An Extension of Muchnik's Theorem

Achim Blumensath[1] and Stephan Kreutzer[2*]

[1] LuFG Mathematische Grundlagen der Informatik
RWTH Aachen / 52056 Aachen / Germany
blume@i7.informatik.rwth-aachen.de
[2] Laboratory for Foundations of Computer Science
University of Edinburgh / Mayfield Road Edinburgh EH9 3JZ / Scotland, U.K.
kreutzer@i7.informatik.rwth-aachen.de

Abstract. One of the strongest decidability results in logic is the theorem of Muchnik which allows one to transfer the decidability of the monadic second-order theory of a structure to the decidability of the MSO-theory of its iteration, a tree built of disjoint copies of the original structure. We present a generalisation of Muchnik's result to stronger logics, namely guarded second-order logic and its extensions by counting quantifiers. We also establish a strong equivalence result between monadic least fixed-point logic (M-LFP) and MSO on trees by showing that whenever M-LFP and MSO coincide on a structure they also coincide on its iteration.

Keywords: Monadic Second-Order Logic, Muchnik's Theorem, Tree Automata, Fixed-Point Logics

1 Introduction

Initiated by the work of Büchi, Läuchli, Rabin, and Shelah in the late 60s, the investigation of monadic second-order logic (MSO) has received continuous attention. The attractiveness of MSO is due to the fact that, on the one hand, it is quite expressive subsuming – besides first-order logic – most modal logics, in particular the modal μ-calculus. On the other hand, MSO is simple enough such that model checking is still decidable for many structures. Hence, one can obtain decidability results for several logics by just considering MSO.

Of particular interest is Rabin's Tree Theorem [7] which states that the monadic theory of the infinite binary tree is decidable. As the unravelings of (countable) Kripke-structures are MSO-interpretable in the infinite binary tree and many modal logics are contained in MSO, one immediately gets decidability results for the satisfiability problem of these logics. However, the complexity bounds obtained in this way are usually far from optimal.

There have been only a few results improving Rabin's theorem. Shelah [9] mentions a result of Stupp [10] which was later improved by Muchnik. Given a

* This research has partially been supported by the European Community Research Training Network "Games and Automata for Synthesis and Validation" (GAMES), (contract HPRN-CT-2002-00283), see www.games.rwth-aachen.de.

structure $\mathfrak{A} = (A, R_0, \ldots, R_s)$ we can construct a new structure $\mathfrak{A}^*$, called the *iteration* of $\mathfrak{A}$, whose universe A^* consists of all finite sequences of elements of A. For every relation R_i of $\mathfrak{A}$, we have the relation

$$R_i^* := \{\, (wa_0, \ldots, wa_{n-1}) \mid w \in A^*,\ \bar{a} \in R_i \,\},$$

and additionally there are two new relations

$$\begin{aligned} \mathrm{suc} &:= \{\, (w, wa) \mid w \in A^*,\ a \in A \,\} \\ \text{and} \quad \mathrm{cl} &:= \{\, waa \mid w \in A^*,\ a \in A \,\}. \end{aligned}$$

Intuitively, $\mathfrak{A}^*$ consists of countably many copies of $\mathfrak{A}$ which are arranged in a tree-like fashion. The theorem of Muchnik states that the monadic theory of $\mathfrak{A}^*$ is decidable if we can decide the theory of $\mathfrak{A}$.

The original proof of Muchnik has never been published. It is mentioned in Semenov [8]. The first published proof is due to Walukiewicz [11], for a recent exposition see Berwanger and Blumensath [1]. Recently, Kuske and Lohrey proved a version for first-order logic where in addition one can factorise A^* by a trace congruence [5].

To transfer the successful method of showing decidability results for modal logics using the decidability of MSO on the binary tree to logics of higher arity – guarded logics for instance – one needs to extend Muchnik's theorem to logics more expressive than MSO. In the present article we establish such a generalisation of Muchnik's theorem to stronger logics, namely to guarded second-order logic and its extension by counting quantifiers. For the proof we employ the usual technique of translating formulae into automata and vice versa.

Finally, we establish a strong equivalence result between monadic least fixed-point logic (M-LFP) and MSO on trees by showing that whenever M-LFP and MSO coincide on a given structure they also coincide on its iteration.

The paper is organised as follows: In Section 2 we introduce iterations and present Muchnik's theorem and some applications. Section 3 contains the definition of an automaton model introduced by Walukiewicz which takes iterations as input. This model has to be restricted in the following section in order to obtain automata whose expressive power exactly matches the logics we are interested in. In Section 5 we prove our extension of Muchnik's theorem and in the final section we present some applications to fixed-point logic.

2 Tree-like Structures

To fix our notation, let $[n] := \{0, \ldots, n-1\}$. A *$\Sigma$-labelled A-tree* is a function $T : A^* \to \Sigma$ which assigns a label $T(w)$ to each vertex $w \in A^*$.

We will use a variant of monadic second-order logic where all first-order variables are eliminated. That is, formulae are constructed from atoms of the form $X \subseteq Y$ and $RX_0 \ldots X_r$ by boolean operations and set quantification. Using slightly nonstandard semantics we say that $R\bar{X}$ holds if $\bar{a} \in R$ for some elements $a_i \in X_i$. Note that we do not require the X_i to be singletons. Obviously, each MSO-formula can be brought into this form.

A relation $S \subseteq A^s$ is *guarded by a relation* R if, for every tuple $\bar{a} \in S$, there is some $\bar{c} \in R$ such that $\bar{a} \subseteq \bar{c}$. A relation S is *guarded* if it is a union of the form $S_0 \cup \cdots \cup S_n$ where each S_i is guarded by some relation R. *Guarded second-order logic*, GSO, extends first-order logic by second-order quantifiers $\exists_R$ and $\forall_R$ that range over relations guarded by a given relation R (see Grädel, Hirsch, and Otto [4] for a detailed definition and further results on GSO).

We will also use the monadic fragment of least fixed-point logic. Let $\varphi(R, x)$ be a first-order formula with a free unary second-order variable R and a free first-order variable x. On any structure $\mathfrak{A}$ with universe A, the formula φ induces an operator F_φ taking any set $P \subseteq A$ to the set $F_\varphi(P) := \{ a \in A \mid (\mathfrak{A}, P) \models \varphi[a] \}$ of elements satisfying the formula if R is interpreted by P. If φ is positive in R, then this operator is monotone, i.e., for all X, Y, if $X \subseteq Y$ then also $F_\varphi(X) \subseteq F_\varphi(Y)$, and therefore has a least fixed point $\mathrm{LFP}(F_\varphi)$. *Monadic least fixed-point logic* (M-LFP) is defined as the extension of FO by the following formula building rule. If $\varphi(R, x)$ is a formula in M-LFP positive in its free second-order variabe R, then $[\mathrm{LFP}_{R,x}\, \varphi(R, x)](x)$ is also a formula in M-LFP defining the least fixed point of the operator induced by φ. See [2] for details on fixed-point logics.

We are interested in the *iteration* of a structure $\mathfrak{A}$ which consists of disjoint copies of $\mathfrak{A}$ arranged in a tree.

Definition 2.1. Let $\mathfrak{A} = (A, R_0, \dots)$ be a τ-structure. The *iteration* of $\mathfrak{A}$ is the structure $\mathfrak{A}^* := (A^*, \mathrm{suc}, \mathrm{cl}, R_0^*, \dots)$ of signature $\tau^* := \tau \cup \{\mathrm{suc}, \mathrm{cl}\}$ where

$$\begin{aligned} \mathrm{suc} &:= \{ (w, wa) \mid w \in A^*,\ a \in A \}, \\ \mathrm{cl} &:= \{ waa \mid w \in A^*,\ a \in A \}, \\ R_i^* &:= \{ (wa_0, \dots, wa_r) \mid w \in A^*,\ \bar{a} \in R_i \}. \end{aligned}$$

Muchnik has shown that this operations preserves the decidability of monadic theories.

