Capturing MSO with One Quantifier

Anuj Dawar, Luc Segoufin

Capturing MSO with one quantifier *

Anuj Dawar[1] and Luc Segoufin[2]

[1] University of Cambridge Computer Laboratory
[2] INRIA and ENS Cachan

Abstract. We construct a single Lindström quantifier Q such that FO(Q), the extension of first-order logic with Q has the same expressive power as monadic second-order logic on the class of binary trees (with distinct left and right successors) and also on unranked trees with a sibling order. This resolves a conjecture by ten Cate and Segoufin. The quantifier Q is a variation of a quantifier expressing the Boolean satisfiability problem.

1 Introduction

Trees as data structures are ubiquitous, serving as a means of representing and structuring data in almost all fields of computer science. In the last two decades there has been a significant amount of research devoted to investigating the power of languages for querying tree-structured data. In this context monadic second-order logic (MSO) has emerged as a standard against which the expressive power of other languages is compared. On the one hand, satisfiability of MSO formulas is decidable on trees, and model-checking is tractable. On the other hand, the language is expressive enough to subsume most practical query languages for tree-structured data. To be precise, the classes of trees definable in MSO are exactly the regular languages and this close correspondence between the logic and tree automata is one of its most attractive features.

In [9], ten Cate and Segoufin consider a logic for querying trees that is intermediate in expressive power between first-order logic (FO) and MSO, that is FO(MTC), the extension of first-order logic with an operator for defining the transitive closure of a definable binary relation (here MTC stands for *monadic transitive closure*, to distinguish from the general transitive closure operator which would allow us to define the transitive closure of any definable $2k$-ary relation). They show that the expressive power of this logic corresponds to a natural extension of the widely studied XML path language XPath, and also characterise it in terms of an automaton model—that of nested tree-walking automata. Among the results they establish is that the expressive power of FO(MTC) is strictly weaker than that of MSO on trees (whether finite or infinite, ranked or unranked).

FO(MTC) can naturally be seen as an extension of FO with a single generalized quantifier in the sense of Lindström [7]. Such quantifiers are a standard

* The research reported here was carried out while the first author was a visitor at ENS Cachan, funded by a Leverhulme Trust Study Abroad Fellowship.

means in abstract model theory (see [1]) of defining a minimal extension of a logic adding the ability to define a particular property. Note, in contrast, that FO(TC)—the extension of first-order logic with the general transitive closure operator, well studied in descriptive complexity theory (see [5])—does not extend FO with a single quantifier but with an infinite family of *vectorized* quantifiers generated from a single one (as in [3]). In the conclusions of [9], ten Cate and Segoufin ask the question whether there is any finite set of Lindström quantifiers $Q_1, \ldots, Q_n$ such that the extension of FO with these quantifiers would have exactly the expressive power of MSO on trees[3]. In this paper, we answer this question by constructing a single Lindström quantifier Q such that FO(Q) has exactly the same expressive power as MSO on finite trees. We first establish this for binary trees (with distinguished left and right successors) and then, in Section 5 consider the case of (sibling-ordered) unranked trees. The quantifier that we construct, which we call qSAT, is a version of a Boolean satisfiability quantifier. It is obtained by modifying a representation of satisifiability as a class of finite relational structures originally given by Lovász and Gács [8]. The precise definition is given in Section 3.

2 Preliminaries

We write $\mathbb{N}$ for the natural numbers, and we fix an arbitrary finite alphabet Σ for the remainder of this paper. We work with finite trees, either binary or unranked, over Σ. A *binary tree* t over Σ is a finite set $T \subseteq \{0,1\}^*$ of strings that is prefix closed and such that for any string w, $w0 \in T$ iff $w1 \in T$, along with a labelling function $\lambda : T \to \Sigma$. An *unranked tree* t over Σ is a finite set $T \subseteq \mathbb{N}^*$, which is prefix closed and such that if $wj \in T$ for some $w \in \mathbb{N}^*$ and $j \in \mathbb{N}$ then $wi \in T$ for all $i < j$, along with a labelling function $\lambda : T \to \Sigma$. In either case, we refer to the elements of T as nodes, to the empty sequence ε as the *root* of the tree t and any maximal sequence in the set T as a *leaf* of t. A subtree s of a binary tree $t = (T, \lambda)$ is the substructure induced by a set of nodes $S \subseteq T$ such that for some $x \in T$ and some set of tree nodes $W \subseteq \{0,1\}^*$, $S = \{xw \mid w \in W\}$.

