Automata for Enriched Trees and Applications

Achim Blumensath

21st February 2025

We study trees where each successor set is equipped with some additional structure. We introduce a family of automaton models for such trees and prove their equivalence to certain fixed-point logics. As a consequence we obtain characterisations of various variants of monadic second-order logic in terms of automata and fixed-point logics. Finally, we use our machinery to give a simplified proof of the Theorem of Muchnik and we derive several variants of this theorem for other logics.

Keywords. coalgebra, automata, fixed-point logics, Muchnik iteration

1 Introduction

Translations between logical formulae and automata provide a versatile tool both for applications (such as model-checking algorithms) and for purely theoretical considerations (like, e.g., studying questions of expressive power). To give a concrete example: two of the strongest decidability results in logic are proved by automata-theoretic methods. The first one is the Theorem of Muchnik [16] which states that a certain operation preserves the decidability of monadic second-order theories; the second one is a Theorem of Puppis [12] on the decidability of certain trees with non-regular labelling.

In most settings, the logic corresponding to automata turns out to be monadic second-order logic, while many weaker logics can be characterised by automata

of special types. But a closer look reveals that it would be more appropriate to say that the logics corresponding to automata are certain fixed-point logics. The fact that these logics have the same expressive power as monadic second-order logic seems to be coincidental.

A systematic study of the correspondence between tree automata and fixed-point logics can be found in the thesis of Carreiro [4] building on previous work of Walukiewicz [7, 16] and Venema [15, 9, 8]. In this article, we improve upon those results in two ways. First of all, the development in [4] is not uniform: each equivalence between a logic and an automaton model is proved on a case-by-case basis using ad-hoc methods. While all definitions and proofs share a similar structure, the details of each case are different. We were able to abstract away these details and to derive a general proof that allows us to derive all results of [4] in a uniform way.

The second improvement concerns the classes of structures supported by our framework. [4] only considers trees (possibly infinite). For some applications, this is not sufficient since they require working with trees where the successors of each vertex are equipped with some additional structure. This can simply be an left-to-right ordering of the successors, or it can be more substantial like, e.g., a probability measure. For the above mentioned Theorem of Muchnik, for instance, one works with trees expanded by an arbitrary number of additional relations, with the restriction that these relations only relate siblings. Our translation generalises in a straightforward way to trees enriched with such an additional structure. As a consequence we are able to give characterisations of a variety of logics via suitable automaton models that hold not only for trees, but also for such 'enriched' trees.

Finally, let me mention that, while most of the constructions and arguments used below can be considered standard, it was surprisingly tricky to get the details right.

The overview of this article is as follows. We set up our framework in Sections 2 and 3. In the first of these sections, we introduce the class of enriched trees we are working with, while the second one introduces a general notion of a logic and a way to extend such logics by fixed-point operators. Sections 4, 5, and 6 contain our translation between logics and automata. Section 4 recalls some results about parity games, which are needed in Section 5 to prove the correctness of our translation. Section 6 contains automata constructions corresponding to various set quantifiers. In Section 7, we use the results from Section 5 to give automata-theoretic characterisations of many variants of monadic second-order logic. Finally, Section 8 contains an application consisting of a new, much simplified proof of the Theorem of Muchnik and of some variants of this theorem

for other logics, some of them already known, some new.

2 Enriched Trees

In order to model transition systems with arbitrary enrichments, it is convenient to use a coalgebraic approach, similar to that one in [15]. That is, we choose a functor $\mathbb{S}$ that returns the set of all possible enriched successor structures and then we model a transition system as a function $\mathrm{suc} : S \to \mathbb{S}S$ mapping each state $s \in S$ to the structure of its successors.

Let us start by defining the class of functors we will use. We will restrict ourselves to so called *polynomial functors,* as these have particularly nice properties and they cover all the applications we have in mind. A polynomial functor maps a set X to a set of X-labelled objects. The formal definition is as follows.

Definition 2.1. A functor $\mathbb{S} : \mathsf{Set} \to \mathsf{Set}$ is *polynomial* if it is of the form

$$\mathbb{S}X = \sum_{i \in I} X^{D_i}, \quad \text{for fixed sets } I \text{ and } D_i,\ i \in I.$$

Such a functor maps a function $f : X \to Y$ to the function $\mathbb{S}f : \mathbb{S}X \to \mathbb{S}Y$ applying f to each label. Formally,

$$\mathbb{F}f(s) := f \circ s\,, \quad \text{for } s : D_i \to X\,.$$

Thus, elements of $\mathbb{S}X$ are functions $s : D_i \to X$, for some i. We denote the domain of such a function by $\mathrm{dom}(s) := D_i$.

(b) Let $\mathbb{S}$ be a polynomial functor. Two elements $s, t \in \mathbb{S}X$ have *the same shape,* in symbols $s \simeq_{\mathrm{sh}} t$, if s, t correspond to the same index $i \in I$, i.e, if $s : D_i \to X$ and $t : D_i \to X$, for some $i \in I$. ⌟

Examples. (a) The functor $\mathbb{S}X := X^*$ returning the set of finite words over the alphabet X is polynomial since it can be written as

$$\mathbb{S}X = \sum_{n<\omega} X^n.$$

(b) Fixing a signature Σ, the functor $\mathbb{S}$ mapping a set X to the set of all countable X-labelled Σ-structures is polynomial since

$$\mathbb{S}X = \sum_{\mathfrak{A}} X^A,$$

where the sum ranges over all countable Σ-structures $\mathfrak{A}$ and A denotes the universe of the structure $\mathfrak{A}$.

(c) Let $\mathcal{D}$ be the set of all pairs $\langle A, \mu \rangle$ where A is a finite set and μ is a probability measure on A. There exists a polynomial functor

$$\mathbb{S}X := \sum_{\langle A,\mu\rangle \in \mathcal{D}} X^A$$

mapping a set X to the set of all X-labellings of some $\langle A, \mu \rangle \in \mathcal{D}$. ⌟

We can now define transition systems as $\mathbb{S}$-coalgebras for some polynomial functor $\mathbb{S}$.

Definition 2.2. Let $\mathbb{S} : \mathsf{Set} \to \mathsf{Set}$ be a polynomial functor and $\Sigma \in \mathsf{Set}$ an alphabet.

(a) A *Σ-labelled $\mathbb{S}$-enriched transition system* is a structure of the form

$$\mathfrak{S} = \langle S, \mathrm{suc}, \lambda, v_0 \rangle$$

where S is the set of *states*, $v_0 \in S$ is the *initial state*, $\lambda : S \to \Sigma$ is a *labelling* of the states, and $\mathrm{suc} : S \to \mathbb{S}S$ is a function assigning a *successor structure* $\mathrm{suc}(v) \in \mathbb{S}S$ to each state $v \in S$. We call the elements of $\mathrm{dom}(\mathrm{suc}(v))$ *directions* at the state v.

As for polynomial functors, we will use the notation $\mathrm{dom}(s)$ to denote the set of states of a transition system s and we represent s by the labelling function $s : \mathrm{dom}(s) \to \Sigma$, leaving the successor function $\mathrm{suc} : \mathrm{dom}(s) \to \mathbb{S}\,\mathrm{dom}(s)$ implicit.

(b) A *homomorphism* $\varphi : s \to t$ between transition systems s and t is a function $\varphi : \mathrm{dom}(s) \to \mathrm{dom}(t)$ satisfying

$$t(\varphi(v)) = s(v) \quad \text{and} \quad \mathrm{suc}(\varphi(v)) = \mathbb{S}\varphi(\mathrm{suc}(v)), \quad \text{for all } v \in \mathrm{dom}(s).$$

(c) For a transition system $\mathfrak{S}$ and a state $v \in S$, we denote by $\mathfrak{S}, v$ the transition system obtained from $\mathfrak{S}$ by changing the initial state to v. ⌟

Examples. (a) For ordinary 'non-enriched' transition systems, we can use a functor of the form

$$\mathbb{S}X := \sum_{\kappa < \lambda} X^\kappa,$$

where λ is some fixed cardinal and the sum ranges over all cardinals κ less than λ.

(c) For Markov chains, we use a polynomial functor $\mathbb{S}$ where the index set I is a set of probability measures.

(d) In Section 8 below we will define an operation $\mathfrak{A}^*$ called a Muchnik iteration. This operation constructs an infinite $\mathbb{S}$-enriched tree where $\mathbb{S}$ returns a set of Σ-structures. ⌟

Remark. (a) We could model transition systems as coalgebras for the combined functor $\mathbb{S} \times \Sigma$. But for our purposes it is more convenient to separate the successors and the labelling.

(b) One shortcoming of the notion of a polynomial functor is the fact that the index set I – which corresponds to the class of objects we want to label – is a set and not a proper class. Therefore we will frequently have to introduce arbitrary cut-offs for the class of successor structures. For instance, instead of the class of all trees, we can only use classes of finitely branching ones, or countably branching ones. While inconvenient, this is usually not a problem since we can always choose the cut-off point large enough to cover the transition system under consideration. ⌟

Trees can now be defined as unravellings of transition systems.

Definition 2.3. Let s be an $\mathbb{S}$-enriched transition system.

(a) A (finite) *path* in s is a finite sequence of the form

$$v_0, d_0, v_1, d_1, \dots, v_{n-1}, d_{n-1}, v_n$$

where $v_0, \dots, v_n \in \mathrm{dom}(s)$ are states, each $d_i \in \mathrm{dom}(\mathrm{suc}(v_i))$ is a direction at v_i, and $v_{i+1} = \mathrm{suc}(v_i)(d_i)$.

(b) The *unravelling* of s is the transition system $\mathrm{un}(s)$ whose domain consists of all finite paths in s starting at the initial state of s, the labelling assigns to each path $\langle v_0, \dots, v_n \rangle$ the label $s(v_n)$ of the last state, and the successor function is defined by $\mathrm{suc}(\langle v_0, \dots, v_n \rangle) \simeq_{\mathrm{sh}} \mathrm{suc}(v_n)$ and

$$\mathrm{suc}(\langle v_0, \dots, v_n \rangle)(d) := \langle v_0, \dots, v_n, d, \mathrm{suc}(v_n)(d) \rangle .$$

(c) s is an $\mathbb{S}$-*enriched tree* if it is isomorphic to its unravelling $\mathrm{un}(s)$. We denote the *root* of s by $\langle\rangle$ (the empty sequence), and we write $\mathbb{T}_{\mathbb{S}}\Sigma$ for the set of all Σ-labelled $\mathbb{S}$-enriched trees. ⌟

3 Logics

Since we are dealing with many different logics, we will use an abstract notion of a logic introduced in [2].

Definition 3.1. (a) A *logic* is a triple $\langle L, \mathcal{M}, \models \rangle$ where L is a set of *formulae,* $\mathcal{M}$ a set of *models,* and $\models \subseteq \mathcal{M} \times L$ a *satisfaction relation.* Frequently, we denote a logic simply by its set L of formulae, leaving $\mathcal{M}$ and $\models$ implicit.

(b) A *morphism of logics* $\langle \lambda, \mu \rangle : \langle L, \mathcal{M}, \models \rangle \to \langle L', \mathcal{M}', \models' \rangle$ is a pair of functions $\lambda : L \to L'$ and $\mu : \mathcal{M}' \to \mathcal{M}$ satisfying

$$M' \models \lambda(\varphi) \quad \text{iff} \quad \mu(M') \models \varphi\,, \quad \text{for all } \varphi \in L \text{ and } M' \in \mathcal{M}'\,.$$

Usually, we denote both components of a morphism with the same identifier. ⌟

Example. Let Σ be a fixed signature and let $\mathcal{C}_\Sigma$ be the class of all countable Σ-structures. Then $\langle \mathrm{FO}[\Sigma], \mathcal{C}_\Sigma, \models \rangle$ forms a logic where $\mathrm{FO}[\Sigma]$ denotes the set of all first-order formulae over the signature Σ. ⌟

As the preceding example shows, our notion of a logic requires us to fix a signature. Frequently this is inconvenient since we would like to consider several different signatures at once. Therefore we introduce families of logics parametrised by their signature.

Definition 3.2. (a) A *family of logics* is a functor L from the category of finite sets to the category of logics.

(b) A family of logics L is *over* a functor $\mathbb{S}$ if, for every set Σ, the class of models of $L[\Sigma]$ is equal to $\mathbb{S}\Sigma$ and, for every function $f : \Sigma \to \Gamma$, the morphism $L[f] : L[\Sigma] \to L[\Gamma]$ is of the form $L[f] = \langle \lambda, \mu \rangle$ with $\mu = \mathbb{S}f$. ⌟

Examples. For transition systems, we can use the following families of logics.

(a) For each set Q, we obtain a logic $\langle \mathrm{MSO}[Q], \mathcal{M}_Q, \models \rangle$ where $\mathrm{MSO}[Q]$ is the set of monadic second-order formulae over the signature $\{E\} + \{\, P_q \mid q \in Q \,\}$ (without free variables), and $\mathcal{M}_Q$ is the set of all transition systems of the form $\mathfrak{S} = \langle S, E, (P_q)_{q \in Q} \rangle$ where E is the edge relation and the P_q are unary predicates.

Given a function $f : Q \to Q'$, we obtain a morphism $\mathrm{MSO}[f] : \mathrm{MSO}[Q] \to \mathrm{MSO}[Q']$ mapping a formula $\varphi \in \mathrm{MSO}[Q]$ to the formula $\mathrm{MSO}[f](\varphi)$ obtained from φ by renaming every predicate P_q to $P_{f(q)}$. The corresponding function on transition systems maps a system $\mathfrak{S} = \langle S, E, (P_{q'})_{q' \in Q'} \rangle$ to the system

$$\mathrm{MSO}[f](\mathfrak{S}) := \langle S, E, (P_{f(q)})_{q \in Q} \rangle\,.$$

(b) We obtain similar families $\mathrm{FO}[Q]$, $\mathrm{WMSO}[Q]$, $\mu\mathrm{ML}[Q]$, for *first-order logic, weak monadic second-order logic,* and *the modal μ-calculus.* (We will define these logics formally in Section 7 below. Note that our notation $\mu\mathrm{ML}$ is slightly inconsistent with our notation for the fixed-point logics μL defined below.) ⌟

Below we will mostly work with families of logics over functors of the form $\mathbb{S} \circ \wp$ where $\mathbb{S}$ is a polynomial functor and $\wp$ the (covariant) power-set functor. This corresponds to models whose elements are labelled by sets of symbols. In particular, logics whose models are Σ-structures are of this form since each element in a structure can belong to several predicates.

Definition 3.3. Let L be a family of logics over $\mathbb{S} \circ \wp$ where $\mathbb{S}$ is some polynomial functor.

(a) A formula $\varphi \in L[X]$ is *monotone* in a symbol $x \in X$ if, given two models $s \simeq_{\mathrm{sh}} s'$ satisfying

$$s'(v) = s(v) \quad \text{or} \quad s'(v) = s(v) \cup \{x\}\,, \quad \text{for all } v \in \mathrm{dom}(s)\,,$$

we have

$$s \vDash \varphi \quad \text{implies} \quad s' \vDash \varphi\,.$$

(a) A formula $\varphi \in L[X]$ is *antitone* in x if, given two models $s \simeq_{\mathrm{sh}} s'$ satisfying

$$s'(v) = s(v) \quad \text{or} \quad s'(v) = s(v) \cup \{x\}\,, \quad \text{for all } v \in \mathrm{dom}(s)\,,$$

we have

$$s' \vDash \varphi \quad \text{implies} \quad s \vDash \varphi\,.$$ ⌟

Example. A first-order formula φ over the signature Σ is monotone in a relation $R \in \Sigma$ if, and only if, φ is equivalent to some formula φ' where every occurrence of R in φ' is under an even number of negation signs. ⌟

For some of the applications below, we will have to put additional restrictions on certain symbols in a formula. For instance, we might require some of the symbols to occur only positively in the formula and others only negatively. For this reason, we add two more parameters to our logics, leading to logics $L[X, U, V]$ parametrised by three sets: the set X of all symbols, the set U of symbols with the additional restriction, and the set V of symbols with the opposite/dual restriction.

Definition 3.4. Let $\mathcal{C}$ be the category of all triples $\langle X, U, V\rangle$ of sets with $U, V \subseteq X$ where the morphisms $f : \langle X, U, V\rangle \to \langle X', U', V'\rangle$ are functions $f : X \to X'$ satisfying

$$x \notin U \Rightarrow f(x) \notin U' \quad \text{and} \quad x \notin V \Rightarrow f(x) \notin V' .$$

A family of logics *with polarities* is a functor L from $\mathcal{C}$ to the category of logics. We say that such a family is *over* a functor $\mathbb{S}$ if, for every triple $\langle X, U, V\rangle$, the class of models of $L[X, U, V]$ is equal to $\mathbb{S}X$ and, for every morphism $f : \langle X, U, V\rangle \to \langle X', U', V'\rangle$, the morphism $L[f] : L[X] \to L[X']$ is of the form $L[f] = \langle \lambda, \mu\rangle$ with $\mu = \mathbb{S}f$. ⌟

Example. For first-order logic, we obtain a family of logics with polarities where $\mathrm{FO}^+[X, U, V]$ contains all formulae $\varphi \in \mathrm{FO}[X]$ such that every predicate in U occurs only positively in φ and every predicate in V only negatively. ⌟

Besides monotonicity, there are two other restrictions we are interested in.

Definition 3.5. Let L be a family of logics over $\mathbb{S} \circ \wp$.