Theorem 2.2 (Muchnik). *For every sentence* $\varphi \in \mathrm{MSO}$ *one can effectively construct a sentence* $\hat{\varphi} \in \mathrm{MSO}$ *such that* $\mathfrak{A} \models \hat{\varphi}$ *iff* $\mathfrak{A}^* \models \varphi$ *for all structures* $\mathfrak{A}$.

This theorem is one of the strongest decidability results known for monadic second-order logic. In particular, it implies Rabin's Tree Theorem.

Example 2.3. Consider the structure $\mathfrak{A}$ with universe $\{0, 1\}$ and two unary predicates $L = \{0\}$ and $R = \{1\}$. MSO model checking for $\mathfrak{A}$ is decidable since $\mathfrak{A}$ is finite. According to Muchnik's Theorem, model checking is also decidable for $\mathfrak{A}^*$. $\mathfrak{A}^*$ is similar to the binary tree. The universe is $\{0, 1\}^*$, and the relations are

$$\begin{aligned} L^* &= \{ w0 \mid w \in \{0, 1\}^* \}, \\ R^* &= \{ w1 \mid w \in \{0, 1\}^* \}, \\ \mathrm{suc} &= \{ (w, wa) \mid a \in \{0, 1\},\ w \in \{0, 1\}^* \}, \\ \mathrm{cl} &= \{ waa \mid a \in \{0, 1\},\ w \in \{0, 1\}^* \}. \end{aligned}$$

In order to prove that model checking for the binary tree is decidable it is sufficient to define its relations in $\mathfrak{A}^*$:

$$S_0xy := \mathrm{suc}(x,y) \wedge L^*y, \qquad S_1xy := \mathrm{suc}(x,y) \wedge R^*y.$$

Similarly the decidability of SωS can be obtained directly without the need to interpret the infinitely branching tree into the binary one.

Example 2.4. Let $\mathfrak{A} := (\omega, \leq)$. The iteration $\mathfrak{A}^*$ has universe ω^* and relations

$$\begin{aligned} \leq^* &= \{\, (wa, wb) \mid a \leq b,\ w \in \omega^* \,\}, \\ \mathrm{suc} &= \{\, (w, wa) \mid a \in \omega,\ w \in \omega^* \,\}, \\ \mathrm{cl} &= \{\, waa \mid a \in \omega,\ w \in \omega^* \,\}. \end{aligned}$$

As final example let us mention that the unraveling of a graph $\mathfrak{G}$ can be defined in $\mathfrak{G}^*$.

Example 2.5. The iteration $\mathfrak{G}^* := (V^*, \mathrm{suc}, \mathrm{cl}, E^*)$ of a graph $\mathfrak{G} = (V, E)$ consists of all finite sequences $w \in V^*$ of vertices. We will construct an MSO-definition of those sequences which are paths in the original graph $\mathfrak{G}$. A word $w \in V^*$ is a path in $\mathfrak{G}$ if for all prefixes of the form uab with $u \in V^*$ and $a, b \in V$ there is an edge $(a, b) \in E$. The prefix relation $\preceq$ is MSO-definable being the transitive closure of the suc relation. Given a prefix $y := uab$ the word $z := uaa$ can be obtained using the clone relation as follows:

$$\psi(y, z) := \exists u\big(\mathrm{suc}(u, y) \wedge \mathrm{suc}(u, z) \wedge \mathrm{cl}(z)\big).$$

Thus, the set of paths in $\mathfrak{G}$ can be defined by

$$\varphi(x) := \forall y \forall z (y \preceq x \wedge \psi(y, z) \rightarrow E^*yz).$$

3 Tree automata

By $\mathcal{B}^+(X)$ we denote the set of (infinitary) positive boolean formulae over X, i.e., all formulae constructed from X with disjunction and conjunction. An interpretation of a formula $\varphi \in \mathcal{B}^+(X)$ is a set $I \subseteq X$ of atoms we consider true.

The main tool used for the investigation of MSO are automata on A-trees. Since A is not required to be finite we need a model of automaton which can work with trees of arbitrary degree. In addition the clone relation cl makes it necessary that the transition function depends on the current position in the input tree. Walukiewicz [11] introduced a type of automaton which satisfies our needs. Since it is fairly general we have to restrict it in the next section.

Definition 3.1. A *tree automaton* is a tuple $\mathcal{A} = (Q, \Sigma, A, \delta, q_0, W)$ where the input is a Σ-labelled A-tree, Q is the set of states, q_0 is the initial state, $W \subseteq Q^\omega$ is the acceptance condition, and

$$\delta : Q \times \Sigma \rightarrow \mathcal{B}^+(Q \times A)^{A^*}$$

is the transition function which assigns to each state q and input symbol c a function $\delta(q, c) : A^* \rightarrow \mathcal{B}^+(Q \times A)$. Frequently we will write $\delta(q, c, w)$ instead of $\delta(q, c)(w)$.

Note that the transition function and acceptance condition of these automata are not finite. To obtain finite automata we will represent the transition function by an MSO-formula and consider only parity acceptance conditions in the next section.

In order to define the language accepted by such an automaton we introduce games.

Definition 3.2. A *game* $\mathcal{G} = (V_0, V_1, E, W)$ is a graph whose universe $V := V_0 \uplus V_1$ is partitioned into positions for, respectively, player 0 and player 1. $W \subseteq V^\omega$ is the winning condition. We assume that every position has an outgoing edge.

The game $\mathcal{G}$ starts at a given position v_0. In each turn the player the current position v belongs to selects an outgoing edge $(v, u) \in E$ and the game continues in position u. The resulting sequence $\pi \in V^\omega$ is called a *play*. Player 0 wins a play π if $\pi \in W$. Otherwise, player 1 wins.

A *strategy* for player i is a function σ that assigns to every prefix $v_0, \dots, v_n$ of a play with $v_n \in V_i$ a successor $v_{n+1} = \sigma(v_0, \dots, v_n)$ such that $(v_n, v_{n+1}) \in E$. σ is *positional* if $\sigma(wv) = \sigma(w'v)$ for all sequences wv, $w'v$ whose last position is the same. A *winning strategy* is a strategy σ such that, whenever player i plays according to σ, then the resulting play is winning for him, regardless of the moves of the opponent.

Below the winning conditions will mostly have the following form:

Definition 3.3. A function $\Omega : \Sigma \to [n]$ induces a *parity condition* $W \subseteq \Sigma^\omega$ that consists of all sequences $(c_i)_{i<\omega} \in \Sigma^\omega$ such that the least number appearing infinitely often in the sequence $(\Omega(c_i))_{i<\omega}$ is even.

A *parity automaton* is a tree automaton $\mathcal{A} = (Q, \Sigma, M, \delta, q_0, W)$ where W is a parity condition. In this case we sometimes write $\mathcal{A} = (Q, \Sigma, M, \delta, q_0, \Omega)$. Similarly, a *parity game* $\mathcal{G} = (V_0, V_1, E, v_0, \Omega)$ is a game with a parity winning condition.

The importance of parity winning conditions stems from the fact that all games with a parity condition are determined and the corresponding winning strategies are positional [3, 6].

Theorem 3.4 (Determinacy of parity games). *For every parity game $\mathcal{G} = (V_0, V_1, E, \Omega)$ there exists a partition $W_0 \uplus W_1$ of the universe such that player i has a positional winning strategy σ_i for all plays starting in a position $v \in W_i$.*

Furthermore, Walukiewicz [11] has shown that the winning region W_0 of a parity game (V_0, V_1, E, Ω) can be defined by a μ-calculus formula. In monadic fixed-point logic it takes the form

$$\mathrm{LFP}_{Z_n,x} \cdots \mathrm{GFP}_{Z_1,x} \bigvee_{k \leq n} \eta_k(x, \bar{Z})$$

with $\eta_k := \Omega_k x \land [V_0 x \to \exists y(Exy \land Z_k y)] \land [V_1 x \to \forall y(Exy \to Z_k y)]$

where $\Omega_k = \Omega^{-1}(k)$ is the set of positions of priority k.

