Weak Monadic Second-Order Logic on Infinitely Branching Trees

Diplomarbeit von Elisabeth Jacobi

Betreut von Priv.-Doz. Dr. habil. Achim Blumensath

Eingereicht am
Fachbereich Mathemaik der
Technischen Universität Darmstadt
im Juli 2013

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Weak Monadic Second-Order Logic on Infinitely Branching Trees

Vorgelegte Diplomarbeit von Elisabeth Jacobi

1. Gutachten: Priv.-Doz. Dr. habil. Achim Blumensath
2. Gutachten: Professor Dr. Thomas Streicher

Tag der Einreichung: 17. Juli 2013

Erklärung zur Diplomarbeit

Hiermit versichere ich, die vorliegende Diplomarbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 17. Juli 2013

(Elisabeth Jacobi)

Danksagung

Ich danke allen Menschen, die mich bei der Entstehung dieser Arbeit unterstützt haben, insbesondere meinem Betreuer Achim Blumensath für seinen Rat und seine Geduld. Des weiteren danke ich Tobias für Motivation und Ablenkung, meinen Eltern für steten Druck und Glauben, meinen besten Freunden für ihre unterstützende Gleichgültigkeit sowie all meinen Mathefreunden für Gespräch und Mathefreuden.

Contents

1 Introduction

This thesis considers the expressive power of weak monadic second-order logic (WMSO) over infinitely branching trees. To this end we mainly use alternating automata. Basically, automata are used to specify and verify properties of structures. They have various applications in software verification, model checking, language processing, complexity theory and decidability theory.

The connection between automata and logic has been known since the 1960s. First of all came Büchi's theorem [Büc60] stating that monadic second-order logic (MSO) with one successor (S1S) is decidable and that finite automata on ω-words and MSO have the same expressive power. He also showed that the transformation from automata to formulae and vice versa is effective.

Later, automata on binary trees as a model for MSO with two successors (S2S) were considered. Rabin obtained some remarkable results in that field. He showed that it is neccessary to use Muller acceptance when considering automata on trees [Rab69]. Then the correspondance between automata and MSO-formulae could be lifted from ω-words to the domain of infinite binary trees. It has also been extended to k-ary trees. Another theorem of Rabin, called Rabin Basis Theorem [Rab72], shows the decidability of the emptyness problem $T_\omega(\mathcal{A}) = \emptyset$? for the set of trees recognized by a Muller automaton $\mathcal{A}$. Rabin's Tree Theorem [Rab69] shows that the family of finitely branching trees recognizable by a Muller automaton is closed under complement.

Considering infinitely branching trees, Muchnik's Theorem ([BB01]) is a generalisation of Rabin's Tree Theorem and thus one of the strongest decidability results known for MSO. Given a structure $\mathfrak{A}$, one can construct its iteration $\mathfrak{A}^*$ which is a tree whose vertices are finite sequences of elements of $\mathfrak{A}$. Muchnik's Theorem says that model checking is decidable for $\mathfrak{A}$ if and only if it is decidable for $\mathfrak{A}^*$.

The equivalenvce of MSO and automata was subsequently studied for other logics, like WMSO. In the 70s, Rabin considered WMSO on binary trees. He showed that a tree language L is definable in WMSO if and only if L and its complement $\bar{L}$ are recognizable by a nondeterministic Büchi automaton [Rab70]. Muller, Schupp and Shelah [MSS92] introduced weak alternating automata and showed their equivalence to WMSO on finitely branching trees. McNaughton showed in [McN66] that nondeterministic automata can be effectively transormed into deterministic ones. This has important consequences. One of them is that MSO and WMSO are equivalent on ω-words. So far WMSO has only been considered over finitely branching trees.

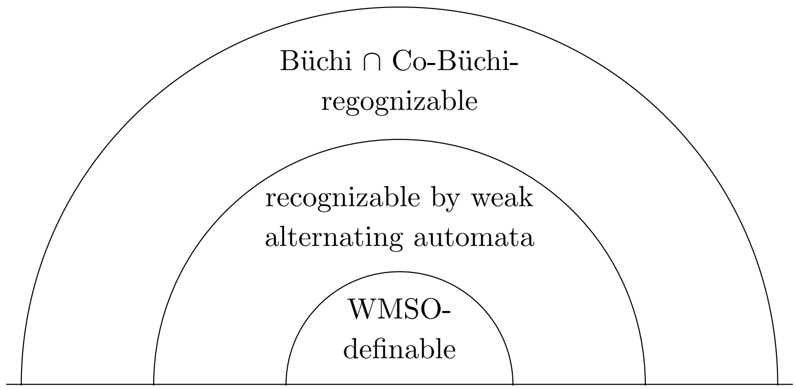


Figure 1.1: The hirachy of languages of infinitely branching trees

In this thesis we investigate to which extent these results generalise to infinitely branching trees. Our motivation stems from an attempt to prove Muchnik's Theorem for WMSO. But since automata and logic do not match in this case, the proof for MSO can not be transfered to WMSO.

The thesis is organized as follows. In the second chapter we give the basic definitions and introduce some proof techniques like games on trees and back-and-forth strategies for WMSO.

Chapter 3 presents the transformation form WMSO-formulae φ defining a class of trees to weak alternating parity automata $\mathcal{A}_\varphi$ recognizing these trees. Since these automata run on infinitely branching trees, we define their transition functions via WMSO in the style of the MSO-automata introduced by Walukiewicz [Wal02].

In Chapter 4, we show that WMSO cannot distinguish between a tree with an infinite path and a tree with infinitely many finite paths which implies that alternating automata are stronger than WMSO on infinitely branching trees.

In Chapter 5 we show that weak alternating automata on infinitely branching trees can be converted into Büchi automata.

Chapter 6 contains a presentation of Rabin's result [Rab70] on finitely branching trees. It says that a tree language L is WMSO-definiable if and only if L and its complement $\bar{L}$ can be recognised by a Büchi automaton.

