6

Properties of Binary Transitive Closure Logics over Trees

STEPHAN KEPSER

Abstract

Binary transitive closure logic (FO^* for short) is the extension of first-order predicate logic by a transitive closure operator of binary relations. Deterministic binary transitive closure logic (FO^{D*}) is the restriction of FO^* to deterministic transitive closures. It is known that these logics are more powerful than FO on arbitrary structures and on finite ordered trees. It is also known that they are at most as powerful as monadic second-order logic (MSO) on arbitrary structures and on finite trees. We will study the expressive power of FO^* and FO^{D*} on trees to show that several MSO properties can be expressed in FO^{D*} (and hence FO^*).

The following results will be shown.

- A linear order can be defined on the nodes of a tree.
- The class EVEN of trees with an even number of nodes can be defined.
- On arbitrary structures with a tree signature, the classes of trees and finite trees can be defined.
- There is a tree language definable in FO^{D*} that cannot be recognised by any tree walking automaton.
- FO^* is strictly more powerful than tree walking automata.

These results imply that FO^{D*} and FO^* are neither compact nor do they have the Löwenheim-Skolem-Upward property.

6.1 Introduction

The question about the best suited logic for describing tree properties or defining tree languages is an important one for model theoretic syntax

FG-2006.
Organizing Committee:, Paola Monachesi, Gerald Penn, Giorgio Satta, Shuly Wintner.
Copyright © 2007, CSLI Publications.

as well as for querying treebanks. Model theoretic syntax is a research program in mathematical linguistics concerned with studying the descriptive complexity of grammar formalisms for natural languages by defining their derivation trees in suitable logical formalisms. Since the very influential book by Rogers (1998) it is monadic second-order logic (MSO) or even more powerful logics that are used to describe linguistic structures.

With the advent of XML and query languages for XML documents, in particular XPath, the interest in logics for querying treebanks rose dramatically. There is now a large interest in this topic in computer science. Independent of that, but temporarily parallel, large syntactically annotated treebanks became available in linguistics. They provide nowadays a rich and important source for the study of language. But in order to access this source, suitable query languages for treebanks are required.

One of the simplest properties that are known to be inexpressible in first-order predicate logic (FO henceforth) is the transitive closure of a binary relation. It is therefore a natural move to extend FO by a binary transitive closure operator. And this move has been done before in the definition of query languages for relational databases, in particular for the SQL3 standard. But it seems that the expressive power of FO plus binary transitive closures (FO^* for short) to define tree properties is not much studied yet. This is somewhat surprising, because there is reason to believe that FO^* is more user friendly than MSO. Most users of query languages, in particular linguists, understand the concept of a transitive closure very well and know how to use it. It is a lot more difficult to use set variables to describe tree properties. An example for this claim is the fact MSO is capable of defining binary transitive closures, as shown by Moschovakis (1974). A formula expressing the transitive closure in MSO is given at the end of the next section. It is questionable that ordinary users (without profound knowledge of MSO) would be able to find this formula.

There exists a more restricted version of transitive closure, namely deterministic transitive closure (FO^{D*}). The deterministic transitive closure of a binary relation is the transitive closure of the functional or deterministic part of the relation. We propose to seriously consider FO^{D*} as a language for defining tree properties. We do so by showing that several important MSO definable properties can be defined in FO^{D*}. One such example is the ability to define a linear order on the nodes of a tree. The order resembles depth-first left-to-right traversal of a tree. A linear order is a powerful concept that can be used defining additional properties. For example, it is used to count the number of nodes in a tree modulo a given natural number. An instance is the definition of the class EVEN of all trees with an even number of nodes in FO^{D*}.

Arguably an important reason for Rogers' choice of MSO is its ability to axiomatise trees. I.e., there exists a set of axioms such that an arbitrary structure (of a suitable signature) is a tree – finite or infinite – iff it is a model of the axioms. It is known that this characterisation of trees cannot be done using FO. But the full expressive power of MSO may not really be needed for the axiomatisation, because we show that arbitrary trees and finite trees can be axiomatised in FO^{D*}. This capability of axiomatising finite and infinite trees implies that FO^{D*} (and hence also FO^*) is neither compact nor does it possess the Löwenheim-Skolem-Upward property.

There exists a tree automaton concept that defines serial instead of parallel processing of nodes in a tree, namely tree walking automata (TWA). As the name implies, a tree is processed by walking up and down in it and inspecting nodes serially. One may therefore believe that these automata could be the automaton-theoretic correspondant of FO^*. But we show here that FO^* is more powerful. Every tree language that is recognised by a TWA can be defined in FO^*. The relationship towards FO^{D*} is less clear. There are FO^{D*}-definable tree languages that cannot be recognised by any TWA.