(a) A formula $\varphi \in L[X]$ is *jointly discrete* in a set of symbols $C \subseteq X$ if it is monotone in every $x \in C$ and if $s \vDash \varphi$ implies $s' \vDash \varphi$, for some s' that is obtained from s by removing all but one occurrence of the labels in C. Formally, $s' \simeq_{\mathrm{sh}} s$ and that there are $v_{\mathrm{o}} \in \mathrm{dom}(s)$ and $c \in C$ such that

$$s'(v_{\mathrm{o}}) = s(v_{\mathrm{o}}) \setminus (C \setminus \{c\}) \quad \text{and} \quad s'(v) = s(v) \setminus C , \quad \text{for all } v \neq v_{\mathrm{o}} .$$

Similarly, φ is *jointly co-discrete* in C if it is monotone in every $x \in C$ and if, given a model s,

$$\forall v_{\mathrm{o}} \forall c [s_{v_{\mathrm{o}},c} \vDash \varphi] \quad \text{implies} \quad s \vDash \varphi ,$$

where $s_{v_{\mathrm{o}},c}$ is defined by $s_{v_{\mathrm{o}},c} \simeq_{\mathrm{sh}} s$ and

$$s_{v_{\mathrm{o}},c}(v_{\mathrm{o}}) = s(v_{\mathrm{o}}) \cup (C \setminus \{c\}) \quad \text{and} \quad s_{v_{\mathrm{o}},c}(v) = s(v) \cup C , \quad \text{for all } v \neq v_{\mathrm{o}} .$$

By $L_{\mathrm{d}}[X, U, V]$ we denote the set of all $L[X]$-formulae that are jointly discrete in U and jointly co-discrete in V.

(b) A formula $\varphi \in L[X]$ is *continuous* in a symbol $x \in X$ if it is monotone in x and if $s \vDash \varphi$ implies $s' \vDash \varphi$, for some s' obtained from s by removing all but finitely

many occurrences of the label x. Formally, $s' \simeq_{\mathrm{sh}} s$ and there is some finite set $P \subseteq \mathrm{dom}(s)$ such that

$$s'(v) = \begin{cases} s(v) & \text{if } v \in P, \\ s(v) \setminus \{x\} & \text{if } v \notin P. \end{cases}$$

Similarly, φ is *co-continuous* in x if it is monotone in x and if, given a model s,

$$\forall P[s_P \models \varphi] \quad \text{implies} \quad s \models \varphi,$$

where the quantifier ranges over all finite sets $P \subseteq \mathrm{dom}(s)$ and the model s_P is defined by $s_P \simeq_{\mathrm{sh}} P$ and

$$s_P(v) = \begin{cases} s(v) & \text{if } v \in P, \\ s(v) \cup \{x\} & \text{if } v \notin P. \end{cases}$$

By $L_{\mathrm{c}}[X, U, V]$ we denote the set of all $L[X]$-formulae that are continuous in every $x \in U$ and co-continuous in every $x \in V$. ⌟

Example. (a) The FO-formula

$$\varphi := \forall x Px \wedge \exists y Qy \wedge \exists y Ry$$

is jointly co-discrete in $\{P\}$, jointly discrete both in $\{Q\}$ and in $\{R\}$, but not jointly discrete in $\{Q, R\}$.

(b) The formula

$$\psi := \exists^{\infty} x Px \wedge \exists y Qy$$

is continuous in Q, but not in P. ⌟

We can define the usual logical connectives in our abstract setting.

Definition 3.6. Let L be a family of logics with polarities over $\mathbb{S} \circ \wp$.

(a) A formula $\psi \in L[X, U, V]$ is a *dual* of $\varphi \in L[X, V, U]$ if

$$s^{\mathrm{op}} \models \psi \quad \text{iff} \quad s \not\models \varphi,$$

where $s^{\mathrm{op}}(v) := X \setminus s(v)$. In this case we write $\varphi^{\mathrm{op}} := \psi$.

(b) A formula $\vartheta \in L[X, U, V]$ is the *conjunction* of two formulae $\varphi, \psi \in L[X, U, V]$ if

$$s \vDash \vartheta \quad \text{iff} \quad s \vDash \varphi \text{ and } s \vDash \psi\,.$$

In this case we write $\varphi \wedge \psi := \vartheta$.

(c) A formula $\vartheta \in L[X, U, V]$ is the *disjunction* of two formulae $\varphi, \psi \in L[X, U, V]$ if

$$s \vDash \vartheta \quad \text{iff} \quad s \vDash \varphi \text{ or } s \vDash \psi\,.$$

In this case we write $\varphi \vee \psi := \vartheta$. ⌟

Remark. Note that the formulae $\varphi \vee \psi$, $\varphi \wedge \psi$, and φ^{op} are well-defined up to logical equivalence (if they exist). Also note that the symbols in φ^{op} have their polarity reversed. ⌟

Examples. For the first-order logic, we have

$$\begin{aligned}\big[Px \vee \exists y(Exy \wedge Qy)\big]^{\mathrm{op}} &= Px \wedge \forall y(Exy \to Qy)\,,\\ \big[\exists x[Px \wedge \forall y[y \neq x \to Qy]\big]^{\mathrm{op}} &= \forall x[Px \vee \exists y[y \neq x \wedge Qy]]\,.\end{aligned}$$ ⌟

The following observations follow immediately from the definitions.

Lemma 3.7. *Let L be a family of logics for $\mathbb{S} \circ \wp$.*

(a) *If $\varphi \in L$ is monotone in x, so is φ^{op}.*

(b) *$\varphi \in L$ is jointly discrete in C if, and only if, φ^{op} is jointly co-discrete in C.*

(c) *$\varphi \in L$ is continuous in x if, and only if, φ^{op} is co-continuous in x.*

Next, let us introduce a variant μL of the modal μ-calculus where the modal operators are defined by L-formulae.

Definition 3.8. Let L be a family of logics with polarities over $\mathbb{S} \circ \wp$.

(a) We denote by μL the following variant of the modal μ-calculus for $\mathbb{S}$-enriched transition systems. Given a set Σ of labels and two disjoint sets X, Y of *fixed-point variables,* we denote by $\mu L[\Sigma; X, Y]$ the smallest set of formulae satisfying the following conditions.

- $a \in \mu L[\Sigma; X, Y]$, for every $a \in \Sigma$.
- $x \in \mu L[\Sigma; X, Y]$, for every $x \in X \cup Y$.

- $\varphi, \psi \in \mu L[\Sigma; X, Y]$ implies $\varphi \wedge \psi, \varphi \vee \psi \in L[\Sigma; X, Y]$ and $\neg\varphi \in \mu L[\Sigma; Y, X]$.
- Let $\Theta \subseteq \mu L[\Sigma; X, Y]$ be a finite set of formulae and let Θ_+ be the set of all $\vartheta \in \Theta$ containing a symbol from X and Θ_- the set of all $\vartheta \in \Theta$ containing a symbol from Y. For every $\varphi \in L[\Theta, \Theta_+, \Theta_-]$, we have $\bigcirc\varphi \in \mu L[\Sigma; X, Y]$.
- If $\varphi \in \mu L[\Sigma; X + \{x\}, Y]$ is monotone in x, then $\mu x.\varphi \in L[\Sigma; X, Y]$. Similarly, if $\varphi \in \mu L[\Sigma; X, Y + \{y\}]$ is monotone in y, then $\nu y.\varphi \in \mu L[\Sigma; X, Y]$.

We will always tacitly assume that fixed-point variables used by distinct fixed-point operators in a formula are distinct.

The semantics is defined as follows. For a given $\mathbb{S}$-enriched transition system $\mathfrak{S} = \langle S, \mathrm{suc}, \lambda \rangle$, an μL-formula $\varphi(x_0, \ldots, x_{n-1}) \in \mu L[\Sigma; X, Y]$, and values $P_0, \ldots, P_{n-1} \subseteq S$ for the free fixed-point variables $x_0, \ldots, x_{n-1} \in X \cup Y$, we define the set $[\![\varphi]\!]_{\bar{P}} \subseteq S$ of states satisfying φ inductively as follows.

$$
\begin{aligned}
[\![a]\!]_{\bar{P}} &:= \lambda^{-1}(a)\,, \\
[\![x_i]\!]_{\bar{P}} &:= P_i\,, \\
[\![\varphi \wedge \psi]\!]_{\bar{P}} &:= [\![\varphi]\!]_{\bar{P}} \cap [\![\psi]\!]_{\bar{P}}\,, \\
[\![\varphi \vee \psi]\!]_{\bar{P}} &:= [\![\varphi]\!]_{\bar{P}} \cup [\![\psi]\!]_{\bar{P}}\,, \\
[\![\neg\varphi]\!]_{\bar{P}} &:= S \setminus [\![\varphi]\!]_{\bar{P}}\,, \\
[\![\bigcirc\psi]\!]_{\bar{P}} &:= \left\{ v \in S \;\middle|\; \mathbb{S}f(\mathrm{suc}(v)) \vDash \psi \right\} \\
&\qquad \text{where } f : S \to \wp(\Theta) \text{ maps } v \in S \text{ to } \{\, \vartheta \in \Theta \mid v \in [\![\vartheta]\!]_{\bar{P}} \,\}\,, \\
[\![\mu x.\psi]\!]_{\bar{P}} &\text{ is the least fixed-point of the function} \\
&\qquad F_\psi : \wp(S) \to \wp(S) : Q \mapsto [\![\psi]\!]_{\bar{P}Q}\,, \\
[\![\nu x.\psi]\!]_{\bar{P}} &\text{ is the greatest fixed-point of } F_\psi\,.
\end{aligned}
$$

Finally, we set $\mu L[\Sigma] := \mu L[\Sigma; \varnothing, \varnothing]$ and

$$\mathfrak{S}, v \vDash \varphi \quad :\text{iff} \quad v \in [\![\varphi]\!]_{\langle\rangle}\,.$$

(b) A formula $\varphi \in \mu L$ is *pure* if, for every subformula of the form $\mu x.\psi$ or $\nu x.\psi$, we have $\psi \in \mu L[\Sigma; X, \varnothing] \cup L[\Sigma; \varnothing, X]$, for some X. We denote the corresponding fragment of μL by $\mu_{\mathrm{p}} L$.

(c) A formula $\varphi \in \mu L$ is *alternation-free* if

- for every subformula of the form $\mu x.\psi$, we have $\psi \in \mu L[\Sigma; X, \varnothing]$, for some X, and
- for every subformula of the form $\nu x.\psi$, we have $\psi \in \mu L[\Sigma; \varnothing, X]$, for some X.

We denote the corresponding fragment of μL by $\mu_{\mathrm{af}} L$. ⌟

Examples. Let E_1 be the logic whose formulae are boolean combinations of statements of the form

$$\mathsf{E}C := \text{‘There exists a position whose set of labels is equal to } C\text{.’},$$

for $C \subseteq \Sigma$. Similarly, let E_ω be the logic consisting of boolean combinations of statements of the form

$$\mathsf{E}_k C := \text{‘There exist at least } k \text{ positions whose set of labels is equal to } C\text{.’},$$

for $C \subseteq \Sigma$ and $k < \omega$.

(a) The fixed-point extension $\mu\mathrm{E}_1$ coincides with the standard modal μ-calculus. Using $L := \mathrm{E}_\omega$, we obtain a graded version $\mu\mathrm{E}_\omega$ of the μ-calculus.

(b) The $\mu_{\mathrm{af}}\mathrm{E}_1$-formula

$$\varphi := \mu x\big[a \vee \bigcirc[\mathsf{E}\{x\}]\big]$$

checks whether there is a reachable vertex labelled by a.

(c) The $\mu_{\mathrm{p}}\mathrm{E}_1$-formula

$$\varphi := \nu x.\mu y\big[\big(a \wedge \bigcirc[\mathsf{E}\{x\}]\big) \vee \bigcirc[\mathsf{E}\{y\}]\big]$$

checks for the existence of a path with infinitely many letters a.

(c) The $\mu_{\mathrm{p}}\mathrm{E}_\omega$-formula

$$\varphi := \nu x.\mu y\big[\bigcirc[\mathsf{E}_2\{x\}] \vee \bigcirc[\mathsf{E}_1\{y\}]\big]$$

checks for an embedding of the infinite binary tree.

(e) To state that a given tree has exactly one vertex labelled a we can use the $\mu\mathrm{FO}_{\mathrm{c}}$-formula

$$\mu x.\big[\big(a \wedge \bigcirc[\forall u.P_\vartheta(u)]\big) \wedge \big(\neg a \wedge \bigcirc\big[\exists u[P_x(u) \wedge \forall v(v \neq u \to P_\vartheta(v))]\big]\big)\big],$$

with $\vartheta := \nu y.[\neg a \wedge \bigcirc[\forall u.P_y(u)]]$. ⌟

Finally, let us introduce a normal form for μL-formulae that will come in handy below in our translation of formulae into automata.

Definition 3.9. Let $\varphi \in \mu L[\Sigma; X, Y]$.

(a) φ is in *negation normal form* if the only negations in φ appear in a subformula of the form $\neg a$ with $a \in \Sigma$.

(b) φ is *guarded* if, for each subformula of the form $\sigma x.\psi$ with $\sigma \in \{\mu, \nu\}$, every occurrence of the variable x in ψ is inside a subformula $\heartsuit\vartheta$ starting with a modal operator. ⌟

Let us start with an observation that is useful to convert a formula into negation normal form.

Remark. (a) Fixed-point formulae satisfy the usual negation law.

$$\neg\nu x.\varphi \equiv \mu x.\neg\varphi[x \mapsto \neg x]\,.$$

(b) For modal operators, we obtain the relation

$$\neg\heartsuit\psi \equiv \heartsuit\mathbb{S}\wp c(\psi^{\mathrm{op}})\,,$$

where $c : \mu L \to \mu L$ is the function exchanging each label ϑ by $\neg\vartheta$. For the proof, note that we have

$$s \vDash \mathbb{S}c(\psi^{\mathrm{op}}) \quad \text{iff} \quad \mathbb{S}c(s) \vDash \psi^{\mathrm{op}} \quad \text{iff} \quad s \nvDash \psi\,.$$

Hence,

$$\begin{aligned}\llbracket\neg\heartsuit\psi\rrbracket_{\bar{P}} &= S \smallsetminus \llbracket\heartsuit\psi\rrbracket_{\bar{P}} \\ &= S \smallsetminus \{\, v \in S \mid \mathbb{S}f(\mathrm{suc}(v)) \vDash \psi \,\} \\ &= \{\, v \in S \mid \mathbb{S}f(\mathrm{suc}(v)) \nvDash \psi \,\} \\ &= \{\, v \in S \mid \mathbb{S}f(\mathrm{suc}(v)) \vDash \mathbb{S}c(\psi^{\mathrm{op}}) \,\} = \llbracket\heartsuit\mathbb{S}c(\psi^{\mathrm{op}})\rrbracket_{\bar{P}}\,.\end{aligned}$$ ⌟

Lemma 3.10. *Let L be a logic over $\mathbb{S} \circ \wp$ that is closed under duals.*

(a) *μL is closed under duals.*

(b) *Every μL-formula is equivalent to one that is guarded and in negation normal form.*

Proof. (a) Let $\varphi \in \mu L$. By (b), we can assume that φ is in negation normal form. Then φ^{op} is the formula obtained from φ by

- replacing every disjunction by a conjunction and vice versa,
- replacing every μ-operator by a ν and vice versa.

(b) To transform a given fromula φ into negation normal form we can use the two laws from the above remark. It therefore remains to show how to make a formula in negation normal form guarded. For each subformula $\mu x.\psi$, let ψ' be the formula obtained from ψ by replacing every unguarded occurrence of x by false. Then we have

$$[\![\psi']\!]_{\bar{P}Q} \subseteq [\![\psi]\!]_{\bar{P}Q} \subseteq Q \cup [\![\psi']\!]_{\bar{P}Q} ,$$

which implies that

$$[\![\mu x.\psi']\!]_{\bar{P}} \subseteq [\![\mu x.\psi]\!]_{\bar{P}} \subseteq [\![\mu x.(x \vee \psi')]\!]_{\bar{P}} = [\![\mu x.\psi']\!]_{\bar{P}} .$$

Hence, $\mu x.\psi$ is equivalent to $\mu x.\psi'$. For greatest fixed points $\nu x.\psi$, we can similarly replace all unguarded x by true instead. □

4 Parity Games

Let us recall some material about parity games that will be used below in the translation of automata into formulae.

Definition 4.1. (a) A *parity game* is a games played by two players (Player $\Diamond$ and Player $\Box$) who move a token along the edges of a directed graph. The game starts in a fixed vertex of the graph and in each turn one of the players chooses an outgoing edge along which to move the token. To determine the moving player, we assign a player to each vertex of this graph. The player assigned to the current vertex chooses the outgoing edge.

The game ends if one of the player cannot make a move (because there are no outgoing edges), in which case that player looses. Otherwise, the choices of the players determine an infinite path through the game, called a *play* of the game. To determine the winner of such an infinite play, we label each vertex by a number, its *priority,* and the least priority seen infinitely often during the play determines the winner: Player $\Diamond$ wins if it is even; otherwise Player $\Box$ wins.

Formally, we represent a parity game as a structure of the form $\mathcal{G} = \langle V_\Diamond, V_\Box, E, \Omega \rangle$ where $V := V_\Diamond + V_\Box$ is the set of *positions,* $V_\Diamond$ are the positions for Player $\Diamond$, $V_\Box$ are the positions for Player $\Box$, $E \subseteq V \times V$ is the *edge relation,* and $\Omega : V \to \omega$ the *priority function.* Given an infinite play $(v_n)_{n<\omega}$ of such a game, Player $\Diamond$ wins if $(v_n)_{n<\omega}$ satisfies the *parity condition:*

$$\liminf_{n<\omega} \Omega(v_n) \text{ is even.}$$

(b) A *(positional) strategy* for Player τ in a parity game ($\tau \in \{\Diamond, \Box\}$) is a function $\sigma : V_\tau \to E$ assigning an outgoing edge to each position of Player τ. Such a strategy is *winning* if Player τ wins every play where all of his choices are according to σ.