The last chapter considers the question of on which trees MSO and WMSO are equivalent. Therefore, we define a topological rank for a tree that counts the nesting of infinite branches. First, we show for infinitely branching tress that MSO and WMSO are not equivalent, even for Cantor-Bendixon rank 1. Then we show that MSO and WMSO are equivalent on finitely branching trees of finite Cantor-Bendixon rank.

2 Preliminaries

Let us start with giving some definitions that will be used for this thesis. We define trees first and then specify the logic to talk about them. Further we introduce parity automata, as well as acceptance via games and some graph theory. The last part deals with Ehrenfeucht-Fraïssé techniques. The reader should be familiar with the basics of fist-order logic and some basic automata theory.

2.1 Trees

By $\mathcal{P}(A)$ we denote the power set of the set A. The natural numbers are denoted by $\mathbb{N}$ and contain 0.

Definition 1. Let A be a not neccessarily finite set. The set of finite sequences or *finite words* is denoted by A^* and the set of *infinite words* over A is denoted by A^ω. Let $\varepsilon \in A^*$ be the *empty word* and let $A^\infty = A^* \cup A^\omega$. For two words $u, v \in A^*$ the *prefix order* $\preccurlyeq$ is defined by

$$u \preccurlyeq v \quad \text{if and only if} \quad \text{there exists } w \in A^* \text{ such that } v = uw.$$

A subset of A^* is said to be *prefix-closed* if it contains the prefixes of all of its elements.

Definition 2. Let Σ be a non-empty finite alphabet whose elements are called *labels* and let A be a set. A Σ-labelled *tree* over A is a map

$$t : \mathrm{dom}(t) \to \Sigma,$$

where $\mathrm{dom}(t)$ is a nonempty subset of A^* which is closed under the prefix order. We denote by $T_{A,\Sigma}$ the set of all Σ-trees over A. If the context is clear, we write T_Σ.

The elements of $\mathrm{dom}(t)$ are called the *vertices* or *nodes* of t. If $x \in \mathrm{dom}(t)$ is a vertex, any vertex of the form xy for $y \in \mathrm{dom}(t)$ is called a *successor* of x. The set of successors of a vertex x is $\mathrm{Suc}(x)$.

A tree is said to be finite if its domain is a finite set. If there exists a vertex with infinitely many successors, the tree is called *infinitely branching.*

This thesis deals with infinitely branching labeled trees that have finite sets of labels. The corresponding relational structure is of the form

$$T = (\mathrm{dom}(t), \mathrm{Root}^T, \preccurlyeq^T, (P_a^T)_{a\in\Sigma}).$$

These relations have the following meaning. $\mathrm{Root}^T = \{\varepsilon\}$ only contains the root of T, $\preccurlyeq$ is the prefix order on $\mathrm{dom}(t)$ and P_a^T contains all $v \in \mathrm{dom}(t)$ where $t(v) = a$ holds.

Definition 3. Let Σ and Δ be two alphabets and let $\eta : \Sigma \to \Delta$ be a map. It induces a map, also denoted by η, from $T_{D,\Sigma}$ into $T_{D,\Delta}$ defined by $t \mapsto \eta \circ t$. Such a map is called a *projection* and $\eta \circ t$ denotes the result of replacing the label of each vertex of t by its image under η.

2.2 Automata on Trees

2.2.1 Weak Monadic Second-Order Logic on Trees

Definition 4. *Weak monadic second-order logic*, in short WMSO, is an extension of first-order logic. It allows us to quantify over set variables that range over finite sets, denoted by capital letters $X, Y, \ldots$.

Atomic formulae of $\mathrm{WMSO}(\sigma)$ are of the form

(i) $t_1 = t_2$ for terms t_1, t_2 ,

(ii) $R(t_1, \ldots, t_n)$ for terms $t_1, \ldots, t_n$ and $R \in \sigma$,

(iii) $x \in X$ where x is an individual and X a set variable.

Further, the set of $\mathrm{WMSO}(\sigma)$-formulae is closed under the usual connectives $\vee, \neg$, and under first-order and set quantifiers.

The quantifier rank of a formula is the maximal number of nested first- and second-order quantifiers in that formula.

In general, WMSO-formulae can contain first-order and set variables. But it is possible to transform every WMSO-formula into a formula containing only set variables. Following Thomas [Tho97] we call that version of weak monadic second-order logic WMSO_0 and use it throughout this thesis. Most of the time we drop the subscript 0. This translation results in new atomic formulae

$$X \subseteq Y, \quad \mathrm{Disj}(X, Y), \quad R(X_1, \ldots, X_n)$$

where the first says that X is a subset of Y and $\mathrm{Disj}(X, Y)$ means that X and Y are disjoint. Formulae of the form $R(X_1, \ldots, X_n)$ are satisfied if there exist elements $x_1 \in X_1, \ldots, x_n \in X_n$ such that $(x_1, \ldots, x_n) \in R$.

The set of all trees where φ holds is denoted by $L(\varphi)$ and is called the tree language defined by φ.

For the transition functions that are defined in the following subsection, it suffices to use positive WMSO.

Definition 5. Let σ be a signature. *Positive* weak monadic second-order logic (WMSO^{+}) over σ consists of those formulae $\varphi \in \mathrm{WMSO}(\sigma)$, where all predicates $R \in \sigma$ occur under an even number of negations.

2.2.2 Tree Automata

We introduce several kinds of tree automata which vary according to the allowed transitions and acceptance conditions. They are all special cases of parity automata. The acceptance condition of a parity automaton is defined by a priority function, assigning a priority to every state and a parity condition saying that the least priority of states occuring infinitely often has to be even.