6.2 Preliminaries

Let M be a set. We write $\wp(M)$ for the power set of M. Let $R \subseteq M \times M$ be a binary relation over M. The *transitive closure* $TC(R)$ of R is the smallest set containing R and for all $x, y, z \in M$ such that $(x, y) \in TC(R)$ and $(y, z) \in TC(R)$ we have $(x, z) \in TC(R)$. I.e.,

$$TC(R) := \bigcap \{W \mid \; R \subseteq W \subseteq M \times M, \forall x, y, z \in M : \\ (x, y), (y, z) \in W \implies (x, z) \in W\}.$$

Deterministic transitive closure is the transitive closure of a deterministic, i.e., functional relation. For an arbitrary binary relation R we define its *deterministic reduct* by

$$R_D := \{(x, y) \in R \mid \forall z : (x, z) \in R \implies y = z\}.$$

Now

$$DTC(R) := TC(R_D).$$

We consider labelled ordered unranked trees. A tree is ordered if the set of child nodes of every node is linearly ordered. A tree is unranked if there is no relationship between the label of a node and the number of its children. For brevity we just write *tree* for *labelled ordered unranked* *tree*. In Sections 6.3 and 6.5 we only consider finite trees, in Section 6.4 we also consider infinite trees.

Definition 1 A *tree domain* is a non-empty subset $T \subseteq \mathbb{N}^*$ such that for all $u, v \in \mathbb{N}^* : uv \in T \implies u \in T$ (closure under prefixes) and for all $u \in \mathbb{N}^*$ and $i \in \mathbb{N} : ui \in T \implies uj \in T$ for all $j < i$ (closure under left sisters).

Let $\mathcal{L}$ be a set of labels. A *tree* is a pair (T, Lab) where T is a tree domain and $Lab : T \to \mathcal{L}$ is a node labelling function.

A tree is *finite* iff its tree domain is finite.

We remark that a tree domain is at most countable, since it is a subset of a countable union of countable sets.

The languages to talk about trees will be extensions of first-order logic. Their syntaxes is as follows. Let $X = \{x, y, z, w, u, x_1, x_2, x_3, \dots\}$ be a denumerable infinite set of variables. The atomic formulae are $L(x)$ for each label $L \in \mathcal{L}$, $x \to y$, $x \downarrow y$, and $x = y$. Complex formulae are constructed from simpler ones by means of the boolean connectives, existential and universal quantification, and transitive closure. I.e., if ϕ and ψ are formulae, then $\neg\phi, \phi \wedge \psi, \phi \vee \psi, \exists x{:}\phi, \forall x{:}\phi$, and $[\mathrm{TC}_{x_1,x_2}\ \phi](x, y)$, resp., $[\mathrm{DTC}_{x_1,x_2}\ \phi](x, y)$, are formulae.

The semantics of the first-order part of the language is standard. Let (T, Lab) be a tree. A variable assignment $a : X \to T$ assigns variables to nodes in the tree. The root node has the empty address ϵ. Now $[\![L(x)]\!]^a = \mathrm{T}$ iff $Lab(a(x)) = L$. $[\![x \downarrow y]\!]^a = T$ iff $a(y) = a(x)i$ for some $i \in \mathbb{N}$, i.e., $\downarrow$ is the parent relation. $[\![x \to y]\!]^a = \mathrm{T}$ iff there is a $u \in T$ and $i \in \mathbb{N}$ such that $a(x) = ui$ and $a(y) = ui + 1$, i.e., $\to$ is the immediate sister relation.

Boolean connectives and quantification have their standard interpretation. Now, $[\![[\mathrm{TC}_{x_1,x_2}\ \phi](x, y)]\!]^a = \mathrm{T}$ iff

$$(a(x), a(y)) \in TC(\{(b, d) \mid [\![\phi]\!]^{ab/x_1 d/x_2} = \mathrm{T}\})$$

where $ab/x_1d/x_2$ is the variable assignment that is identical to a except that x_1 is assigned to b and x_2 to d. If ϕ is a formula with free variables x_1, x_2, it can be regarded as a binary relation $\phi(x_1, x_2)$. Then $[\mathrm{TC}_{x_1,x_2}\ \phi]$ is the transitive closure of this binary relation. This language is abbreviated FO^*.

And $[\![[\mathrm{DTC}_{x_1,x_2}\ \phi](x, y)]\!]^a = \mathrm{T}$ iff

$$(a(x), a(y)) \in DTC(\{(b, d) \mid [\![\phi]\!]^{ab/x_1 d/x_2} = \mathrm{T}\}).$$

This language is abbreviated $\mathrm{FO}^{\mathrm{D}*}$. It is simple to see that everything expressible in $\mathrm{FO}^{\mathrm{D}*}$ can also be expressed in FO^*, because

$$[\mathrm{DTC}_{x_1,x_2}\ \phi(x_1,x_2)](x,y) \quad \leftrightarrow \\ [\mathrm{TC}_{x_1,x_2}\ \phi(x_1,x_2) \wedge \forall z \phi(x_1,z) \rightarrow z = x_2](x,y).$$

It is an open question whether there are tree languages definable in FO^* that cannot be defined in $\mathrm{FO}^{\mathrm{D}*}$.