In order to define queries over trees in logic, such as first-order or second-order logic, we consider two vocabularies of relations—one for binary trees and one for unranked trees. In the former case, we have two binary relations lsucc (for left successor) and rsucc (for right successor) which are interpreted in a tree t by $\mathrm{lsucc}(x, y)$ if, and only if, $y = x0$ and $\mathrm{rsucc}(x, y)$ if, and only if, $y = x1$. In addition, for each $\sigma \in \Sigma$, we have a unary relation (which we also write σ) so that $\sigma(x)$ holds just in case $\lambda(x) = \sigma$.

In the case of unranked trees, in addition to the unary relations Σ, we have two binary relations succ (the parent relation) and $\prec$ (the sibling order) which

[3] As written in [9], the question asks for a set of such quantifiers with expressive power equivalent to FO(MTC). This is clearly a typographical error and MSO is what is meant.

are defined by $\mathrm{succ}(x, y)$ just in case $y = xz$ for some $z \in \mathbb{N}$ and $x \prec y$ just in case $x = zi$ and $y = zj$ for some $z \in \mathbb{N}^*$ and some $i, j \in \mathbb{N}$ with $i < j$.

The formulas of first-order logic (FO) and monadic second-order logic (MSO) are defined as usual, starting with atomic formulas using the predicate symbols $\Sigma \cup \{\mathrm{lsucc}, \mathrm{rsucc}\}$ (in the case of binary trees) and $\Sigma \cup \{\mathrm{succ}, \prec\}$ in the case of unranked trees and closing under Boolean operations and quantification over elements for FO and over *sets* of elements for MSO. We always assume that the equality predicate is available. For a tree t and a sentence ϕ of any logic, we write $t \models \phi$ to denote that t makes ϕ true in the usual way. In general, for a relational signature τ, we write $\mathrm{Str}(\tau)$ for the collection of finite τ-structures. For a τ-structure $\mathbf{A}$, we write A for its universe, and if ϕ is a formula with free first-order variables, we write $\phi^{\mathbf{A}}$ for the relation defined by the formula ϕ when interpreted in $\mathbf{A}$.

We also sometimes write $x < y$ for nodes x and y in a tree t to denote that $y = xz$ for a non-empty string z, i.e. x is an *ancestor* of y. Note that this relation is definable in MSO as it is the transitive closure of succ (or $\mathrm{lsucc} \cup \mathrm{rsucc}$, in the case of binary trees). This relation is not, in general, definable in FO. Thus, the absence of the ancestor relation from our vocabulary makes the main result adjoining a single quantifier to FO to achieve the expressive power of MSO stronger. We do, however, need the sibling order $\prec$.

A tree automaton is a tuple $A = (Q, s, F, \delta)$ where Q is a finite set of *states*, $s \in Q$ is the *initial state*, $F \subseteq Q$ is the set of *accepting states* and $\delta \subseteq Q \times \Sigma \times Q \times Q$ is the *transition relation*. A *run* of an automaton A on a binary tree $t = (T, \lambda)$ starting with state q is a map $\rho : T \to Q$ such that: $\rho(\varepsilon) = q$; and if $x, y, z \in T$ are such that y is the left successor of x and z is the right successor of x, $\rho(x) = q_1$, $\rho(y) = q_2$, $\rho(z) = q_3$ and $\lambda(x) = \sigma$ then $(q_1, \sigma, q_2, q_3) \in \delta$. We say ρ is an *accepting run* starting with state q if for all leaves x of t, $\rho(x) \in F$. We simply say ρ is *accepting* if it is an accepting run starting from s. We say that A *accepts* t if there is some accepting run of A on t. We also use the term *partial run* of A to depth i from node x to mean a run on the subtree of t rooted at x and including all descendants of x at distance at most i. Note that our automata are top-down in the sense that it is the root that is labelled by the initial state and the leaves by final states in an accepting run. The bottom-up automaton model where leaves are labelled by initial states and the root by a final state yields is known to be equivalent.