(c) A game $\mathcal{G}$ is *positionally determined* if there exist two positional strategies $\sigma_\Diamond$ and $\sigma_\Box$ and a partition $V = W_\Diamond + W_\Box$ of the positions ($W_\Diamond$ and $W_\Box$ may be empty) such that $\sigma_\Diamond$ is winning for Player $\Diamond$, for every game that starts in some position from $W_\Diamond$, and similarly $\sigma_\Box$ is winning for Player $\Box$, for all starting positions in $W_\Box$. ⌟

The usefulness of parity games stems from the following well-known fact.

Theorem 4.2 (Emerson, Jutla, Mostowski [6, 11]). *Every parity game is positionally determined.*

The second result we will need is the fact that we can compute the winning regions of a parity game by a formula of the modal μ-calculus μML. To do so, we encode a parity game $\mathcal{G} = \langle V_\Diamond, V_\Box, E, \Omega \rangle$ as a transition system with predicates $V_\Diamond, V_\Box, \Omega_k$, for each priority k.

Definition 4.3. For a μML-formula ψ and $k < \omega$, we set

$$\begin{aligned} \mathrm{step}\,\psi &:= [V_\Diamond \wedge \Diamond\psi] \vee [V_\Box \wedge \Box\psi]\,, \\ \mathrm{win} &:= \sigma_0 x_0 \cdots \sigma_{k-1} x_{k-1}.\, \mathrm{step} \bigvee_{i<k} (\Omega_i \wedge x_i)\,, \end{aligned}$$

where σ_k is equal to μ if k is odd, and equal to ν if k is even. ⌟

Proposition 4.4 (Emerson, Jutla [6]). *For all $k < \omega$, the μML-formula* win *defines the winning region for Player $\Diamond$ on all parity games with at most k priorities.*

The games obtained from automata have a special form: the players strictly alternate in making moves.

Definition 4.5. A parity game $\mathcal{G} = \langle V_\Diamond, V_\Box, E, \Omega \rangle$ is *strictly alternating* if, for every edge $\langle u, v \rangle \in E$,

$$u \in V_\Diamond \Leftrightarrow v \in V_\Box \quad \text{and} \quad u \in V_\Diamond \Rightarrow \Omega(u) \leq \Omega(v)\,. \qquad ⌟$$

For games of this special form, we obtain the following corollary to Proposition 4.4.

Definition 4.6. For $k < \omega$, we set

$$\mathrm{win}^2_k := \sigma_0 x_0 \cdots \sigma_{k-1} x_{k-1} . \Diamond\Box \bigvee_{i<k} (\Omega_i \wedge x_i) ,$$

where σ_k is equal to μ if k is odd, and equal to ν if k is even. ⌟

Corollary 4.7. *For all $k < \omega$, the μML-formula win^2_k defines the winning region (restricted to positions of Player $\Diamond$) for Player $\Diamond$ on all strictly alternating parity games with at most k priorities.*

5 Automata

When defining the transition relation for an automaton working on trees whose branching degree is unbounded, we cannot simply list all allowed states for the successors since this would be an infinite amount of data. To obtain a finite automaton we need to adopt some formalism that can be used to specify the transition relation in a finite way. Following an idea of Walukiewicz [16], we use logical formulae for this task. Generalising the work of Carreiro [4], we will show that we can translate μL-formulae into automata where the transition relation is defined by L-formulae.

Definition 5.1. Let $\mathbb{S} : \mathsf{Set} \to \mathsf{Set}$ be a polynomial functor and L a family of logics with polarities over $\mathbb{S} \circ \wp$.

(a) Given a formula $\varphi \in L[Q]$ we say that a state $q \in Q$ *occurs* in φ if $\varphi \notin L[Q \setminus \{q\}]$.

(b) Let $\delta : Q \times \Sigma \to L[Q]$ be a function. We associate with δ a directed graph whose set of vertices is Q and there is an edge $p \to q$ if q occurs in $\delta(p, a)$, for some $a \in \Sigma$. We say that $q \in Q$ is *reachable* from $p \in Q$ if this graph has a path from p to q. A *component* of δ is a strongly connected component of the associated graph.

(c) An *(alternating) L-automaton over $\mathbb{S}$-enriched trees* is a tuple

$$\mathcal{A} = \langle Q, \Sigma, \delta, q_0, \Omega \rangle$$

where Q is a finite set of *states*, Σ is a finite *input alphabet*, $q_0 \in Q$ the *initial state*, $\Omega : Q \to \omega$ a *priority function*, and $\delta : Q \times \Sigma \to L[Q]$ is the *transition function*, which we assume satisfies the following property: for every state $q \in Q$, there exists two disjoint sets $C, D \subseteq Q$ such that

- $C \cup D$ is the component containing q,
- $\delta(q, a) \in L[Q, C, D]$, for all $a \in \Sigma$, and
- $\delta(q, a)$ is monotone in all states $p \in C \cup D$.

(d) A *run* of such an automaton $\mathcal{A}$ on an $\mathbb{S}$-enriched tree t is a function $\rho : \mathrm{dom}(t) \to \wp(Q \times Q)$ with the following properties. (A pair $\langle p, q\rangle \in \rho(v)$ represents the fact that (a copy of) the automaton is in state p at the predecessor of v and in state q at the vertex v itself.) Given a state $p \in Q$, we write

$$\rho_{/p}(v) := \{\, q \in Q \mid \langle p, q\rangle \in \rho(v) \,\}\,.$$

We say that ρ is a run if $\rho(\langle\rangle) = \{\langle q_0, q_0\rangle\}$ and

$$\mathbb{S}\rho_{/p}(\mathrm{suc}(v)) \vDash \delta(p, t(v))\,, \quad \text{for all } p \in \bigcup_{q\in Q} \rho_{/q}(v)\,.$$

(e) Let ρ be a run and $\beta = (v_i)_{i<\omega}$ an infinite branch of t. A *trace* of ρ *along* β is a sequence $(q_i)_{i<\omega}$ of states starting with the initial state q_0 such that

$$\langle q_i, q_{i+1}\rangle \in \rho(v_{i+1})\,, \quad \text{for all } i < \omega\,.$$

A run ρ is *accepting* if all traces $(q_i)_i$ along every branch satisfy the parity condition:

$$\liminf_{i<\omega} \Omega(q_i) \text{ is even.}$$

The *language recognised* by $\mathcal{A}$ is the set

$$[\![\mathcal{A}]\!] := \{\, t \in \mathbb{T}_{\mathbb{S}}\Sigma \mid \mathcal{A} \text{ has an accepting run on } t \,\}\,. \qquad \lrcorner$$

Examples. Let $\mathbb{S}X := X^*$ be the functor for finitely branching transition systems.

(a) The $\mu_{\mathrm{af}}\mathsf{E}_1$-formula

$$\varphi := \mu x\big[a \vee \bigcirc[\mathsf{E}_1 x]\big]$$

checks whether there is a reachable vertex labelled by a. We can translate it into an E_1-automaton with a single state q with priority $\Omega(q) := 1$ where the transition function is

$$\delta(q, a) := \mathsf{true} \quad \text{and} \quad \delta(q, b) := \mathsf{E}_1 q\,.$$

(b) The following FO-automaton recognises the language of all trees $t \in \mathbb{T}_{\mathbb{S}}\{a, b\}$ where every subtree contains at least one letter a. We use two states p and q, where p looks for the letter a while q checks the condition for every subtree. The state q is initial, the transition function is

$$\begin{aligned}
\delta(p, a) &:= \mathsf{true}\,, \\
\delta(p, b) &:= \exists x P_p x\,, \\
\delta(q, c) &:= \forall x P_q x \wedge \exists x P_p x\,, \quad \text{for } c \in \{a, b\}\,,
\end{aligned}$$

and the priorities are $\Omega(p) := 1$ and $\Omega(q) := 0$.

(c) The $\mu_{\mathrm{p}}\mathrm{E}_1$-formula

$$\varphi := \nu x.\mu y\big[\big(a \wedge \bigcirc[\mathsf{E}_1 x]\big) \vee \bigcirc[\mathsf{E}_1 y]\big]$$

checks for the existence of a path with infinitely many letters a. We can translate it into an E_1-automaton with two states q_a and q_b. The priorities are $\Omega(q_a) := 0$ and $\Omega(q_b) := 1$ and the transition function is

$$\delta(q_c, d) := \mathsf{E}_1 q_d\,, \quad \text{for } c, d \in \{a, b\}\,. \qquad \lrcorner$$

It will turn out that the two fragments $\mu_{\mathrm{p}}L$ and $\mu_{\mathrm{af}}L$ of μL can be characterised by L-automata of the following form.

Definition 5.2. Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, \Omega\rangle$ be an L-automaton.

(a) $\mathcal{A}$ is *pure* if, for every component $C \subseteq Q$ of $\mathcal{A}$, we have

$$\begin{aligned}
&\delta(q, a) \in L[Q, C, \varnothing]\,, \quad \text{for all } q \in C \text{ and } a \in \Sigma\,, \\
\text{or} \quad &\delta(q, a) \in L[Q, \varnothing, C]\,, \quad \text{for all } q \in C \text{ and } a \in \Sigma\,.
\end{aligned}$$

(b) $\mathcal{A}$ is *weak* if

$$\Omega(p) \leq \Omega(q)\,, \quad \text{for all states } q \text{ that occur in } \delta(p, a) \text{ for some } a \in \Sigma\,,$$

and

$$\begin{aligned}
\Omega(q) \text{ is odd} \quad &\Rightarrow \quad \delta(q, a) \in L[Q, C, \varnothing]\,, \\
\Omega(q) \text{ is even} \quad &\Rightarrow \quad \delta(q, a) \in L[Q, \varnothing, C]\,,
\end{aligned}$$

where $C \subseteq Q$ is the component of $\mathcal{A}$ containing q. $\lrcorner$

Remark. Every weak L-automaton is equivalent to a (non-weak) L-automaton that only uses the priorities 0 and 1. (We can just replace the priority function Ω by the function $\Omega'(q) := \Omega(q) \bmod 2$.) ⌟

The main result of this section is the following equivalence between automata and formulae. In fact, the two formalisms are close enough that, similar to the modal μ-calculus, we can consider automata a normal form for formulae.

Theorem 5.3. *Let L be a family of logics with polarities over $\mathbb{S}$ that is closed under finite disjunctions, finite conjunctions, and duals.*

(a) *A language $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$ is μL-definable if, and only if, it is recognised by an L-automaton.*

(b) *A language $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$ is $\mu_{\mathrm{p}} L$-definable if, and only if, it is recognised by a pure L-automaton.*

(c) *A language $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$ is $\mu_{\mathrm{af}} L$-definable if, and only if, it is recognised by a weak L-automaton.*

Furthermore, the above translations between formulae and automata are effective provided that disjunctions, conjunctions, and duals of L-formulae are computable.

The remainder of this section is devoted to the proof, which is basically the same as the corresponding proof for the modal μ-calculus. We only have to additionally check that the transition functions and modal operators we construct are well-formed and of the correct type. We split the proof into Propositions 5.8 and 5.9 below.

Before doing so, it is useful to give an alternative definition of acceptance via a parity game.

Definition 5.4. Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_{\mathrm{o}}, \Omega \rangle$ be an L-automaton and $t \in \mathbb{T}_{\mathbb{S}}\Sigma$ an input tree. The *acceptance game* for $\mathcal{A}$ on t is the parity game $\mathcal{G}(\mathcal{A}, t) := \langle V_\Diamond, V_\Box, E, \Omega' \rangle$ with positions

$$V_\Diamond := \mathrm{dom}(t) \times Q\,,$$
$$V_\Box := \{\, \langle v, s \rangle \in \mathrm{dom}(t) \times \mathbb{S}\mathcal{P}(Q) \mid s \simeq_{\mathrm{sh}} \mathrm{suc}(v) \,\}\,.$$

The priorities are

$$\Omega'(\langle v, q \rangle) := \Omega(q) \quad \text{and} \quad \Omega'(\langle v, s \rangle) := \max \mathrm{rng}\, \Omega\,,$$

for $v \in \mathrm{dom}(t)$, $q \in Q$, and $s \in \mathbb{S}Q$.

Finally, the edge relation is defined as follows. Let $u, v \in \mathrm{dom}(t)$, $q \in Q$, and $s \in \mathbb{S}\wp(Q)$. There are edges

$$\begin{aligned}
\langle v, q\rangle &\to \langle v, s\rangle \quad &&: \text{iff} \quad s \vDash \delta(q, t(v))\,,\\
\langle v, s\rangle &\to \langle u, q\rangle \quad &&: \text{iff} \quad u = \mathrm{suc}(v)(d) \quad \text{and} \quad q \in s(d)\,,\\
& && \qquad \text{for some } d \in \mathrm{dom}(\mathrm{suc}(v))\,.
\end{aligned}$$

⌟

Proposition 5.5. *An L-automaton $\mathcal{A}$ accepts a tree t if, and only if, Player $\Diamond$ has a winning strategy in the game $\mathcal{G}(\mathcal{A}, t)$.*

The proof is entirely standard: every accepting run can be used to define a winning strategy and every winning strategy an accepting run.

For the two directions of the proof of Theorem 5.3, we generalise the standard translation between the modal μ-calculus and tree automata. We start with the translation of automata into formulae. To simplify the construction we will use a variant of μL with *simultaneous* fixed points.

Definition 5.6. The variant of μL with *simultaneous fixed points* has fixed-point formulae of the form

$$\mu_k \bar{x}.\bar{\psi} \quad \text{and} \quad \nu_k \bar{x}.\bar{\psi}\,,$$

where $\bar{x}$ is an n-tuple of (pairwise distinct) fixed-point variables, $\bar{\psi}$ an n-tuple of μL-formulae that are monotone in the variables $\bar{x}$, and $k < n$ is an index. The semantics $[\![\mu_k \bar{x}.\bar{\psi}]\!]_{\bar{P}}$ of such a formula is defined as follows. Let $\bar{T}$ be the least fixed point of the operation $F : \wp(S)^n \to \wp(S)^n$ defined by

$$F(\bar{Q}) := \left\langle [\![\psi_k]\!]_{\bar{P}\bar{Q}} \right\rangle_{k<n}.$$

Then $[\![\mu_k \bar{x}.\bar{\psi}]\!]_{\bar{P}} := T_k$. ⌟

Lemma 5.7. *Every $\mu L[\Sigma; X, Y]$-formula with simultaneous fixed points can be translated to one without. Furthermore, if the given formula is alternation-free or pure, so is the resulting formula.*

Proof. This is a standard construction, which we will recall for convenience. Consider a formula of the form $\varphi = \mu_k \bar{x}.\bar{\psi}$ where $\bar{x}$ and $\bar{\psi}$ are n-tuples. (The case of a greatest fixed point is handled analogously.) We may assume by induction that the formulae ψ_i do not contain simultaneous fixed points. We transform φ into a μL-formula in two steps.

First, we modify the formula such the variables $x_0, \dots, x_{i-1}$ are not free in ψ_i, for $i < n$. To do so, we replace each occurrence of x_0 in ψ_i, for $i > 0$, by the formula $\mu x_0.\psi_0$. Next we replace each x_1 in (the modified version of) ψ_i, for $i > 1$, by $\mu x_1.\psi_1$ Continuing this way, we replace each x_j in ψ_i, for $i > j$, by $\mu x_j.\psi_j$.

We denote the resulting formulae again by $\psi_0, \dots, \psi_{n-1}$. In the second step, we eliminate the remaining variables. We start with the formula

$$\varphi_{n-1} := \mu x_{n-1}.\psi_{n-1}\,.$$

Next, we construct

$$\varphi_{n-2} := \mu x_{n-2}.\psi'_{n-2}\,,$$

where ψ'_{n-2} is the formula obtained from ψ_{n-2} by replacing each occurrence of x_{n-1} by φ_{n-1}. Continuing in this way, we set

$$\varphi_j := \mu x_j.\psi'_j\,,$$

where ψ'_j is the formula obtained from ψ_j by replacing each occurrence of x_i with $i > j$ by φ_i. The resulting formulae $\varphi_0, \dots, \varphi_{n-1}$ belong to μL and define the respective components of the fixed point. In particular, φ_k is equivalent to $\mu_k \bar{x}.\bar{\psi}$.

Finally, note that our construction preserves purity and alternation freeness. □

The two directions of the proof of Theorem 5.3 can now be proved as follows.

Proposition 5.8.

(a) *Every language recognised by an L-automaton is μL-definable.*

(b) *Every language recognised by a weak L-automaton is $\mu_{\mathrm{af}} L$-definable.*

(c) *Every language recognised by a pure L-automaton is $\mu_{\mathrm{p}} L$-definable.*

Proof. Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, \Omega \rangle$ be an L-automaton, $t \in \mathbb{T}_{\mathbb{S}}\Sigma$ an input tree, and $\mathcal{G} = \langle V_\Diamond, V_\Box, E, \Omega \rangle$ the associated acceptance game. It is sufficient to find a formula φ of the respective logic such that

$$t \vDash \varphi \quad \text{iff} \quad \text{the initial position of } \mathcal{G} \text{ is winning for Player } \Diamond.$$

More precisely, for each state $q \in Q$, we will construct a formula $\varphi_q \in \mu L$ such that, for every vertex $v \in \mathrm{dom}(t)$,

$$t, v \vDash \varphi_q \quad \text{iff} \quad \text{Player } \Diamond \text{ wins when starting in the position } \langle v, q \rangle\,.$$

We proceed by induction on the number of components of $\mathcal{A}$ that are reachable from q. Hence, let C be a component with $q \in C$ and let $P \subseteq Q$ be the set of all states $p \in Q \setminus C$ reachable from some $q \in C$. By inductive hypothesis, we already know the formulae φ_p, for $p \in P$. By Corollary 4.7, the μML-formula

$$\mathrm{win}^2 := \sigma_0 x_0 \cdots \sigma_{k-1} x_{k-1}.\Diamond\Box \bigvee_{i<k} (\Omega_i \wedge x_i)$$

defines the winning region for Player $\Diamond$ in $\mathcal{G}$. It follows that the μML-formula

$$\chi_C := \sigma_0 x_0 \cdots \sigma_{k-1} x_{k-1}.\Diamond\Box\Big[\bigvee_{p\in C} (Q_p \wedge x_{\Omega(p)}) \vee \bigvee_{p\in P} (Q_p \wedge \chi_P)\Big]$$

defines the winning region restricted to positions with a state in C. (χ_P is the corresponding formula for states in P and Q_p is the predicate checking that the current state is p. Hence, $\Omega_k \equiv \bigvee_{p\in\Omega^{-1}(k)} Q_p$.)

