A Bounding Quantifier

Mikołaj Bojańczyk*

Uniwersytet Warszawski, Wydział MIM, Banacha 2, Warszawa, Poland

Abstract. The logic MSOL+$\mathbb{B}$ is defined, by extending monadic second-order logic on the infinite binary tree with a new *bounding quantifier* $\mathbb{B}$. In this logic, a formula $\mathbb{B}X.\varphi(X)$ states that there is a finite bound on the size of sets satisfying $\varphi(X)$. Satisfiability is proved decidable for two fragments of MSOL+$\mathbb{B}$: formulas of the form $\neg\mathbb{B}X.\varphi(X)$, with φ a $\mathbb{B}$-free formula; and formulas built from $\mathbb{B}$-free formulas by nesting $\mathbb{B}$, $\exists$, $\vee$ and $\wedge$.

1 Introduction

Using monadic second-order logic over infinite trees one cannot express properties such as: "there exists bigger and bigger sets such that..." or "there is a bound on the size of sets such that...". In this paper we present decision procedures for an extension of MSOL where such properties are definable.

The need for such cardinality constraints occurs naturally in applications. For instance, a graph that is interpreted in the full binary tree using monadic second-order logic (MSOL) is known to have bounded tree-width if and only if it does not contain bigger and bigger complete bipartite subgraphs [1]. Another example: a formula of the two-way μ-calculus [12] has a finite model if and only if it has a tree model in which there is a bound on the size of certain sets [2]. Sometimes boundedness is an object of interest in itself, cf. [4], where pushdown games with the bounded stack condition are considered.

In light of these examples, it seems worthwhile to consider the logic MSOL+$\mathbb{B}$ obtained from MSOL by adding two new quantifiers $\mathbb{B}$ and $\mathbb{U}$, which express properties like the ones just mentioned. Let $\psi(X)$ be a formula expressing some property of a set X in a labeled infinite tree. The formula $\mathbb{B}X.\psi(X)$ is satisfied in those trees t where there is a finite bound – which might depend on t – on the size of sets F such that the tree $t[X := F]$ satisfies $\psi(X)$. We also consider the dual quantifier $\mathbb{U}$, which states that there is no finite bound.

Adding new constructions to MSOL has a long history. A notable early example is a paper of Elgot and Rabin [6], where the authors investigated what predicates P can be added to MSOL over $\langle\mathbb{N}, \leq\rangle$ while preserving decidability of the theory. Among the positive examples they gave are monadic predicates representing the sets $\{i! : i \in \mathbb{N}\}$, $\{i^k : i \in \mathbb{N}\}$ and $\{k^i : i \in \mathbb{N}\}$. This line of

* Supported by the European Community Research Training Network GAMES and Polish KBN grant No. 4 T11C 042 25.

research was recently continued by Carton and Thomas in [5], where the list was extended by so called *morphic* predicates.

A construction similar to our bounding quantifier can be found in [7], where Klaedtke and Ruess consider extending MSOL on trees and words with cardinality constraints of the form:

$$|X_1| + \cdots + |X_r| < |Y_1| + \cdots + |Y_s|.$$

Although MSOL with these cardinality constraints is in general undecidable, the authors show a decision procedure for a fragment of the logic, where, among other restrictions, quantification is allowed only over finite sets. Interestingly, MSOL+$\mathbb{B}$ is definable using cardinality constraints, although it does fall outside the aforementioned fragment and cannot be described using the techniques in [7]:

$$\mathbb{B}X.\psi \quad \text{iff} \quad \exists Y.\text{Finite}(Y) \land \forall X.(\psi(X) \to |X| < |Y|).$$

Finally, a quantifier that also deals with cardinality can be formulated based on the results of Niwiński in [8]. It is not however the size of sets satisfying $\psi(X)$, but the number of such sets that is quantified. More precisely, a binary tree t satisfies $\exists^{\mathfrak{c}} X.\psi(X)$ if there are continuum sets F such that $t[X := F]$ satisfies $\psi(X)$. This quantifier, it turns out, is definable in MSOL, and thus its unrestricted use retains decidability.

Our bounding quantifier $\mathbb{B}$, however, is *not* definable in MSOL. Using the bounding quantifier, one can define nonregular languages and hence the question: is satisfiability of MSOL+$\mathbb{B}$ formulas decidable? In this paper we investigate this question and, while being unable to provide an exhaustive answer, we present decision procedures for two nontrivial fragments of MSOL+$\mathbb{B}$.

This investigation leads us to identify a class of tree languages, new to our knowledge, which we call quasiregular tree languages. A set of infinite trees is *L-quasiregular* if it coincides with the regular language L over the set of regular trees and, moreover, is the sum of some family of tree regular languages. The intuition behind an L-quasiregular language is that it is a slight non-regular variation over the language L, yet in most situations behaves the same way as L.

On the one hand, quasiregular languages are simple enough to have decidable emptiness: an L-quasiregular language is nonempty iff L is nonempty. On the other hand, quasiregular languages are powerful enough to allow nontrivial applications of the bounding quantifier: they are closed under bounding quantification, existential quantification, conjunction and disjunction. This yields the decidability result:

Theorem 43
The satisfiability problem for existential bounding formulas*, i. e. ones built from an MSOL core by application of* $\mathbb{B}$, $\exists$, $\land$ *and* $\lor$ *is decidable.*

Unfortunately quasiregular languages do not capture all of MSOL+$\mathbb{B}$. For instance, they are not closed under complementation, hence Theorem 43 gives no insight into properties that use the dual quantifier $\mathbb{U}$.