Definition 6. An *alternating parity tree automaton* over the alphabet Σ is a tuple

$$\mathcal{A} = (Q, \Sigma, \delta, q_I, \Omega),$$

where Q is a finite set of states, q_I the initial state,

$$\delta : Q \times \Sigma \to \mathrm{WMSO}^{+}(\{P_q \mid q \in Q\})$$

is the *transition function* that determines the successor states for a given input tuple

$$(q, a) \in Q \times \Sigma$$

and

$$\Omega : Q \to \mathbb{N}$$

is a priority function. We say that a sequence of states $(q_n)_{n<\omega}$ satisfies the *parity condition* Ω if and only if

$$\liminf_{n\to\infty} \Omega(q_n) \text{ is even.}$$

A parity automaton is called

— *weak alternating*, if there exists a preorder $\sqsubseteq$ on Q such that $q \sqsubseteq p$, for every p appearing in $\delta(q, a)$, and $p \sqsubseteq q$ and $q \sqsubseteq p$ implies $\Omega(p) = \Omega(q)$;

— *Büchi*, if $\Omega : Q \to \{0, 1\}$;

— *co-Büchi*, if $\Omega : Q \to \{1, 2\}$.

A *run* of the automaton $\mathcal{A}$ on a tree t is a map

$$r : \mathrm{dom}(t) \to \mathcal{P}(Q \times Q)$$

that satisfies the following conditions:

(i) $r(\varepsilon) = \{(q_I, q_I)\}$;

(ii) for $v \in \mathrm{dom}(t)$ and $p \in Q$ we define a structure

$$\mathfrak{D}^r_{v,p} := (\mathrm{Suc}(v), (P_q)_{q\in Q}), \text{ where } P_q := \{d \in \mathrm{Suc}(v) \mid (p,q) \in r(d)\}.$$

Every $(p,q) \in r(v)$ has to satisfy $\mathfrak{D}^r_{v,q} \models \delta(q, t(v))$.

A run of $\mathcal{A}$ on t is *successful* if for every path $(w_n)_{n<\omega} \in \mathrm{dom}(t)$ and every sequence $(q_n)_{n<\omega} \in Q^\omega$ such that $q_0 = q_I$ and $(q_n, q_{n+1}) \in r(w_{n+1})$, the sequence $(q_n)_{n<\omega}$ satisfies the parity condition. We say that a parity automaton $\mathcal{A}$ *accepts* a tree t, if there exists an accepting run r on t. The set of all $\mathcal{A}$-runs on t is $\mathrm{Run}(\mathcal{A}, t)$. The collection of trees defined by $\mathcal{A}$ is denoted by $L(\mathcal{A})$.

A *nondeterministic* automaton is an automaton as above where a run is a map

$$r : \mathrm{dom}(t) \to Q.$$

For every $w \in \mathrm{dom}(t)$, we define the structure

$$\mathfrak{D}^r_w := (\mathrm{Suc}(w), (P_q)_{q\in Q}),$$

where

$$P_q := \{d \in \mathrm{Suc}(w) \mid r(d) = q\}.$$

A run has to satisfy

$$\mathfrak{D}^r_w \models \delta(r(w), t(w))$$

for every $w \in \mathrm{dom}(t)$.

A run of a nondeterministic automaton is accepting, if $r(\varepsilon) = q_I$ and for every branch π of t the sequence $r(w)_{w\in\pi}$ satisfies the parity condition.

2.2.3 Games on Trees

We introduce a game played on trees which simulates the possible runs of an automaton on the tree.

Definition 7. A *parity game* is a game G played on a graph in which the winning condition is a parity condition. Formally, the game is defined as

$$G = \langle V = V_1 \cup V_2, E, \Omega : V \to \mathbb{N}\rangle,$$

where V is the set of vertices, E the edge relation and Ω a priority function. V_1 are the game positions of Player I and V_2 are those of Player II. The game starts at the initial position $v_0 \in V$. Each player chooses in turn a successor of the current vertex. The game position v has the successor v' if and only if $(v, v') \in E$. If the current position is in V_2, then it is Player II's turn to make a move, otherwise Player I makes a move. This leads to an infinite path of game postitions.

Such a path $(v_n)_{n<\omega}$ is called a *play* and it is *winning*, if it satisfies the parity condition. That is, if $\liminf_{n\to\infty} \Omega(v_n)$ is even. In this case, Player I wins the play, otherwise Player II wins.

A *strategy* for player I is a function

$$f : V^*V_1 \to V$$

from the set of words into V such that $f(w)$ is a successor of w in G. Player I has followed the strategy f in the play $(v_n)_{n<\omega}$ if, for all $n < \omega$,

$$v_n \in V_1 \quad \text{implies} \quad f(v_0 \dots v_n) = v_{n+1}.$$

A strategy for player II is, in a dual way, $f : V^*V_2 \to V$.

If the strategy depends only on the last vertex of the path $v_0 \dots v_n$, it is called a *memoryless strategy*.

A strategy f is called *winning* for player I if he wins all plays in which he follows f.

In [PP04, Tho97] we find this theorem:

Theorem 8. *In each vertex of a parity game, one of the players has a memoryless winning strategy.*

We can define acceptance of tree automata in terms of parity games.

Definition 9. Let $\mathcal{A} = (Q, q_I, \Sigma, \delta, \Omega)$ be an automaton and $T \in T(D, \Sigma)$ a tree. We define the *parity game for alternating automata* $\mathcal{G}(\mathcal{A}, t)$ as follows. The sets of vertices are

$$V_1 = Q \times \mathrm{dom}(t) \quad \text{and} \quad V_2 = \{((S_d)_{d\in\mathrm{Suc}(w)}, w) \mid S_d \subseteq Q, w \in \mathrm{dom}(t)\}.$$

Let

$$\mathfrak{D}_w((S_d)_{d\in\mathrm{Suc}(w)}) := (\mathrm{Suc}(w), (P_q)_{q\in Q})$$

such that

$$P_q = \{d \in \mathrm{Suc}(w) \mid q \in S_d\}.$$

(i) A vertex $(p, w) \in V_1$ has the successor $((S_d)_{d\in\mathrm{Suc}(w)}, w)$ if and only if

$$\mathfrak{D}_w((S_d)_{d\in\mathrm{Suc}(w)}) \models \delta(p, t(w));$$

(ii) a vertex $((S_d)_{d\in\mathrm{Suc}(w)}, w) \in V_2$ has the successors (q, d) for each $d \in \mathrm{Suc}(w)$ with $q \in S_d$;

(iii) the initial position is (q_I, ε).