(a), (b) We will inductively translate every subformula $\psi(x_0, \ldots, x_{n-1})$ of χ_C into an μL-formula $\psi_q^*(\tilde{x}_0, \ldots, \tilde{x}_{n-1})$ such that, for all $v \in \mathrm{dom}(t)$ and $P_0, \ldots, P_{k-1} \subseteq V_\Diamond$,

$$\mathcal{G}, \langle v, q\rangle \vDash \psi(P_0, \ldots, P_{k-1}) \quad \text{iff} \quad t, v \vDash \psi_q^*(\bar{P}_0, \ldots, \bar{P}_{k-1}),$$

where $\bar{P}_i := (P_i^p)_{p\in Q}$ with

$$P_i^p := \{\, v \in \mathrm{dom}(t) \mid \langle v, p\rangle \in P_i \,\}.$$

Note that to each variable x_i in χ_C there corresponds a Q-tuple $(x_{i,q})_{q\in Q}$ in the translation. We start with

$$\Big[\bigvee_{p\in C} (Q_p \wedge x_{\Omega(p)}) \vee \bigvee_{p\in P} (Q_p \wedge \chi_P)\Big]_q^* := \begin{cases} x_{\Omega(q),q} & \text{if } q \in C, \\ \varphi_q & \text{if } q \in P. \end{cases}$$

For the modal operators, we set

$$\Big[\Diamond\Box\Big(\bigvee_{p\in C} (Q_p \wedge x_{\Omega(p)}) \vee \bigvee_{p\in P} (Q_p \wedge \chi_P)\Big)\Big]_q^* := \bigvee_{a\in\Sigma} \big[a \wedge \bigcirc\mathbb{S} f(\delta(q,a))\big],$$

where $X := \{\, x_{\Omega(p),p} \mid p \in C \,\}$ and the function $f : Q \to \mu L[\Sigma; X, X]$ is defined by

$$f(p) := \begin{cases} x_{\Omega(p),p} & \text{if } p \in C, \\ \varphi_p & \text{if } p \in P. \end{cases}$$

Note that the above formula is well-formed since we have $\delta(q,a) \in L[Q, C_1, C_2]$, for some partition $C = C_1 + C_2$ independent of a, which implies that

$$\bigcirc\mathbb{S}f(\delta(q,a)) \in \mu L[\Sigma; X_1, X_2] \quad \text{where} \quad X_i \coloneqq \{\, x_{\Omega(p),p} \mid p \in C_i \,\}\,.$$

Furthermore, in case (b), we have $\delta(q,a) \in L[Q, C, \varnothing] \cup L[Q, \varnothing, C]$, which implies that

$$\bigcirc\mathbb{S}f(\delta(q,a)) \in \mu_{\mathrm{p}} L[\Sigma; X, \varnothing] \cup \mu_{\mathrm{p}} L[\Sigma; \varnothing, X]\,.$$

Hence, the resulting formula is pure.

Finally, we translate the fixed-point operators using simultaneous fixed-points as

$$(\mu x_i.\psi)^*_q \coloneqq \mu_q (x_{i,p})_{p \in C \cap \Omega^{-1}(i)}.(\psi^*_p)_{p \in C \cap \Omega^{-1}(i)}\,, \quad \text{if } i \text{ is odd,}$$

or $$(\nu x_i.\psi)^*_q \coloneqq \nu_q (x_{i,p})_{p \in C \cap \Omega^{-1}(i)}.(\psi^*_p)_{p \in C \cap \Omega^{-1}(i)}\,, \quad \text{if } i \text{ is odd.}$$

Note that these formulae are well-formed since every state $p \in C$ occurs positively in $\delta(q,a)$. This implies that every variable $x_{i,p}$ occurs positively in $\psi_{p'}$, for $p, p' \in C \cap \Omega^{-1}(i)$.

It remains to prove the correctness of our translation. We proceed by induction on the formula ψ. Since most steps are straightforward, we only consider the case of the modal operator. Hence, suppose that $\psi = \Diamond\Box\vartheta$. By definition of $\mathcal{G}$, we have

$$\mathcal{G}, \langle v, q\rangle \vDash \Diamond\Box\vartheta$$

if, and only if, there exists a labelling $s \in \mathbb{S}Q$ such that $s \simeq_{\mathrm{sh}} \mathrm{suc}(v)$,

$$s \vDash \delta(q, t(v)) \quad \text{and} \quad \mathcal{G}, \langle \mathrm{suc}(v)(d), s(d)\rangle \vDash \vartheta\,, \quad \text{for all } d \in \mathrm{dom}(\mathrm{suc}(v))\,.$$

By inductive hypothesis, this is equivalent to the existance of some $s \in \mathbb{S}Q$ satisfying

$$s \vDash \delta(q, t(v)) \quad \text{and} \quad t, \mathrm{suc}(v)(d) \vDash \vartheta^*_{s(d)}\,, \quad \text{for all } d \in \mathrm{dom}(\mathrm{suc}(v))\,.$$

This last statement holds if, and only if,

$$t, v \vDash \bigcirc\mathbb{S}f(\delta(q, t(v)))_{p \in Q}(\bar{P}_0, \ldots, \bar{P}_{n-1})\,,$$

which is equivalent to

$$t, v \vDash \bigvee_{a \in \Sigma} \big[a \wedge \bigcirc\mathbb{S}f(\delta(q,a))_{p \in Q}(\bar{P}_0, \ldots, \bar{P}_{n-1})\big]\,.$$

(c) As above, we can use the μML-formula

$$\chi_C := \sigma_0 x_0 \cdots \sigma_{k-1} x_{k-1}.\Diamond\Box\Big[\bigvee_{p\in C}(Q_p \wedge x_{\Omega(p)}) \vee \bigvee_{p\in P}(Q_p \wedge \chi_P)\Big]$$

to define the winning region restricted to positions with a state in C. Since the automaton is weak, all states in C have the same priority k. Consequently, the formula simplifies to

$$\chi'_C := \sigma_k x.\Diamond\Box\Big[\bigvee_{p\in C}(Q_p \wedge x) \vee \bigvee_{p\in P}(Q_p \wedge \chi'_P)\Big].$$

As above, we inductively translate every subformula ψ of χ'_C into into an $\mu_{\mathrm{p}}L$-formula ψ^*_q, starting with

$$\Big[\bigvee_{p\in C}(Q_p \wedge x) \vee \bigvee_{p\in P}(Q_p \wedge \chi'_P)\Big]^*_q := \begin{cases} x_q & \text{if } q \in C, \\ \varphi_q & \text{if } q \in P, \end{cases}$$

and

$$\Big[\Diamond\Box\Big(\bigvee_{p\in C}(Q_p \wedge x_{\Omega(p)}) \vee \bigvee_{p\in P}(Q_p \wedge \chi'_P)\Big)\Big]^*_q := \bigvee_{a\in\Sigma}\big[a \wedge \circ\mathbb{S}f(\delta(q,a))\big],$$

where $X := \{\, x_p \mid p \in C \,\}$ and the function $f : Q \to \mu_{\mathrm{p}}L[\Sigma; X, X]$ is defined by

$$f(p) := \begin{cases} x_p & \text{if } p \in C, \\ \varphi_p & \text{if } p \in P. \end{cases}$$

Note that this formula is well-formed since, depending on whether or not k is odd, we have $\delta(q,a) \in L[Q, C, \varnothing]$ or $\delta(q,a) \in L[Q, \varnothing, C]$, which implies that

$$\circ\mathbb{S}f(\delta(q,a)) \in \begin{cases} \mu_{\mathrm{p}}L[\Sigma; X, \varnothing] & \text{if } k \text{ is odd}, \\ \mu_{\mathrm{p}}L[\Sigma; \varnothing, X] & \text{if } k \text{ is odd}. \end{cases}$$

Finally, we translate the fixed-point operator by

$$(\mu x.\psi)^*_q := \mu_q(x_p)_{p\in Q}.(\psi^*_p)_{p\in Q}, \quad \text{if } k \text{ is odd},$$
$$\text{or} \qquad (\nu x.\psi)^*_q := \nu_q(x_p)_{p\in Q}.(\psi^*_p)_{p\in Q}, \quad \text{if } k \text{ is odd},$$

which is alternation-free. □

Example. To understand the construction in the following proof, let us consider the $\mu\mathrm{E}_1$-formula

$$\varphi := \nu x.\mu y\big[\big(a \wedge \bigcirc[\mathsf{E}x]\big) \vee \bigcirc[\mathsf{E}y]\big]$$

which checks for the existence of a path with infinitely many letters a. This formula has the following subformulae.

$$\begin{array}{llll} a\,, & x\,, & \psi_0 := \bigcirc[\mathsf{E}y]\,, & \psi_2 := a \wedge \psi_1\,, \qquad \psi_4 := \mu y.\psi_3\,, \\ & y\,, & \psi_1 := \bigcirc[\mathsf{E}x]\,, & \psi_3 := \psi_2 \wedge \psi_0\,, \qquad \varphi\,. \end{array}$$

We translate φ into the automaton with states

$$Q := \{x, y, \varphi\}\,,$$

priorities

$$\Omega(\varphi) := 0\,, \quad \Omega(x) := 0\,, \quad \Omega(y) := 1\,,$$

and transitions

$$\delta(\varphi, c) = \delta(x, c) = \delta(y, c) := \begin{cases} \mathsf{E}x \vee \mathsf{E}y & \text{if } c = a\,, \\ \mathsf{E}y & \text{if } c \neq a\,. \end{cases}$$

⌟

Proposition 5.9. *Let L be a family of logics over $\mathbb{S}$ that is closed under finite disjunctions, finite conjunctions, and duals.*

(a) *Every μL-definable language is recognised by an L-automaton.*

(b) *Every $\mu_{\mathrm{p}} L$-definable language is recognised by a pure L-automaton.*

(c) *Every $\mu_{\mathrm{af}} L$-definable language is recognised by a weak L-automaton.*

Proof. For technical reasons, we will present our translation for formulae with free fixed-point variables. Therefore we have to work with trees equipped with additional information specifying the values of the variables. We will represent such a tree as a tree over the extended alphabet $\Sigma \times \wp(V)$, where V is the set of variables. The labels of such a tree are therefore pairs $\langle a, U\rangle$ with $a \in \Sigma$ and $U \subseteq V$, where the second component specifies to which of the variables $x \in V$ the current vertex belongs. Our notation for such trees is

$$t, \tilde{P} \in \mathbb{T}_{\mathbb{S}}[\Sigma \times \wp(V)] \quad \text{where} \quad t \in \mathbb{T}_{\mathbb{S}}\Sigma \text{ and } P_x \subseteq \mathrm{dom}(t), \text{ for } x \in V\,.$$

Fix a formula $\varphi \in \mu L[\Sigma; X_\mathrm{f}, Y_\mathrm{f}]$. By Lemma 3.10, we may assume that φ is guarded and in negation normal form. For each fixed-point variable x bound in φ, we denote by $\sigma_x x.\psi_x$ the subformula where x is bound. Set

$$X_\mathrm{b} := \{\, x \mid \psi_x \in \mu L[\Sigma; U, V] \text{ for some } U, V \text{ with } x \in U \,\},$$
$$Y_\mathrm{b} := \{\, x \mid \psi_x \in \mu L[\Sigma; U, V] \text{ for some } U, V \text{ with } x \in V \,\}.$$

Note that, since φ is in negation normal form, we have $\psi \in L[\Sigma; X_\mathrm{f} + X_\mathrm{b}, Y_\mathrm{f} + Y_\mathrm{b}]$, for all subformulae ψ of φ.

We define the *alternation-depth* of a variable x as the length n of the longest sequence $\sigma_0 y_0.\psi_0, \ldots, \sigma_n y_n.\psi_n$ of subformulae in Q such that

- $\sigma_0 y_0.\psi_0 \in Q$,
- $\sigma_n y_n.\psi_n = \sigma_x x.\psi_x$, and
- $\sigma_{i+1} y_{i+1}.\psi_{i+1}$ is a subformula of $\sigma_i y_i.\psi_i$ and $\sigma_{i+1} \neq \sigma_i$, for all $i < n$.

We construct an automaton $\mathcal{A}_\varphi$ whose set of states

$$Q \subseteq L[\Sigma; X_\mathrm{f} + X_\mathrm{b}, Y_\mathrm{f} + Y_\mathrm{b}]$$

is the set of all subformulae of φ. (We treat different occurrences of the same subformula as different subformulae.) Each state $\psi \in Q$ checks whether a subtree satisfies the formula ψ. We use the second component i only to signal when we iterate a fixed point. The initial state is φ and the priorities are given by

$$\Omega(\psi) := \begin{cases} 2i_x + 1 & \text{if } \psi = x \in X_\mathrm{b} \cup Y_\mathrm{b} \text{ and } \sigma_x = \mu, \\ 2i_x & \text{if } \psi = x \in X_\mathrm{b} \cup Y_\mathrm{b} \text{ and } \sigma_x = \nu, \\ 2k & \text{otherwise}, \end{cases}$$

where i_x is the alternation-depth of the variable x and k is the maximal alternation-depth of a variable in φ.

We define the transition function δ inductively starting with the letters and the modal operators.

$$\delta\big(a, \langle c, U\rangle\big) := \begin{cases} \mathsf{true} & \text{if } a = c, \\ \mathsf{false} & \text{if } a \neq c, \end{cases} \qquad \text{for } a \in \Sigma,$$

$$\delta\big(\neg a, \langle c, U\rangle\big) := \begin{cases} \mathsf{false} & \text{if } a = c, \\ \mathsf{true} & \text{if } a \neq c, \end{cases} \qquad \text{for } a \in \Sigma,$$

$$\delta\big(x, \langle c, U\rangle\big) := \begin{cases} \mathsf{true} & \text{if } a \in U, \\ \mathsf{false} & \text{if } a \notin U, \end{cases} \qquad \text{for } x \in X_\mathrm{f} \cup Y_\mathrm{f},$$

$$
\begin{aligned}
\delta\big(\psi_0 \vee \psi_1, \langle c, U\rangle\big) &:= \delta\big(\psi_0, \langle c, U\rangle\big) \vee \delta\big(\psi_1, \langle c, U\rangle\big)\,,\\
\delta\big(\psi_0 \wedge \psi_1, \langle c, U\rangle\big) &:= \delta\big(\psi_0, \langle c, U\rangle\big) \wedge \delta\big(\psi_1, \langle c, U\rangle\big)\,,\\
\delta\big(\mu x.\psi_0, \langle c, U\rangle\big) &:= \delta\big(\psi_0, \langle c, U\rangle\big)\,,\\
\delta\big(\nu x.\psi_0, \langle c, U\rangle\big) &:= \delta\big(\psi_0, \langle c, U\rangle\big)\,,\\
\delta\big(\bigcirc\psi, \langle c, U\rangle\big) &:= \psi\,,\\
\delta(x, \langle c, U\rangle) &:= \delta(\psi_x, \langle c, U\rangle)\,, \qquad \text{for } x \in X_\mathrm{b} \cup Y_\mathrm{b}\,.
\end{aligned}
$$

(Note that, since φ is guarded, the definition of $\delta(\psi_x, \langle c, U\rangle)$ does not depend on the definition of $\delta(x, \langle c, U\rangle)$. So the above definition does not create a cyclic dependency.)

Before proving that the resulting automaton $\mathcal{A}_\varphi$ recognises the correct language, let us show that it is well-formed and that is has the correct type. Fix a component C of $\mathcal{A}_\varphi$. Set

$$
Z := \{\, x \in X_\mathrm{b} \cup Y_\mathrm{b} \mid \sigma_x x.\psi_x \in C \,\}\,.
$$

Then

$$
C \subseteq \mu L[\Sigma; X_\mathrm{f} + (Z \cap X_\mathrm{b}), Y_\mathrm{f} + (Z \cap Y_\mathrm{b})\,.
$$

Let $\Phi_+ \subseteq C$ be the set of all subformulae of φ that contain some variable from $Z \cap X_\mathrm{b}$ and let $\Phi_- \subseteq C$ the the corresponding set for variables in $Z \cap Y_\mathrm{b}$. It follows that

$$
\bigcirc\psi \in C \quad \text{implies} \quad \psi \in L[\mu L, \Phi_+, \Phi_-]\,.
$$

Since every transition formula is a boolean combination of such formulae ψ, it follows that

$$
\delta\big(\psi, \langle c, U\rangle\big) \in L[Q, \Phi_+, \Phi_-]\,, \quad \text{for all } \psi \in C\,.
$$

Consequently, $\mathcal{A}_\varphi$ is well-formed.

Furthermore, if φ is pure, we have $Z \subseteq X_\mathrm{b}$ or $Z \subseteq Y_\mathrm{b}$. Consequently, $\Phi_+ = \varnothing$ or $\Phi_- = \varnothing$, and it follows that $\mathcal{A}_\varphi$ is pure.

In order to obtain a weak automaton we have to slightly modify the above construction. Note that, if our formula φ is weak, we have $\sigma_x = \sigma_y$, for all variables $x, y \in C$. Consequently, all such variables x have the same alternation depth and therefore the same priority $\Omega(x)$. Since every loop contains a state $x \in X_\mathrm{b} \cup Y_\mathrm{b}$

and the other states have maximal priority, it follows that we can set all priorities in C to the same value $\Omega(x)$ without changing the behaviour of the automaton. The resulting automaton is weak.