For this reason, we also conduct a separate analysis of the $\mathbb{U}$ quantifier (note that satisfiability for $\mathbb{U}$ is related to *validity* for $\mathbb{B}$). By inspection of an underlying automaton, we prove:

Theorem 47
Satisfiability is decidable for formulas of the form $\mathbb{U}X.\psi$, where ψ is MSOL.

We are, however, unable to extend this result in a fashion similar to Theorem 43, by allowing for non-trivial nesting.

The plan of the paper is as follows. In Section 2, we briefly survey possible applications of the bounding quantifier. After the preliminaries in Section 3, we introduce the quantifier in Section 4. In Sections 4.1 and 4.2 we prove decidability for bounding existential formulas, while in Section 4.3, we prove decidability for formulas which use the unbounding quantifier outside an MSOL formula.

Acknowledgements I would like to thank Igor Walukiewicz, Damian Niwiński and Thomas Colcombet for their valuable suggestions.

2 Applications

In this section we briefly and informally overview three possible applications. We would like to emphasize that in none of these cases does using the bounding quantifier give *new* results, it only simplifies proofs of existing ones.

The first application comes from graph theory. Sometimes a graph $G = (V, E)$ can be interpreted in the unlabeled full binary tree $\{0,1\}^*$ via two formulas: a formula $\alpha(x)$ true for the vertices used to represent a vertex from V and a formula $\beta(x, y)$ representing the edge relation E. From [1], it follows that such a graph $G(\alpha, \beta)$ is of bounded tree-width if and only if there is a fixed bound N on the size n of full bipartite subgraphs $K_{n,n}$ of $G(\alpha, \beta)$. Given two sets $F, G \subseteq \{0,1\}^*$ one can express using MSOL that these sets represent the left and right parts of a bipartite subgraph. The property that there exist bigger and bigger sets F, G encoding a bipartite graph can then, after some effort, be expressed as a formula of the form $\mathbb{U}Z.\psi(Z)$, where the unboundedness of only a single set Z is required. The validity of such a formula in the unlabeled tree can be verified using either one of the Theorems 43 and 47, hence we obtain conceptually simple decidability proof for the problem: "does a graph represented in the full binary tree have bounded tree-width?" [1]

Another application is in deciding the winner in a certain type of pushdown game. A *pushdown game* is a two-player game obtained from a *pushdown graph*. The vertices of the graph are the configurations $(q, \gamma) \in Q \times \Gamma^*$ of a pushdown automaton of state space Q and stack alphabet Γ, while the edges represent the transitions. The game is obtained by adding a partition of Q into states Q_0 of player 0 and states Q_1 of player 1, along with a *winning condition*, or set of plays in $(Q \times \Gamma^*)^{\mathbb{N}}$ that are winning for the player 0. In [4], the authors consider the *bounded stack* winning condition, where a play is winning for player 0 if there is a fixed finite bound on the size of the stacks appearing in it. Using a natural interpretation of the pushdown game in a binary tree, the fact that player 0 wins the game from a fixed position v is equivalent to the satisfiability of a formula

$$\exists S_0 \, \forall S_1 \, \mathbb{B}X. \; \psi(S_0, S_1, X, v)$$

in which $\psi(S_0, S_1, X, v)$ says that X represents a stack appearing in the unique play starting in vertex v and concordant with the strategies S_0 and S_1. We are able to quantify over strategies due to memoryless determinacy of the relevant game. Moreover, by a closer inspection of the game, one can show that $\forall S_1$ can be shifted inside the $\mathbb{B}$ quantifier, yielding an existential bounding formula whose satisfiability is decidable by Theorem 43.

Finally, the bounding quantifier can be applied to the following decision problem [3,2]: "Is a given formula ψ of the modal μ-calculus with backward modalities satisfiable in some finite structure?" In [3] it is shown that the answer is yes iff a certain nonregular language L_ψ of infinite trees is nonempty. This language expresses the property that certain paths in a tree are of bounded length, and can easily be expressed using an existential quasiregular formula.

3 Preliminaries

In this section we define the basic notions used in the paper: infinite trees, regular languages of infinite trees and regular trees.

A *finite A-sequence* is a function $\boldsymbol{a} : \{0, \ldots, n\} \to A$, while an *infinite A-sequence* is a function $\boldsymbol{a} : \mathbb{N} \to A$. We use boldface letters to denote sequences. Given a function $f : A \to B$ and an A-sequence $\boldsymbol{a}$, $f \circ \boldsymbol{a}$ is a well defined B-sequence. Often we will forsake the functional notation and write $\boldsymbol{a}_i$ instead of $\boldsymbol{a}(i)$. The length $|\boldsymbol{a}| \in \mathbb{N} \cup \{\infty\}$ of a sequence is the size of its domain. We use A^* to denote the set of finite A-sequences and $A^\mathbb{N}$ for the set of infinite A-sequences. The concatenation of two sequences $\boldsymbol{a}$ and $\boldsymbol{b}$, denoted by $\boldsymbol{a} \cdot \boldsymbol{b}$, is defined in the usual fashion.