Let $(\alpha_n, w_n)_{n<\omega}$ be a play. Let $I \subseteq \omega$ be the set of indices n such that $(\alpha_n, w_n) \in V_1$. The play is *winning* if $(\alpha_n)_{n\in I}$ satisfies the parity condition.

Now we can formulate in terms of games what acceptance of a tree by an automaton means. The proof is formulated for alternating automata but it can be adapted easily for nondeterministic automata.

Lemma 10. *Let $\mathcal{A}$ be a parity automaton and t a tree. $\mathcal{A}$ accepts t if and only if Player I has a winning strategy for the game $\mathcal{G}(\mathcal{A}, t)$.*

Proof. ($\Rightarrow$) Let r be an accepting run of $\mathcal{A}$ on t. We define a memoryless strategy $f : V_1 \to V_2$ for Player I for every $(q, w) \in V_1$ by

$$f(q, w) = ((S_d^q)_{d\in \mathrm{Suc}(w)}, w),$$

where

$$S_d^q = \{q' \in Q \mid (q, q') \in r(d)\}.$$

Let $(\alpha_n, w_n)_{n<\omega}$ be a play in $\mathcal{G}(\mathcal{A}, t)$ where Player I followed f. We want to show that it is a winning play. Consider the subseqence where $(\alpha_{2i}, w_{2i}) = (q_{2i}, w_{2i}) \in V_1$. For every i, the position (q_{2i+2}, w_{2i+2}) arises from the set $((S_d^q)_{d\in\mathrm{Suc}(w_{2i})}, w_{2i})$ which is chosen according to r as a successor of (q_{2i}, w_{2i}). Thus, since $(q_n)_{n<\omega}$ with $(q_n, q_{n+1}) \in r(d)$ satisfies the parity condition, every $(q_n, w_n)_{n<\omega}$ does this, too. Therefore, the play $(\alpha_n, w_n)_{n<\omega}$ is winning for Player I.

($\Leftarrow$) Let $f : V_1 \to V_2$ be a memoryless winning strategy for Player I in $\mathcal{G}(\mathcal{A}, t)$. We define a run $r : \mathrm{dom}(t) \to \mathcal{P}(Q \times Q)$ by setting $r(\varepsilon) = \{(q_I, q_I)\}$. For every other $w \in \mathrm{dom}(t)$ suppose that $f(q, w) = ((S_d^q)_{d\in\mathrm{Suc}(w)}, w)$ for each $q \in Q$. We define

$$r(d) = \bigcup_{q\in Q} \{(q, q') \mid q' \in S_d^q\},$$

for $d \in \mathrm{Suc}(w)$.

To prove that r is successful, let $(w_n)_{n<\omega}$ be a path of t with $w_0 = \varepsilon$ and a corresponding sequence $(q_n)_{n<\omega}$ such that $(q_n, q_{n+1}) \in r(w_{n+1})$. By the choice of r which is according to the winning strategy f, every (q_n, w_n) appears also in the play (α_n, w_n) and satisfies the parity condition. Therefore, r is successful.

□

2.3 Back-And-Forth Arguments

This section provides some technical preparations which are needed in Chapter 4 to prove the equivalence of tree structures of different size.

After recalling some definitions and results, we present two operations to split trees, firstly at the root, secondly at an arbitrary vertex and prove the compatibility with WMSO.

2.3.1 WMSO_m-Equivalence of Structures

Definition 11. Let WMSO^m be the set of WMSO-formulae with quantifier rank $\leq m$. $\mathrm{free}(\varphi)$ is the set of free variables of $\varphi \in \mathrm{WMSO}$.

For a structure $\mathfrak{A}$,

$$\mathrm{Th}_m(\mathfrak{A}) := \{\varphi \in \mathrm{WMSO}^m \mid \mathfrak{A} \models \varphi, \mathrm{free}(\varphi) = \emptyset\}$$

is called the *m-theory* of $\mathfrak{A}$. Two structures $\mathfrak{A}, \mathfrak{B}$ are said to be *m-equivalent*, $\mathfrak{A} \equiv_m \mathfrak{B}$, if and only if

$$\mathrm{Th}_m(\mathfrak{A}) = \mathrm{Th}_m(\mathfrak{B}).$$

Two structures $\mathfrak{A}, \mathfrak{B}$ of signature σ with universes A, B are *isomorphic*, denoted by $\mathfrak{A} \cong \mathfrak{B}$, if and only if there exists a bijection $\iota : A \to B$ preserving relations and constants in σ, i.e.

(i) for n-ary $R \in \sigma$ and $a_1, \ldots, a_n \in A$:

$$R^{\mathfrak{A}}(a_1 \ldots a_n) \quad \text{if and only if} \quad R^{\mathfrak{B}}(\iota(a_1) \ldots \iota(a_n));$$

(ii) for $c \in \sigma$ we have $\iota(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$.

Lemma 12. *For stuctures $\mathfrak{A}, \mathfrak{B}$ the following are equivalent:*

(i) $\mathfrak{A} \equiv_{m+1} \mathfrak{B}$

(ii) for every finite subset $P \subseteq A$ there exists a finite set $Q \subseteq B$ such that $(\mathfrak{A}, P) \equiv_m (\mathfrak{B}, Q)$ and
for every finite subset $Q \subseteq B$ there exists a finite set $P \subseteq A$ such that $(\mathfrak{A}, P) \equiv_m (\mathfrak{B}, Q)$.

