleft^* u, \neg u \triangleleft^* t, \neg u \prec t} \prec_1 \quad \frac{S, t \prec u, t \triangleleft^* s, u \triangleleft^* v}{S, t \prec u, t \triangleleft^* s, u \triangleleft^* v, s \prec v} \prec_2$$

$$\frac{S, t \prec u, v \triangleleft^* t}{S, t \prec u, v \triangleleft^* t, v \prec u \mid S, t \prec u, v \triangleleft^* t, v \triangleleft^+ t, v \triangleleft^+ u} \prec_3$$

$$\frac{S, t \prec u, v \triangleleft^* u}{S, t \prec u, v \triangleleft^* u, t \prec v \mid S, t \prec u, v \triangleleft^* u, v \triangleleft^+ t, v \triangleleft^+ u} \prec_4$$

$$\frac{S, t \prec u, u \prec v}{S, t \prec u, u \prec v, t \prec v} \prec_t \quad \frac{S, \neg t \prec u}{S, \neg t \prec u, u \prec t \mid S, \neg t \prec u, t \triangleleft^* u \mid S, \neg t \prec u, u \triangleleft^* t} \neg \prec$$

Figure 5. Tree Rules



**Proposition 2 (Termination)** *All tableau for finite sets of formulae can be extended to tableau in which no rule applies to the final configuration.*

This follows from the fact that the size of any tableau for finite sets of formulae has a finite upper bound. The proof is in (Rogers & Vijay-Shanker, 1992).

**Proposition 3 (Soundness and Completeness)**

*A saturated tableau for a finite set of formulae  $\Sigma$  exists iff there is a consistent quasi-tree which satisfies  $\Sigma$ .*

**Proof:** The forward implication (soundness) follows from lemma 7. Completeness follows from the fact that if  $\Sigma$  is satisfiable there is no closed tableau for  $\Sigma$  (corollary 1), and thus, by proposition 2 and lemma 8, there must be a saturated tableau for  $\Sigma$ .  $\square$

## 7 Extracting Trees from Quasi-trees

Having derived some quasi-tree satisfying a set of relationships, we would like to produce a “minimal” representative of the trees it characterizes. In section 3.1 we define the conditions under which a quasi-tree is a tree. Working from those conditions we can determine in which ways a quasi-tree  $M$  may fail to be a tree, namely:

- $(\mathcal{P}^M)^*$  is a proper subset of  $\mathcal{D}^M$ ,
- $\mathcal{L}^M$  and/or  $\mathcal{D}^M$  may be partial, ie: for some  $t, u$ ,  $M \not\models (t \prec u \vee \neg t \prec u)$  or  $M \not\models (t \triangleleft^* u \vee \neg t \triangleleft^* u)$ .

The case of partial  $\mathcal{L}^M$  is problematic in that, while it is possible to choose a unique representative, its choice must be arbitrary. For our applications this is not significant since currently in TAGs left-of is fully specified and in parsing it is always resolved by the input. Thus we make the assumption that in every quasi-tree  $M$  from which we need to extract a tree, left-of will be complete. That is, for all terms  $t, u$ :  $M \models t \prec u \vee \neg t \prec u$ . Thus  $M \not\models t \triangleleft^* u \vee \neg t \triangleleft^* u \Rightarrow M \models u \triangleleft^* t$ .

Suppose  $M \models u \triangleleft^* t$  and  $M \not\models (t \triangleleft^* u \vee \neg t \triangleleft^* u)$ , and that  $\mathcal{I}^M(u) = x$  and  $\mathcal{I}^M(t) = y$ . In D-theory, this case never arises, since proper domination, rather than domination, is primitive. It is clear that the TAG applications require that  $x$  and  $y$  be identified, ie:  $\langle y, x \rangle$  should be added to  $\mathcal{D}^M$ . Thus we choose to complete  $\mathcal{D}^M$  by extending it. Under the assumption that  $\mathcal{L}$  is complete this simply means: if  $M \not\models \neg t \triangleleft^* u$ ,  $\mathcal{D}^M$  should be extended such that  $M \models t \triangleleft^* u$ . That  $M$  can be extended in this way consistently follows from lemma 3. That the result of completing  $\mathcal{D}^M$  in this way is unique follows from the fact that, under these conditions, extending  $\mathcal{D}^M$  does not extend either  $\mathcal{A}^M$  or  $\mathcal{L}^M$ . The details can be found in (Rogers & Vijay-Shanker, 1992).

In the resulting quasi-tree domination has been resolved into equality or proper domination. To

arrive at a tree we need only to expand  $\mathcal{P}^M$  such that  $(\mathcal{P}^M)^* = \mathcal{D}^M$ . In the proof of lemma 4 we show that this will be the case in any quasi-tree  $T$  closed under:

$$\begin{aligned} \langle x, z \rangle \in \mathcal{A}^T \text{ and } (\forall y)[\langle x, y \rangle \notin \mathcal{A}^T \text{ or } \langle y, z \rangle \notin \mathcal{A}^T] \\ \Rightarrow \langle x, z \rangle \in \mathcal{P}^T \\ \langle x, y \rangle \notin \mathcal{L}^T \text{ and } \langle y, x \rangle \notin \mathcal{L}^T \cup \mathcal{A}^T \\ \Rightarrow \langle x, y \rangle \in \mathcal{D}^T. \end{aligned}$$

The second of these conditions is our mechanism for completing  $\mathcal{D}^M$ . The first amounts to taking immediate domination as the parent relation — precisely the mechanism for finding the standard referent. Thus the tree we extract is both the circumscriptive reading of (Vijay-Shanker, 1992) and the standard referent of (Marcus, Hindle & Fleck, 1983).

## References

- Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley.
- Beth, E. W. (1959). *The Foundations of Mathematics*. Amsterdam: North-Holland.
- Fitting, M. (1990). *First-order Logic and Automated Theorem Proving*. New York: Springer-Verlag.
- Marcus, M. P. (1980). *A Theory of Syntactic Recognition for Natural Language*. MIT Press.
- Marcus, M. P. (1987). Deterministic parsing and description theory. In P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, & P. Bennett (Eds.), *Linguistic Theory and Computer Applications*. Academic Press.
- Marcus, M. P., Hindle, D., & Fleck, M. M. (1983). D-theory: Talking about talking about trees. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA.
- Rogers, J. & Vijay-Shanker, K. (1992). A formalization of partial descriptions of trees. Technical Report TR92-23, Dept. of Comp. and Info. Sci., University of Delaware, Newark, DE.
- Vijay-Shanker, K. (1992). Using descriptions of trees in a tree-adjoining grammar. *Computational Linguistics*. To appear.
- Vijay-Shanker, K. & Schabes, Y. (1992). Structure sharing in lexicalized tree-adjoining grammars. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'92)*, Nantes.