Let Σ be some finite set, called the *alphabet*. An *infinite Σ-tree* is a function $t : \{0,1\}^* \to \Sigma$. Therefore, all infinite trees have the same domain. We denote the set of infinite Σ-trees by $\text{Trees}^\infty(\Sigma)$. An *infinite tree language over Σ* is any subset of $\text{Trees}^\infty(\Sigma)$. Since we will only consider infinite trees in this paper and the next one, we will omit the word infinite and simply write Σ-tree and tree language. A node is any element of $\{0,1\}^*$. We order nodes using the prefix relation $\leq$. Given $v \in \{0,1\}^*$, the *subtree of t rooted in v* is the tree $t|_v$ defined by:

$$t|_v(w) = t(v \cdot w).$$

A *regular tree* is a tree with finitely many distinct subtrees; the class of all regular trees is denoted by REG. An *infinite path* is any infinite sequence of nodes $\boldsymbol{\pi}$ such that:

$$\boldsymbol{\pi}_0 = \varepsilon \quad \boldsymbol{\pi}_1 = \boldsymbol{\pi}_0 \cdot a_0 \quad \boldsymbol{\pi}_2 = \boldsymbol{\pi}_1 \cdot a_1 \quad \cdots \qquad a_i \in \{0,1\} \; .$$

Given two nodes $v < w$, we define the set $\text{Bet}(v, w)$ of elements between v and w as $v \cdot \{0,1\}^* \setminus w \cdot \{0,1\}^*$.

Let Σ be an alphabet and $*$ a letter outside Σ. A *Σ-context* is any $\Sigma \cup \{*\}$-tree C where the label $*$ occurs only once, in a position called the *hole* of C. We don't require this position to be a leaf, since there are no leaves in an infinite tree, but all nodes below the hole are going to be irrelevant to the context. The *domain* $\mathrm{dom}(C)$ of a context C is the set of nodes that are not below or equal to the hole. Given a Σ-tree t and a context $C[]$ whose hole is v, the tree $C[t]$ is defined by:

$$C[t](w) = \begin{cases} t(u) & \text{if } w = v \cdot u \text{ for some } u \in \{0,1\}^*; \\ C(w) & \text{otherwise.} \end{cases}$$

The composition of two contexts C and D is the unique context $C{\cdot}D$ such that $(C{\cdot}D)[t] = C[D[t]]$ holds for all trees t. We do not use multicontexts for infinite trees.

3.1 Nondeterministic Tree Automata and Regular Tree Languages

As in the case of finite trees, regular languages of infinite trees can be defined both using automata and monadic second-order logic. The two approaches are briefly described in this section.

Definition 31 [Parity condition] A sequence $\boldsymbol{a} \in A^{\mathbb{N}}$ of numbers belonging to some finite set of natural numbers A is said to satisfy the *parity condition* if the smallest number occurring infinitely often in $\boldsymbol{a}$ is even.

A *nondeterministic tree automaton with the parity condition* is a tuple

$$\mathcal{A} = \langle Q, \Sigma, q_I, \delta, \Omega \rangle$$

where Q is a finite set of *states*, Σ is the finite *input alphabet*, $q_I \in Q$ is the *initial state*, $\delta \subseteq Q \times \Sigma \times Q \times Q$ is the *transition relation* and $\Omega : Q \to \mathbb{N}$ is the *ranking function*. Elements of the finite image $\Omega(Q)$ are called *ranks*. A *run* of $\mathcal{A}$ over a Σ-tree t is any Q-tree ρ such that

$$\langle \rho(v), t(v), \rho(v{\cdot}0), \rho(v{\cdot}1) \rangle \in \delta \qquad \text{for every } v \in \{0,1\}^*.$$

The run ρ is *accepting* if for every infinite path $\boldsymbol{\pi}$, the sequence of ranks $\Omega \circ \rho \circ \boldsymbol{\pi}$ satisfies the parity condition. The automaton *accepts a tree t from state $q \in Q$* if there is some accepting run with state q labeling the root. A tree is *accepted* if it is accepted from the initial state q_I. The *language of $\mathcal{A}$*, denoted $L(\mathcal{A})$, is the set of trees accepted by $\mathcal{A}$; such a language is said to be *regular*. An automaton is *nonempty* if and only if its language is.

We say two trees s and t are *equivalent for an automaton $\mathcal{A}$*, which is denoted $s \simeq_{\mathcal{A}} t$, if for every state q of $\mathcal{A}$, the tree s is accepted from q if and only if the tree t is. If the trees s and t are equivalent for $\mathcal{A}$, then they cannot be distinguished by a context, i.e. for every context $C[]$, the tree $C[s]$ is accepted by $\mathcal{A}$ if and only if the tree $C[t]$ is.

We now proceed to define the logical approach to regular languages of infinite trees. Consider an alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$. As in the finite tree case, with a Σ-tree, we associate a relational structure

$$\underline{t} = \langle \{0,1\}^*, S_0, S_1, \leq, \underline{\sigma}_1^t, \ldots, \underline{\sigma}_n^t \rangle.$$

The relations are interpreted as follows: S_0 is the set of left sons $\{0,1\}^*0$, S_1 is the set of right sons $\{0,1\}^*1$, $\leq$ is the prefix ordering, while $\underline{\sigma}_i^t$ is the set of nodes that are labeled by the letter σ_i.

With a sentence ψ of monadic second-order logic we associate the language $L(\psi)$ of trees t such that $\underline{t}$ satisfies ψ. Such a language is said to be *MSOL-definable*. A famous result of Rabin [9] says that a language of infinite trees is MSOL-definable if and only if it is regular.