It is known since the work Thatcher and Wright [10] and Doner [4] that the class of tree languages accepted by automata is exactly the same as those definable by sentences of MSO (see [11] for an exposition). We formally state one direction of this equivalence for future use.

Theorem 2.1 ([10, 4]). *For any sentence ϕ of* MSO *there is a tree automaton A such that for any binary tree t, $t \models \phi$ if, and only if, A accepts t.*

Let τ and $\tau' = \{R_1, \ldots, R_m\}$ be relational signatures, where R_i is a relation symbol of arity r_i. A sequence $\Psi = \psi_1(\bar{x}_1, \bar{y}), \ldots, \psi_m(\bar{x}_m, \bar{y})$ of formulas of signature τ, where ψ_i has free variables among $x_1, \ldots, x_{r_i}$ and $\bar{y}$ defines an interpretation Ψ that takes a pair $(\mathbf{A}, \bar{a})$ consisting of a τ structure $\mathbf{A}$ and

a tuple $\bar{a}$ from its universe A interpreting the variables $\bar{y}$ to a τ'-structure $\Psi(\mathbf{A}, \bar{a}) = (A, \psi_1^{\mathbf{A},\bar{a}}, \ldots, \psi_m^{\mathbf{A},\bar{a}})$. When $\bar{y}$ is empty, we say that Ψ is an interpretation without parameters.

The following definition of a generalized quantifier is essentially due to Lindström [7].

Definition 2.2. *Let K be a collection of structures of some fixed signature τ, which is closed under isomorphisms, i.e. if $\mathbf{A} \in K$ and $\mathbf{A} \cong \mathbf{B}$ then $\mathbf{B} \in K$. With K we associate the* quantifier Q_K*, which can be adjoined to first-order logic to form an extension* $\mathrm{FO}(Q_K)$*, which is defined by closing* FO *under the following rule for building formulas:*

If $\Psi = (\psi_1, \ldots, \psi_k)$ is an interpretation from τ to τ' then $Q_K \bar{x} \Psi$ is a formula of $\mathrm{FO}(Q_K)$ of signature τ whose free variables are the parameters of Ψ.

The semantics of is given by the following rule: for a τ-structure $\mathbf{A}$ and a valuation $\bar{a}$ for $\bar{y}$,

$$(\mathbf{A}, \bar{a}) \models Q_K \bar{x} \Psi \iff \Psi(\mathbf{A}, \bar{a}) \in K.$$

Where it causes no confusion, we write K both for the quantifier Q_K and the class of structures that defines it.

It should be noted that there are definitions of first-order interpretation in the literature that are more general than what we define. In particular, in our definition, the universe of the interpreted structure $\Psi\mathbf{A}$ is always the same as the universe of $\mathbf{A}$. We do not allow relativization (which restricts the universe to a definable subset), quotienting (where the universe is obtained by taking the quotient of $\mathbf{A}$ under a definable congruence) or vectorizations (where the universe of the interpreted structure is a set of tuples from $\mathbf{A}$). One reason for restricting ourselves in this way is that the simple notion is sufficient for our purpose. Another is that, while relativization and quotienting are harmless, MSO definability is not closed under vectorized interpretations. There are other general notions of interpretation that preserve MSO definability (such as the MSO transductions of Courcelle (see [2]), but we do not need this generality here.