Proof. (i)$\Rightarrow$(ii): Let $P \subseteq A$ be finite and set $\varphi := \bigwedge \mathrm{Th}_m(\mathfrak{A}, P)$. Then $\mathfrak{A} \models \exists X\varphi$ and by assumption also $\mathfrak{B} \models \exists X\varphi$. Consequently, there exists a finite set $Q \subseteq B$ such that $\mathfrak{B} \models \varphi(Q)$. Hence, $(\mathfrak{B}, Q) \equiv_m (\mathfrak{A}, P)$. The other direction follows by symmetry of $\equiv_m$.

(ii)$\Rightarrow$(i): Since every formula in WMSO_0^{m+1} is a boolean combination of formulae of the form $\exists X\varphi$ with $\varphi \in \mathrm{WMSO}^m$, it is sufficient to prove that $\mathfrak{A} \models \exists X\varphi \Rightarrow \mathfrak{B} \models \exists X\varphi$, for $\varphi \in \mathrm{WMSO}^m$. Let $\varphi \in \mathrm{WMSO}_0^m$ and $\mathfrak{A} \models \exists X\varphi$. Choose a set $P \subseteq A$ with $\mathfrak{A} \models \varphi(P)$. There exists $Q \subseteq B$ such that $(\mathfrak{A}, P) \equiv_m (\mathfrak{B}, Q)$. Hence, $\mathfrak{A} \models \varphi(P)$ implies $\mathfrak{B} \models \varphi(Q)$. Therefore, $\mathfrak{B} \models \exists X\varphi(X)$. □

Corollary 13. *There is a one-to-one correspondence between m-theories $\mathrm{Th}_m(\mathfrak{A})$ and sets of the form $\{\mathrm{Th}_{m-1}(\mathfrak{A}, P) \mid P \subseteq A\}$.*

Lemma 14. *The number of WMSO_m-theories for a fixed finite signature τ is finite.*

Proof. For a τ-structure $\mathfrak{A}$, we inductively define sets of theories

$$\mathrm{Th}'_0(\mathfrak{A}) := \mathrm{Th}_0(\mathfrak{A}) \quad \text{and} \quad \mathrm{Th}'_{m+1}(\mathfrak{A}) := \{\mathrm{Th}'_m(\mathfrak{A}, P) \mid P \subseteq A \text{ finite}\}.$$

By Corollary 13, $\mathrm{Th}_m(\mathfrak{A})$ is uniquely determined by $\mathrm{Th}'_m(\mathfrak{A})$. Now we can see that $\mathrm{Th}'_m(\mathfrak{A})$ is finite, because in every iteration step the set is included in the power set of the set before and $\mathrm{Th}_0(\mathfrak{A})$ is finite due to finite signature. □

2.3.2 Composition of Trees and their Theories

Now we present some lemmas showing the preservation of m-equivalence for three composition operations on trees.

In [EF95, p.39] we find the first one:

Lemma 15. *The disjoint union preserves* WMSO^m*-eqivalence, i.e. for trees S and T*

$$S \equiv_m T, S' \equiv_m T' \implies S \,\dot{\cup}\, S' \equiv_m T' \,\dot{\cup}\, T'.$$

In the remainder of the chapter, we present operations to split trees into their subtrees such that the theories of the subtrees are preserved. Therefore we first introduce the definition of an interpretation that translates between two structures.

Definition 16. Let τ and $\sigma = \{R_1, \ldots, R_m\}$ be two vocabularies with $\rho(R_i)$ the arity of R_i.

A *definition scheme* $\Phi = \langle \phi, \psi_1, \ldots \psi_m \rangle$ is a list of formulae of $\mathrm{WMSO}(\tau)$ such that ϕ has exactly one free first order variable and each ψ_i has $\rho(R_i)$ distinct free first order variables.

Given a list Φ as above, the *interpretation* Φ^* is a partial function from τ-structures to σ-stuctures and is defined by $\Phi^*(\mathfrak{A}) = \mathfrak{A}_\Phi$ where

(i) the universe of $\mathfrak{A}_\Phi$ is the set

$$A_\Phi = \{a \in A : \mathfrak{A} \models \phi(a)\},$$

(ii) the interpretation of R_i in $\mathfrak{A}_\Phi$ is the set

$$\mathfrak{A}_\Phi(R_i) = \{\bar{a} \in A_\Phi^{\rho(R_i)} : \mathfrak{A} \models \psi_i(\bar{a})\}.$$

The image $\Phi^*(\mathfrak{A})$ is defined, if the universe A_ϕ is nonempty.

The *syntactic interpretation* of Φ is a function $\Phi^\sharp : \mathrm{WMSO}(\tau) \to \mathrm{WMSO}(\sigma)$ from $\mathrm{WMSO}(\tau)$-formulae to $\mathrm{WMSO}(\sigma)$-formulae, inductively defined by the following properties:

(i) For $R_i \in \sigma$ and $\theta = R_i(x_1, \dots, x_m)$, we put

$$\Phi^\sharp(\theta) = \psi_i(x_1, \dots, x_m) \wedge \bigwedge_i \phi(x_i).$$

For equality and a set variable Z we have

$$\Phi^\sharp(x = y) = x = y$$

and

$$\Phi^\sharp(Z(x)) = Z(x) \wedge \phi(x).$$

(ii) For the boolean connectives, let θ_1 and θ_2 be a WMSO(τ)-formulae.

a) $\Phi^\sharp(\theta_1 \vee \theta_2) = \Phi^\sharp(\theta_1) \vee \Phi^\sharp(\theta_2)$.

b) $\Phi^\sharp(\neg\theta_1) = \neg\Phi^\sharp(\theta_1)$.