4 The Bounding Quantifier

The logic MSOL+$\mathbb{B}$ is obtained from MSOL by adding two quantifiers: the *bounding quantifier* $\mathbb{B}$, and its dual *unbounding quantifier* $\mathbb{U}$, which we define here using infinitary disjunction and conjunction:

$$\mathbb{B}^i X.\varphi := \forall X.(\varphi(X) \Rightarrow |X| < i) \qquad \mathbb{U}^i X.\varphi := \exists X.(\varphi(X) \land |X| \geq i)$$

$$\mathbb{B}X.\ \varphi := \bigvee_{i \in \mathbb{N}} \mathbb{B}^i X.\varphi \qquad \mathbb{U}X.\ \varphi := \bigwedge_{i \in \mathbb{N}} \mathbb{U}^i X.\varphi$$

MSOL+$\mathbb{B}$ defines strictly more languages than MSOL (see Fact 44), hence it is interesting to consider decidability of the following problem:

Is a given formula of MSOL+$\mathbb{B}$ satisfiable in some infinite tree?

The remainder of this paper is devoted to this question. Although unable to provide a decision procedure for the whole logic, we do identify two decidable fragments. The first, existential bounding formulas, is proved decidable in Sections 4.1 and 4.2, while the second, formulas of the form $\mathbb{U}X.\psi$ with ψ in MSOL, is proved decidable in Section 4.3.

4.1 Quasiregular Tree Languages

Before we proceed with the proof of Theorem 43, we define the concept of a quasiregular tree language. We then demonstrate some simple closure properties of quasiregular tree languages and, in Section 4.2, show that quasiregular tree languages are closed under bounding quantification. These closure properties, along with the decidable nonemptiness of quasiregular languages, yield the decision procedure for existential bounding formulas found in Theorem 43.

For technical reasons, we will find it henceforth convenient to work on trees where the alphabet is the powerset $P(\Sigma)$ of some set Σ. The same results would hold for arbitrary alphabets, but the notation would be more cumbersome. By $\mathrm{Val}(\Sigma)$ we denote the set of $P(\Sigma)$-trees. Elements of the set Σ will be treated as set variables, the intuition being that a tree in $\mathrm{Val}(\Sigma)$ represents a valuation of the variables in Σ. Given a tree $t \in \mathrm{Val}(\Sigma)$ and a set $F \subseteq \{0,1\}^*$, the tree

$$t[X := F] \in \mathrm{Val}(\Sigma \cup \{X\})$$

is defined by adding the element X to the labels of all nodes in F and removing it, if necessary, from all the other nodes.

Bounding quantification for an arbitrary tree language $L \subseteq \mathrm{Val}(\Sigma)$ is defined as follows. A tree $t \in \mathrm{Val}(\Sigma \setminus \{X\})$ belongs to the language $\mathbb{B}X.L$ if there is some finite bound on the size of sets F such that the tree $t[X := F]$ belongs to L.

We now give the key definition of a quasiregular tree language.

Definition 41 (Quasiregular Language) Let L be a regular tree language. A tree language K is *L-quasiregular* if

- $K \cap \mathrm{REG} = L \cap \mathrm{REG}$, and
- K is the union of some family of regular tree languages

A tree language is *quasiregular* if it is L-quasiregular for some regular language L. For the rest of Section 4.1 we will use the letter L for regular languages and the letter K for quasiregular ones.

Lemma 1. *If K is L-quasiregular, then $K \subseteq L$.*

Proof
Let $\{L_i\}_{i \in I}$ be the family of regular tree languages whose union is K. We will show that each language L_i is a subset of L. Indeed, over regular trees L_i is a subset of L, since K and L agree over regular trees. This implies the inclusion $L_i \subseteq L$ for arbitrary trees, since otherwise the regular language $L_i \setminus L$ would be nonempty and therefore, by Rabin's Basis Theorem [10], contain a regular tree. $\square$

The following easy fact shows that emptiness is decidable for quasiregular tree languages given an appropriate presentation:

Fact 42 If K is L-quasiregular, then K is nonempty iff L is nonempty.

Proof
If L is nonempty, then it contains by Rabin's Basis Theorem a regular tree and hence K must contain this same tree. The other implication follows from Lemma 1. $\square$

In particular, every nonempty quasiregular language contains a regular tree. For a variable X we define the *projection function Π_X* which given a tree returns the tree with X removed from all the labels. Projection is the tree language operation corresponding to existential quantification, as testified by the following equation:

$$L(\exists X.\psi) = \Pi_X(L(\psi)).$$

A set $F \subseteq \{0,1\}^*$ is *regular* if the unique tree $t[X := F] \in \mathrm{Val}(\{X\})$ is regular. Equivalently, F is regular if it is a regular word language. The following is a standard result:

Lemma 2. *If a regular tree t belongs to the projection $\Pi_X(L)$ of a regular language L, then $t[X := F]$ belongs to L for some regular set F.*

Proof
Since t is a regular tree, the set $\{t\}$ is a regular tree language and so is $\Pi_X^{-1}(\{t\})$. Therefore the intersection $L \cap \Pi_X^{-1}(\{t\})$ is regular and nonempty and, by Rabin's Basis Theorem, contains some regular tree. Obviously, the X component in this tree must be a regular set. □

Now we are ready to show some basic closure properties of quasiregular languages:

Lemma 3. *Quasiregular languages are closed under projection, intersection and union.*

Proof
The cases of intersection and union are trivial; we will only do the proof for projection. Let K be L-quasiregular. We will show that the projection $\Pi_X(K)$ is $\Pi_X(L)$-quasiregular. First we prove that $\Pi_X(K)$ is the union of a family of regular languages. By assumption, K is the union some family of regular tree languages $\{L_i\}_{i \in I}$. But then

$$\Pi_X(K) = \Pi_X(\bigcup_{i \in I} L_i) = \bigcup_{i \in I} \Pi_X(L_i)$$

and, since regular tree languages are closed under projection, $\Pi_X(K)$ is the union of some family of regular tree languages.