With our definition, MSO definability is closed under first-order interpretations in the sense that if K is definable by an MSO sentence and Ψ is an interpretation, then the class $\{\mathbf{A} \mid \Psi\mathbf{A} \in K\}$ is also MSO-definable. An immediate consequence is the following lemma, which we state for future reference.

Lemma 2.3. *If K is definable by an* MSO *sentence, then every formula of* $\mathrm{FO}(Q_K)$ *is equivalent to a formula of* MSO*.*

3 Satisfiability Quantifier

The quantifier we define is based on a representation of the Boolean satisfiability problem as a class of relational structures. We first consider a classical representation due to Lovász and Gács [8], who showed that this class of structures is NP-complete under (vectorized) first-order interpretations.

Definition 3.1. *Let τ_{SAT} denote the vocabulary (V, C, P, N) where V and C are unary relation symbols and P and N are binary relation symbols. We denote by SAT the class of τ_{SAT}-structures $\mathbf{A}$ in which:*

1. *$V^{\mathbf{A}}$ and $C^{\mathbf{A}}$ partition the universe A;*
2. *$P^{\mathbf{A}}, N^{\mathbf{A}} \subseteq V^{\mathbf{A}} \times C^{\mathbf{A}}$.*
3. *there is a set $S \subseteq V^{\mathbf{A}}$ such that for each $c \in C^{\mathbf{A}}$ there is a $v \in V^{\mathbf{A}}$ such that: either $v \in S$ and $P(v, c)$ or $v \notin S$ and $N(v, c)$.*

The idea is that a structure in SAT represents a propositional formula in CNF. V is the set of variables and C the set of clauses. $P(v, c)$ holds if the variable v appears positively in the clause c and $N(v, c)$ holds if v appears negatively in c. The third condition in the definition ensures that $\mathbf{A} \in \mathsf{SAT}$ only if it represents a *satisfiable* formula. It is immediate from the definition that SAT is definable by a sentence of MSO, since each of the three conditions is easily expressed as an MSO formula.

While SAT is a natural quantifier, expressing a well-known problem, we find it convenient to consider a modification of it, which makes our proof considerably easier. Let τ_{qSAT} be the vocabulary (Cl, Pos, Neg) where Cl is a binary relation and Pos and Neg are ternary relations. For a τ_{qSAT}-structure $\mathbf{A} = (A, Cl, Pos, Neg)$, write flat($\mathbf{A}$) for the τ_{SAT}-structure whose universe is $A \uplus Cl$ (i.e. the disjoint union of A and Cl), which interprets the unary relations V and C by A and Cl respectively, where P is interpreted as the set of pairs (a, c) such that if $c = (a_1, a_2) \in Cl$, then $Pos(a, a_1, a_2)$ holds and similarly N is interpreted as the set of pairs (a, c) such that if $c = (a_1, a_2) \in Cl$, then $Neg(a, a_1, a_2)$ holds.

Definition 3.2. *We define qSAT to be the class of τ_{qSAT}-structures $\mathbf{A}$ such that* flat($\mathbf{A}$) $\in \mathsf{SAT}$.

In other words, while in the τ_{SAT} representation of Boolean formulas, we explicitly have elements for each variable and clause, in the τ_{qSAT} Representation, the universe consists just of the set of variables and the clauses are coded by pairs of variables. This limits us to Boolean formulas where the number of clauses is at most n^2 (where n is the number of variables) but this suffices for our purpose. The reason for considering this more convoluted definition is that in defining an interpretation of τ_{SAT} in a tree t, we are limited to constructing instances where the number of variables *and* clauses is at most the number of nodes in the tree. On the other hand, in interpreting τ_{qSAT}, we can effectively construct instances of quadratic size. This simplifies our argument.