(iii) For existential quantification of FO-variables, we put

$$\Phi^\sharp(\exists y\theta) = \exists y(\phi(y) \wedge \Phi^\sharp(\theta)).$$

(iv) For second order quantification

$$\Phi^\sharp(\exists U\theta_1) = \exists U[\forall v(U(v) \to \phi(v)) \wedge \Phi^\sharp(\theta_1)].$$

Lemma 17. *Let Φ be an interpretation in* WMSO*.*

(i) If $\theta \in$ WMSO, then $\Phi^\sharp(\theta)$ is in WMSO*.*

(ii) If Φ is of quantifier rank q and θ is a formula of quantifier rank r, then the quantifier rank of $\Phi^\sharp(\theta)$ is bounded by $r \cdot q$.

Proof. (i) follows form property (v) of Definition 16 and (ii) follows by Definition of $\Phi^\sharp$. □

Without further assumptions we obtain the fundamental property of the connection between Φ^* and $\Phi^\sharp$.

Theorem 18. *Let $\Phi = \langle \phi, \psi_1, \dots, \psi_m \rangle$ be an interpretation in* WMSO *form τ to σ. Then semantic and syntactic interpretation, Φ^* and $\Phi^\sharp$, are linked as follows: Given a τ-structure $\mathfrak{A}$ such that $\Phi^*(\mathfrak{A})$ is defined and an* WMSO(σ)*-formula θ, then*

$$\mathfrak{A} \models \Phi^\sharp(\theta) \quad \textit{iff} \quad \Phi^*(\mathfrak{A}) \models \theta.$$

Theorem 18 together with Lemma 17 gives us what is needed here:

Lemma 19. *Let Φ be an interpretation of quantifier rank q and let $\mathfrak{A}$ be a τ-structure. Then $\mathrm{Th}_m^{\mathrm{WMSO}}(\Phi^*(\mathfrak{A}))$ depends only on $\mathrm{Th}_{m \cdot q}^{\mathrm{WMSO}}(\mathfrak{A})$.*

In the remainder of this chapter we present various operations to divide trees that are compatible with the theories of the respective subtrees.

First, we describe an operation on trees that takes the disjoint union of two trees and fuses their roots.

Definition 20. For two trees T, S define

$$T \oplus S = (V_{T\oplus S}, \mathrm{Root}^{T\oplus S}, \preccurlyeq^{T\oplus S}),$$

where

(i) $V_{T\oplus S} := (V_T - \mathrm{Root}^T) \,\dot\cup\, (V_S - \mathrm{Root}^S) \,\dot\cup\, \{r\}$

(ii) $\mathrm{Root}^{T\oplus S} := \{r\}$

(iii) $\preccurlyeq^{T\oplus S} := \preccurlyeq^T \,\dot\cup \preccurlyeq^S \cup \{(r,v) \mid v \in V_{T\oplus S}\}$

Lemma 21.

$$T \equiv_m T' \ , \ S \equiv_m S' \ \Rightarrow \ T \oplus S \equiv_m T' \oplus S'.$$

Proof. $S \oplus T = \mathrm{fuse}_{\mathrm{Root}}(T \dot\cup S)$ is build up from the disjoint union of the two structures and then fusing their roots. Both the disjoint union and the fusion operation are compatible with the WMSO-theory, cf. Lemma 15 and [Mak04]. $\square$

The next operation takes the disjoint union of the trees and fuses them at a distinguished vertex.

Definition 22. For τ-trees S, T and a vertex $l \in S$, we define the operation

$$S \oplus_l' T = (V_{S\oplus_l' T}, \mathrm{Root}^{S\oplus_l' T}, \preccurlyeq^{S\oplus_l' T}),$$

such that

(i) $V_{S\oplus' T} := V_S \,\dot\cup\, V_T - \mathrm{Root}^T$

(ii) $\mathrm{Root}^{S\oplus_l' T} := \mathrm{Root}^S$

(iii) $\preccurlyeq^{S\oplus_l' T} := \preccurlyeq^S \cup \preccurlyeq^T \cup \{(x,y) \mid x \preccurlyeq^S l, y \in T\}$

Lemma 23. $\mathrm{Th}_m(S \oplus_l' T)$ *is uniquely determined by* $\mathrm{Th}_m(T)$ *and* $\mathrm{Th}_m(\langle S, l\rangle)$.

Proof. Note that $S \oplus_l' T = \Phi^*(\langle S, l\rangle \,\dot\cup\, T)$ where

$$\Phi := \langle \phi, \psi_{\mathrm{Root}}(x), \psi_{\preccurlyeq}(x,y)\rangle,$$

where

$$
\begin{aligned}
\phi_S &:= S(x) \vee (T(x) \wedge \neg \operatorname{Root}(x)) \\
\psi_{\operatorname{Root}}(x) &:= S(x) \wedge \operatorname{Root}(x) \\
\psi_{\preccurlyeq}(x, y) &:= x \preccurlyeq y \vee (S(x) \wedge T(y) \wedge x \preccurlyeq l)
\end{aligned}
$$

Lemma 19 then implies that $\operatorname{Th}_m(\Phi^*(\langle S, l\rangle \mathbin{\dot\cup} T))$ depends only on $\operatorname{Th}_m(\langle S, l\rangle \mathbin{\dot\cup} T)$, which, by Lemma 15, only depends on $\operatorname{Th}_m(\langle S, l\rangle)$ and $\operatorname{Th}_m(T)$. $\square$

3 Translating Formulae into Automata

Following the previous chapter, we use the signature $\sigma = (\mathrm{Root}, \preccurlyeq, (P_a)_{a\in\Sigma})$ for formulae over trees. Let φ be a WMSO(σ)-formula. The goal of this section is to show that for every WMSO-definable tree language L_φ we can construct a weak alternating automaton recognizing that language.