FO^* is amongst the smallest extension of first-order logic. It is known that the transitive closure of a binary relation is *not* first-order definable (Fagin, 1975). But when talking about trees, people frequently want to talk about paths in a tree. And a path is the transitive closure of certain base steps. $\mathrm{FO}^{\mathrm{D}*}$ and FO^* have at most the expressive power of monadic second-order logic (MSO). It is an old result, which goes back at least to Moschovakis (1974, p. 20), that the transitive closure of every MSO-definable binary relation is also MSO-definable. The following formula is due to Courcelle (1990). Let R be an MSO-definable binary relation. Then

$$\forall X \ \ (\forall z, w(z \in X \wedge R(z,w) \implies w \in X) \wedge \forall z(R(x,z) \implies z \in X)) \\ \implies y \in X$$

is a formula with free variables x and y that defines the transitive closure of R. It follows that every tree language definable in FO^* can be defined in MSO. Whether the two logics are equivalent, seems an open question. For $\mathrm{FO}^{\mathrm{D}*}$, the question is settled. Recently, Bojanczyk et al. (2006) have shown that the expressive power of MSO for defining tree languages properly extends the expressive power of $\mathrm{FO}^{\mathrm{D}*}$.

6.3 Definability of Order

One of the abstract insights from descriptive complexity theory is that order is a very important property of structures. The relationship between certain logics and classical complexity classes is frequently restricted to *ordered* structures, i.e., structures where the carrier is linearly ordered. The reason for this restriction is to be found in the fact that computation is an ordered process. Definability and non-definability results for certain logics over ordered structures frequently do not extend to unordered structures. It is therefore an important property of a logic, if the logic itself is capable of expressing order without recourse to an extended signature. The probably best known logic with this property is Σ^1_1, the extension of first-order logic by arbitrary relation variables that are existentially quantified. It is obviously possible to define order in Σ^1_1, because we can say there is a binary relation that has all the properties of a linear order. These properties are known to be first-order properties. It is hence the ability to say “there is a binary relation” that is the key.

There is no way that $\mathrm{FO}^{\mathrm{D}*}$ or FO^* could define order on arbitrary finite structures. But if we only consider ordered trees as models, $\mathrm{FO}^{\mathrm{D}*}$ can define order. Indeed it is possible to give a definition of the depth-first left-to-right order of nodes in a tree (and some variants).

Proposition 1 *There is an explicit definition of a linear order of the nodes in a tree in* $\mathrm{FO}^{\mathrm{D}*}$.

Proof. Define the proper dominance relation of trees $Dom(x, y)$ as $[\mathrm{DTC}_{y,x}\ x \downarrow y](y, x)$. The idea of how to define dominance deterministically by walking upwards from the descendants to the ancestors goes back to Etessami and Immerman (1995). Similarly but simpler, define the sister relation $Sis(x, y)$ as $[\mathrm{DTC}_{x,y}\ x \rightarrow y](x, y)$. Now define $x < y$ as

$$\begin{aligned} Dom(x,y) \vee (\exists w, v :\ & Sis(w,v) \wedge \\ & (w = x \vee Dom(w,x)) \wedge (v = y \vee Dom(v,y))). \end{aligned}$$

The first disjunct expresses the “depth-first” part of the order. The more complicated second disjunct formalises the “left-to-right” part. It expresses that there is a common ancestor of nodes x and y and node x is to be found on a left branch while y is to be found on a right branch. Care is taken that mutual domination is excluded. Hence the two disjuncts are mutually exclusive. Since the dominance and the sisterhood steps are both irreflexive, the whole relation $<$ is irreflexive. Furthermore for each pair of distinct nodes in a tree, either one dominates the other, or there is a common ancestor such that one node is on a left branch while the other is on a right branch. Hence the relation is total. Transitivity can easily be checked by considering the four cases involved in expanding $x < y$ and $y < z$. □

The proposition basically states that ordered trees are ordered structures in any logic at least as powerful as $\mathrm{FO}^{\mathrm{D}*}$. Note that the root node is the smallest element of the order. If the tree is finite, the largest element is the leaf of the rightmost branch of the tree. The root node is FO-definable via $\neg\exists y : y \downarrow x$. The largest element Max of the order is $\mathrm{FO}^{\mathrm{D}*}$-definable by $\exists x \neg \exists y : x < y$. The successor y of a node x in the linear order ($Succ(x, y)$) is also $\mathrm{FO}^{\mathrm{D}*}$-definable: $x < y \wedge \neg\exists z : x < z \wedge z < y$. Using a linear order it is possible to count modulo some natural number on trees. That is for $n, k \in \mathbb{N}$ we can define the class of finite trees such that each tree in the class has $d \times n + k$ nodes (for some $d \in \mathbb{N}$). As an example, we define the class EVEN of trees with an even number of nodes (i.e, $n = 2, k = 0$).