To prove the correctness of our construction, let $\mathcal{A}_\psi$ be the automaton obtained by translating the formula $\psi \in Q$. Note that each of these automata is equal to part of the automaton $\mathcal{A}_\varphi$ for the whole formula φ, except that the transition function differs for states of the form $\langle x, i\rangle$ with $x \in X_\mathrm{b} + Y_\mathrm{b}$.

We show by induction on $\psi \in Q$ that

$$t, \bar{P} \in [\![\mathcal{A}_\psi]\!] \quad \text{iff} \quad t \in [\![\psi]\!]_{\bar{P}}$$

(where $\bar{P}$ are the value of the free variables in ψ). Most cases are straightforward. Let us give the proof for the least fixed-point operator. Hence, let us consider a subformula $\mu x.\psi \in Q$ and suppose that we have already proved the claim for the formula ψ.

($\Rightarrow$) Fix a tree $t, \bar{P} \in \mathbb{T}_\mathbb{S}(\Sigma \times \wp(X_\mathrm{f} + Y_\mathrm{f}))$. Suppose that σ is a winning strategy for Player $\Diamond$ in the game $\mathcal{G}(\mathcal{A}_{\mu x.\psi}, t, \bar{P})$. Since no infinite play conforming to σ contains infinitely many positions with the state x, we can define the following ordinal rank for the positions in the game. If, from a postion $\langle v, q\rangle$, no position of the form $\langle u, x\rangle$ is reachable when following the strategy σ, we assign the rank 0 to $\langle v, q\rangle$. Inductively, the rank of an arbitrary position $\langle v, q\rangle$ of Player $\Diamond$ is the least ordinal α such that every successor of the position $\sigma(\langle v, q\rangle)$ has a rank less than α.

By induction on α we prove that, if Player $\Diamond$ has a winning strategy σ of rank at most α in the game $\mathcal{G}(\mathcal{A}_{\mu x.\psi}, t, \bar{P})$, then $t \in [\![\mu x.\psi]\!]_{\bar{P}}$. Let Q be the set of all vertices $v \in \mathrm{dom}(t)$ such that the position $\langle v, x\rangle$ of $\mathcal{G}(\mathcal{A}_{\mu x.\psi}, t, \bar{P})$ has rank less than α. Then σ induces a winning strategy in $\mathcal{G}(\mathcal{A}_\psi, t, \bar{P}Q)$, which implies that $t \in [\![\psi]\!]_{\bar{P},Q}$. By inductive hypothesis, we further have $Q \subseteq [\![\mu x.\psi]\!]_{\bar{P}}$. By monotonicity of ψ, it follows that $t \in [\![\psi(\mu x.\psi)]\!]_{\bar{P}}$, which is equivalent to $t \in [\![\mu x.\psi]\!]_{\bar{P}}$,

($\Leftarrow$) Fix a tree $t, \bar{P} \in \mathbb{T}_\mathbb{S}(\Sigma \times \wp(X_\mathrm{f} + Y_\mathrm{f}))$. For an ordinal α, let $F^\alpha := F^\alpha_\psi(\varnothing)$ be the α-th stage of the fixed-point induction for the formula ψ on $t, \bar{P}$. By induction on α, we show that $v \in F^\alpha$ implies $(t, \bar{P})|_v \in [\![\mathcal{A}_{\mu x.\psi}]\!]$, where $(t, \bar{P})|_v$ denotes the subtree of $t, \bar{P}$ rooted at v. For $\alpha = 0$, the claim is trivial. If α is a limit ordinal, we have $F^\alpha = \bigcup_{\beta<\alpha} F^\beta$ and the claim follows immediately by inductive hypothesis. For the successor step, suppose that we have already proved the claim for α. Let $v \in F^{\alpha+1}$. By the inductive hypothesis for ψ, there exists an accepting run of $\mathcal{A}_\psi$ on $t, \bar{P}F^\alpha$. Furthermore, for every $u \in F^\alpha$, the inductive hypothesis for α provides

a run of $\mathcal{A}_{\mu x.\psi}$ on $(t, \tilde{P})|_u$. Combining these runs, we obtain a run of $\mathcal{A}_{\mu x.\psi}$ on $(t, \tilde{P})|_v$. □

6 Projection

We would like to translate various variants of monadic second-order logic into automata. To be able to do so, we need to prove that the resulting classes of automata are closed under projections of various kinds.

Definition 6.1. Let $\mathbb{F}$ be a polynomial functor and Σ, Γ two sets.

(a) The *projection* of $s \in \mathbb{F}(\Sigma \times \Gamma)$ is

$$\mathrm{pr}_\Sigma(s) := f \circ s\,,$$

where $f : \Sigma \times \Gamma \to \Sigma$ is the projection to the first component.

The *projection* of a language $K \subseteq \mathbb{F}(\Sigma \times \Gamma)$ is the language

$$\mathrm{pr}_\Sigma[K] := \{\, \mathrm{pr}_\Sigma(s) \mid s \in K \,\}\,.$$

(b) Fix a distinguished element $\gamma_\circ \in \Gamma$ and a class

$$\mathcal{P} \subseteq \left\{\, P \;\middle|\; P \subseteq \mathrm{dom}(t), \text{ for some } t \in \mathbb{F}1 \,\right\}.$$

(Usually, $\Gamma = \wp(X)$ is a power set and $\gamma_\circ = \varnothing$ the empty set.)

The *$\mathcal{P}$-projection* of $K \subseteq \mathbb{T}(\Sigma \times \Gamma)$ is the language $\mathrm{pr}^{\mathcal{P}}_{\Sigma,\gamma_\circ}[K] := \mathrm{pr}_{\Sigma,\gamma_\circ}[K_{\mathcal{P}}]$, where

$$K_{\mathcal{P}} := \left\{\, s \in K \;\middle|\; \text{there is some } P \in \mathcal{P} \text{ such that } \mathrm{dom}(s) \smallsetminus t^{-1}[\Sigma \times \{\gamma_\circ\}] \subseteq P \,\right\}.$$

If $\mathbb{F} = \mathbb{T}_{\mathbb{S}}$ and the set $\mathcal{P}$ consists of all *finite sets, well-founded sets, finitely branching sets, chains,* or *finite chains,* we speak of, respectively, the *finite projection,* the *well-founded projection,* the *finitely branching projection,* the *chain projection,* or the *finite chain projection* of K.

If $\mathbb{F} = \mathbb{S}$ and the set $\mathcal{P}$ consists of all *finite sets* or all *singletons,* we speak of, respectively, the *finite projection* or the *singleton projection.*

(c) We say that a property P of languages is *closed under $\mathcal{P}$-projections* if, whenever a language K has property P, so does every $\mathcal{P}$-projection of K. Similarly, we say that a family of logics L is closed under $\mathcal{P}$-projections if, whenever a language K is L-definable, so is every $\mathcal{P}$-projection of K. Given a formula $\varphi \in L[\Sigma \times \Gamma]$ defining K, we denote by $\exists^{\mathcal{P}}_{\Gamma,\gamma_\circ}\varphi$ the formula defining the corresponding $\mathcal{P}$-projection. ⌟

The automaton construction below does not work for arbitrary logics L. We have to make a few restrictions. First of all, the construction does not respect polarities. Which means that we will work with families of logics *without* polarities. Furthermore, we require two basic closure properties.

Definition 6.2. Let L be a family of logics over $\mathbb{S} \circ \wp$.

(a) We say that L is closed under *label restrictions* if, for all $\Delta \subseteq \wp(\Sigma)$, there exists an $L[\Sigma]$-formula χ_Δ defining the set $\mathbb{S}\Delta$.

(b) L is closed under *boolean label substitutions* if there exists a family $\hat{L}$ of logics over $\mathbb{S}$ such that $L[Q] = \hat{L}[\wp(Q)]$, for all sets Q. ⌟

Lemma 6.3. *Let L be a family of logics over $\mathbb{S} \circ \wp$ that is closed under boolean label substitutions, Let Σ, Γ, Δ be sets, $\psi \in L[\Sigma]$ a formula, and let $(\vartheta_c)_{c\in\Sigma}$ be a family of boolean combinations of elements of $\Delta \times \Gamma$. There exists an $L[\Delta \times \wp(\Gamma)]$-formula $\psi[c \mapsto \vartheta_c]_{c\in\Sigma}$ such that*

$$s \vDash \psi[c \mapsto \vartheta_c]_{c\in\Sigma} \quad \text{iff} \quad \mathbb{S}\tau(s) \vDash \psi\,,$$

where $\tau : \wp(\Delta \times \wp(\Gamma)) \to \wp(\Sigma)$ is the function

$$\tau(P) := \{\, c \in \Sigma \mid \langle a, B\rangle \in P,\ \{a\} \times B \text{ satisfies } \vartheta_c \,\}\,.$$

Proof. We have to show that $\mathbb{S}\tau : \mathbb{S}\wp(\Delta \times \wp(\Gamma)) \to \mathbb{S}\wp(\Sigma)$ is (the second component of) a morphism $L[\Sigma] \to L[\Delta \times \wp(\Gamma)]$ of logics. Note that

$$\tau = g \circ \wp(f)\,,$$

where

$$f : \Delta \times \wp(\Gamma) \to \wp(\Delta \times \Gamma) : \langle a, B\rangle \mapsto \{a\} \times B\,,$$
$$g : \wp\wp(\Delta \times \Gamma) \to \wp(\Sigma) : P \mapsto \{\, c \in \Sigma \mid A \in P,\ A \text{ satisfies } \vartheta_c \,\}\,.$$

Let $\hat{L}$ be the extension of L to a logic over $\mathbb{S}$. We obtain morphisms of logics

$$\mathbb{S}\wp(f) : L[\wp(\Delta \times \Gamma)] \to L[\Delta \times \wp(\Gamma)]\,,$$
$$\mathbb{S}g : \hat{L}[\Sigma] \to \hat{L}[\wp(\Delta \times \Gamma)]\,.$$

The composition $\mathbb{S}\tau = \mathbb{S}(g \circ \wp(f))$ induces the desired morphism of logics $L[\Sigma] \to L[\Delta \times \wp(\Gamma)]$. □

Definition 6.4. Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, \Omega \rangle$ be an L-automaton.

(a) For $s, t \in \mathbb{S}\wp(X)$, we write

$$s \leq t \quad :\text{iff} \quad s \simeq_{\text{sh}} t \quad \text{and} \quad s(d) \subseteq t(d)\,, \quad \text{for all } d \in \text{dom}(s)\,.$$

(b) $\mathcal{A}$ is *partially non-deterministic* if there exists a partition $Q = Q_{\text{alt}} + Q_{\text{nd}}$ of its states such that,

- for $q \in Q_{\text{alt}}$, the formula $\delta(q, a)$ does not contain any state from Q_{nd},
- for all $q \in Q_{\text{nd}}$ and $a \in \Sigma$,

$$\begin{aligned} s \vDash \delta(q, a) \quad \text{implies} \quad & \text{there is some } s_0 \leq s \text{ such that} \\ & s_0 \vDash \delta(q, a) \text{ and, for all } d \in \text{dom}(s) \\ & |s_0(d)| = 1 \text{ or } s_0(d) \subseteq Q_{\text{alt}}\,. \end{aligned}$$

We call the elements of Q_{alt} the *alternating states* of $\mathcal{A}$ and those of Q_{nd} its *non-deterministic states.*

(c) Suppose that $\mathcal{A}$ is partially non-deterministic. Let $Q = Q_{\text{alt}} + Q_{\text{nd}}$ be the corresponding partition of its states, and let ρ be a run on the input tree t.

We say that ρ is *economical* if, for all $v \in \text{dom}(\rho)$,

$$|\rho(v)| = 1 \quad \text{or} \quad \rho(v) \subseteq Q \times Q_{\text{alt}}\,.$$

The *non-deterministic part* of an economical run ρ on t is the set

$$P := \{\, v \in \text{dom}(\rho) \mid \rho(v) \in \wp(Q_{\text{nd}} \times Q_{\text{nd}}) \,\}\,.$$

The complement of P is the *alternating part* of ρ.

Given a class $\mathcal{P}$ of prefixes as in Definition 6.1, we say that $\mathcal{A}$ is *partially non-deterministic of shape* $\mathcal{P}$ if, for every $t \in [\![\mathcal{A}]\!]$, there exists an accpeting economical run ρ whose non-deterministic part belongs to $\mathcal{P}$. ⌟

Lemma 6.5. *Let $\mathcal{A}$ be a partially non-deterministic automaton. For every $t \in [\![\mathcal{A}]\!]$, there exists an accpeting run ρ of $\mathcal{A}$ on t that is economical.*

Proof. Let ρ be a minimal (with respect to $\subseteq$) accepting run of $\mathcal{A}$ on t. We claim that ρ is economical. For a contradiction, suppose otherwise. Fix a vertex $v \in \text{dom}(t)$ such that

$$|\rho(v)| > 1 \quad \text{and} \quad \rho(v) \cap (Q \times Q_{\text{nd}}) \neq \varnothing\,.$$

Choose v such that $|v|$ is minimal, let u be the predecessor of v, and fix a pair $\langle p, q\rangle \in \rho(v)$ with $q \in Q_{\mathrm{nd}}$.

We distinguish two cases. If $p \in Q_{\mathrm{alt}}$, the transition formula $\delta(p, t(u))$ does not mention the state $q \in Q_{\mathrm{nd}}$. Hence, we can remove the pair $\langle p, q\rangle$ from $\rho(v)$ and still obtain a valid run. A contradiction to the minimality of ρ.

Hence, we have $p \in Q_{\mathrm{nd}}$. Since $\mathcal{A}$ is partiall non-deterministic, it follows that there exists some $s \simeq_{\mathrm{sh}} \mathrm{suc}(u)$ with

- $s \vDash \delta(p, t(u))$,
- $|s(d)| = 1$ or $s(d) \subseteq Q_{\mathrm{alt}}$,
- $q' \in s(d) \Rightarrow \langle p, q'\rangle \in \rho(\mathrm{suc}(u)(d))$.

Let d be the direction such that $v = \mathrm{suc}(u)(d)$. If $s(d) = \{q'\}$, we could remove everything from $\rho(v)$ except for the pair $\langle p, q'\rangle$. If $s(d) \subseteq Q_{\mathrm{alt}}$, we could remove the pair $\langle p, q\rangle$ from $\rho(v)$. In both cases we obtain a contradiction to the minimality of ρ. □

By definition, every partially non-deterministic L-automaton is an alternating L-automaton. Conversely, we can translate every alternating L-automaton into a partially non-deterministic one. To do so, we make use of the following consequence of the theorem of McNaughton-Pappert [10].

Proposition 6.6. *For every finite ω-semigroup $\mathfrak{S} = \langle S, S_\omega\rangle$ and every element $a \in S_\omega$, there exists a deterministic ω-automaton $\mathcal{A}$ such that*

$$[\![\mathcal{A}]\!] = \{\, w \in S^\omega \mid \pi(w) = a \,\}.$$

Proposition 6.7. *Let $T \in \{\mathrm{general}, \mathrm{pure}, \mathrm{weak}\}$ be a type of automaton, let L be a family of logics without (!) polarities for $\mathbb{S} \circ \wp$ that is closed under conjunctions, disjunctions, label restrictions, and boolean label substitutions, and let $L' \in \{L, L_{\mathrm{c}}, L_{\mathrm{d}}\}$. Suppose that L and L' satisfy one of the following conditions.*

- *$L' = L$ and L is closed under projections.*
- *$L' = L_{\mathrm{c}}$ and L is closed under finite projections.*
- *$L' = L_{\mathrm{d}}$ and L is closed under singleton projections.*

For every alternating L'-automaton $\mathcal{A}$ of type T, there exists an L'-automaton $\mathcal{B}$ of type T such that $[\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]$ and $\mathcal{B}$ is partially non-deterministic whose shape is given by the following table.

type	*general*	*pure*	*weak*
L	*all trees*	*all trees*	*well-founded trees*
L_{c}	*finitely-branching trees*	*finitely-branching trees*	*finite trees*
L_{d}	*chains*	*chains*	*finite chains*

Proof. We start with the cases where T = general or T = pure, and then we explain how to modify this proof for weak automata. Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_{\mathrm{o}}, \Omega \rangle$ be an alternating L'-automaton, possibly pure. We construct a partially non-deterministic automaton $\mathcal{B}$ whose alternating part is just $\mathcal{A}$, while the non-deterministic part guesses a run ρ of $\mathcal{A}$ and verifies that it is accepting. To do so, $\mathcal{B}$ has to check that every trace of ρ satisfies the parity condition. Unfortunately, this condition is not a parity condition itself. Hence, we first have to construct a deterministic ω-automaton $\mathcal{C}$ that reads a branch of ρ and checks all the traces along that branch. Then we simply have to run $\mathcal{C}$ along all the branches of ρ and use the states of $\mathcal{C}$ for our parity condition.