Theorem 24. *For every formula $\varphi \in \mathrm{WMSO}(\sigma)$ there exists a weak alternating automaton $\mathcal{A}_\varphi$ such that*

$$L(\mathcal{A}_\varphi) = L(\varphi).$$

The proof is by induction on the structure of φ. For the inductive step we have to define $L(\varphi)$ for formulae with free variables. Let $\varphi(X_1, \ldots, X_n) \in \mathrm{WMSO}(\sigma)$ with free variables $X_1, \ldots, X_n$ and $t : \mathrm{dom}(t) \to \Sigma$ be a tree. To specify the values of $X_1, \ldots, X_n \in \mathrm{dom}(t)$, we use an extended alphabet

$$\Sigma_{\bar{P}} = \Sigma \times \mathcal{P}(V_\varphi),$$

where Σ is the set of labels of $t : \mathrm{dom}(t) \to \Sigma$ and V_φ is the set of free variables of φ. Thereby we can represent $\langle P_1, \ldots, P_n\rangle$ by the new labelling

$$t_{\bar{P}} : \mathrm{dom}(t) \to \Sigma_{\bar{P}}$$

such that for a vertex $x \in \mathrm{dom}(t)$,

$$t_{\bar{P}}(x) = (t(x), V), \quad \text{where } V = \{X_i \mid x \in P_i\}.$$

With the above notations we get

Lemma 25. *Let $T = (\mathrm{dom}(t), \mathrm{Root}, \leq, (P_a)_{a\in\Sigma})$. For every atomic $\varphi \in$ WMSO there exists a weak alternating automaton $\mathcal{A}_\varphi$ such that*

$$T \models \varphi(P_1, \ldots, P_n) \textit{ if and only if } t_{\bar{P}} \in L(\mathcal{A}_\varphi).$$

Proof. For atomic φ, we build $\mathcal{A}_\varphi = (Q, \Sigma, \delta, \Omega)$ as follows.

$X \subseteq Y$: For every vertex $x \in \mathrm{dom}(t)$ it has to be true that $x \in X \to x \in Y$. Let $Q = \{q_0\}$.

$$\delta(q, (a, V)) = \begin{cases} \forall x P_{q_0}(x) & \text{if } X \notin V \text{ or } Y \in V, \\ \bot & \text{else} \end{cases}$$

For the time $\mathcal{A}_\varphi$ stays in q_0, $X \subseteq Y$ is true and $\mathcal{A}$ is accepting. Thus the parity condition is $\Omega(q_0) = 0$.

$\mathrm{Disj}(X, Y)$: The automaton has to check if $x \notin X$ or $x \notin Y$ for every $x \in T$. Let the state set be $Q := \{q_0\}$ and $\Omega(q_0) = 0$.

$$\delta(q, (a, V)) = \begin{cases} \forall x P_{q_0}(x) & \text{if } Y \notin V \text{ or } X \notin V \\ \bot & \text{else} \end{cases}$$

$\mathrm{Root}(X)$: This automaton accepts or rejects immediately after reading the first letter of $t(x)$. There is one state q with $\Omega(q) = 1$.

$$\delta(q, (a, V)) = \begin{cases} \top & \text{if } X \in V \\ \bot & \text{else} \end{cases}$$

$\mathrm{Suc}(X, Y)$: In this case $\mathcal{A}_\varphi$ has to verify for some $x \in X$ whether some successor of x is also in Y. This requires two states, therefore $Q = \{q_0, q_1\}$.

$$\delta(q_0, (a, V)) = \begin{cases} \exists x (P_{q_0}(x) \vee P_{q_1}(x)) & \text{if } X \in V, \\ \exists x P_{q_0}(x) & \text{if } X \notin V \end{cases}$$

$$\delta(q_1, (a, V) = \begin{cases} \top & \text{if } Y \in V, \\ \bot & \text{else} \end{cases}$$

We define $\Omega(q) = 1$ for all $q \in Q$. $\mathcal{A}_\varphi$ stops accepting or rejecting if it ends with $\top$ or $\bot$, respectively.

$P_c(X)$: The automaton has to check if x is in X and $t(x) = c$ for $c \in \Sigma$. Thus, we need Q to be $\{q_0\}$.

$$\delta(q, (a, V)) = \begin{cases} \top & \text{if } a = c \wedge X \in V \\ \exists x P_{q_0}(x) & \text{else} \end{cases}$$

Obviously, q_0 is not accepting this time, thus $\Omega(q_0) = 1$.

$\square$

To prove closure under boolean connectives we need automata that recognize those classes of trees that came up from the negation of a formula

$$L(\neg\varphi) = \{T \in T_\Sigma \mid T \models \neg\varphi\} = \overline{L(\varphi)}$$

and the disjunction of two formulas

$$L(\varphi \vee \psi) = \{T \in T_\Sigma \mid T \models \varphi \vee \psi\} = L(\varphi) \cup L(\psi).$$

Therefore we introduce an automaton recognizing the complement of a tree language and one recognizing the sum of two tree languages.

For these two cases the proof is analogous to the proof already done for MSO in [BB01].

Definition 26. The *sum* of two weak alternating automata $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, q_1^I, \Omega_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, q_2^I, \Omega_2)$ is the automaton

$$\mathcal{A}_1 + \mathcal{A}_2 := (Q_1 \,\dot\cup\, Q_2 \,\dot\cup\, \{q_I\}, \Sigma, A, \delta, q_I, \Omega),$$

where

$$\delta^+(q,a) = \begin{cases} \delta_i(q,a) & \text{if } q \in Q_i, \\ \delta(q_I,a) = \delta_1(q_1^I,a) \vee \delta_2(q_2^I,a) & \text{if } q = q_I, \end{cases}$$

and $\Omega : Q_1 \cup Q_2 \cup \{q_I\} \to \mathbb{N}$, such that $\Omega(q) = \Omega_i(q)$ if $q \in Q_i$.