Proposition 2 *The class of finite trees with an even number of nodes* *is* $\mathrm{FO}^{\mathrm{D}*}$*-definable.*

Proof. We only consider the case where a tree has more than two nodes. The formula

$$\exists w : Succ(Root, w) \wedge [\mathrm{DTC}_{x,y} \exists z : Succ(x, z) \wedge Succ(z, y)](w, Max)$$

expresses that we go in one step from the root to its successor w. From w we can reach the last element of the order by an arbitrary number of two successor steps. If we take the two-successors-step path through the linear order from the root to the maximum, we have an odd number of nodes, since a path of n double-successor-steps has $n + 1$ nodes. □

Corollary 3 $\mathrm{FO}^{\mathrm{D}*}$ *has* no *normal form of the type* $[\mathrm{DTC}_{x,y}\ \phi(x, y)](r, r)$ *where* $\phi(x, y)$ *is an FO formula and* r *the root. The same is true mutatis mutandis for* FO^*.

Proof. With a single application of a DTC-operator we can define trees with a linear order. If FO with a single DTC-operator is interpreted over finite successor structures, then it is equivalent to FO with order. But over finite orderings, EVEN is not definable in FO. □

The above corollary is stated here because it contrasts with a fundamental result in descriptive complexity theory. Let FO(TC) be the extension of FO by transitive closure operators of arbitrary width, that is the transitive closure of binary relations on tuples of arbitrary width. Let FO(DTC) be its deterministic counterpart. Immerman (1999) showed that both FO(TC) and FO(DTC) on ordered structures have a normal form consisting of a single outer application of the (deterministic) transitive closure operator on an otherwise FO formula.

6.4 Definability of Tree Structures

In previous and all following sections we assume that we only consider tree models as defined in the Preliminaries section. But in this section we take a more general view, a view that has its origin in model theoretic syntax. The aim is to find whether it is possible to give an axiomatisation of those structures linguists are interested in. This task has two subparts. The first consists of defining trees, or more precisely finite trees, as the intended models. The second part consists of axiomatising linguistic principles such as the Binding theory in the given logic. We will only be concerned with the first part here. This section is inspired by the book by Rogers (1998). More specifically we show that the main results of Chapter 3 carry over to $\mathrm{FO}^{\mathrm{D}*}$. We will frequently cite this chapter in the current section.

The language of this section is deterministic binary transitive closure logic with equality over the following base relations:

$\triangleleft$	parent relation
$\triangleleft^*$	dominance relation
$\triangleleft^+$	proper dominance relation
$\prec$	left-of relation

We also assume there to be a set $\mathcal{L}$ of unary predicate symbols representing linguistic labels. We write $\mathrm{FO}^{\mathrm{D}*\triangleleft}$ for this language to indicate that the base relations differ from the ones in the other sections of this paper.

A model for $\mathrm{FO}^{\mathrm{D}*\triangleleft}$ is a tuple (U, P, D, L, Lab) where U is a non-empty domain, P, D and L are binary relations over U interpreting $\triangleleft, \triangleleft^*$ and $\prec$. And $Lab : \mathcal{L} \to \wp(U)$ interpretes each label as a subset of U.

Since the intended models of this language are trees, we have to restrict the class of models by giving axioms of trees. Many properties of trees can be defined by first-order axioms. The following 12 axioms are cited from (Rogers, 1998, p. 15f.).

A1 $\exists x \forall y : x \triangleleft^* y$
(Connectivity wrt dominance)

A2 $\forall x, y : (x \triangleleft^* y \land y \triangleleft^* x) \to x = y$
(Antisymmetry of dominance)

A3 $\forall x, y, z : (x \triangleleft^* y \land y \triangleleft^* z) \to x \triangleleft^* z$
(Transitivity of dominance)

A4 $\forall x, y : x \triangleleft^+ y \leftrightarrow (x \triangleleft^* y \land x \neq y)$
(Definition of proper dominance)

A5 $\forall x, y : x \triangleleft y \leftrightarrow (x \triangleleft^+ y \land \forall z : (x \triangleleft^* z \land z \triangleleft^* y) \to (z \triangleleft^* x \lor y \triangleleft^* z))$
(Definition of immediate dominance)