Again, it is quite easy to see that the class of structures qSAT is definable in MSO. Indeed, the definition is obtained as a conjunction of the wellformedness condition:

$$\forall x, y, z(Pos(x, y, z) \vee Neg(x, y, z)) \Rightarrow Cl(y, z)$$

with the satisfiability condition:

$$\exists S \forall x, y(Cl(x, y) \Rightarrow \exists s(S(s) \wedge Pos(s, x, y)) \vee (\neg S(s) \wedge Neg(s, x, y))).$$

Thus, by Lemma 2.3 we immediately have the following lemma.

Lemma 3.3. *Every formula of* FO(qSAT) *is equivalent to a formula of* MSO.

Note that this holds in general, not just on trees.

4 Capturing MSO

In this section, we begin by showing that FO(qSAT) has the same expressive power as MSO on binary trees. Lemma 3.3 established one direction of this equivalence. For the other, we aim to show that for any MSO sentence ϕ, the class of binary trees t such that $t \models \phi$ is reducible, by a first-order interpretation, to the class qSAT. The basic idea of the construction is similar to the proof that any MSO sentence is equivalent, on the class of binary trees, to an *existential* MSO sentence with exactly one second-order quantifier (see [11]).

Fix an MSO sentence ϕ and let $A = (Q, q_1, F, \delta)$ be a tree automaton accepting the set of trees $\{t \mid t \models \phi\}$. Without loss of generality we assume that A is complete: that is, for any state q and any $\sigma \in \Sigma$, there are states s and t with $(q, \sigma, s, t) \in \delta$. Also, we assume $Q = \{q_1, \ldots, q_k\}$ with q_1 being the initial state.

Let t be a tree and let ρ be a run of A on t. Let S_ρ be the set of nodes defined inductively as follows. The root of t is in S_ρ. If x is a node of t with x in S_ρ and $\rho(x) = q_i$ then all descendants of x at distance i are in S_ρ. No other nodes are in S_ρ.

Now, given a binary tree $t = (T, \lambda)$ and a set $S \subseteq T$, we can say that $S = S_\rho$ for some *accepting* run ρ of A if, and only if, the following conditions are satisfied:

1. The root is in S. The left and right successors of the root are in S.
2. For any node x in S, other than root, there is an ancestor of x at distance less than k from x that is in S. We call the ancestor of x that is closest to x and in S the *S-predecessor of x*.
3. If x is in S and y, the S-predecessor of x, is at distance i from x then all descendants of y at distance i are in S and no descendant of y at distance less than i is in S. In this case we say that y is an *i-node*.
4. For every i-node x in S, if $y_1, \ldots, y_n$ are the descendants of x at distance i from x then there is a run of A starting in x in state q_i and reaching y_j in state q_{α_j} such that for all $j \leq n$:
   (a) if the subtree of t rooted at y_j has depth less than α_j then there is an accepting run of A starting from y_j in state q_{α_j}; and
   (b) if the subtree rooted at y_j has depth at least α_j then y_j is a α_j-node.
   Moreover, if y is a leaf of t at distance less than i from x then the run reaches an accepting state in y.

Note that each of the conditions above can be expressed by a first-order formula with a unary relation for S. This is because each of the conditions is only about the local neighbourhood (to distance at most k) of a node x. This shows, in particular, that the class of trees accepted by A is defined by a formula $\exists S\theta$ where θ is first-order. Our aim here is slightly different. We want to use this construction to obtain from a tree t, a *propositional* formula θ_t which is satisfiable

if, and only if, there is an accepting run of A on t. The variables of θ_t are exactly the nodes T so any subset S of T determines a truth assignment to the variables making the variables in S true and all other variables false. Then, each of the conditions above translates into a set of clauses on the variables T. We now show that this translation can be achieved by means of a first-order interpretation.