The automaton $\mathcal{C}$ is based on the ω-semigroup $\mathfrak{S} = \langle S, S_\omega \rangle$ with domains

$$S := \wp(Q \times Q) \quad \text{and} \quad S_\omega := \wp(Q)\,,$$

where the product is defined as follows. For $A, B, A_{\mathrm{o}}, A_1, \ldots \in S$ and $U \in S_\omega$, we set

$$\begin{aligned} A \cdot B &:= \big\{\, \langle p, r \rangle \;\big|\; \langle p, q \rangle \in A \text{ and } \langle q, r \rangle \in B \,\big\}\,, \\ A \cdot U &:= \big\{\, p \;\big|\; \langle p, q \rangle \in A \text{ and } q \in U \,\big\}\,, \\ \prod_{i<\omega} A_i &:= \big\{\, p_{\mathrm{o}} \in Q \;\big|\; \text{there are } p_{\mathrm{o}}, p_1, \cdots \in Q \text{ with } \langle p_i, p_{i+1} \rangle \in A_i, \text{ for all } i, \\ &\qquad\qquad \text{such that } \liminf_i \Omega(p_i) \text{ is even} \,\big\}\,. \end{aligned}$$

By Proposition 6.6 there exists a deterministic ω-automaton $\mathcal{C} = \langle W, S, \eta, w_{\mathrm{o}}, \Phi \rangle$ recognising the language

$$[\![\mathcal{C}]\!] = \{\, (A_i)_i \in S^\omega \mid q_{\mathrm{o}} \in \textstyle\prod_i A_i \,\}\,.$$

We construct a partially non-deterministic automaton $\mathcal{B} := \langle Q', \Sigma, \delta', q'_{\mathrm{o}}, \Omega' \rangle$ as follows. For the alternating part, we use the original states of $\mathcal{A}$, while the non-deterministic part uses states in $W \times S$. For technical reasons, when switching from non-deterministic to alternating mode, the automaton goes through a state in $Q \times Q$ (the first component contains the previous state and the second component

the current one). Hence, we set

$$Q'_{\mathrm{alt}} := Q + Q \times Q \quad \text{and} \quad Q'_{\mathrm{nd}} := W \times S\,.$$

Initial state and priority function are given by

$$q'_{\mathrm{o}} := \big\langle w_{\mathrm{o}},\, \{\langle q_{\mathrm{o}}, q_{\mathrm{o}}\rangle\}\big\rangle\,,$$
$$\Omega'(p) := \begin{cases} \Omega(p) & \text{if } p \in Q\,, \\ \Omega(q') & \text{if } p = \langle p', q'\rangle \in Q \times Q\,, \\ \Phi(w) & \text{if } p = \langle w, A\rangle \in W \times S\,, \end{cases}$$

Finally, the transition function is defined as follows. For the alternating part, we set

$$\delta'(p, a) := \delta(p, a)\,, \qquad \text{for } p \in Q\,,$$
$$\delta'(\langle p, q\rangle, a) := \delta(q, a)\,, \qquad \text{for } \langle p, q\rangle \in Q \times Q\,.$$

To define the transition function for the non-deterministic part, we have to deal with the problem that a (non-economical) run might assign several non-deterministic states to the same vertex. What we do in this case is to guess one of these states and ignore the others. Hence, given a labelling $s \in \mathbb{S}\wp(Q')$, we use an inverse projection to guess a labelling $s' \in \mathbb{S}\Delta$ where

$$\Delta := \Big\{\, \langle P, P^\dagger\rangle \in \wp(Q') \times \wp(Q'_{\mathrm{nd}}) \;\Big|\; P^\dagger \subseteq P,\ |P^\dagger| \leq 1 \,\Big\}\,,$$

and then we ignore all non-deterministic states that do not belong to P_{o}. To distinguish given states in $p \in P \cap Q'_{\mathrm{nd}}$ from the guessed ones in $P^\dagger$, we denote the latter states by $p^\dagger$. Hence, the formula $p^\dagger$ checks whether $p \in P^\dagger$ while p checks whether $p \in P$. This leads to the transition formula

$$\delta'(\langle w, A\rangle, a) := \exists^{\mathcal{P}}_{\wp(Q'),\varnothing}\Big[\chi_\Delta \wedge \bigwedge_{\langle p', p\rangle \in A} \hat{\delta}_p\Big]$$

where $\chi_\Delta \in L[\wp(Q') \times \wp(Q'_{\mathrm{nd}})]$ is a formula stating that all labels belong Δ (which

exists since L is closed under label restrictions), and

$$\mathcal{P} := \begin{cases} \text{all sets} & \text{if } L' = L\,, \\ \text{finite sets} & \text{if } L' = L_{\mathrm{c}}\,, \\ \text{singletons} & \text{if } L' = L_{\mathrm{d}}\,, \end{cases}$$

$$\hat{\delta}_p := \delta(p,a)\Big[q \mapsto (\vartheta \wedge \langle p,q\rangle^\dagger) \vee \bigvee_{B \ni \langle p,q\rangle} \langle \eta(w,A),B\rangle^\dagger\Big]_{q\in Q}\,,$$

$$\vartheta := \bigwedge_{q\in Q'_{\mathrm{nd}}} \neg q^\dagger\,.$$

($\delta(p,a)[q \mapsto \vartheta_q]_{q\in Q}$ is the formula from Lemma 6.3.)

The automaton $\mathcal{B}$ is a well-formed L'-automaton since every non-deterministic state occurs positively in the formula $\delta'(\langle w,A\rangle,a)$ and

$$\delta'(\langle w,A\rangle,a) \in L'[Q',Q'_{\mathrm{nd}},\varnothing]\,.$$

(We can remove all non-deterministic states that are different from the guessed ones. And there are only finitely many guessed states if $L' = L_{\mathrm{c}}$, and only a single one if $L' = L_{\mathrm{d}}$.) Furthermore it follows that, if $\mathcal{A}$ is pure, then so is $\mathcal{B}$.

Let us check that $\mathcal{B}$ is partially non-deterministic with the desired shape and with alternating states $Q'_{\mathrm{alt}} = Q + Q\times Q$ and non-deterministic states $Q'_{\mathrm{nd}} = W\times S$. To check that $\mathcal{B}$ is partially non-deterministic, suppose that

$$s \vDash \delta'(q,a)\,, \quad \text{for } q \in Q'_{\mathrm{nd}} \text{ and } a \in \Sigma\,.$$

By definition of the transition formula for non-deterministic states, it follows that there is some $s_{\mathrm{o}} \le s$ such that

$$s_{\mathrm{o}} \vDash \delta'(q,a) \text{ and, for all } d \in \mathrm{dom}(s),\ |s_{\mathrm{o}}(d)| = 1 \text{ or } s_{\mathrm{o}}(d) \subseteq Q'_{\mathrm{alt}}\,.$$

The fact that the shape of $\mathcal{B}$ is either the class of all trees, of all finitely-branching ones, or of all chains, follows by choice of $\mathcal{P}$ in the transition formula above.

It remains to prove that $[\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]$. First, note that we have

$$s \vDash \delta'(\langle w,A\rangle,a)$$

if, and only if, there exists some $s_{\mathrm{o}} \le s$ such that

- $|s_{\mathrm{o}}(d)| = 1$ or $s_{\mathrm{o}}(d) \subseteq Q\times Q\,,\quad$ for all $d \in \mathrm{dom}(s)\,,$

- $\{\, d \in \mathrm{dom}(s) \mid s_{\mathrm{o}}(d) \in Q_{\mathrm{nd}} \,\} \in \mathcal{P}$
- $\mathbb{S}\tau_p(s_{\mathrm{o}}) \vDash \delta(p,a)\,, \quad$ for all $\langle p', p\rangle \in A$, where

$$\tau_p(P) := \begin{cases} f_p[P] & \text{if } P \subseteq Q \times Q\,, \\ \{\, q \in Q \mid \langle w', B\rangle \in P,\ w' = \eta(w,A) \text{ and } \langle p, q\rangle \in B \,\} & \\ & \text{if } P \subseteq W \times S\,. \end{cases}$$

and $f_p(B) := \{\, q \in Q \mid \langle p, q\rangle \in B \,\}$.

It follows that s is a model of $\delta'(\langle w, A\rangle, a)$ if, and only if, there is some $s' \in \mathbb{S}[\wp(Q \times Q)]$ such that

$$s'(d) \subseteq s(d) \cap Q_{\mathrm{alt}} \quad \text{or} \quad \langle \eta(w,A), s'(d)\rangle \in s(d)\,,$$

and

$$\mathbb{S}f_p(s') \vDash \delta(p,a)\,, \quad \text{for all } \langle p', p\rangle \in A\,,$$

where $f_p : Q \times Q \to Q$ is defined as above.

($\subseteq$) Let $\rho' : \mathrm{dom}(t) \to \wp(Q' \times Q')$ be an accepting economical run of $\mathcal{B}$ on some tree t. We define a function $\rho : \mathrm{dom}(t) \to \wp(Q \times Q)$ by

$$\rho(v) := \begin{cases} \rho'(v) & \text{if } \rho'(v) \subseteq Q \times Q\,, \\ \big\{\, \langle p', q\rangle \;\big|\; \langle\langle p, p'\rangle, q\rangle \in \rho'(v) \,\big\} & \text{if } \rho'(v) \subseteq (Q \times Q) \times Q\,, \\ B & \text{if } \rho'(v) = \{\langle w, A\rangle\} \times B \text{ for some } B \subseteq Q \times Q\,, \\ B & \text{if } \rho'(v) = \{\langle w, A, u, B\rangle\} \subseteq (W \times S) \times (W \times S)\,, \end{cases}$$

By the above remark, it follows that ρ is an accepting run of $\mathcal{A}$ on t.

($\supseteq$) Let $\rho : \mathrm{dom}(t) \to \wp(Q \times Q)$ be an accepting run of $\mathcal{A}$ on some tree t. W.l.o.g. we may assume that, for every vertex $v \in \mathrm{dom}(t)$ with predecessor u and every pair $\langle q, q'\rangle \in \rho(v)$, there is some $\langle p, p'\rangle \in \rho(u)$ with $p' = q$. (Otherwise, we can remove from $\rho(v)$ all pairs $\langle q, q'\rangle$ not satisfying this condition.) Given a prefix-closed subset $P \subseteq \mathrm{dom}(t)$, we will construct a run $\rho' : \mathrm{dom}(t) \to \wp(Q' \times Q')$ whose non-deterministic part is P. Let $\sigma : \mathrm{dom}(t) \to W$ be the function with

$$\sigma(v) := \begin{cases} w_{\mathrm{o}} & \text{if } v = \langle\rangle \text{ is the root,} \\ \eta(\sigma(u), \rho(u)) & \text{if } v \text{ has a predecessor } u\,. \end{cases}$$

We define $\rho' : \mathrm{dom}(t) \to \wp(Q' \times Q')$ as follows. For a vertex $v \in \mathrm{dom}(t)$ with predecessor u, we set

$$\rho'(v) := \begin{cases} \{\langle \sigma(u), \rho(u), \sigma(v), \rho(v)\rangle\} & \text{if } v \in P\,, \\ \{\langle \sigma(u), \rho(u)\rangle\} \times \rho(v) & \text{if } v \notin P \text{ but } u \in P\,, \\ \rho(v) & \text{if } u \notin P\,. \end{cases}$$

By the above remark, it follows that ρ' is an accpeting economical run of $\mathcal{B}$ on t whose non-deterministic part is equal to P.

It remains to consider the weak cases. Let $\mathcal{B}'$ be the automaton obtained from the automaton $\mathcal{B}$ constructed above by increasing the priorities of all states in Q'_{at} by 2 and setting the priorities of all states in Q'_{nd} to 1. Since no accpeting run of $\mathcal{B}'$ has an infinite branch with non-deterministic states only, it follows that the shape of $\mathcal{B}'$ consists of all well-founded trees in the shape of $\mathcal{B}$. Furthermore, the automaton $\mathcal{B}'$ is weak since, as we remarked above,

$$\delta'(\langle w, A\rangle, a) \in L'[Q', Q'_{\mathrm{nd}}, \varnothing]\,. \qquad \square$$

Theorem 6.8. *Let $T \in \{\mathrm{general}, \mathrm{pure}, \mathrm{weak}\}$ be a type of automaton, let L be a family of logics without (!) polarities for $\mathbb{S} \circ \wp$ that is closed under conjunctions, disjunctions, label restrictions, and boolean label substitutions, and let $L' \in \{L, L_{\mathrm{c}}, L_{\mathrm{d}}\}$. Suppose that L and L' satisfy one of the following conditions.*

- *$L' = L$ and L is closed under projections.*
- *$L' = L_{\mathrm{c}}$ and L is closed under finite projections.*
- *$L' = L_{\mathrm{d}}$ and L is closed under singleton projections.*

The class of languages recognised by L'-automata of type T is closed under $\mathcal{P}$-projections where $\mathcal{P}$ is the class of trees from the following table.

type	*general*	*pure*	*weak*
L	*all trees*	*all trees*	*well-founded trees*
L_{c}	*finitely-branching trees*	*finitely-branching trees*	*finite trees*
L_{d}	*chains*	*chains*	*finite chains*

Proof. Let $K \subseteq \mathbb{T}_{\mathbb{S}}[\Sigma \times \Gamma]$ be a language of the form $K = [\![\mathcal{A}]\!]$ for some L'-automaton $\mathcal{A}$ of type T and fix a distinguished element $\gamma_{\mathrm{o}} \in \Gamma$. By Proposition 6.7, there exists a partially non-deterministic L'-automaton $\mathcal{B} = \langle Q, \Sigma \times \Gamma, \delta, q_{\mathrm{o}}, \Omega\rangle$

such that $[\![\mathcal{B}]\!] = K$ and the shape of $\mathcal{B}$ is equal to $\mathcal{P}$. Let $\mathcal{C}$ be the automaton obtained from $\mathcal{B}$ by changing the transition relation to

$$\delta'(q,a) := \begin{cases} \delta(q,\langle a,\gamma_o\rangle) & \text{if } q \in Q_{\mathrm{alt}}\,, \\ \bigvee_{\gamma\in\Gamma} \delta(q,\langle a,\gamma\rangle) & \text{if } q \in Q_{\mathrm{nd}}\,, \end{cases} \qquad \text{for } q \in Q \text{ and } a \in \Sigma\,.$$

We claim that $[\![\mathcal{C}]\!] = \mathrm{pr}^{\mathcal{P}}_{\Sigma,\gamma_o}[K]$.

($\supseteq$) Let $t = \mathrm{pr}^{\mathcal{P}}_{\Gamma,\gamma_o}(s)$, for some $s \in K$. By assumption, there exists an accepting economical run ρ of $\mathcal{B}$ on s such that the set

$$P := \{\, v \in \mathrm{dom}(s) \mid s(v) \notin \Sigma \times \{\gamma_o\} \,\}$$

is included in the non-deterministic part of ρ. Then ρ is also an accepting run of $\mathcal{C}$ on t.

($\subseteq$) Let ρ be an accepting economical run of $\mathcal{C}$ on t and let P be its non-deterministic part. Since the shape of $\mathcal{C}$ is equal to the shape of $\mathcal{B}$, we have $P \in \mathcal{P}$. By definition of δ' we can choose, for every $v \in P$, some element $c_v \in \Gamma$ such that

$$s_{v,p} \vDash \delta(p,\langle t(v), c_v\rangle)\,, \quad \text{for all } v \in P \text{ with } \rho(v) = \{\langle q,p\rangle\}\,,$$

where $s_{v,p} \in \mathbb{S}\wp(Q)$ is the successor structure obtained from ρ as in Definition 5.1. We define a tree $s \in \mathbb{T}_{\mathbb{S}}(\Sigma\times\Gamma)$ by

$$s(v) := \begin{cases} \langle t(v),\gamma_o\rangle & \text{if } v \notin P\,, \\ \langle t(v), c_v\rangle & \text{if } v \in P\,. \end{cases}$$

Then $t = \mathrm{pr}_{\Sigma,\gamma_o}(s)$ and ρ is an accepting run of $\mathcal{B}$ on s. Hence, $s \in K$ and $t \in \mathrm{pr}_{\Sigma,\gamma_o}[K]$. □

7 Monadic Second-Order Logic

In this section, we use the machinery we have set up to derive characterisations of certain variants of monadic second-order logic. Let us start by introducing the logics we will work with.

Definition 7.1. Let Q be an alphabet and Σ a signature.

(a) The logic $\mathrm{E}_\infty[Q]$ has formulae that are boolean combination of statements of the form $\mathsf{E}_k\varphi$ where $k < \omega$ or $k = \infty$, and φ is a boolean combination of

elements of Q. Such a formula holds in a structure $s \in \mathbb{S}\mathcal{P}(Q)$ if, and only if, there are at least k elements $v \in \mathrm{dom}(s)$ such that $s(v)$ satisfies φ.

(b) We denote by $\mathrm{E}_\omega[Q]$ the fragment of $\mathrm{E}_\infty[Q]$ that only uses the quantifier E_k with $k < \omega$. Similarly, $\mathrm{E}_1[Q]$ is the fragment of $\mathrm{E}_\infty[Q]$ that only uses the quantifier E_k with $k = 1$. Finally, we denote by $\mathrm{C}[Q]$ the extension of $\mathrm{E}_\infty[Q]$ by statements of the form $\mathsf{C}_{k,m}\varphi$ stating that the number of elements satisfying φ is finite and congruent k modulo m.

(c) For a relational signature Σ, we denote by $\mathbb{S}_\Sigma$ the functor mapping a set Q to the class of all structures of signature $\Sigma + \{ P_q \mid q \in Q \}$ where the (unary) predicates P_q form a partition of the universe.

Let $\mathfrak{A}$ be a structure over a signature of the form $\Sigma + \{E\} + \{ P_q \mid q \in Q \}$ (where E is binary) such that the predicates P_q form a partition of the universe A. We identify $\mathfrak{A}$ with with the Q-labelled $\mathbb{S}_\Sigma$-enriched transition system $\langle A, \mathrm{suc}, \lambda \rangle$ where

$$\lambda(a) := q \quad : \text{iff} \quad a \in P_q\,, \quad \text{for } a \in A \text{ and } q \in Q\,,$$

and $\mathrm{suc}(a)$ is the $(\Sigma + \{ P_q \mid q \in Q \})$-reduct of the substructure of $\mathfrak{A}$ induced by the set

$$\{ b \in A \mid \langle a, b \rangle \in E \}\,.$$

(d) Let $\mathcal{P}$ be a property of sets. We denote by $\mathrm{MSO}_{\mathcal{P}}[\Sigma; Q]$ the version of monadic second-order logic where quantification is restricted to sets included in some set satisfying $\mathcal{P}$, and where the models are structures with the signature $\Sigma + \{E\} + \{ P_q \mid q \in Q \}$. For particular choices of $\mathcal{P}$ we obtain the following variants.

logic	name	$\mathcal{P}$
MSO	monadic second-order logic	all sets
WMSO	weak monadic second-order logic	finite sets
CL	chain logic	chains
WCL	weak chain logic	finite chains
$\mathrm{MSO}_{\mathrm{wf}}$	—	well-founded trees
$\mathrm{MSO}_{\mathrm{fb}}$	—	finitely-branching trees

(e) *Guarded second-order logic* GSO is a variant of full second-order logic where all second-order quantifiers are restricted to range over *guarded* relations only. A relation R is called *guarded* if every tuple $\bar{a} \in R$ is included (as a set) in some

tuple $\bar{c}$ from a relation of the given structure. (We consider equality as a binary relation in this context, which implies that every unary relation is guarded.)