Definition 3.5. Let $\mathcal{A} = (Q, \Sigma, A, \delta, q_0, W)$ be an automaton where the formulae $\delta(q, c)$ are in disjunctive normal form. For each tree $T : A^* \to \Sigma$, we define the game $\mathcal{G}(\mathcal{A}, T)$ as follows:

(a) The set of vertices consists of $V_0 := Q \times A^*$ and $V_1 := \mathscr{P}(Q \times A) \times A^*$.
(b) The initial position is (q_0, ε).
(c) Each node $(q, w) \in V_0$ with $\delta(q, T(w), w) = \bigvee_i \bigwedge \Phi_i$ has the successors (Φ_i, w) for each i. The successors of some node $(\Phi, w) \in V_1$ are the nodes (q, wa) for $(q, a) \in \Phi$.
(d) A play $(q_0, w_0), (\Phi_0, w_0), (q_1, w_1), (\Phi_1, w_1), \ldots$ is winning if the sequence $q_0 q_1 \ldots$ is in W.

The language $L(\mathcal{A})$ recognised by $\mathcal{A}$ is the set of all trees T such that player 0 has a winning strategy for the game $\mathcal{G}(\mathcal{A}, T)$.

In order to obtain automata whose expressive power corresponds to a given logic we have to restrict our model to only allow transition functions $\delta(q, c)$ in a given class $\mathcal{T}$. Walukiewicz has derived conditions on $\mathcal{T}$ which ensure that the class of automata obtained in this way is still closed under boolean operations and projections. Using slightly different operations, we follow the presentation of Berwanger and Blumensath [1].

Besides disjunctions, conjunctions, and duals of formulae $\mathcal{T}$ has to be closed under the following operations:

Definition 3.6. Let $\varphi \in \mathcal{B}^+(Q \times A)$.

(a) The *collection* of φ is defined as follows. Let $\bigvee_i \bigwedge_k (q_{ik}, a_{ik})$ be the disjunctive normal form of φ.

$$\operatorname{collect}(\varphi) := \bigvee_i \bigwedge_{a \in A} (Q_i(a), a) \in \mathcal{B}^+(\mathscr{P}(Q) \times A)$$

where $Q_i(a) := \{\, q_{ik} \mid a_{ik} = a \,\}$.

(b) Let $q' \in Q'$. The *shift* of φ by the state q' is the formula $\mathrm{sh}_{q'} \varphi \in \mathcal{B}^+(Q' \times Q \times A)$ obtained from φ by replacing all atoms (q, a) by (q', q, a).

Theorem 3.7. *Let $\mathcal{T}$ be a class of functions $f : A^{<\omega} \to \mathcal{B}^+(Q \times A)$ where A and Q may be different for each $f \in \mathcal{T}$. If $\mathcal{T}$ is closed under disjunction, conjunction, dual, shift, and collection then the class of automata with transition functions $\delta : Q \times \Sigma \to \mathcal{T}$ is closed under union, complement, and projection, and every such automaton can be transformed into a nondeterministic one.*

4 $\mathfrak{L}$-automata

The type of automata defined in the previous section is much too powerful. In order to prove our extension of Muchnik's Theorem we have to find a subclass whose expressive power on the class of trees obtained from relational structures by the operation of iteration corresponds exactly to the logic in question. Since, in general, a version of this theorem for one logic does not imply the corresponding version for another logic, even if the latter is strictly weaker, we have to state the theorem for each logic separately. To avoid duplicating the proofs we introduce the following notions.

Definition 4.1. A logic $\mathfrak{L}$ *extends* MSO if it contains MSO and is closed under boolean operations and set quantification.

If $\mathfrak{L}$ is a logic extending MSO then we denote by $\mathfrak{L}+\mathrm{GSO}$ the extension of $\mathfrak{L}$ by guarded second-order quantification, $\mathfrak{L}(\exists^\omega)$ extends $\mathfrak{L}$ with the predicate $|X| \geq \aleph_0$, and $\mathfrak{L}+\mathrm{C}$ denotes the extension of $\mathfrak{L}$ by predicates $|X| \equiv k \pmod m$ for all $k, m < \omega$. We adopt the convention that $|X| \equiv k \pmod m$ is false for infinite sets X. In particular, this implies that $\mathfrak{L}+\mathrm{C}$ is at least as expressive as $\mathfrak{L}(\exists^\omega)$.

Definition 4.2. The following class of logics is considered below.

$$\mathcal{L} := \{\mathrm{MSO},\ \mathrm{GSO},\ \mathrm{MSO}(\exists^\omega),\ \mathrm{GSO}(\exists^\omega),\ \mathrm{MSO}(\exists^\omega)+\mathrm{C},\ \mathrm{GSO}(\exists^\omega)+\mathrm{C}\}.$$

Definition 4.3. Let $\mathfrak{A} = (A, \bar{R})$, $S \subseteq (A^*)^s$, and $w \in A^*$. Define

$$S|_w := \{\, a \in A \mid wa \in S \,\}.$$

A relation S is called *local* if $S = \bigcup\{\, wS_w \mid w \in A^* \,\}$, i.e., if every tuple $\bar{c} \in S$ is of the form $(wa_0, \ldots, wa_{n-1})$ for some $w \in A^*$, and $a_0, \ldots, a_{n-1} \in A$.

Remark 4.4. If $S \subseteq A^*$ is guarded by R^* then S is local.

Let $\mathfrak{L}$ be a logic extending MSO. In order to evaluate $\mathfrak{L}$-formulae over the iteration of some structure we translate them into automata where the transition function is defined by $\mathfrak{L}$-formulae. This is done in such a way that the resulting class of automata is expressively equivalent to $\mathfrak{L}$.

Definition 4.5. Let $\mathfrak{L}$ be an extension of MSO, $\mathfrak{A}$ a structure, $\bar{S}$ relations over A^*, $\varphi(X, \bar{Y}; \bar{Z}) \in \mathfrak{L}$, and $n < \omega$. The function

$$\langle\!\langle \varphi; \bar{S} \rangle\!\rangle_{\mathfrak{A}} : A^* \to \mathcal{B}^+([n] \times A)$$

is defined by

$$\begin{aligned}
\langle\!\langle \varphi; \bar{S} \rangle\!\rangle_{\mathfrak{A}}(\varepsilon) \ &:= \bigvee \Big\{ \bigwedge \{\, (q, b) \mid b \in Q_q \,\} \ \Big| \ Q_0, \ldots, Q_{n-1} \subseteq A \text{ such that } \\
&\qquad\qquad \mathfrak{A} \models \varphi(\emptyset, \bar{Q}; \bar{S}|_\varepsilon) \Big\}, \\
\langle\!\langle \varphi; \bar{S} \rangle\!\rangle_{\mathfrak{A}}(wa) &:= \bigvee \Big\{ \bigwedge \{\, (q, b) \mid b \in Q_q \,\} \ \Big| \ Q_0, \ldots, Q_{n-1} \subseteq A \text{ such that } \\
&\qquad\qquad \mathfrak{A} \models \varphi(\{a\}, \bar{Q}; \bar{S}|_{wa}) \Big\}.
\end{aligned}$$

Let $\mathcal{T}^n_{\mathfrak{A}}$ be the set of all functions of the form $\langle\!\langle \varphi; \bar{S} \rangle\!\rangle_{\mathfrak{A}}$.

Definition 4.6. Let $\mathfrak{L}$ be an extension of MSO. An $\mathfrak{L}$-*automaton* is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, \Omega)$ where $Q = [n]$ for some $n \in \omega$ and $\delta : Q \times \Sigma \to \mathfrak{L}$. $\mathcal{A}$ accepts a Σ-labelled structure $\mathfrak{A}^*$ if the automaton $\mathcal{A}_{\mathfrak{A}} := (Q, \Sigma, A, \delta_{\mathfrak{A}}, q_0, \Omega)$ does so, where $\delta : Q \times \Sigma \to \mathcal{T}^n_{\mathfrak{A}}$ is defined by $\delta_{\mathfrak{A}}(q, c) := \langle\!\langle \delta(q, c) \rangle\!\rangle_{\mathfrak{A}}$.

In order to translate formulae into automata, the latter must be closed under all operations available in the respective logic.