Lemma 4.1. *For any tree automaton A, there is a first-order interpretation Θ such that for any binary tree t, $\Theta t \in \mathsf{qSAT}$ if, and only if, A accepts t.*

Proof. The instance Θt of qSAT that we construct has as its universe (and therefore the set of variables), the nodes T of t. The clauses are indexed, as required by the definition of qSAT, by pairs of variables. The number of clauses is bounded by $c|T|$ for some constant c (depending on A) and we find it convenient to index the clauses by pairs $(x, y) \in T^2$ where y is an ancestor of x at distance at most c. The distance of y from x effectively serves as an integer index. For any positive integer i, we write $y = \mathrm{anc}_i(x)$ to denote that y is the ancestor of x at distance i. Note that for fixed i this is expressible as a first-order formula with free variables x and y. We also fix an injective mapping of tuples of natural numbers as natural numbers and write, for instance, $\langle l, m, n\rangle$ for the number that codes the triple (l, m, n).

To represent condition 1, for each $x \in \{\varepsilon, 0, 1\}$ we have a clause indexed by (x, x) which is just x (i.e. a single positive occurrence of the variable x).

To represent condition 2 we have, for each node x that is not in $\{\varepsilon, 0, 1\}$, a clause indexed by (x, x) that is $x \to (y_1 \vee \cdots y_k)$ where $y_i = \mathrm{anc}_i(x)$.

To represent condition 3 for any node x, and any i with $1 \leq i \leq k$, let $w_1, \ldots, w_l$ be the descendants of x at distance exactly i from x and $z_1, \ldots, z_m$ be the descendants of x at distance less than i from x. Note that $l, m \leq 2^k$. Then, for each such i, and each j and j' with $1 \leq j, j' \leq l$ we have the clause $x \wedge w_j \to w_{j'}$, indexed by (x, y) for $y = \mathrm{anc}_{\langle 1,i,j,j'\rangle}(x)$. Also for each j and j' with $1 \leq j \leq l$ and $1 \leq j' \leq m$ we have the clause $x \wedge w_j \to \neg z_{j'}$, indexed by (x, y) for $y = \mathrm{anc}_{\langle 2,i,j,j'\rangle}(x)$.

To represent condition 4, for each node x and each $1 \leq i \leq k$, let z be the lexicographically smallest descendant of x at distance i if there is one and let $w_1, \ldots, w_n$ enumerate all the descendants of x at distance i. Consider any run ρ of the automaton A on the subtree rooted at x starting in state q_i, and let $\rho(w_j) = q_l$. We write α_{ρ,w_j} for the propositional formula that is:

- `true` if w_j has no descendants at distance l and there is a run of A starting in q_l on the subtree rooted at w_j which ends in a final state on all leaves; and
- z', where z' is a descendant of w_j at distance l from w_j otherwise.

We now construct the propositional formula:

$$x \wedge z \to \bigvee_{\rho} \bigwedge_{w} \alpha_{\rho,w} \tag{1}$$

where ρ ranges over all partial runs of A on the subtree rooted at x starting in state q_i, and up to depth i such that for any descendant u of x that is at distance less than i from x and is a leaf $\rho(u) \in F$; and w ranges over $\{w_1, \ldots, w_n\}$.

Let $d_1, \ldots, d_r$ be the clauses when the formula (1) is converted to CNF. Note that r is bounded by a function of k. Then, we include the clause d_l indexed by the pair (x, y) where $y = \mathrm{anc}_{\langle i,l \rangle}(x)$.

Note that in the above, clauses are indexed by pairs (x, y) with y an ancestor of x at distance at most c, where c is a function of k. The interpretation Θ takes the tree t to an instance (T, Cl, Pos, Neg) of qSAT where Cl is the set of indices defined above. It is easy to see that Cl is definable by a first-order formula because the distance between x and y is bounded. The only variables that appear in a clause indexed by (x, y) are at distance at most $2k$ from x. Since the number of such nodes is bounded (by a function of k) and a total order on this set is definable in first-order logic, any relation on these is first-order definable. Moreover, whether or not a variable is included in the clause and if so, positively or negatively also depends only on the neighbourhood of x to a bounded distance. In particular, this means that the relations Pos and Neg are easily defined by first-order formulas. The construction above really defines clauses only for nodes x that are far enough away from the root. In particular, if x is at distance less than c from the root, it may not have enough ancestors to code the number of clauses required. However, there are only a bounded number of such nodes and we can deal with them exhaustively inside a first-order formula.