(f) Finally, the *counting* variants of MSO and GSO are the extensions of the respective logic by predicates of the form

$$|X| < \infty \wedge |X| \equiv k \pmod{m}, \quad \text{for } k < m < \omega .$$

We denote these two logics by CMSO and CGSO. ⌟

We use a variant of $\mathrm{MSO}_{\mathcal{P}}[\Sigma; Q]$ without first-order variables where the atomic formulae are of the form

$$X \subseteq Y \quad \text{and} \quad RZ_0 \dots Z_{k-1},$$

for relation symbols $R \in \Sigma + \{ P_q \mid q \in Q \}$ and monadic variables $X, Y, Z_0, \dots, Z_{k-1}$. A formula of the form $R\bar{Z}$ holds if there are elements $v_i \in Z_i$ such that the tuple $\bar{v}$ belongs to the relation R. It is straightforward to inductively translate every MSO-formula into one of this special form (see, e.g., [14, 1]).

We will prove below that every $\mathrm{MSO}_{\mathcal{P}}$-formula is equivalent to a formula from a suitable fixed-point logic. For the inductive proof, we will have to deal with $\mathrm{MSO}_{\mathcal{P}}$-formulae $\varphi(\bar{X})$ with free monadic variables $\bar{X}$. In order to make the values of these variables accessible to the fixed-point formula we annotate each vertex v of the given tree with the set of variables it belongs to. Thus, given a tree $t \in \mathbb{T}_{\mathbb{S}} Q$ and values $P_0, \dots, P_{n-1} \subseteq \mathrm{dom}(t)$ for $\bar{X}$, we construct the tree $t_{\bar{P}} \in \mathbb{T}_{\mathbb{S}}(\wp(Q) \times \wp(\bar{X}))$ with $\mathrm{dom}(t_{\bar{P}}) = \mathrm{dom}(t)$ and labelling

$$t_{\bar{P}}(v) := \langle t(v), U_v \rangle \quad \text{where} \quad U_v := \{ X_i \mid v \in P_i \} .$$

The base case of the induction is given by the following lemma.

Lemma 7.2. *For every atomic* $\mathrm{MSO}[\Sigma; Q]$*-formula* $\varphi(\bar{X})$*, there exists an* $\mu_{\mathrm{af}}\mathrm{FO}[Q \times \wp(\bar{X})]$*-formula* ψ *such that*

$$t \vDash \varphi(\bar{P}) \quad \text{iff} \quad t|_{\bar{P}} \vDash \psi, \quad \textit{for all } t \textit{ and } \bar{P} .$$

Proof. For the labelling of the tree we use the predicates P_q and P_X, for $q \in Q$ and monadic variables X. Furthermore, we use the predicates P'_ϑ, for $\vartheta \in \mu_{\mathrm{af}}\mathrm{FO}$, to check inside of a modal operator $\bigcirc$ whether the corresponding successor satisfies the μ_{af}FO-formula ϑ.

If $\varphi = (X \subseteq Y)$, we use the formula

$$\psi := \nu x.\big[(P_X \to P_Y) \wedge \bigcirc[\forall d.P'_x d]\big].$$

If $\varphi = P_q X$, for $q \in Q$, we define

$$\psi := \mu x.\big[\bigcirc[\exists d.P'_x d] \vee (P_X \wedge P_q)\big].$$

If $\varphi = (X \preceq Y)$ where $\preceq$ is the tree order on the vertices, we set

$$\psi := \mu x.\big[\bigcirc[\exists d.P'_x d] \vee \big[P_X \wedge \mu y.(\bigcirc[\exists d.P'_y d] \vee P_Y)\big]\big].$$

Finally, if $\varphi = R\bar{Z}$ where R is one of the relations from the successor structures, we define

$$\psi := \mu x.\big[\bigcirc[\exists z.P'_x z] \vee \bigcirc[\exists \bar{d}.(R\bar{d} \wedge \textstyle\bigwedge_i P'_{P_{Z_i}} d_i)]\big]. \qquad \square$$

Lemma 7.3. *Let $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, \Omega\rangle$ be an automaton.*

(a) *If $\mathcal{A}$ is a weak* MSO*-automaton, the language $[\![\mathcal{A}]\!]$ is $\mathrm{MSO}_{\mathrm{wf}}$-definable.*
(b) *If $\mathcal{A}$ is a pure $\mathrm{WMSO}_{\mathrm{c}}$-automaton, the language $[\![\mathcal{A}]\!]$ is $\mathrm{MSO}_{\mathrm{fb}}$-definable.*
(c) *If $\mathcal{A}$ is a weak $\mathrm{WMSO}_{\mathrm{c}}$-automaton, the language $[\![\mathcal{A}]\!]$ is* WMSO*-definable.*
(d) *If $\mathcal{A}$ is a pure FO_{d}-automaton, the language $[\![\mathcal{A}]\!]$ is* CL*-definable.*
(e) *If $\mathcal{A}$ is a weak FO_{d}-automaton, the language $[\![\mathcal{A}]\!]$ is* WCL*-definable.*

Proof. For each state q of $\mathcal{A}$, we will construct a formula φ_q of the respective logic stating that the given tree has an accepting run which starts in the state q. We proceed by induction on the number of states reachable from q. Let C be the connected component of $\mathcal{A}$ containing q and let D be the set of all states reachable from q that do not belong to C. We distinguish two cases.

First suppose that $\delta(q, a) \in L[C, \varnothing]$, where L is the transition logic in question. We claim that, if there exists an accepting run ρ with initial state q, we can choose ρ such that the set

$$P := \{\, v \mid \rho(v) \cap C \times C \neq \varnothing \,\}$$

forms (a) a well-founded tree, (b) a finitely branching tree, (c) a finite tree, (d) a chain, or (e) a finite chain, respectively. Then we obtain the desired formula φ_q of the respective logic by stating that

- there exists a set P and a family $(Z_{p',p})_{p,p' \in C}$ of sets (of the kind allowed by our logic) encoding the restriction of ρ to P,
- the root satisfies $Z_{q,q}$,
- for every infinite branch β that is contained entirely in P, every trace along β satisfies the parity condition,
- the formula

$$(\forall v \in P) \bigwedge_{p,p' \in C} [Z_{p',p} v \to \hat{\delta}_p(v)]$$

  holds where $\hat{\delta}_p(v)$ is the formula obtained from $\delta(p, t(v))$ by (I) relativising the formula to the set of successors of v; (II) replacing all states $r \in C$ by the formula $Z_{p,r}$; and (III) replacing all states $r \in D$ by the formula φ_r (which exists by inductive hypothesis).

It therefore remains to prove the above claim. Fix an accepting run ρ of $\mathcal{A}$ on the given input tree. We construct an accepting run ρ_o that has the above property. We choose $\rho_o(v) \subseteq \rho(v)$ by induction on the distance of v from the root.

We call an entry $\langle p', p \rangle \in \rho(v)$ *unreachable* if either v is the root and $p \neq q$, or if v has a predecessor u and there is no $\langle r', r \rangle \in \rho(u)$ with $r = p'$. Clearly, (recursively) removing all unreachable entries from an accepting run results again in an accepting run. Furthermore note that, if ρ is a run without unreachable entries, the corresponding set P is prefix-closed.

(a) If $\mathcal{A}$ is weak, $\delta(q, a) \in L[C, \varnothing]$ implies that $\Omega(q)$ is odd. Let ρ_o be the run obtained from ρ by recursively removing all unreachable entries. Then the associated set P does not contain any infinite branches. Hence, it forms a well-formed tree.

(b) Suppose that $\mathcal{A}$ is a pure $\mathrm{MSO_c}$-automaton. For every vertex v and every entry $\langle p', p \rangle \in \rho(v)$, it follows that there exists a finite set $U_{v,p}$ of successors such that the truth value of the transition formula $\delta(p, t(v))$ does not change when we remove all entries of the form $\langle p, r \rangle$ from $\rho(w)$, for successors w of v that do not belong to $U_{v,p}$. Let ρ_o be the resulting run and P the corresponding set. It follows that every vertex $v \in P$ has only finitely many successors in P (those in $\bigcup_{p \in C} U_{v,p}$). Hence, P forms a finitely-branching tree.

(c) Combining the arguments in (a) and (b), we obtain a finite set P.

(d) Given an accpeting run ρ, we will construct an accpeting run ρ_o and a path $(v_i)_i$ such that each set $\rho_o(v_i)$ contains a single entry from $C \times C$ while $\rho_o(u)$ is disjoint from $C \times C$, for all vertices u that do not lie on the path.

We start with the root v_0 and $\rho_0(v_0) := \{\langle q, q\rangle\}$. For the inductive step, suppose that we have already defined v_i and $\rho_0(v_i)$. Let $\langle p', p\rangle \in \rho_0(v_i)$ be the unique entry from $C \times C$. Since the transition formula $\delta(p, t(v_i))$ belongs to $\mathrm{MSO_d}[C, \varnothing]$, we can find some successor u of v_i and some state $r \in C$ such that the truth value of $\delta(p, t(v_i))$ does not change when we remove all the states from C from every successer, except for the state r at u. If $\langle p, r\rangle \in \rho(u)$, we set

$$v_{i+1} := u \quad \text{and} \quad \rho_0(v_{i+1}) := \big(\rho(v_{i+1}) \setminus (C \times C)\big) \cup \{\langle p, r\rangle\}\,.$$

Otherwise, we stop the construction of the path at the vertex v_i. Finally, we set

$$\rho_0(u) := \rho(u) \setminus (C \times C)\,, \quad \text{for all vertices } u \text{ not on the path.}$$

The run obtained from ρ_0 by removing all unreachable entries has the desired properties.

(e) Combining the arguments in (a) and (d), we obtain a finite chain P.

It remains to consider the case where $\delta(q, a) \in L[\varnothing, C]$. Let $\mathcal{A}^{\mathrm{op}}$ be the automaton for the complement. Recall that $\mathcal{A}^{\mathrm{op}}$ has the same states as $\mathcal{A}$, but their priorities are increased by 1, and the transition formulae are the duals of the formulae from $\mathcal{A}$. Furthermore, a tree is accepted by $\mathcal{A}^{\mathrm{op}}$ with initial state q if, and only if, the tree is rejected by $\mathcal{A}$ with initial state q. In particular, C is still a component of $\mathcal{A}^{\mathrm{op}}$ and we have $\delta^{\mathrm{op}}(q, a) \in L[C, \varnothing]$ for $q \in C$. Consequently, we can use the above case to find a formula φ_q that defines the set of all trees that are rejected by $\mathcal{A}$ when starting in state q. The negation $\neg\varphi_q$ is the desired formula. □

The first characterisation is basically due to Walukiewicz [16].

Theorem 7.4 (Walukiewicz [16]). *For a language $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$, the following statements are equivalent.*

(1) *K is* MSO*-definable.*

(2) *K is μ_{p}*MSO*-definable.*

(3) *K is recognised by a pure* MSO*-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is $\mu_{\mathrm{p}}\mathrm{E}_\omega$-definable.*

(5) *K is recognised by a pure E_ω-automaton.*

Furthermore, all translations between the above formalisms are effective.

Proof. (3) ⇔ (2) follows by Theorem 5.3.

(2) ⇒ (1) It is straightforward to inductively translate every μMSO-formula into MSO.

(3) ⇒ (5) One can use a standard back-and-forth argument [5, 1] to prove that, over structures with an empty signature, every MSO-formula is equivalent to an E_ω-formula.

(5) ⇒ (3) is trivial and (4) ⇔ (5) follows by Theorem 5.3.

(1) ⇒ (3) By induction on a given MSO-formula φ, we construct an equivalent automaton $\mathcal{A}$. If φ is atomic, the existence of $\mathcal{A}$ follows by Lemma 7.2 and the already established implication (2) ⇒ (3). If φ is a boolean combination of MSO-formulae, the claim follows by inductive hypothesis, the equivalence (2) ⇔ (3), and the fact that the logic μ_pMSO is closed under boolean combinations. Finally, if $\varphi = \exists X \psi$, the claim follows by inductive hypothesis and Theorem 6.8. □

We obtain analogous results for CMSO, GSO, and CGSO. The proofs are the same as that of Theorem 7.4, except that for counting logics, we also have to construct automata for predicates of the form

$$|X| < \infty \wedge |X| \equiv k \pmod{m}.$$

in the implication (1) ⇒ (3).

Theorem 7.5. *For a language $K \subseteq \mathbb{T}_\mathbb{S}\Sigma$, the following statements are equivalent.*

(1) *K is* CMSO*-definable.*

(2) *K is μ_p*CMSO*-definable.*

(3) *K is recognised by a pure* CMSO*-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is μ_p*C*-definable.*

(5) *K is recognised by a pure* C*-automaton.*

Furthermore, all translations between the above formalisms are effective.

Theorem 7.6. *For a language $K \subseteq \mathbb{T}_\mathbb{S}\Sigma$, the following statements are equivalent.*

(1) *K is* GSO*-definable.*

(2) *K is μ_p*GSO*-definable.*

(3) *K is recognised by a pure* GSO*-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is* $\mu_{\mathrm{p}}\mathrm{E}_{\omega}$*-definable.*

(5) *K is recognised by a pure* E_{ω}*-automaton.*

(6) *K is* MSO*-definable.*

Furthermore, all translations between the above formalisms are effective.

Theorem 7.7. *For a language* $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$*, the following statements are equivalent.*

(1) *K is* CGSO*-definable.*

(2) *K is* μ_{p}CGSO*-definable.*

(3) *K is recognised by a pure* CGSO*-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is* μ_{p}C*-definable.*

(5) *K is recognised by a pure* C*-automaton.*

(6) *K is* CMSO*-definable.*

Furthermore, all translations between the above formalisms are effective.

For the special case of ordinary trees, the next two theorems are due to [4].

Theorem 7.8. *For a language* $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$*, the following statements are equivalent.*

(1) *K is* WMSO*-definable.*

(2) *K is* $\mu_{\mathrm{af}}\mathrm{WMSO}_{\mathrm{c}}$*-definable.*

(3) *K is recognised by a weak* $\mathrm{WMSO}_{\mathrm{c}}$*-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is* $\mu_{\mathrm{af}}(\mathrm{E}_{\infty})_{\mathrm{c}}$*-definable.*

(5) *K is recognised by a weak* $(\mathrm{E}_{\infty})_{\mathrm{c}}$*-automaton.*

Furthermore, all translations between the above formalisms are effective.

Proof. The proof is analogous to that of Theorem 7.4, except for the implication (2) ⇒ (1). Instead, we prove the implication (3) ⇒ (1), which follows by Lemma 7.3 (c). □

Theorem 7.9. *For a language* $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$*, the following statements are equivalent.*

(1) *K is* WCL*-definable.*

(2) *K is* $\mu_{\mathrm{af}}\mathrm{FO}_{\mathrm{d}}$*-definable.*

(3) *K is recognised by a weak* FO_{d}*-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is $\mu_{\mathrm{af}}(\mathrm{E}_\omega)_{\mathrm{d}}$-definable.*

(5) *K is recognised by a weak $(\mathrm{E}_\omega)_{\mathrm{d}}$-automaton.*

Furthermore, all translations between the above formalisms are effective.

Proof. Again the proof is analogous to that of Theorem 7.4, except for the implication (2) ⇒ (1). Instead we can prove the implication (3) ⇒ (1) using Lemma 7.3 (e). □

The following results seem to be new.

Theorem 7.10. *For a language $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$, the following statements are equivalent.*

(1) *K is CL-definable.*

(2) *K is $\mu_{\mathrm{p}}\mathrm{FO}_{\mathrm{d}}$-definable.*

(3) *K is recognised by a pure FO_{d}-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is $\mu_{\mathrm{p}}(\mathrm{E}_\omega)_{\mathrm{d}}$-definable.*

(5) *K is recognised by a pure $(\mathrm{E}_\omega)_{\mathrm{d}}$-automaton.*

Furthermore, all translations between the above formalisms are effective.

Proof. Again, we replace the implication (2) ⇒ (1) by (3) ⇒ (1), which follows by Lemma 7.3 (d). □

Theorem 7.11. *For a language $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$, the following statements are equivalent.*

(1) *K is $\mathrm{MSO}_{\mathrm{fb}}$-definable.*

(2) *K is $\mu_{\mathrm{p}}\mathrm{WMSO}_{\mathrm{c}}$-definable.*

(3) *K is recognised by a pure $\mathrm{WMSO}_{\mathrm{c}}$-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is $\mu_{\mathrm{p}}(\mathrm{E}_\infty)_{\mathrm{c}}$-definable.*

(5) *K is recognised by a pure $(\mathrm{E}_\infty)_{\mathrm{c}}$-automaton.*

Furthermore, all translations between the above formalisms are effective.

Proof. Again the proof is similar to the ones above. For (3) ⇒ (1), we can use Lemma 7.3 (b). □

Theorem 7.12. *For a language $K \subseteq \mathbb{T}_{\mathbb{S}}\Sigma$, the following statements are equivalent.*

(1) *K is* $\mathrm{MSO}_{\mathrm{wf}}$*-definable.*

(2) *K is* $\mu_{\mathrm{af}}\mathrm{MSO}$*-definable.*

(3) *K is recognised by a weak* MSO*-automaton.*

For ordinary trees, the following statements are equivalent to those above.