Proposition 4.7. *Let $\mathfrak{L}$ be an extension of* MSO. *$\mathfrak{L}$-automata are closed under boolean operations and projection.*

Proof. The proof follows the same lines as the corresponding one of Walukiewicz [11]. By Theorem 3.7 it is sufficient to show closure under disjunction, conjunction, dual, shift, and collection. To do so we will frequently need to convert between interpretations $I \subseteq Q \times A$ of boolean formulae $\langle\!\langle \varphi; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w) \in \mathcal{B}^+(Q \times A)$ and sets $\bar{Q}$ such that $\mathfrak{A} \models \varphi(C, \bar{Q})$. Given $I \subseteq Q \times A$ define

$$Q_i(I) := \{\, a \in A \mid (q_i, a) \in I \,\}$$

for $i < n$, and given $Q_0, \ldots, Q_{n-1} \subseteq A$ define

$$I(\bar{Q}) := \{\, (q_i, a) \mid a \in Q_i,\ i < n \,\}.$$

Note that $I(\bar{Q}(I)) = I$ and $Q_i(I(\bar{Q})) = Q_i$. Then

$$I \models \langle\!\langle \varphi; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w) \quad \text{iff} \quad \mathfrak{A} \models \varphi(C, \bar{Q}(I); \bar{R})$$

and vice versa. (Here and below C denotes the set consisting of the last element of w.)

(disjunction) For the disjunction of two $\mathfrak{L}$-definable functions we can simply take the disjunction of their definitions since

$$\begin{aligned}
& I \models \langle\!\langle \varphi_0; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w) \vee \langle\!\langle \varphi_1; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w) \\
\text{iff}\ \ & I \models \langle\!\langle \varphi_i; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w) \text{ for some } i \\
\text{iff}\ \ & \mathfrak{A} \models \varphi_i(C, \bar{Q}(I); \bar{R}) \text{ for some } i \\
\text{iff}\ \ & \mathfrak{A} \models \varphi_0(C, \bar{Q}(I); \bar{R}) \vee \varphi_1(C, \bar{Q}(I); \bar{R}) \\
\text{iff}\ \ & I \models \langle\!\langle \varphi_0 \vee \varphi_1; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w).
\end{aligned}$$

(dual) The definition of the dual operation is slightly more involved.

$$\begin{aligned}
& I \models \overline{\langle\!\langle \varphi; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w)} \\
\text{iff}\ \ & Q \times A \setminus I \not\models \langle\!\langle \varphi; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w) \\
\text{iff}\ \ & J \models \langle\!\langle \varphi; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w) \text{ implies } J \cap I \neq \emptyset \\
\text{iff}\ \ & \mathfrak{A} \models \varphi(C, \bar{P}; \bar{R}) \text{ implies } P_i \cap Q_i(I) \neq \emptyset \text{ for some } i \\
\text{iff}\ \ & \mathfrak{A} \models \forall \bar{P}\big(\varphi(C, \bar{P}; \bar{R}) \to \textstyle\bigvee_{i<n} P_i \cap Q_i \neq \emptyset\big)
\end{aligned}$$

(conjunction) follows from (disjunction) and (dual).

(shift) For a shift we simply need to renumber the states. If the pair (q_i, q_k) is encoded as number $ni + k$ we obtain

$$\varphi(C, Q_{ni+0}, \ldots, Q_{ni+n-1}; \bar{R}).$$

(collection) The collection of a formula can be defined the following way:

$$I \models \text{collect}\,\langle\!\langle \varphi; \bar{R} \rangle\!\rangle_{\mathfrak{A}}(w)$$

$$\begin{aligned}
\text{iff}\quad & \text{there are } Q'_S \subseteq Q_S(I) \text{ such that } \bar{Q}' \text{ partitions } A \text{ and } \mathfrak{A} \models \varphi(C, \bar{P}; \bar{R}) \\
& \text{where } a \in P_i \;:\text{iff}\; i \in S \text{ for the unique } S \subseteq [n] \text{ with } a \in Q'_S \\
\text{iff}\quad & \text{there are } \bar{Q}' \text{ partitioning } A \text{ such that } \mathfrak{A} \models \varphi(C, \bar{P}; \bar{R}) \text{ where} \\
& P_i := \textstyle\bigcup_{S:i\in S} Q'_S \\
\text{iff}\quad & \mathfrak{A} \models \varphi(C, \bar{P}; \bar{R}) \text{ for some } P_i \subseteq \textstyle\bigcup_{S:i\in S} Q_S \text{ with} \\
& P_i \cap Q_S = \emptyset \text{ for all } S \text{ with } i \notin S \\
\text{iff}\quad & \mathfrak{A} \models \exists \bar{P}\big(\varphi(C, \bar{P}; \bar{R}) \wedge \textstyle\bigwedge_{i<n} P_i \subseteq \bigcup_{S:i\in S} Q_S \\
& \qquad\qquad \wedge \textstyle\bigwedge_{S\subseteq[n]} \bigwedge_{i\notin S} P_i \cap Q_S = \emptyset\big).
\end{aligned}$$

□

For proper extensions $\mathfrak{L}$ of MSO, we further have to prove that $\mathfrak{L}$-automata are closed under the additional operations available in $\mathfrak{L}$.

Proposition 4.8. *Let $\mathfrak{L}$ be an extension of* MSO. $\mathfrak{L}+$GSO-*automata are closed under guarded quantification.*

Proof. In a formula of the form $\exists_{\text{suc}} T\psi$ every k-tuple $\bar{a} \in T$ is contained in an edge $(w_0, w_1) \in \text{suc}$. We can encode $\bar{a}$ by the element w_1 and a function $h : [k] \to [2]$ such that $a_i = w_{h(i)}$. Consequently, the quantifier $\exists_{\text{suc}} T$ can be replaced by 2^k monadic quantifiers $\exists X_h$ where h ranges over $[2]^{[k]}$.

Similarly, since cl is unary we can rewrite a formula of the form $\exists_{\text{cl}} T\psi$ using a monadic quantifier.

It remains to consider formulae $\exists_{R^*} T\psi(\bar{X}; \bar{S}, T)$ with non-monadic variable T. Let $\mathcal{A} = (Q, \Sigma, \delta_{\mathcal{A}}, q_0, \Omega)$ be a nondeterministic automaton equivalent to ψ. Since T ranges over local relations we have $\mathfrak{A}^* \models \exists T\psi(\bar{P}; \bar{S}, T)$ if and only if there are sets $T_w \subseteq A$ such that $\mathfrak{A}^* \models \psi(\bar{P}; \bar{S}, T)$ where $T := \bigcup_w wT_w$. By induction hypothesis, this is equivalent to $\mathcal{A}$ accepting the structure $(\mathfrak{A}^*, \bar{P}, \bar{S}, T)$.

We claim that this is the case if and only if $(\mathfrak{A}^*, \bar{P}, \bar{S})$ is accepted by the automaton $\mathcal{B} = (Q, \Sigma, \delta_{\mathcal{B}}, q_0, \Omega)$ where $\delta_{\mathcal{B}}(q, c) := \exists T\delta_{\mathcal{A}}(q, c)$. Before we prove that $\mathcal{B}$ is the desired automaton, we first show that it is also nondeterministic.

Suppose otherwise. Then there exists a model I of $\langle\!\langle \exists T\delta(q, c); \bar{S} \rangle\!\rangle_{\mathfrak{A}}(w)$ which is minimal and contains pairs $(q_0, a), (q_1, a) \in I$ for some $q_0 \neq q_1$. Since

$$\mathfrak{A} \models \exists T\delta(q, c)(C, \bar{Q}(I); \bar{S}|_w, T)$$

we find some $T' \subseteq A$ such that

$$\mathfrak{A} \models \delta(q, c)(C, \bar{Q}(I); \bar{S}|_w, T').$$

Setting $T := wT'$ it follows that

$$I \models \langle\!\langle \delta(q, c); \bar{S}, T \rangle\!\rangle_{\mathfrak{A}}(w).$$

As $\mathcal{A}$ is nondeterministic there exists a model $I_0 \subset I$ such that $Q_i(I_0) \cap Q_k(I_0) = \emptyset$ for $i \neq k$. But

$$I_0 \models \langle\!\langle \delta(q, c); \bar{S}, T \rangle\!\rangle_{\mathfrak{A}}(w).$$

implies that

$$I_0 \models \langle\!\langle \exists T \delta(q,c); \bar{S} \rangle\!\rangle_{\mathfrak{A}}(w)$$

in contradiction to the minimality of I.