It is easily checked that the instance of qSAT so defined is satisfiable if, and only if, t is accepted by A.

Theorem 4.2. FO($qSAT$) *has the same expressive power as* MSO *on binary trees.*

Proof. Immediate from Lemmas 3.3 and 4.1.

It should be noted that the interpretation constructed in the proof of Lemma 4.1 is one *without parameters*. Thus, the proof also establishes a normal form for the logic FO(qSAT) on binary trees, in which each formula is of the form $\mathsf{qSAT}\Psi$ for a first-order interpretation Ψ.

5 Unranked Trees

In this section, we sketch an argument to show that, even on unranked trees, the expressive power of FO(qSAT) is the same as that of MSO. One direction of this is immediate by Lemma 3.3. For the other direction, we reduce the question to that of binary trees through the standard encoding of unranked trees as binary trees (see, for instance, [6]). Below, we describe the encoding and briefly sketch the reduction.

We define a *partial binary tree* $t = (T, \lambda)$ where $T \subseteq \{0, 1\}^*$ is a finite prefix-closed set of strings and $\lambda : T \to \Sigma$ is a labelling function. In other words, we do not require that every node has either 0 or 2 successors—a node may also have

just a left or just a right successor. We treat such trees, in the natural way, as structures over the signature $\Sigma \cup \{\text{lsucc}, \text{rsucc}\}$.

For an unranked tree $t = (T, \lambda)$ its binary encoding is the unique partial binary tree $s = (S, \mu)$ for which there is a bijection $h : T \to S$ such that for any $x, y \in T$: if y is the $\prec$-first successor of x then $h(y)$ is the left successor of $h(x)$; and if y is the $\prec$-successor of x then $h(y)$ is the right successor of x.

Now, it is easily seen that there is an MSO interpretation that takes a structure $\mathbf{A}$ that is the binary encoding of an unranked tree t to a structure isomorphic to t. Indeed, we can define the $x \prec y$ as the transitive closure of $\text{rsucc}^{\mathbf{A}}$ and $\text{succ}(x, y)$ by $\exists z \text{lsucc}(x, z) \wedge (z = y \vee z \prec y)$. Both of these are MSO definable. This immediately gives us a translation of MSO formulas on unranked trees into corresponding formulas on their binary encodings as stated in the following proposition.

Proposition 5.1. *For any* MSO *formula ϕ there is an* MSO *formula ψ such that an unranked t satisfies ϕ if, and only if, its binary encoding satisfies ψ.*

We next define the *completion* of a partial binary tree $t = (T, \lambda)$ as the binary tree over the alphabet $\Sigma \cup \{\bot\}$ over the set of strings T' which is the minimal set that includes T and also includes $x0$ iff it includes $x1$, for any $x \in \{0, 1\}^*$ and such that the label of any $x \in T$ is $\lambda(x)$, while the label of any $x \notin T$ is $\bot$. While it is not possible to construct an interpretation (in the sense we have defined it) from partial binary trees to their completions because the universes of the structures are different, it is still possible to translate MSO formulas. More specifically, the standard translation of MSO formulas on binary trees to automata easily yields, for any MSO sentence ϕ a tree automaton A such that ϕ is satisfied on a partial binary tree t if, and only if, A accepts the completion of t. It is then an easy exercise to modify the construction in the proof of Lemma 4.1 to obtain, from A an interpretation that takes the partial binary tree t to an instance of qSAT that is satisfiable if, and only if, A accepts the completion of t.