We also need to show that for every regular tree t,

$$t \in \Pi_X(K) \qquad \text{iff} \qquad t \in \Pi_X(L).$$

The left to right implication follows from Lemma 1. The right to left implication follows from the fact that if $t \in \Pi_X(L)$ then, by Lemma 2, for some regular set F, $t[X = F] \in L$. Since the tree $t[X = F]$ is regular, it also belongs to K and hence t belongs to $\Pi_X(K)$.

□

4.2 Closure Under Bounding Quantification

In this section, we show that quasiregular tree languages are closed under application of the bounding quantifier. This, together with the closure properties described in Lemma 3, yields a decision procedure for the fragment of MSOL+$\mathbb{B}$ that nests $\mathbb{B}$ along with existential quantification, conjunction and disjunction.

Recall that a *chain* is any set of nodes that is linearly ordered by $\leq$. We say that a chain C is a *trace path* of a set of nodes F if the set $F \cap \mathrm{Bet}(v, u)$ is nonempty for all nodes $v < w$ in C.

Lemma 4. *A set of at least 3^n nodes has a trace path of size n. An infinite set has an infinite trace path.*

Let $t \in \mathrm{Val}(\Sigma)$ and $L \subseteq \mathrm{Val}(\Sigma \cup \{X\})$. An (L, X)-*bad chain* in the tree t is an infinite chain C whose every finite subset is a trace path of some set F such that $t[X := F] \in L$. The set of trees containing no (L, X)-bad chain is denoted by $\mathbb{C}X.L$. Bad chains have the desirable property of being MSOL definable, as testified by:

Lemma 5. *If L is regular then $\mathbb{C}X.L$ is regular.*

Lemma 6. *If K is L-quasiregular then* $\mathrm{REG} \cap \mathbb{C}X.K = \mathrm{REG} \cap \mathbb{C}X.L$.

Proof
This follows from the fact that for a finite (and therefore regular) node set F and a regular tree t, the tree $t[X := F]$ is regular, and hence belongs to L if and only if it belongs to K. □

Lemma 7. *Let L be a regular tree language. A regular tree that belongs to $\mathbb{C}X.L$ also belongs to $\mathbb{B}X.L$.*

Proof
Consider a regular tree t with m distinct subtrees. Let $\mathcal{A}$ being some automaton recognizing L with k being the index of the relation $\simeq_{\mathcal{A}}$. Setting n to be $3^{k \cdot m+1}$, we will show that if t does not belong to the language $\mathbb{B}^n X.L$, then an (L, X)-bad chain must exist.

Consider indeed a set F of at least n nodes such that the tree $t[X := F]$ belongs to L. By Lemma 4, this set has a trace path with more than $k{\cdot}m$ nodes. Let $v < w$ be two nodes on this trace path such that

$$t[X := F]|_v \simeq_{\mathcal{A}} t[X := F]|_w \qquad \text{and} \qquad t|_v = t|_w \ .$$

Such two nodes exist by virtue of the trace path's size. Moreover, since v and w are on the trace path, the intersection $F \cap \mathrm{Bet}(v, w)$ is nonempty. Let $u \in \{0, 1\}^*$ be such that $w = v \cdot u$.

We claim that the chain $\{v \cdot u^i : i \in \mathbb{N}\}$ is a bad chain. For this, we will show that for every $i \in \mathbb{N}$, the subchain $C_i = \{v \cdot u^j : j \leq i\}$ can be expanded to a set F_i satisfying $t[X := F_i] \in L$.

This is done by pumping i times the part of the set F between v and w. Consider the following partition of F:

$$F_1 = \{u' : u' < v\} \cap F \qquad F_2 = \mathrm{Bet}(v, w) \cap F \qquad F_3 = \{u' : u' > w\} \cap F$$

One can easily check that the following set F_i contains the subchain C_i:

$$F_i \quad = \quad F_1 \ \cup \bigcup_{j \in \{0,\ldots,i\}} v{\cdot}u^j{\cdot}v^{-1}{\cdot}F_2 \ \cup \ v{\cdot}u^i{\cdot}v^{-1}{\cdot}F_3.$$

Moreover, since for all $j \in \{0, \dots, i\}$, the equivalence

$$t[X := F_j]_{v \cdot u^j} \ \simeq_{\mathcal{A}} \ t[X := F]|_w$$

holds, the tree $t[X := F_i]$ belongs to L. □

Using the Lemma 7 above, we can show that quasiregular tree languages are closed under application of the bounding quantifier.

Lemma 8. *If K is quasiregular, then so is $\mathbb{B}X.K$.*

Proof
If K is quasiregular, then it is a union $\bigcup_{i \in I} L_i$ of some family of regular tree languages. Therefore $\mathbb{B}X.\ K$ is also a union regular tree languages:

$$\mathbb{B}X.K = \bigcup_{i \in I} \bigcup_{j > 0} \mathbb{B}^j X.\ L_i.$$

Let L be such that K is L-quasiregular. We will show:

$$\mathbb{C}X.L \cap \mathrm{REG} = \mathbb{B}X.K \cap \mathrm{REG}.$$

The right to left inclusion follows from Lemma 6 and the simple inclusion $\mathbb{B}X.K \subseteq \mathbb{C}X.K$. The left to right inclusion follows from Lemma 7. □

Putting together the closure properties of quasiregular tree languages proved in this and the previous section, we obtain:

Theorem 43
The satisfiability problem for existential bounding formulas*, i.e. ones built from arbitrary MSOL formulas by application of* $\mathbb{B}$, $\exists$, $\wedge$ *and* $\vee$ *is decidable.*

Proof
By Lemmas 3 and 8, the language $L(\psi)$ of an existential bounding formula is L-quasiregular for some effectively obtained regular tree language L. By Fact 42, the emptiness of $L(\psi)$ is equivalent to the emptiness of L. □

Unfortunately, we cannot hope to extend the quasiregular tree language approach to decide all possible nestings of the bounding quantifier, as certified by the following Fact:

Fact 44 Even for regular L, $\neg\mathbb{B}X.L$ is not necessarily quasiregular.