It remains to prove the above claim.

($\Rightarrow$) Let $\varrho : A^* \to Q$ be the run of $\mathcal{A}$ on $(\mathfrak{A}^*, \bar{P}, \bar{S}, T)$. Let $w \in A^*$ and define $I_w := \{\, (\varrho(wa), a) \mid a \in A \,\}$. For all $w \in A^*$ we have

$$\begin{aligned}
& I_w \models \langle\!\langle \delta(q,c); \bar{S}, T \rangle\!\rangle_{\mathfrak{A}}(w) \\
\Rightarrow\ & \mathfrak{A} \models \delta(q,c)(C, \bar{Q}(I_w); \bar{S}|_w, T_w) \\
\Rightarrow\ & \mathfrak{A} \models \exists T \delta(q,c)(C, \bar{Q}(I_w); \bar{S}|_w, T) \\
\Rightarrow\ & I_w \models \langle\!\langle \exists T \delta(q,c); \bar{S} \rangle\!\rangle_{\mathfrak{A}}(w).
\end{aligned}$$

Consequently, ϱ is also a run of $\mathcal{B}$ on $(\mathfrak{A}, \bar{P}, \bar{S})$.

($\Leftarrow$) Let $\varrho : A^* \to Q$ be the run of $\mathcal{B}$ on $(\mathfrak{A}^*, \bar{P}, \bar{S})$. For $w \in A^*$ define $I_w := \{\, (\varrho(wa), a) \mid a \in A \,\}$ and fix some $T_w \subseteq A^r$ such that

$$\mathfrak{A} \models \delta(q,c)(C, \bar{Q}(I_w); \bar{S}|_w, T_w).$$

Define $T := \bigcup_w wT_w$. Then $I_w \models \langle\!\langle \delta(q,c); \bar{S}, T \rangle\!\rangle_{\mathfrak{A}}(w)$. Hence, ϱ is a run of $\mathcal{A}$ on $(\mathfrak{A}, \bar{P}, \bar{S}, T)$. □

Lemma 4.9. *Let $\mathfrak{L}$ be an extension of* MSO*. There exists an $\mathfrak{L}(\exists^\omega)$-automaton recognising the predicate $|X| \geq \aleph_0$.*

Proof. There are two possible scenarios for infinite sets X_i. The prefix closure $\downarrow X_i$ may contain an infinite path, or there is some $w \in \downarrow X_i$ such that $wa \in \downarrow X_i$ for infinitely many elements $a \in A$. The automaton for the predicate $|X_i| \geq \aleph_0$ has states $Q := \{q_0, q_1\}$ and priority function $\Omega(q_0) := 0$, $\Omega(q_1) := 1$. In state q_0 it looks for infinitely many elements $x \in X_i$, whereas in state q_1 it looks for at least one such element. We define the transition function δ such that

$$\begin{aligned}
\delta_{\mathfrak{A}}(q_0, c, w) &= \bigvee_{a \in A} \big((q_0, a) \wedge (q_1, a)\big) \vee \bigvee_{\substack{A_0 \subseteq A \\ |A_0| \geq \aleph_0}} \bigwedge_{a \in A_0} (q_1, a), \\
\delta_{\mathfrak{A}}(q_1, c, w) &= \begin{cases} \text{true} & \text{if } i \in c, \\ \bigvee_{a \in A} (q_1, a) & \text{otherwise,} \end{cases}
\end{aligned}$$

by setting

$$\begin{aligned}
\delta(q_0, c) &= \exists x (Q_0 x \wedge Q_1 x) \vee |Q_1| \geq \aleph_0, \\
\delta(q_1, c) &= \begin{cases} \text{true} & \text{if } i \in c, \\ \exists x Q_1 x & \text{otherwise,} \end{cases}
\end{aligned}$$

□

Lemma 4.10. *Let $\mathfrak{L}$ be an extension of* MSO*. There exists an $\big(\mathfrak{L}{+}\mathrm{C}\big)$-automaton recognising the predicate $|X| \equiv k \pmod{m}$.*

Proof. Since there is an $\mathfrak{L}(\exists^\omega)$-automaton for $|X_i| \geq \aleph_0$ we may assume that X_i is finite when constructing an automaton for the predicate $|X_i| \equiv k \pmod m$.

Let $Q := \{ q_k \mid k < m \}$ and $\Omega(q_k) := 0$ for all k. We label an element w by q_k if $|X \cap wA^*| \equiv k \pmod m$. If n_k is the number of successors wa such that $|X \cap waA^*| \equiv k \pmod m$ then we have

$$|X \cap wA^*| \equiv \sum_{k<m} kn_k + |X \cap \{w\}| \pmod m.$$

Obviously, we only need to know n_k modulo m. Consequently, we define

$$\delta(q_k, c) = \begin{cases} \bigvee_{\bar{n} \in N_{k-1}} \bigwedge_{l<m} |Q_l| \equiv n_l \pmod m & \text{if } i \in c, \\ \bigvee_{\bar{n} \in N_k} \bigwedge_{l<m} |Q_l| \equiv n_l \pmod m & \text{otherwise,} \end{cases}$$

where

$$N_k := \Big\{ \bar{n} \in [m]^k \;\Big|\; \sum_{l<m} ln_l \equiv k \pmod m \Big\}.$$

□

Using the preceding propositions we can state the equivalence result. We say that an automaton $\mathcal{A}$ is *equivalent* to an $\mathfrak{L}$-formula $\varphi(X_0, \dots, X_{m-1})$ where all free variables are monadic if $L(\mathcal{A})$ consists of those structures whose labelling encode sets $\bar{U}$ such that $\varphi(\bar{U})$ holds. The encoding of $\bar{U}$ is the $\mathscr{P}([m])$-labelled tree T such that $T(w) = \{ i \in [m] \mid w \in X_i \}$ for all $w \in \{0,1\}^*$.

Theorem 4.11. *Let $\mathfrak{L} \in \mathcal{L}$. For every formula $\varphi \in \mathfrak{L}$ there is an equivalent $\mathfrak{L}$-automaton and vice versa.*

Proof. ($\Rightarrow$) By induction on $\varphi(\bar{X})$ we construct an equivalent $\mathfrak{L}$-automaton $\mathcal{A} := (Q, \mathscr{P}([m]), \delta, q_0, \Omega)$. We have already seen that $\mathfrak{L}$-automata are closed under all operations of $\mathfrak{L}$. Hence, it only remains to construct automata for atomic formulae.

($X_i \subseteq X_j$) We have to check for every element w of the input tree T that $i \notin T(w)$ or $j \in T(w)$. Thus, we set $Q := \{q_0\}$ with $\Omega(q_0) := 0$ and define the transition function such that

$$\delta_{\mathfrak{A}}(q_0, c, w) = \begin{cases} \bigwedge_{a \in A}(q_0, a) & \text{if } i \notin c \text{ or } j \in c, \\ \text{false} & \text{otherwise.} \end{cases}$$

for each input structure $\mathfrak{A}^*$. This can be done by setting

$$\delta(q_0, c) := \begin{cases} \forall x Q_0 x & \text{if } i \notin c \text{ or } j \in c, \\ \text{false} & \text{otherwise.} \end{cases}$$

($R^* X_{i_1} \dots X_{i_k}$) Set $Q := \{q_0, \dots, q_k\}$ and $\Omega(q_i) := 1$. The automaton guesses a node in the input tree while in state q_0 and checks whether its children are in the relation R. That is,

$$\delta_{\mathfrak{A}}(q_0, c, w) = \bigvee_{a \in A} (q_0, a) \vee \bigvee \{\, (q_1, a_1) \wedge \cdots \wedge (q_k, a_k) \mid \bar{a} \in R^{\mathfrak{A}} \,\},$$

$$\delta_{\mathfrak{A}}(q_j, c, w) = \begin{cases} \text{true} & \text{if } j \in c, \\ \text{false} & \text{otherwise,} \end{cases} \qquad \text{for } 1 \leq j \leq k.$$