Finally, we note that there is an FO interpretation that takes an unranked tree t and yields (a structure isomorphic to) the binary encoding of t. This is obtained by defining $\text{lsucc}(x, y)$ by the formula $\text{succ}(x, y) \wedge \forall z \neg z \prec x$ and $\text{rsucc}(x, y)$ by $x \prec y \wedge \forall z (y \prec z \Rightarrow (z = y \vee y \prec z))$. This means that for any FO(qSAT) formula ϕ there is an FO(qSAT) formula ψ such that ψ is satisfied in an unranked tree t if, and only if, ϕ is satisfied in the binary encoding of t. This completes the cycle of translations and establishes the following.

Theorem 5.2. FO(qSAT) *has the same expressive power as* MSO *on unranked trees.*

Proof. One direction is immediate from Lemma 3.3. In the other direction, if we have a sentence ϕ of MSO, this translates to a sentence of MSO interpreted on the binary encodings of unranked trees. In turn, this can be turned into an automaton on the completion of the binary encoding, whose acceptance condition is expressed as a FO(qSAT) sentence on binary encodings. This then translates into an FO(qSAT) sentence on unranked trees.

6 Conclusion

We have shown that we can construct a single Lindström quantifier Q such that adding it to first-order logic yields a logic that is able to express all regular tree languages (on either binary or unranked trees). There is one sense in which this is a *completeness* result. It shows that all regular tree languages can be reduced to Q by rather simple first-order reductions, without vectorizations—reductions which MSO is closed under—and at the same time Q is itself definable in MSO. What prevents us from saying that Q is complete for regular tree languages under simple first-order reductions is that Q is not itself a tree language. It might be interesting to find a quantifier that is a tree language that has this property. In other words, is there a tree language that is MSO-complete under simple first-order reductions? One may also ask if similar results hold for natural classes of structures other than trees.

Our quantifier qSAT is a variation of a natural quantifier coding the satisfiability of CNF formulas. As we noted, SAT is perhaps a more natural quantifier coding this problem. Our reasons for using qSAT instead of SAT were technical: the number of clauses in the CNF formulas we construct is potentially greater than (though by no more than a constant factor) the number of nodes in the tree. Perhaps a more sophisticated construction could circumvent this and show that even the quantifier SAT has the property we formulated. It should be noted that qSAT is reducible to SAT by a *vectorized* first-order reduction, indeed one of dimension 2. If this could be achieved by a simple reduction instead, it would indeed establish that $\mathrm{FO}(\mathsf{SAT})$ was as expressive as MSO on trees.

Finally, it is interesting to ask if a similar result holds for the full infinite binary tree. That is, is there a quantifier Q so that $\mathrm{FO}(Q)$ has the same expressive power as MSO. In this case, the expressive power of MSO is strictly greater than that of weak MSO, where set quantification is restricted to finite sets. It seems plausible that one could show that at least the expressive power of weak MSO is captured by a single quantifier.

References

1. J. Barwise and S. Feferman, editors. *Model-Theoretic Logics*. Springer-Verlag, 1985.
2. B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic, a language theoretic approach*. Cambridge University Press, 2012.
3. A. Dawar. Generalized quantifiers and logical reducibilities. *Journal of Logic and Computation*, 5(2):213–226, 1995.
4. J. Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.
5. N. Immerman. *Descriptive Complexity*. Springer, 1999.
6. Leonid Libkin. Logics for unranked trees: An overview. *Logical Methods in Computer Science*, 2, 2006.
7. P. Lindström. First order predicate logic with generalized quantifiers. *Theoria*, 32:186–195, 1966.

8. L. Lovász and P. Gács. Some remarks on generalized spectra. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 23:27–144, 1977.
9. B. ten Cate and L. Segoufin. Transitive closure logic, nested tree walking automata, and XPath. *J. ACM*, 57:18:1–18:41, 2010.
10. J.W. Thatcher and J.B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical systems theory*, 2:57–81, 1968.
11. W. Thomas. Languages, automata and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.