Proof
The language L in question is obtained from a formula ψ with free variables X and Y. This formula states that Y contains no infinite subchains and that X is a subchain of Y.

In a regular tree $t \in \mathrm{Val}(\{X, Y\})$ with n distinct subtrees, a subchain of Y can be of size at most n – otherwise Y has an infinite subchain and ψ does not hold. Therefore $\neg\mathbb{B}X.L(\psi)$ is a nonempty language without a regular tree and cannot be quasiregular. □

4.3 The Unbounding Quantifier

In this section we present a procedure which, given a regular language $L \subseteq \mathrm{Val}(\Sigma)$ and a variable $X \in \Sigma$, decides whether the language $\mathbb{U}X.L$ is nonempty. This implies that satisfiability is decidable for formulas of the form $\mathbb{U}X.\psi(X)$, where ψ is in MSOL. Unfortunately, we are unable to extend this decision procedure to accommodate nesting, the way we did in Theorem 43. On the other hand though, the procedure runs in polynomial time in the size of an input parity automaton.

In order to help the reader's intuition a bit, we will begin our analysis by debunking a natural, yet false, idea: for every regular language L there is some $n \in \mathbb{N}$ such that the language $\mathbb{U}X.L$ is nonempty if and only if the language $\mathbb{U}^n X.L$ is.

The intuition behind this idea would be that a pumping process should inflate arbitrarily a set F satisfying $t[X := F] \in L$ once it has reached some threshold size. The problem, however, is that a tree may contain labels which are not part of the set F, and the pumping might violate this labeling. A suitable counterexample is the following language $L \subseteq \mathrm{Val}(\{X, Y\})$:

X is a subset of Y and Y is a finite set.

Obviously the language $\mathbb{U}X.L$ is empty, yet for every $n \in \mathbb{N}$, the language $\mathbb{U}^n X.L$ is nonempty. We will have to bear such issues in mind in the proofs below, taking care that we pump only the part of the labeling corresponding to X.

Let us fix a set Σ, a regular language $L \subseteq \mathrm{Val}(\Sigma)$ and a variable $X \in \Sigma$ for the rest of this section. We will use $\hat{\Sigma}$ to denote the set $\Sigma \setminus \{X\}$. Analogously to the "language" definition of $\mathbb{B}X.L$ in Section 4.1, we say a tree $t \in \mathrm{Val}(\hat{\Sigma})$ belongs to $\mathbb{U}X.L$ if there is *no* finite bound on the size of sets F such that $t[X := F] \in L$.

An infinite sequence of nodes $\boldsymbol{v}$ is *increasing* if $\boldsymbol{v}_i < \boldsymbol{v}_{i+1}$ holds for all $i \in \mathbb{N}$. A family of node sets $\mathcal{F}$ is *traced* by an increasing sequence $\boldsymbol{v}$ of nodes if for all $i \in \mathbb{N}$,

$$|F \cap \mathrm{Bet}(\boldsymbol{v}_i, \boldsymbol{v}_{i+1})| \geq i \qquad \text{for some } F \in \mathcal{F} .$$

Lemma 9. *A family that contains sets of unbounded size is traced.*

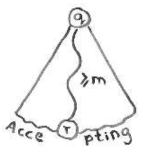
We fix now some nondeterministic parity automaton recognizing L:

$$\mathcal{A} = \langle Q, \Sigma, A, q_I, \delta, \Omega \rangle.$$

Without loss of generality, we assume that every state $q \in Q$ is used in some accepting run. The rest of this section is devoted to an analysis this automaton and to establishing a structural property equivalent to the nonemptiness of the language $\mathbb{U}X.L$.

A *descriptor* is any element of $Q \times \Omega(Q) \times Q$. With a $P(\Sigma)$-context C we associate the set *Trans*(C) consisting of those descriptors (q, m, r) such that there is a run of $\mathcal{A}$ that starts in the root of C in state q and:

- The (finite) path of the run that ends in the hole of C uses states of rank at least m and ends in the state r.
- All the (infinite) paths of the run that do not go through the hole of C satisfy the parity condition.



The intuition is that *Trans*(C) describes the possible runs of $\mathcal{A}$ which go through the context C. The compositions of two descriptors and then of two sets of descriptors are defined below (descriptors which do not agree on the state p do not compose):

$$\begin{aligned} (q,n,p)\cdot(p,m,r) &= (q,\min(n,m),r) && \text{(descriptor)}\\ X\cdot Y &= \{x\cdot y : x\in X, y\in Y\} && \text{(set of descriptors).}\end{aligned}$$

The descriptor set of the context composition $C\cdot D$ can be computed from the composition of the descriptor sets of the contexts C and D:

$$\mathit{Trans}(C\cdot D) = \mathit{Trans}(C)\cdot \mathit{Trans}(D). \qquad (1)$$