A6 $\forall x, z : x \triangleleft^+ z \to ((\exists y : x \triangleleft y \land y \triangleleft^* z) \land (\exists y : y \triangleleft z))$
(Discreteness of dominance)

A7 $\forall x, y : (x \triangleleft^* y \land y \triangleleft^* x) \leftrightarrow (x \not\prec y \land y \not\prec x)$
(Exhaustiveness and exclusiveness)

A8 $\forall w, x, y, z : (x \prec y \land x \triangleleft^* w \land y \triangleleft^* z) \to w \prec z$
(Inheritance of Left-of wrt dominance)

A9 $\forall x, y, z : (x \prec y \land y \prec z) \to x \prec z$
(Transitivity of left-of)

A10 $\forall x, y : x \prec y \to y \not\prec x$
(Asymmetry of left-of)

A11 $\forall x (\exists y : x \triangleleft y) \to (\exists y : x \triangleleft y \land \forall z : x \triangleleft z \to z \not\prec y)$
(Existence of a minimum child)

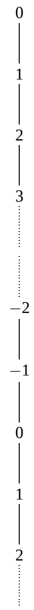

FIGURE 1 A non-standard model of the first-order tree axioms.

A12 $\forall x, z : x \prec z \rightarrow \quad (\exists y : x \prec y \wedge \forall w : x \prec w \rightarrow w \not\prec y) \wedge$
$(\exists y : y \prec z \wedge \forall w : w \prec z \rightarrow y \not\prec w)$
(Discreteness of left-of)

A discussion of these axioms can be found in (Rogers, 1998, p. 16f.). Every tree (finite or infinite) obeys to these axioms. But there are non-standard models, i.e., structures that are models of theses axioms but would not be considered as trees. Actually, it is *not* possible to give a first-order axiomatisation of trees, as was shown by Backofen et al. (1995). The simplest example of a non-standard model can be gained by adapting the well-known example of a non-standard model of FO arithmetics to tree structures. This model is depicted in Figure 1. The carrier is the disjoint union of the natural numbers and the integers. The dominance relation is defined by taking the natural order on natural numbers and integers plus every natural number dominates every integer. Formally: $U = \mathbb{N} \uplus \mathbb{Z}$, $P = \{(n, n+1) \mid n \in \mathbb{N} \cup \mathbb{Z}\}$, $D = \{(n, m) \mid n, m \in \mathbb{N}, n \leq m\} \cup \{(n, m) \mid n, m \in \mathbb{Z}, n \leq m\} \cup \{(n, z) \mid n \in \mathbb{N}, z \in \mathbb{Z}\}$, and $L = \emptyset$. This model is not a tree because the integers are infinitely far away from the root.

The FO axioms demand that the proper dominance relation does

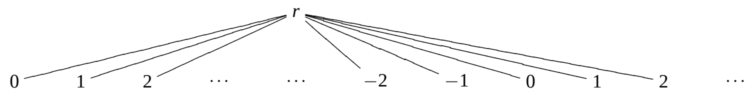

FIGURE 2 Another non-standard model of the first-order tree axioms.

not only contain the immediate dominance relation but also the transitive closure of the immediate dominance. In the non-standard model, proper dominance truly extends the transitive closure of immediate dominance. All natural numbers properly dominate all integers. But this part of the dominance relation is not contained in the transitive closure of immediate dominance. In a proper tree model, the proper dominance is always identical to the transitive closure of immediate dominance. This insight can be expressed in $\mathrm{FO}^{\mathrm{D}*}\triangleleft$ as an axiom.

AT1 $\forall x, y : x \triangleleft^+ y \rightarrow [\mathrm{DTC}_{w,z} z \triangleleft w](y, x)$
(Proper dominance is the transitive closure of immediate dominance)

Another way of reading this axiom is to say that the path from an arbitrary node back to the root is finite.

AT1 together with the first-order axioms does still not suffice to axiomatise proper trees. An example of a non-standard model for which AT1 holds true is given in Figure 2. Formally, we set $U = \{r\} \uplus \mathbb{N} \uplus \mathbb{Z}$, $P = \{(r, z) \mid z \in \mathbb{N} \cup \mathbb{Z}\}$, $D = P \cup \{(i, i) \mid i \in \{r\} \cup \mathbb{N} \cup \mathbb{Z}\}$, and $L = \{(n, m) \mid n, m \in \mathbb{N}, n < m\} \cup \{(n, m) \mid n, m \in \mathbb{Z}, n < m\} \cup \{(n, z) \mid n \in \mathbb{N}, z \in \mathbb{Z}\}$. Consider the sisters of a node. They are ordered by $\prec$, and there is a left-most sister. Now, in a proper tree, the number of sisters to the left is finite for every node. In the model in Figure 2 all integers have infinitely many left sisters. This configuration has to be avoided by means of one more axiom as follows. We can easily define that one node is the immediate sister of another node. The relation $IS(x, y)$ is defined as $\exists z : z \triangleleft x \wedge z \triangleleft y \wedge x \prec y \wedge \neg\exists w : x \prec w \prec y$. Now we can spell out an axiom analogue to AT1.