The corresponding $\mathfrak{L}$-definition is

$$\delta(q_0, c) := \exists x Q_0 x \vee \exists \bar{x}(R\bar{x} \wedge Q_1 x_1 \wedge \cdots \wedge Q_k x_k),$$

$$\delta(q_j, c) = \begin{cases} \text{true} & \text{if } i_j \in c, \\ \text{false} & \text{otherwise,} \end{cases} \qquad \text{for } 1 \leq j \leq k.$$

($SX_{i_1} \dots X_{i_k}$ for a relation variable S) We reuse the automaton for $R^* \bar{X}$. Set $Q := \{q_0, \dots, q_k\}$, $\Omega(q_i) := 1$, and define

$$\delta(q_0, c) := \exists x Q_0 x \vee \exists \bar{x}(S\bar{x} \wedge Q_1 x_1 \wedge \cdots \wedge Q_k x_k),$$

$$\delta(q_j, c) = \begin{cases} \text{true} & \text{if } i_j \in c, \\ \text{false} & \text{otherwise,} \end{cases} \qquad \text{for } 1 \leq j \leq k.$$

($\mathrm{suc}(X_i, X_j)$) Let $Q := \{q_0, q_1\}$ and $\Omega(q_i) := 1$. We guess some element $w \in X_i$ having a successor in X_j.

$$\delta_{\mathfrak{A}}(q_0, c, w) = \begin{cases} \bigvee_{a \in A} (q_0, a) & \text{if } i \notin c, \\ \bigvee_{a \in A} \big((q_0, a) \vee (q_1, a)\big) & \text{otherwise,} \end{cases}$$

$$\delta_{\mathfrak{A}}(q_1, c, w) = \begin{cases} \text{true} & \text{if } j \in c, \\ \text{false} & \text{otherwise.} \end{cases}$$

The corresponding $\mathfrak{L}$-definition is

$$\delta(q_0, c) := \begin{cases} \exists x Q_0 x & \text{if } i \notin c, \\ \exists x(Q_0 x \vee Q_1 x) & \text{otherwise,} \end{cases}$$

$$\delta(q_1, c) := \begin{cases} \text{true} & \text{if } j \in c, \\ \text{false} & \text{otherwise.} \end{cases}$$

($\mathrm{cl}(X_i)$) Let $Q := \{q_0, q_1\}$ and $\Omega(q_i) := 1$. We guess some element wa such that its successor waa is in X_i.

$$\delta_{\mathfrak{A}}(q_0, c, w) = \begin{cases} \bigvee_{a \in A} (q_0, a) & \text{if } w = \varepsilon, \\ \bigvee_{a \in A} (q_0, a) \vee (q_1, b) & \text{if } w = w'b, \end{cases}$$

$$\delta_{\mathfrak{A}}(q_1, c, w) = \begin{cases} \text{true} & \text{if } i \in c, \\ \text{false} & \text{otherwise.} \end{cases}$$

The corresponding $\mathfrak{L}$-definition is

$$\delta(q_0, c) := \exists x Q_0 x \vee \exists x(Cx \wedge Q_1 x),$$
$$\delta(q_1, c) := \begin{cases} \text{true} & \text{if } i \in c, \\ \text{false} & \text{otherwise.} \end{cases}$$

Note that this is the only place where the transition function actually depends on the current vertex.

($\Leftarrow$) Let $\mathcal{A} = (Q, \Sigma, \delta, 0, \Omega)$ be an $\mathfrak{L}$-automaton and fix an input structure $\mathfrak{A}^*$. W.l.o.g. assume that $\mathcal{A}$ is nondeterministic. $\mathfrak{A}^*$ is accepted by $\mathcal{A}$ if there is an accepting run $\varrho : A^* \to Q$ of $\mathcal{A}$ on $\mathfrak{A}^*$. This can be expressed by an $\mathfrak{L}$-formula $\varphi(\bar{X})$ in the following way: we quantify existentially over tuples $\bar{Q}$ encoding ϱ (i.e., $Q_i = \varrho^{-1}(i)$), and then check that at each position $w \in A^*$ a valid transition is used and that each path in ϱ is accepting. □

Before proceeding to the proof of our extension of Muchnik's Theorem let us give an immediate corollary to the equivalence result.

Theorem 4.12. *If $\mathfrak{L}_0$, $\mathfrak{L}_1 \in \mathcal{L}$ then $\mathfrak{L}_0 \le \mathfrak{L}_1$ on $\mathfrak{A}$ implies $\mathfrak{L}_0 \le \mathfrak{L}_1$ on $\mathfrak{A}^*$.*

Proof. Let $\varphi_0 \in \mathfrak{L}_0$ and $\mathcal{A}_0$ be the corresponding $\mathfrak{L}_0$-automaton. For every formula $\delta_0(q, c) \in \mathfrak{L}_0$ there is an equivalent $\mathfrak{L}_1$-formula. Hence, we can translate $\mathcal{A}_0$ into an $\mathfrak{L}_1$-automaton $\mathcal{A}_1$. The corresponding $\mathfrak{L}_1$-formula φ_1 is the desired translation of φ_0 into $\mathfrak{L}_1$. □

5 Muchnik's Theorem

We are now ready to prove the main result of this article.

Theorem 5.1. *Let $\mathfrak{L} \in \mathcal{L}$. For every sentence $\varphi \in \mathfrak{L}$ one can effectively construct a sentence $\hat{\varphi} \in \mathfrak{L}$ such that $\mathfrak{A} \models \hat{\varphi}$ iff $\mathfrak{A}^* \models \varphi$ for all structures $\mathfrak{A}$.*

Corollary 5.2. *Let $\mathfrak{A}$ be a structure. The $\mathfrak{L}$-theory of $\mathfrak{A}^*$ is decidable if and only if we can decide the $\mathfrak{L}$-theory of $\mathfrak{A}$.*

The proof of Muchnik's Theorem is split into several steps. First, let $\mathcal{A} = (Q, \Sigma, \delta, q_0, \Omega)$ be the $\mathfrak{L}$-automaton equivalent to φ. W.l.o.g. assume that $\Omega(i) = i$ for all $i \in Q = [n]$. Note that the input alphabet $\Sigma = \{\emptyset\}$ of $\mathcal{A}$ is unary since φ is a sentence. We construct a formula $\hat{\varphi}$ stating that player 0 has a winning strategy in the game $\mathcal{G}(\mathcal{A}, \mathfrak{A})$. It follows that $\mathfrak{A} \models \hat{\varphi}$ iff $\mathfrak{A}^* \in L(\mathcal{A})$ iff $\mathfrak{A}^* \models \varphi$.

The game structure. We construct $\hat{\varphi}$ by modifying the formula of Walukiewicz so that it can be evaluated in the structure $\mathfrak{A}$. To do so we embed the game $\mathcal{G}(\mathcal{A}, \mathfrak{A})$ in the $\mathfrak{A}$. First, we reduce the second component of a position (X, w) from $w \in A^*$ to a single symbol $a \in A$. Let $\mathcal{G}'(\mathcal{A}, \mathfrak{A})$ be the game obtained from $\mathcal{G}(\mathcal{A}, \mathfrak{A}^*)$ by identifying all nodes of the form (q, wa) and $(q, w'a)$, i.e.:

(a) Let $V_0 := Q \times A$. The vertices of player 0 are $V_0 \cup \{(q_0, \varepsilon)\}$, those of player 1 are $V_1 := \mathscr{P}(Q \times A)$.

(b) The initial position is (q_0, ε).
(c) Let $\langle\!\langle \delta(q,\emptyset) \rangle\!\rangle_{\mathfrak{A}}(a) = \bigvee_i \bigwedge \Phi_i$ for $a \in A \cup \{\varepsilon\}$. The node $(q,a) \in V_0$ has the successors Φ_i for all i. Nodes $\Phi \in V_1$ have their elements $(q,a) \in \Phi$ as successors.
(d) A play $(q_0,a_0), \Phi_0, (q_1,a_1), \Phi_1, \dots$ is winning if the sequence $(q_i)_{i<\omega}$ satisfies the parity condition Ω.