We will also be using descriptors of $P(\hat{\Sigma})$-contexts. For a $P(\hat{\Sigma})$-context C and $k\in\mathbb{N}$, we define $\mathrm{Trans}_k(C)$ to be the set

$$\bigcup_{F:|F|\geq k} \mathit{Trans}(C[X := F]).$$

A *schema* is a pair $R = (R^\bullet, R^\circ)$ of descriptor sets. A schema is meant to describe a $P(\hat{\Sigma})$-context, the intuition being that the R° descriptors can be obtained from any sets F, while the $R^\bullet$ descriptors are obtained from "large" sets. A $P(\hat{\Sigma})$-context C is said to *k-realize* a schema R if $R^\circ \subseteq \mathrm{Trans}_0(C)$ and $R^\bullet \subseteq \mathrm{Trans}_k(C)$. The composition $R\cdot S$ of two schemas $R = (R^\bullet, R^\circ)$ and $S = (S^\bullet, S^\circ)$ is defined to be the schema

$$R\cdot S = (R^\bullet\cdot S^\circ \cup R^\circ\cdot S^\bullet, R^\circ\cdot S^\circ).$$

The following obvious fact describes how composition of schemas corresponds to composition of $P(\hat{\Sigma})$-contexts.

Fact 45 Let R and S be schemas which are respectively k-realized by $P(\hat{\Sigma})$-contexts C and D. The schema $R\cdot S$ is k-realized by the context $C\cdot D$.

We now proceed to define the notion of an infinitary sequence. The intuition here is that an infinitary sequence exhibits the existence of a tree belonging to $\mathbb{U}X.L$, which is obtained by composing all the contexts in $\boldsymbol{C}$:

Definition 46 A sequence of schemas $\boldsymbol{R}$ is *infinitary* if both

- There is a sequence of $P(\hat{\Sigma})$-contexts $\boldsymbol{C}$ such that for every $n\in\mathbb{N}$ the schema $\boldsymbol{R}_n$ is n-realized by the context $\boldsymbol{C}_n$; and

– For some fixed state $r \in Q$ and all $n \in \mathbb{N}$, there is a state sequence $\boldsymbol{q}$ with $\boldsymbol{q}_0 = r$ such that:
  1. For $i < n$, $(\boldsymbol{q}_i, m, \boldsymbol{q}_{i+1}) \in \boldsymbol{R}_i^\circ$ for some rank m;
  2. For $i = n$, $(\boldsymbol{q}_i, m, \boldsymbol{q}_{i+1}) \in \boldsymbol{R}_i^\bullet$ for some rank m;
  3. For $i > n$, $(\boldsymbol{q}_i, m, \boldsymbol{q}_{i+1}) \in \boldsymbol{R}_i^\circ$ for some even rank m.

Lemma 10. *$\mathbb{U}X.L$ is nonempty iff there exists an infinitary sequence.*

Proof
Consider first the right to left implication. Let $\boldsymbol{R}$ be the infinitary sequence with $\boldsymbol{C}$ and $r \in Q$ being the appropriate sequence of contexts and starting state from Definition 46. Let $t \in \mathrm{Val}(\hat{\Sigma})$ be the infinite composition of all successive contexts in $\boldsymbol{C}$:

$$t = \boldsymbol{C}_0 \cdot \boldsymbol{C}_1 \cdot \boldsymbol{C}_2 \cdots$$

Let D be a context such that $(q_I, n, r) \in \mathit{Trans}(D)$ for some $n \in \Omega(Q)$. This context exists by our assumption on $\mathcal{A}$ not having useless states. Using the properties of the sequence $\boldsymbol{R}$ postulated in Definition 46, one can easily verify that the tree $D[t]$ belongs to $\mathbb{U}X.L$.

For the left to right implication, consider a tree t in $\mathbb{U}X.L$. From this tree we will extract an infinitary sequence. Consider the family of node sets

$$\{F \subseteq \{0,1\}^* : t[X := F] \in L\}.$$

By assumption on t, this family contains sets of unbounded size. Therefore, by Lemma 9, it is traced by some increasing sequence $\boldsymbol{v}$. Consider the sequence $\boldsymbol{C}$ of contexts, where $\boldsymbol{C}_i$ is obtained from the tree $t|_{\boldsymbol{v}_i}$ by placing the hole in the node corresponding to $\boldsymbol{v}_{i+1}$. One can verify that the sequence of schemas

$$\boldsymbol{R}_i = (\mathit{Trans}(\boldsymbol{C}_i), \mathrm{Trans}_i(\boldsymbol{C}_i)),$$

along with $r = q_I$, is infinitary. □

Although infinitary sequences characterize the unboundedness of L, they are a little hard to work with. That is why we use a special type of infinitary sequence, which nonetheless remains equivalent to the general case (cf. Lemma 12). Consider a very simple schema R which consists of two loops in R° and a connecting descriptor in $R^\bullet$:

$$R = (R^\circ, R^\bullet) \qquad \text{where} \quad R^\circ = \{(q,k,q),(p,m,p)\} \quad \text{and} \quad R^\bullet = \{(q,n,p)\}.$$

We say the pair of states (q,p) used above is *inflatable* if the sequence $\boldsymbol{R}$ constantly equal R is infinitary, for some choice of ranks k, m and n. Note that in this case, the rank m must be even.

Lemma 11. *The set of inflatable pairs can be computed in polynomial time.*

Proof
For $i \in \mathbb{N}$, consider the set A_i of triples

$$\langle (q_1,n_1,p_1),(q_2,n_2,p_2),(q_3,n_3,p_3)\rangle \in (Q \times \Omega(Q) \times Q)^3$$

such that for some $P(\hat{\Sigma})$-context C:

– $(q_1, n_1, p_1), (q_3, n_3, p_3) \in \text{Trans}_0(C)$;
– $(q_2, n_2, p_2) \in \text{Trans}_i(C)$.