AT2 $\forall x, y, z : (x \triangleleft y \wedge x \triangleleft z \wedge y \prec z) \rightarrow [\mathrm{DTC}_{v,w} IS(v, w)](y, z)$
(Finitely many left sisters)

Theorem 4 *Axioms A1–A12, AT1, and AT2 define the class of tree models.*

The proof is analogous to the proof of Theorem 3.9 in (Rogers, 1998). Consider in particular Footnote 8 on page 23.

Proof. Rogers showed that every tree (in the sense of Definition 1) is a model of axioms A1–A12 and for each node $x \in U$ the sets $A_x = \{(y, x) \in D\}$ of ancestors of x and $L_x = \{y \mid \exists z : (z, x), (z, y) \in$

D and $(y, x) \in L\}$ of left sisters of x are finite (Lemma 3.5). And every tree obviously satisfies axioms AT1 and AT2.

Furthermore, each model of axioms A1–A12 where A_x and L_x are finite for each node $x \in U$ is isomorphic to a tree (Lemma 3.6).

Now suppose a model of A1–A12 satisfies AT1. Then for each node $x \in U$ the set A_x is finite, because it contains the root (A1) and is constructed of parent-child steps (AT1), and a transitive closure of single steps cannot reach a limit ordinal. An analogous argument can be made with respect to models of A1–A12 and AT2. Hence for every model of of A1–A12, AT1, and AT2 and all nodes $x \in U$ we see that the sets A_x and L_x are finite. By the above quoted Lemma 3.6, these models are isomorphic to trees. □

The tree models of Axioms A1–A2, AT1, and AT2 can be finite as well as infinite. But since they are all tree models, they are at most countable. This is because every tree domain is at most countable (see remark after Definition 1). And every tree model is isomorphic to a tree. As an immediate consequence we get that $\mathrm{FO}^{\mathrm{D}*}$ does *not* have the Löwenheim-Skolem-Upward property. This property states that if a theory (i.e., potentially infinite set of sentences) has a model of size ω it has models of arbitrary infinite cardinalities. It is a typical property of FO logic.

Corollary 5 *The logics* $\mathrm{FO}^{\mathrm{D}*}$ *and* FO^{*} *do* not *have the Löwenheim-Skolem-Upward property.*

Linguists are mostly (if not exclusively) concerned with finite trees. Hence it would be nice if we could restrict the class of models further down to finite trees. This can indeed be done. Rogers (1998) defines a linear order on the nodes of a tree as follows. Node $x < y$ iff $x \triangleleft^{+} y \vee x \prec y$. By Axiom A7, each pair of nodes is either a member of the dominance relation or a member of the left-of relation. Hence this defines indeed a linear order. Actually, the order is the same as the one in the previous section: depth-first left-to-right tree traversal. As in the previous section we use $Succ(x, y)$ for y being the immediate successor of x in the order. Finiteness can now be defined in two steps. Firstly we demand the linear order to be the deterministic transitive closure of the immediate successor relation. The consequence of this demand is that for every element in the order there is only a finite number of nodes that are smaller than this element. Secondly we demand the order to have a maximal element. If the maximal element has only a finite number of elements smaller than it, the tree is obviously finite.

AF $\forall x, y : x < y \implies [\mathrm{DTC}_{x,y}\ Succ(x, y)](x, y) \quad \wedge$
$\exists x \forall y : y < x \vee y = x.$
(Finiteness of the order $<$)

Theorem 6 *Axioms A1–A12, AT1, AT2, and AF define the class of* finite *tree models.*

Proof. By Theorem 4, every model of the Axioms A1–A12, AT1, and AT2 is isomorphic to a tree model. If a model is finite, then AF is certainly true. For the converse, assume that $\forall x, y : x < y \implies [\mathrm{DTC}_{x,y}\ Succ(x, y)](x, y)$. By definition of the DTC-operator, the set $\{y \mid y < x\}$ of elements smaller than x is finite for every node x. If the order has additionally a maximal element m, then it is finite. □

This theorem implies that another property of FO, namely compactness, does not extend to $\mathrm{FO}^{\mathrm{D}*}$.

Corollary 7 *The logics* $\mathrm{FO}^{\mathrm{D}*}$ *and* FO^* *are* not *compact.*

FO, on the other hand, is not capable of defining the class of finite trees. It is well known that compactness and definability of finiteness of models mutually exclude each other.