(4) *K is* $\mu_{\mathrm{af}}\mathrm{E}_\omega$*-definable.*

(5) *K is recognised by a weak* E_ω*-automaton.*

Furthermore, all translations between the above formalisms are effective.

Proof. Again the proof is similar to the ones above. For (3) ⇒ (1), we can use Lemma 7.3 (1). □

Open Question. *Is there an automaton characterisation of* $\mathrm{MSO}^{\mathcal{P}}$ *where* $\mathcal{P}$ *is the set of thin trees?*

8 The Muchnik Iteration

As an application of the machinery developed in this article we consider the so-called Muchnik iteration which, given some Σ-structure $\mathfrak{A}$, creates an infinite tree of copies of $\mathfrak{A}$. It can be seen as a generalisation of the unravelling operation to arbitrary Σ-structures.

Definition 8.1. Let $\mathfrak{A} = \langle A, \bar{R} \rangle$ be a Σ-structure.

(a) The *Muchnik iteration* of $\mathfrak{A}$ is the $(\Sigma + \{E, \mathrm{cl}\})$-structure $\mathfrak{A}^* = \langle A^*, E, \mathrm{cl}, \bar{R}^* \rangle$ whose universe consists of all finite sequence over A and

$$\begin{aligned} R_i^* &:= \{ \langle wa_0, \dots, wa_{n-1} \rangle \mid w \in A^*,\ \bar{a} \in R \}, \\ E &:= \{ \langle w, wa \rangle \mid w \in A^*,\ a \in A \}, \\ \mathrm{cl} &:= \{ waa \mid w \in A^*,\ a \in A \}. \end{aligned}$$

We call cl the *clone predicate.*

(b) We regard $\mathfrak{A}^*$ as an $\mathbb{S}$-enriched transition system for the functor $\mathbb{S}X := X^A$ by choosing for $\mathrm{suc}(w)$ (with $w \in A^*$) the substructure of $\mathfrak{A}^*$ induced by the set

$$\{ wa \mid a \in A \}.$$ ⌟

The following theorem, originally due to Muchnik, is one of the strongest decidability results for MSO known. The theorem was announced in [13], but

the original proof has never been published. Walukiewicz [16] introduced MSO-automata to give a new independent proof of this theorem. Here we present a new, much simplified proof that also applies to several other logics.

Theorem 8.2 (Muchnik, Walukiewicz [16])**.** *Given an* MSO*-formula φ, we can compute an* MSO*-formula φ^* such that*

$$\mathfrak{A}^* \models \varphi \quad \text{iff} \quad \mathfrak{A} \models \varphi^*, \quad \textit{for all structures } \mathfrak{A}.$$

The proof of this theorem is split into two lemmas. Let us start with a bit of terminology.

Definition 8.3. (a) A *system* of logics is a functor L mapping each finite relational signature Σ to a logic $L[\Sigma]$ whose class of models is the class of all Σ-structures.

(b) Given such a system L and a signature Σ, we construct a family of logics L_Σ by

$$L_\Sigma[Q] := L[\Sigma + \{ P_q \mid q \in Q \}], \quad \text{for every set} Q,$$

where the P_q are unary predicates. ⌟

Remark. (a) I apologise for any confusion caused by defining both families of logics and systems of logics, but I was simply not able to come up with better terminology.

(b) Clearly, all of the classical logics like first-order logic, monadic second-order logic, etc. are systems of logics. ⌟

Next, let us extend the logic μL by a mechanism for *loop-detection.*

Definition 8.4. Let L be a family of logics with polarities over $\mathbb{S} \circ \wp$.

(a) We denote by $\mu^{\circlearrowleft} L$ the following variant of μL. Given a set Σ of labels and two disjoint sets X, Y of fixed-point variables, we define the syntax and semantics of $\mu^{\circlearrowleft} L[\Sigma; X, Y]$ using the same rules for as $\mu L[\Sigma; X, Y]$, except the one for the modal operators, which is modified as follows. In the syntax, we allow an additional label $* \in 1$.

- Let $\Theta \subseteq \mu L[\Sigma; X, Y]$ be a finite set of formulae and let Θ_+ be the set of all $\vartheta \in \Theta$ containing a symbol from X and Θ_- the set of all $\vartheta \in \Theta$ containing a symbol from Y. For every $\varphi \in L[\Theta + 1, \Theta_+, \Theta_-]$, we have $\bigcirc\varphi \in \mu L[\Sigma; X, Y]$.

The label $*$ is used to mark loops, that is, if the vertex v is a successor of itself, we label it by $*$. Formally, we set

$$[\![\bigcirc\psi]\!]_{\bar{P}} := \{\, v \in S \mid \mathbb{S}f_v(\mathrm{suc}(v)) \vDash \psi \,\}$$

where the function $f_v : S \to \wp(\Theta)$ maps $u \in S$ to

$$\{\, \vartheta \in \Theta \mid u \in [\![\vartheta]\!]_{\bar{P}} \,\} \cup \begin{cases} 1 & \text{if } u = v\,, \\ \varnothing & \text{otherwise}\,, \end{cases}$$

(b) We denote by $\mu_{\mathrm{p}}^{\circlearrowleft} L$ and $\mu_{\mathrm{af}}^{\circlearrowleft} L$ the corresponding *pure* and *alternation-free* fragments of $\mu^{\circlearrowleft} L$. ⌟

Definition 8.5. Given a Σ-structure $\mathfrak{A}$, we denote by $\widehat{\mathfrak{A}}$ the $(\Sigma + \{E\})$-structure $\langle \mathfrak{A} + 1, E\rangle$ with edge relation $E := (A + 1) \times A$. ⌟

Note that the unravelling of $\widehat{\mathfrak{A}}$ coincides with $\mathfrak{A}^*$, except that it does not have the clone predicate.

The first step in the proof of Theorem 8.2 consists in proving a variant for the logic $\mu^{\circlearrowleft} L$.

Lemma 8.6. *Let L be a system of logics with polarities over $\mathbb{S} \circ \wp$. For every μL-formula φ, there exists a $\mu^{\circlearrowleft} L$-formula φ^* such that*

$$\mathfrak{A}^* \vDash \varphi \quad \text{iff} \quad \widehat{\mathfrak{A}} \vDash \varphi^*, \quad \textit{for every } \Sigma\textit{-structure } \mathfrak{A}\,.$$

Furthermore, if $\varphi \in \mu_{\mathrm{p}} L$ or $\varphi \in \mu_{\mathrm{af}} L$, we can choose $\psi \in \mu_{\mathrm{p}}^{\circlearrowleft} L$ and $\psi \in \mu_{\mathrm{af}}^{\circlearrowleft} L$, respectively.

Proof. Let $\varphi(\bar{x})$ be a μL-formula, possibly with free fixed-point variables $\bar{x}$. By induction on φ, we construct an $\mu^{\circlearrowleft} L$-formula $\varphi^*(\bar{x})$ such that

$$\mathfrak{A}^* \vDash \varphi(\rho^{-1}[\bar{P}]) \quad \text{iff} \quad \widehat{\mathfrak{A}} \vDash \varphi^*(\bar{P})\,,$$

for every Σ-structure $\mathfrak{A}$ and all sets $\bar{P}$ in $\widehat{A}$, where $\rho : A^* \to \widehat{A} = A + 1$ is the function mapping the empty word $\langle\rangle$ to the unique element $* \in 1$ and every other word to its last letter. The only two non-trivial steps in the induction is the case of modal operators and fixed-points.

First, suppose that $\varphi = \bigcirc\psi$. Fix a finite set Θ of μL-formulae such that $\psi \in L_{\Sigma+\{\mathrm{cl}\}}[\Theta]$. By inductive hypothesis, we can translate every $\vartheta \in \Theta$ into an equivalent $\mu\overset{\circlearrowleft}{} L$-formula ϑ^*. Let Θ^* be the resulting set and let ψ^* be the formula obtained from ψ by replacing each ϑ by ϑ^*. Since

$$L_{\Sigma+\{\mathrm{cl}\}}[\Theta] = L[\Sigma + \{\mathrm{cl}\} + \Theta] = L_\Sigma[\Theta + \{\mathrm{cl}\}],$$

it follows that $\varphi^* := \bigcirc\psi^* \in \mu\overset{\circlearrowleft}{} L$. Furthermore, the fact that

$$\mathfrak{A}^* \vDash \varphi(\rho^{-1}[\bar{P}]) \quad \text{iff} \quad \widehat{\mathfrak{A}} \vDash \varphi^*(\bar{P})$$

follows immediately from the inductive hypothesis.

It remains to consider a fixed-point formula $\mu y.\psi(\bar{x}, y)$. Let ψ^* be the $\mu\overset{\circlearrowleft}{} L$-formula obtained by inductive hypothesis. We set $\varphi^* := \mu y.\psi^*$. To see that φ^* has the desired properties, let $(Q_i)_i$ be the fixed-point induction of ψ on the structure $\mathfrak{A}^*$. By induction on i, it then follows that $Q_i = \rho^{-1}[P_i]$, where P_i is the i-th stage of the fixed-point induction of ψ^* on $\widehat{\mathfrak{A}}$. □

Finally, we have to translate $\mu\overset{\circlearrowleft}{} L$ back into the logics we are actually interested in.

Lemma 8.7.

(a) *Let L be* MSO *or* GSO. *For every $\mu\overset{\circlearrowleft}{} L$-formula φ, there exists an L-formula ψ such that*

$$\mathfrak{S} \vDash \varphi \quad \text{iff} \quad \mathfrak{S} \vDash \psi\,, \quad \textit{for every transition system } \mathfrak{S}\,.$$

(b) *For every $\mu_{\mathrm{af}}\overset{\circlearrowleft}{}$ WMSO_c-formula φ, there exists a* WMSO*-formula ψ such that*

$$\mathfrak{S} \vDash \varphi \quad \text{iff} \quad \mathfrak{S} \vDash \psi\,, \quad \textit{for every transition system } \mathfrak{S}\,.$$

Proof. (a) Given a $\mu\overset{\circlearrowleft}{} L$-formula $\varphi(\bar{x})$, possibly with free fixed-point variables $\bar{x}$, we inductively construct an L-formula $\psi(z, \bar{X})$ such that

$$\mathfrak{S}, s \vDash \varphi(\bar{P}) \quad \text{iff} \quad \mathfrak{S} \vDash \psi(s, \bar{P})\,,$$

for all transition systems $\mathfrak{S}$ and all sets $\bar{P}$. Most steps of the induction are trivial. For a fixed point $\mu y.\varphi(\bar{x}, y)$ we can express in L that $z \in Y$, where Y is the least set satisfying

$$\forall y[y \in Y \leftrightarrow \psi(y, \bar{X}, Y)]\,.$$

For a modal operator $\bigcirc\vartheta$, we can use the relativisation of the formula ϑ to the set U of all successors of z. The additional predicate used for loop-detection is equal to $U \cap \{z\}$, which is obviously definable.

(b) We proceed analogously to (a), the only exception being the translation of fixed-point operators. Hence, suppose that $\varphi = \mu x_0 \cdots \mu x_{n-1} \psi(\bar{x})$ where $\psi \in \mathrm{WMSO}_c[\bar{x}, \bar{x}, \varnothing]$ is continuous in $\bar{x}$. By induction on n we prove that, for every transition system $\mathfrak{S}$, there exists a finite set $U \subseteq S$ such that

$$\mathfrak{S} \models \mu x_0 \cdots \mu x_{n-1} \psi \quad \text{iff} \quad \mathfrak{S} \models \mu x_0 \cdots \mu x_{n-1} [U \wedge \psi] .$$

Then it follows that we can define the fixed-point in WMSO by stating that $z \in Y$, where $Y \subseteq U$ is the least set satisfying

$$\forall y [y \in Y \leftrightarrow y \in U \wedge \psi(y, \bar{X}, Y)] .$$

First, suppose that $n = 1$. Let $F : \wp(S) \to \wp(S)$ be the function associated with the formula ψ, let $H_i := F^i(\varnothing)$ be the i-th stage of the fixed-point induction, and let H_∞ be the limit. Since ψ is continuous in x_0, it follows that, for every $i < \omega$ and every element $u \in H_{i+1}$, there exists some finite set $W_u \subseteq H_i$ with $u \in F(W_u)$.

Let us show that this implies that $H_\infty = H_\omega$. For a contradiction, suppose otherwise. Then there exists some element $u \in H_{\omega+1} \setminus H_\omega$. Let k be the maximal number such that W_u contains some element of $H_{k+1} \setminus H_k$. Since W_u is finite, it follows that $k < \omega$. Hence, $u \in F(W_u) \subseteq F(H_k) \subseteq H_\omega$. A contradiction.

Setting

$$U_u := \{u\} \cup \bigcup_{w \in W_u} U_w ,$$

it now follows by induction on i that, for every $u \in H_i$, there is some finite set $U_u \subseteq S$ with

$$\mathfrak{S}, u \models \mu x_0 [U_u \wedge \psi] .$$

Hence, the set $U := U_v$ has the desired properties.

For the inductive step, suppose that $n > 1$. We can use the above case to find a finite set W such that

$$\mathfrak{S} \models \mu x_0 [W \wedge \mu x_1 \cdots \mu x_{n-1} \psi] .$$

Furthermore, we can use the inductive hypothesis to find, for every $w \in W$ and every finite $P \subseteq W$, some finite set $V_{P,w}$ with

$$\begin{aligned} & \mathfrak{S}, w \vDash \mu x_1 \cdots \mu x_{n-1} \psi(P, x_1, \ldots, x_{n-1}) \\ \text{iff} \quad & \mathfrak{S}, w \vDash \mu x_1 \cdots \mu x_{n-1} \big[V_{P,w} \wedge \psi(P, x_1, \ldots, x_{n-1}) \big] . \end{aligned}$$

Setting $U := \bigcup_{P,w} V_{P,w}$ it follows that

$$\mathfrak{S}, u \vDash \mu x_1 \cdots \mu x_{n-1} \big[U \wedge \psi(H_i, x_0, \ldots, x_{n-1}) \big] , \quad \text{for all } u \in W \cap H_{i+1} .$$

Consequently, we have

$$\mathfrak{S} \vDash \mu x_0 \cdots \mu x_{n-1} \big[U \wedge \psi \big] . \qquad \square$$

Combining these two lemmas we obtain the following theorem. The case L = MSO is the origninal Theorem of Muchnik, the cases for CMSO, GSO, and CGSO were proved in [3], and the case L = WMSO is new.

Theorem 8.8. *Let L be one of* MSO, CMSO, GSO, CGSO, *or* WMSO. *Given an L-formula φ, we can compute an L-formula φ^* such that*

$$\mathfrak{A}^* \vDash \varphi \quad \text{iff} \quad \mathfrak{A} \vDash \varphi^* , \quad \textit{for all structures } \mathfrak{A} .$$

Proof. We give the proof for L = MSO. The other cases are entirely analogous. We can use Theorem 7.4 to translate a given MSO-formula φ into a μMSO-formula ψ. Let ψ^* be the $\mu^{\circlearrowleft}$MSO-formula obtained from ψ via Lemma 8.6. We obtain the desired MSO-formula φ^* by applying Lemma 8.7 to ψ^*. $\square$

References

[1] A. Blumensath, *Monadic Second-Order Model Theory*. book in preparation, `https://www.fi.muni.cz/~blumens/MSO.pdf`.

[2] ———, *Algebraic Language Theory for Eilenberg–Moore Algebras*, Logical Methods in Computer Science, 17 (2021), pp. 6:1–6:60.

[3] A. Blumensath and S. Kreutzer, *An Extension to Muchnik's Theorem*, Journal of Logic and Computation, 15 (2005), pp. 59–74.

[4] F. Carreiro, *Fragments of Fixpoint Logics*, PhD Thesis, Institute for Logic, Language and Computation, Amsterdam, 2015.

[5] H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer Verlag, 1995.

[6] 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.

[7] D. Janin and I. Walukiewicz, *On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic*, in Proc. of the 7th International Conference on Concurrency Theory, CONCUR 1996, 1996, pp. 263–277.

[8] C. Kissig and Y. Venema, *Complementation of Coalgebra Automata*, in Algebra and Coalgebra in Computer Science, Third International Conference, CALCO 2009, Udine, Italy, September 7–10, 2009. Proceedings, A. Kurz, M. Lenisa, and A. Tarlecki, eds., vol. 5728 of Lecture Notes in Computer Science, Springer, 2009, pp. 81–96.

[9] C. Kupke and Y. Venema, *Coalgebraic Automata Theory: Basic Results*, Log. Methods Comput. Sci., 4 (2008), pp. 1–43.

[10] R. McNaughton, *Testing and generating infinite sequences by a finite automaton*, Information and Control, 9 (1966), pp. 521–530.

[11] A. W. Mostowski, *Games with forbidden positions*, Tech. Rep. 78, Uniwersytet Gdański, Poland, Instytut Matematyki, 1991.

[12] G. Puppis, *Automata for Branching and Layered Temporal Structures: An Investigation into Regularities of Infinite Transition Systems*, vol. 5955 of Lecture Notes in Computer Science, Springer, 2010.

[13] A. L. Semenov, *Decidability of monadic theories*, LNCS, 176 (1984), pp. 162–175.

[14] W. Thomas, *Languages, Automata, and Logic*, in Handbook of Formal Languages, G. Rozenberg and A. Salomaa, eds., vol. 3, Springer, New York, 1997, pp. 389–455.

[15] Y. Venema, *Automata and Fixed Point Logic: A Coalgebraic Perspective*, Inf. Comput., 204 (2006), pp. 637–678.

[16] I. Walukiewicz, *Monadic second-order logic on tree-like structures*, Theoretical Computer Science, 275 (2002), pp. 311–346.