Lemma 5.3. *Player 0 has a winning strategy from the vertex (q, wa) in the game $\mathcal{G}(\mathcal{A}, \mathfrak{A}^*)$ if and only if he has one from the vertex (q,a) in the game $\mathcal{G}'(\mathcal{A}, \mathfrak{A})$.*

Proof. The unravelings of $\mathcal{G}(\mathcal{A}, \mathfrak{A}^*)$ and $\mathcal{G}'(\mathcal{A}, \mathfrak{A})$ from the respective vertices are isomorphic. □

In the second step we encode the game $\mathcal{G}'(\mathcal{A}, \mathfrak{A})$ as the structure

$$\mathfrak{G}(\mathcal{A}, \mathfrak{A}) := \big(V_0 \cup V_1,\ E,\ \mathrm{eq}_2,\ V_0,\ V_1,\ (S_q)_{q\in Q}, R_0, \dots\big)$$

where (V_0, V_1, E) is the graph of the game,

$$\begin{aligned} &\mathrm{eq}_2(q,a)(q',a') &&:\text{iff } a = a', \\ &S_q(q',a) &&:\text{iff } q = q', \\ &R_i(q_0,a_0)\dots(q_r,a_r) &&:\text{iff } (a_0,\dots,a_r) \in R_i^{\mathfrak{A}}. \end{aligned}$$

Note that these relations only contain elements of V_0. Let $\mathfrak{G}(\mathcal{A}, \mathfrak{A})|_{V_0}$ denote the restriction of $\mathfrak{G}(\mathcal{A}, \mathfrak{A})$ to V_0.

Finally, we can embed $\mathfrak{G}(\mathcal{A}, \mathfrak{A})|_{V_0}$ in $\mathfrak{A}$ via an interpretation.

Definition 5.4. Let $\mathfrak{A} = (A, R_0, \dots, R_r)$ and $\mathfrak{B}$ be structures. An $\mathfrak{L}$-*interpretation* of $\mathfrak{A}$ in $\mathfrak{B}$ is a sequence of $\mathfrak{L}$-formulae $\mathcal{I} := \langle k, (\vartheta_{\bar\imath}^R)_{R,\bar\imath} \rangle$ where, given R of arity r, the indices $\bar\imath$ range over $[k]^r$, such that

(i) $A \cong B \times [k]$,
(ii) $R_j \cong \{\, ((a_1,i_1),\dots,(a_r,i_r)) \mid \mathfrak{B} \models \vartheta_{\bar\imath}^{R_j}(\bar a) \,\}$.

The use of interpretations is made possible by the following property. By MSO_0^+ we denote the set of quantifier-free, positive MSO-formulae.

Lemma 5.5. *Let $\mathcal{I}$ be an MSO_0^+-interpretation and $\varphi \in \mathfrak{L}$ for $\mathfrak{L} \in \mathcal{L}$. There is a formula $\varphi^{\mathcal{I}} \in \mathfrak{L}$ such that $\mathcal{I}(\mathfrak{A}) \models \varphi$ iff $\mathfrak{A} \models \varphi^{\mathcal{I}}$ for every structure $\mathfrak{A}$.*

To construct $\varphi^{\mathcal{I}}$ one simply replaces each relation in φ by its definition.

Lemma 5.6. *There is an MSO_0^+-interpretation $\mathcal{I}$ with $\mathfrak{G}(\mathcal{A}, \mathfrak{A})|_{V_0} = \mathcal{I}(\mathfrak{A})$ for all structures $\mathfrak{A}$.*

Proof. Let $\mathcal{I}$ be defined by $\vartheta_{ik}^{\mathrm{eq}_2}(X,Y) := X = Y$ and

$$\vartheta_k^{S_i}(X) := \begin{cases} \text{true} & \text{if } i = k, \\ \text{false} & \text{otherwise,} \end{cases} \qquad \vartheta_{\bar k}^{R_i}(\bar X) := \begin{cases} R\bar X & \text{if } k_0 = \dots = k_r, \\ \text{false} & \text{otherwise.} \end{cases}$$
□

In order to speak about all of $\mathfrak{G}(\mathcal{A},\mathfrak{A})$ in its restriction to V_0 we treat elements $\Phi \in V_1 = \mathscr{P}(V_0)$ as sets $\Phi \subseteq V_0$. All we have to do is to define the edge relation. We split E into three parts

$$E_0 \subseteq V_0 \times V_1, \quad E_1 \subseteq V_1 \times V_0, \quad \text{and} \quad E_2 \subseteq \{(q_0,\varepsilon)\} \times V_1$$

which we have to define separately by formulae $\varepsilon_0(x,Y)$, $\varepsilon_1(X,y)$, and $\varepsilon_2(Y)$.

Lemma 5.7. *There are $\mathfrak{L}$-formulae $\varepsilon_0(x,Y)$, $\varepsilon_1(X,y)$, and $\varepsilon_2(Y)$ defining the edge relations E_0, E_1 and E_2 respectively.*

Proof. Since $\bigl(\Phi,(q,a)\bigr) \in E_1$ iff $(q,a) \in \Phi$ we set $\varepsilon_1(Y,x) := Yx$.

The definition of ε_0 is more involved. Let $\delta_q(C,\bar{Q}) := \langle\!\langle \delta(q,\emptyset)\rangle\!\rangle_{\mathfrak{A}}$. We have

$$((q,a),\Phi) \in E_0 \quad \text{iff} \quad \mathfrak{A} \models \delta_q(\{a\},\bar{Q})$$

where $Q_i := \{\, b \mid (i,b) \in \Phi \,\}$. In order to evaluate δ_q we need to define $\mathfrak{A}$ inside $\mathfrak{G}(\mathcal{A},\mathfrak{A})$. Since the latter consists of $|Q|$ copies of $\mathfrak{A}$ with universes $(S_q)_{q\in Q}$, we pick one such copy and relativise δ_q to it. For simplicity we choose S_q corresponding to the first component of (q,a).

$$((q,a),\Phi) \in E_0 \quad \text{iff} \quad \mathfrak{G}(\mathcal{A},\mathfrak{A})|_{V_0} \models \delta_q^{S_q}\bigl(\{(q,a)\},\bar{Q}'\bigr)$$

where $Q'_i := \{\,(q,b) \mid (i,b) \in \Phi\,\}$. This condition can be written as

$$\mathfrak{G}(\mathcal{A},\mathfrak{A})|_{V_0} \models \exists C \exists \bar{Q} \Bigl(\delta_q^{S_q}(C,\bar{Q}) \wedge C = \{(q,a)\} \wedge \bigwedge_{i\in Q} Q_i = \{\,(q,b) \mid (i,b) \in \Phi\,\}\Bigr).$$

Thus, we define

$$\varepsilon_0(x,Y) := \bigvee_{q\in Q} \bigl(S_q x \wedge \varepsilon_0^q(x,Y)\bigr)$$

where

$$\varepsilon_0^q(x,Y) := \exists C \exists \bar{Q} \Bigl(\delta_q^{S_q}(C,\bar{Q}) \wedge C = \{x\} \wedge \bigwedge_{i\in Q} Q_i = \{\,(q,b) \mid (i,b) \in Y\,\}\Bigr).$$

Obviously, $Q_i = \{\,(q,b) \mid (i,b) \in Y\,\}$ can be expressed by an FO-formula using eq_2. In the same way we define

$$\varepsilon_2(Y) := \exists \bar{Q} \Bigl(\delta_{q_0}^{S_{q_0}}(\emptyset,\bar{Q}) \wedge \bigwedge_{i\in Q} Q_i = \{\,(q_0,b) \mid (i,b) \in Y\,\}\Bigr).$$