Using a dynamic algorithm, the set A_i can be computed in time polynomial on i and the size of the state space Q. By a pumping argument, one can show that the pair (q, p) is inflatable if and only if

$$\langle (q, n_1, q), (q, n_2, p), (p, n_3, p) \rangle \in A_{|Q|+1} \qquad \text{for some even } n_3.$$

□

We now proceed to show Lemma 12, which shows that one can consider inflatable pairs instead of arbitrary infinitary sequences.

Lemma 12. *There is an infinitary sequence iff there is an inflatable pair.*

Proof
An inflatable pair is by definition obtained from an infinitary sequence, hence the right to left implication. For the other implication, consider an infinitary sequence $\boldsymbol{R}$ along with the appropriate sequence of contexts $\boldsymbol{C}$. With every two indices $i < j$, we associate the schema $R[i, j]$ obtained by composing the schemas $\boldsymbol{R}_i \cdot \cdots \cdot \boldsymbol{R}_{j-1}$. Since there is a finite number of schemas, by Ramsey's Theorem [11] there is a schema R and a set of indices $I = \{i_1 < i_2 < \cdots\} \subseteq \mathbb{N}$ such that $R[i, j] = R$ for every $i < j$ in I. Naturally, in this case $R \cdot R = R$ and, by Fact 45, the sequence constantly equal R is an infinitary sequence which realizes the sequence of contexts $\boldsymbol{D}$ defined by

$$\boldsymbol{D}_j = \boldsymbol{C}_{i_j} \cdot \cdots \cdot \boldsymbol{C}_{i_{j+1}-1}.$$

We will now show how to extract an inflatable pair from this sequence. Let $q \to p$ be the relation holding for those states $q, p \in Q$ such that (q, m, p) belongs to R° for some rank m. Since $R \cdot R = R$, the relation $\to$ is transitive. Let $(q, m, p) \in R^\bullet$ be a descriptor used for infinitely many n in clause 2 of Definition 46. We claim:

- $r \to q'$, $q' \to q'$ and $q' \to q$, for some $q' \in Q$. This follows from transitivity of $\to$ if we take n in Definition 46 to be big enough to find a loop.
- $p \to p'$ and $(p', m, p') \in R^\circ$ for some $p' \in Q$ and even rank m. This is done as above.

Consider finally the sequence of contexts $\boldsymbol{E}$ defined by

$$\boldsymbol{E}_i = \boldsymbol{D}_i \cdot \boldsymbol{D}_{i+1} \cdot \boldsymbol{D}_{i+2} \ .$$

By Fact 45, for all $i \in \mathbb{N}$ the schema $R \cdot R \cdot R = R$ is i-realized by the context $\boldsymbol{E}_i$. One can easily verify that the sequence $\boldsymbol{E}$ witnesses the fact that (q', p') is an inflatable pair. □

From Lemmas 10, 11 and 12 we immediately obtain:

Theorem 47
Satisfiability is decidable for formulas of the form $\mathbb{U}X.\psi$, where ψ is MSOL.

Note that the appropriate algorithm is in fact polynomial in the size of a parity automaton recognizing ψ.

5 Closing Remarks

The results in this paper can only be thought of as initiating research regarding of the bounding quantifier: we have not shown satisfiability decidable for the whole logic. The Theorems 43 and 47 can thus be improved by showing satisfiability decidable (or undecidable) for larger fragments than the ones considered above. Moreover, a better complexity assessment for Theorem 43 would be welcome.

References

1. K. Barthelmann. When can an equational simple graph be generated by hyperedge replacement? In *MFCS*, volume 1450 of *Lecture Notes in Computer Science*, pages 543–552, 1998.
2. M. Bojańczyk. Two-way alternating automata and finite models. In *International Colloquium on Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 833–844, 2002.
3. M. Bojańczyk. The finite graph problem for two-way alternating automata. *Theoretical Computer Science*, 298(3):511–528, 2003.
4. A. Bouquet, O. Serre, and I. Walukiewicz. Pushdown games with unboundedness and regular conditions. In *Foundations of Software Technology and Theoretical Computer Science*, volume 2914 of *Lecture Notes in Computer Science*, pages 88–99, 2003.
5. O. Carton and W. Thomas. The monadic theory of morphic infinite words and generalizations. In *Mathematical Foundations of Computer Science*, volume 1893 of *Lecture Notes in Computer Science*, pages 275–284, 2000.
6. C. C. Elgot and M. O. Rabin. Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. *Journal of Symbolic Logic*, 31:169–181, 1966.
7. F. Klaedtke and H. Ruess. Parikh automata and monadic second–order logics with linear cardinality constraints. Technical Report 177, Institute of Computer Science at Freiburg University, 2002.
8. D. Niwiński. On the cardinality of sets of infinite trees recognizable by infinite automata. In *Mathematical Foundations of Computer Science*, volume 520 of *Lecture Notes in Computer Science*, 1991.
9. M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the AMS*, 141:1–23, 1969.
10. M. O. Rabin. *Automata on Infinite Objects and Church's Problem.* American Mathematical Society, Providence, RI, 1972.
11. F. P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, 30:264–285, 1930.
12. M. Vardi. Reasoning about the past with two-way automata. In *International Colloquium on Automata, Languages and Programming*, number 1443 in Lecture Notes in Computer Science, pages 628–641, 1998.