6.5 Transitive Closure Logics and Tree Walking Automata

Tree walking automata were introduced by Aho and Ullman (1971) as sequential automata on trees. At every moment of its run, a TWA is in a single node of the tree and in one of a finite number of states. It walks around the tree choosing a neighboring node based on the current state, the label of the current node, and the child number of the current node.

More formally, we consider trees of maximal branching degree k. The following definition is mainly cited from (Bojanczyk and Colcombet, 2005). Every node v has a type. The possible values are Types $= \{r, 1, 2, \ldots, k\} \times \{l, i\}$ where r stands for the root, $j \in \{1, \ldots, k\}$ states that v is the j-th child, l states that v is a leaf, i that v is an internal node. A direction is an element of Dir $= \{\uparrow, \downarrow_1, \ldots, \downarrow_k, stay\}$ where $\uparrow$ stands for 'move to the parent', $\downarrow_j$ 'move to the j-th child, and *stay* to 'stay at the current node'. A TWA is a quintuple $(S, \Sigma, \delta, s_0, F)$ where S is a finite set of states, Σ is the alphabet of node labels, $s_0 \in S$ is the initial state and $F \subseteq S$ is the set of final states. The transition relation δ is of the form

$$\delta \subseteq (S \times \text{Types} \times \Sigma) \times (S \times \text{Dir}).$$

A configuration is a pair of a node and a state. A run is a sequence of configurations where every two consecutive configurations are consistent with the transition relation. A run is accepting iff it starts and ends at the root of the tree, the first state is s_0 and the last state is a member of F. The TWA accepts a tree iff there is an accepting run. The set of Σ-trees recognised by a TWA is the set of trees for which there is an accepting run.

Bojanczyk and Colcombet (2005) showed that TWA cannot recognise all regular tree languages. This means that MSO and tree automata are strictly more powerful than TWA. In an extension of their proof we will show that even FO* is more powerful than TWA.

Theorem 8 *The classes of tree languages definable in* FO* *strictly extend the classes of tree languages recognisable by TWA.*

Proof. The proof consists of two parts. We will first show that every TWA-recognisable tree language is FO*-definable. Secondly we will show that there is an FO$^{\text{D}*}$-definable tree language that cannot be recognised by any TWA.

The first part of the proof is based on recent results by Neven and Schwentick (2003). They showed that a tree language is recognisable by a TWA if and only if it is definable by a formula of the following type: $[\text{TC}_{x,y}\ \phi(x,y)](r,r)$ where r is a constant for the root of a tree, ϕ is an FO formula with additional unary $depth_m$ predicates. Apart from the $depth_m$ predicates, these formulae are obviously in FO*. Now, $depth_m(x)$ is true iff x is a multiple of m steps away from the root. For every m, the predicate $depth_m$ can be defined by an FO*-formula: $[\text{TC}_{x_0,x_m}\ \exists x_1,\dots x_{m-1} : x_0 \downarrow x_1 \wedge \cdots \wedge x_{m-1} \downarrow x_m](r,x)$ is a predicate that is true on a node x just in case there is a $k \in \mathbb{N}$ such that x is at depth $k \times m$. Thus every TWA-recognisable tree language is FO*-definable.

To show the second half of the theorem, we will indicate that the separating language L given by Bojanczyk and Colcombet (2005) can be defined in FO$^{\text{D}*}$. The authors consider binary trees. They show (in Fact 1) that L can be defined in first-order logic with the following three basic relations: left and right child, and ancestor relation. Now, left and right child are obviously FO*-definable relations. And the ancestor relation is – as in the previous sections – FO$^{\text{D}*}$-definable by $[\text{DTC}_{y,x}\ x \downarrow y](y,x)$. □

Corollary 9 *There exists an* FO$^{\text{D}*}$*-definable tree language that is* not *TWA-recognisable.*

Please note that there exists an alternative proof of Theorem 8. Engelfriet and Hoogeboom (2006) have recently shown that transitive closure logics correspond to certain pebble automata. (A pebble automaton is a TWA enhanced by a finite sets of pebbles.) More precisely, the deterministic pebble automata have exactly the same expressive power as deterministic binary transitive closure logic. And non-deterministic pebble automata have the same expressive power as binary transitive closure logic where each transitive closure operator is under the scope of an even number of negations. Since a TWA is a pebble automaton with 0 pebbles, the first half of above theorem follows from the equivalence results of (Engelfriet and Hoogeboom, 2006). The second half of the theorem follows from new results by Bojanczyk et al. (2006) who show that each additional pebble extends the expressive power of a pebble automaton. Bojanczyk et al. (2006) also provide an alternative proof of Corollary 9. As a result, either TWA and DPA are incomparable, or TWA are less powerful than DPA.