□

The winning set. It remains to evaluate the formula

$$\mathrm{LFP}_{Z_1,x} \cdots \mathrm{GFP}_{Z_n,x} \bigvee_{i\le n} \eta_i(x,\bar{Z})$$

with $\eta_i := S_i x \wedge [V_0 x \rightarrow \exists y(Exy \wedge Z_i y)] \wedge [V_1 x \rightarrow \forall y(Exy \rightarrow Z_i y)]$

which defines the winning set in the original game graph $\mathcal{G}'(\mathcal{A},\mathfrak{A})$. Since in the given game the nodes of V_0 and V_1 are strictly alternating, we remain in V_0 if we take two steps each time.

$$\eta'_i := S_i x \wedge V_0 x \wedge \exists y\bigl(V_1 x \wedge Exy \wedge \forall z(Eyz \rightarrow Z_i z)\bigr)$$

Lemma 5.8. *The formulae* $\mathrm{GFP}_{Z_1,x} \bigvee_{i\leq n} \eta_i$ *and* $\mathrm{GFP}_{Z_1,x} \bigvee_{i\leq n} \eta'_i$ *define the same subset of* V_0 *in* $\mathfrak{G}(\mathcal{A},\mathfrak{A})$ *for each assignment of the free variables.*

Finally, interpreting elements of V_1 by subsets of V_0, as explained above, we obtain

$$\eta''_i := S_i x \wedge V_0 x \wedge \exists Y\big(Y \subseteq V_0 \wedge \varepsilon_0(x,Y) \wedge \forall z(\varepsilon_1(Y,z) \rightarrow Z_i z)\big)$$

Again, the equivalence of η'_i and η''_i is checked easily. Thus, we can state that player 0 has a winning strategy in $\mathcal{G}'(\mathcal{A},\mathfrak{A})$ from position (q_0,ε) by

$$\hat{\varphi} := \exists Y\big[\varepsilon_2(Y) \wedge \forall x\big(\varepsilon_0(Y,x) \rightarrow \mathrm{LFP}_{Z_1,x} \cdots \mathrm{GFP}_{Z_n,x} \textstyle\bigvee_{i\leq n} \eta''_i\big)\big].$$

This concludes the proof of Theorem 5.1.

6 Least fixed-point logic

We conclude this article by deriving conditions which imply that monadic second-order logic collapses to monadic fixed-point logic.

Theorem 6.1. *Let* $\mathfrak{A}$ *be a structure where* MSO (*effectively*) *collapses to* M-LFP. *For every formula* $\varphi(\bar{X}) \in$ MSO *one can* (*effectively*) *construct a formula* $\hat{\varphi}(\bar{X}) \in$ M-LFP *such that* $\mathfrak{A}^* \models \varphi(\bar{P})$ *iff* $\mathfrak{A}^* \models \hat{\varphi}(\bar{P})$.

Proof. The proof is analogous to the one of Muchnik's Theorem. Let $\mathcal{A} = (Q,\Sigma,\delta,q_0,\Omega)$ be the MSO-automaton equivalent to φ. We construct an LFP-formula $\hat{\varphi}$ stating that player 0 has a winning strategy in the game $\mathcal{G}(\mathcal{A},\mathfrak{A})$. Hence, $\mathfrak{A}^* \models \hat{\varphi}(\bar{P})$ iff $\mathfrak{A}^* \in L(\mathcal{A})$ iff $\mathfrak{A}^* \models \varphi(\bar{P})$.

This time, we embed the game $\mathcal{G}(\mathcal{A},\mathfrak{A})$ directly into $\mathfrak{A}^*$. We consider the following variant of $\mathcal{G}(\mathcal{A},\mathfrak{A})$ which obviously is equivalent.

(a) The sets of vertices are $V_0 := Q \times A^*$ and $V_1 := \bigcup_{w\in A^*} \mathscr{P}(Q \times wA)$.
(b) The initial position is (q_0,ε).
(c) If $\langle\!\langle \delta(q,\emptyset)\rangle\!\rangle_{\mathfrak{A}}(w) = \bigvee_i \bigwedge \Phi_i$ for $w \in A^*$ then the successors of a node $(q,a) \in V_0$ are the sets $\{\, (p,wa) \mid (p,a) \in \Phi_i \,\}$ for all i. Nodes $\Phi \in V_1$ have their elements $(q,w) \in \Phi$ as successors.
(d) A play $(q_0,w_0),\Phi_0,(q_1,w_1),\Phi_1,\ldots$ is winning if the sequence $(q_i)_{i<\omega}$ satisfies the parity condition Ω.

In the same was as above we can encode this game as a structure $\mathfrak{G}(\mathcal{A},\mathfrak{A})$ such that $\mathfrak{G}(\mathcal{A},\mathfrak{A})|_{V_0}$ can be interpreted in $\mathfrak{A}^*$. Again, elements of V_1 are encoded as subsets of V_0. Note that, for each such subset Φ, the set $\{\, wa \mid (p,wa) \in \Phi$ for some $p \,\}$ is first-order definable with the parameter w.

Let $\varepsilon_0(x,Y)$ be the formula defining the relation $E \cap V_0 \times V_1$ and $\varepsilon_1(X,y)$ the one defining $E \cap V_1 \times V_0$.

It remains to evaluate the formula $\mathrm{LFP}_{Z_1,x} \cdots \mathrm{GFP}_{Z_n,x} \bigvee_{i\leq n} \eta_i(x,\bar{Z})$ with which defines the winning set in the original game graph $\mathcal{G}'(\bar{A},\mathfrak{A})$. Again, we can replace η_i by

$$\eta''_i := S_i x \wedge V_0 x \wedge \exists Y\big(Y \subseteq V_0 \wedge \varepsilon_0(x,Y) \wedge \forall z(\varepsilon_1(Y,z) \rightarrow Z_i z)\big)$$

Note that this formula is local to one copy of $\mathfrak{A}$ as it only speaks about the vertex $x = (q, w)$ and elements of the (definable) set $Q \times wA$. Consequently, there is some LFP-formula $\chi_i(x, \bar{Z})$ equivalent to η_i'', and we can write the winning formula as $\hat{\varphi} := \mathrm{LFP}_{Z_1,x} \cdots \mathrm{GFP}_{Z_n,x} \bigvee_{i \leq n} \chi_i$. □

Remark 6.2. The preceding theorem and its proof also hold for the logics MSO($\exists^\omega$) and M-LFP($\exists^\omega$), and for MSO + C and M-LFP + C.

References

1. D. Berwanger and A. Blumensath, *The monadic theory of tree-like structures*, in Automata, Logic, and Infinite Games, E. Gr"adel, W. Thomas, and T. Wilke, eds., no. 2500 in LNCS, Springer Verlag, 2003, pp. 285–301.
2. H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer, 2nd ed., 1999.
3. E. A. Emerson and C. S. Jutla, *Tree automata, mu-calculus and determinacy (extended abstract)*, in Proc. of the 32nd Annual Symp. on Foundations of Computer Science, FoCS, 1991, pp. 368–377.
4. E. Grädel, C. Hirsch, and M. Otto, *Back and forth between guarded and modal logics*, in Proc. 15th IEEE Symp. on Logic in Computer Science, 2000, pp. 217–228.
5. D. Kuske and M. Lohrey, *Decidable theories of graphs, factorized unfoldings and cayley-graphs*, Tech. Rep. 37, University of Leicester, 2002.
6. A. W. Mostowski, *Games with forbidden positions*, Tech. Rep. 78, Uniwersytet Gdański, Poland, Instytut Matematyki, 1991.
7. M. O. Rabin, *Decidability of second-order theories and automata on infinite trees*, Trans. Amer. Math. Soc., 141 (1969), pp. 1–35.
8. A. L. Semenov, *Decidability of monadic theories*, LNCS, 176 (1984), pp. 162–175.
9. S. Shelah, *The monadic second order theory of order*, Annals of Mathematics, 102 (1975), pp. 379–419.
10. J. Stupp, *The lattice model is recursive in the original model.* manuscript, The Hebrew University, Jerusalem, 1975.
11. I. Walukiewicz, *Monadic second-order logic on tree-like structures*, Theoretical Computer Science, 275 (2002), pp. 311–346.