6.6 Conclusion

We showed a range of properties of FO^{D*} and FO^* to indicate that they should seriously be considered as logics for defining tree languages. Although the addition of binary transitive closure to first-order logic can be seen as a small one, FO^{D*} is capable of expressing important second-order properties over trees. It is possible to define a linear order over the nodes in a tree. And using this order one can count modulo any natural number. On arbitrary structures with appropriate signature one can axiomatise the classes of trees and finite trees. These axiomatisations showed that FO^{D*} is neither compact nor does it have the Löwenheim-Skolem-Upward property. Furthermore although tree walking automata look like they might serve as an automaton model for FO^*, it turns out that FO^* is more powerful than TWA.

A word about complexity issues may be in place. FO^{D*} and FO^* have quite a good data complexity. By translating FO^* formulae into MSO formulae and using the equivalence between MSO and tree automata one can see that FO^* has a linear time data complexity. And since FO^* is a sublogic of FO(TC), it also has NLOGSPACE data complexity whereas FO^{D*} has LOGSPACE data complexity. A straight-forward implementation of transitive closure yields a PTIME query complexity. It is unclear to the author whether this result can be improved upon.

The main open question is of course whether FO^* is strictly less powerful than MSO. It is also interesting to study the relationship of FO^* to modal languages for trees like PDL_{Tree} (Kracht, 1995). Marx (2004) basically showed that PDL_{Tree} is at most as powerful as FO^*_3, where FO^*_3 is the restriction of FO^* where every formula has at most

3 different variables. ten Cate (2006) recently showed that queries in XPath with Kleene star and loop predicate have the same expressive power as FO^*_3.

One may also ask what happens if we introduce the transitive closure of arbitrary relations, not just binary ones. This logic (abbreviated FO(TC)) was introduced by Immerman (see Immerman, 1999) to logically describe the complexity class NLOGSPACE. Tiede and Kepser (2006) have recently shown that FO(TC) is more expressive than MSO over trees. The statement remains true even if one only considers *deterministic* transitive closures.

Acknowledgements

The author wishes to thank four anonymous referees whose comments helped improving the quality of the paper.
This research was funded by a grant of the German Research Foundation (DFG SFB-441).

Stephan Kepser
Collaborative Research Centre 441
University of Tübingen
Germany

References

Aho, Alfred V. and Jeffrey D. Ullman. 1971. Translations on a context-free grammar. *Information and Control* 19:439–475.

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

Bojanczyk, Mikolaj and Thomas Colcombet. 2005. Tree-walking automata do not recognize all regular languages. In H. N. Gabow and R. Fagin, eds., *The 37th ACM Symposium on Theory of Computing (STOC 2005)*, pages 234–243. ACM.

Bojanczyk, Mikołaj, Mathias Samuelides, Thomas Schwentick, and Luc Segoufin. 2006. Expressive power of pebble automata. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds., *Automata, Languages and Programming, ICALP 2006*, LNCS 4051, pages 157–168. Springer.

Courcelle, Bruno. 1990. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, ed., *Handbook of Theoretical Computer Science*, vol. B, chap. 5, pages 193–242. Elsevier.

Engelfriet, Joost and Hendrik Jan Hoogeboom. 2006. Nested pebbles and transitive closure. In B. Durand and W. Thomas, eds., *STACS 2006*, vol. LNCS 3884, pages 477–488. Springer.

Etessami, Kousha and Neil Immerman. 1995. Reachability and the power of local ordering. *Theoretical Computer Science* 148(2):261–279.

Fagin, Ronald. 1975. Monadic generalized spectra. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 21:89–96.

Immerman, Neil. 1999. *Descriptive Complexity*. Springer.

Kracht, Marcus. 1995. Syntactic codes and grammar refinement. *Journal of Logic, Language, and Information* 4(1):41–60.

Marx, Maarten. 2004. XPath with conditional axis relations. In E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, and E. Ferrari, eds., *Advances in Database Technology – EDBT 2004*, vol. LNCS 2992, pages 477–494. Springer.

Moschovakis, Yiannis. 1974. *Elementary Induction on Abstract Structures*. North-Holland Publishing Company.

Neven, Frank and Thomas Schwentick. 2003. On the power of tree-walking automata. *Information and Computation* 183(1):86–103.

Rogers, James. 1998. *A Descriptive Approach to Language-Theoretic Complexity*. CSLI Publications.

ten Cate, Balder. 2006. Expressivity of XPath with transitive closure. In J. van den Bussche, ed., *Proceedings of PODS 2006*, pages 328–337.

Tiede, Hans-Jörg and Stephan Kepser. 2006. Monadic second-order logic over trees and transitive closure logics. In G. Mints, ed., *3th Workshop on Logic, Language, Information, and Computation*.