Lemma 27. *The class of tree languages recognised by weak alternating tree automata is closed under union.*

Proof. $\mathcal{A}_1 + \mathcal{A}_2$ satisfies the requirements of a weak alternating tree automaton since there is a preorder on the state set as follows.

$$\begin{aligned} q_I \sqsubseteq p &\qquad \text{for every } p \in Q_1 \cup Q_2; \\ p \sqsubseteq q &\qquad \text{if } p \sqsubseteq^{Q_1} q \text{ or } p \sqsubseteq^{Q_2} q. \end{aligned}$$

Let f_1, f_2 be winning strategies for player I in $G(\mathcal{A}_1, t)$ and $G(\mathcal{A}_2, t)$, respectively. In the game $G(\mathcal{A}_1 + \mathcal{A}_2, t)$, player I can choose from the initial state q_I whether to play in $\mathcal{A}_1$ or $\mathcal{A}_2$. Thus if player I follows f_1, he wins $G(\mathcal{A}_1 + \mathcal{A}_2, t)$ as well as if he follows f_2.

Vice versa, a winning strategy for $G(\mathcal{A}_1 + \mathcal{A}_2, t)$ is still a successful strategy on $G(\mathcal{A}_1, t)$ or $G(\mathcal{A}_2, t)$ depending which subtree player I chooses in his first move.

Thus, for the resulting automaton $\mathcal{A}_1 + \mathcal{A}_2$, we have that $L(\mathcal{A}_1 + \mathcal{A}_2) = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$. ☐

The automaton recognizing trees described by $\neg\varphi$ is defined by dualisation of acceptance condition and transition function.

The construction of the dual automaton and the proof of Lemma 30 are adapted from [Wal02].

Definition 28. The *complement* of $\mathcal{A} = (Q, \Sigma, \delta, q_I, \Omega)$ is the automaton

$$\bar{\mathcal{A}} = (Q, \Sigma, \bar\delta, q_I, \bar\Omega).$$

$\bar\delta(q,a) = \overline{\delta(q,a)}$ is the dual of $\delta(q,a)$, where $\vee$ and $\wedge$, as well as existential and universal quantification are exchanged. The dual acceptance condition is $\bar\Omega(q) = \Omega(q) + 1$.

Lemma 29. *Let $\varphi \in \mathrm{WMSO}^+(\{P_q \mid q \in Q\})$. If $(D, (P_q)_{q\in Q}) \models \varphi$ and $(D, (P'_q)_{q\in Q}) \models \bar{\varphi}$, then there are $d \in D$ and $q \in Q$ such that $d \in P_q \cap P'_q$.*

Proof. Set

$$P''_q := D \setminus P'_q.$$

Then $(D, (P'_q)_{q\in Q}) \models \bar{\varphi}$ implies $(D, (P''_q)_{q\in Q}) \models \neg\varphi$. Since φ is monotone, it follows that there is some $q \in Q$ such that

$$P_q \nsubseteq P''_q.$$

Hence, there is some element $d \in P_q \setminus P''_q = P_q \cap P'_q$. □

Consider a play p in the game $G(\mathcal{A}, t)$ and a play $\bar{p}$ in $G(\bar{\mathcal{A}}, t)$. Note that the game positions V_1 of player I in $G(\mathcal{A}, t)$ are the same as those in $\bar{V}_1$ of $G(\bar{\mathcal{A}}, t)$. The game positions of player II, V_2 and $\bar{V}_2$ are of the form $((S_d)_{d\in\mathrm{Suc}(w)}, w)$ or $((\bar{S}_d)_{d\in\mathrm{Suc}(w)}, w)$, respectively, such that the sets S_d (respectively $\bar{S}_d$) contain successor states for vertex $w \in t$. Since $S_d = \{q \in Q \mid d \in P_q\}$, we obtain the next statement direclty from Lemma 29.

Lemma 30. *Let $\mathcal{A}$ be an automaton and $\bar{\mathcal{A}}$ its complement. For every Σ-labelled tree t we have*

$$t \in L(\mathcal{A}) \quad \textit{if and only if} \quad t \notin L(\bar{\mathcal{A}}).$$

Proof. To prove the implication from left to the right, assume $t \in L(\mathcal{A})$. This means that there is a winning strategy f for player I in the game $G(\mathcal{A}, t)$. We will show how to use this strategy to construct a winning strategy for player II in $G(\bar{\mathcal{A}}, t)$. This implies that player I does not have a winning stragtegy in $G(\bar{\mathcal{A}}, t)$, hence t is not accepted by $\mathcal{A}$.

We construct the winning strategy for player II in $G(\bar{\mathcal{A}}, t)$ by induction on the length of the play. The initial postition in both plays is (q_I, ε). Our induction hypothesis is that we have two finite plays $p = v_0 \dots v_{n_1}$ in $G(\mathcal{A}, t)$ and $\bar{p} = \bar{v}_0 \dots \bar{v}_{n-1}$ in $G(\bar{\mathcal{A}}, t)$ that are of length n, such that the subsequences of p and $\bar{p}$ in V_1 are the same. The play p followed strategy f. Let (q, w) be the last position in p as well as in $\bar{p}$. Now player I chooses in $G(\bar{\mathcal{A}}, t)$ the successors $((\bar{S}_d)_{d\in Suc(w)}, w)$ of (q, w). We are going to consult the strategy f to find an appropriate answer for player II.

As p was obtained using strategy f we know hat $f(p)$ is defined. Hence, let

$$f(p) = ((S_d)_{d\in\mathrm{Suc}(w)}, w).$$

By Lemma 29, we know that there exists $d \in \mathrm{Suc}(w)$ such that

$$S_d \cap \bar{S}_d \neq \emptyset.$$

Choose $q' \in S_d \cap \bar{S}_d$. Player II in $G(\bar{\mathcal{A}}, t)$ plays (q', d). So we put

$$\bar{f}(\bar{p}(q, w)((\bar{S}_d)_{d\in\mathrm{Suc}(w)}, w)) = (q', d).$$

This is also a possible choice for player I in $G(\mathcal{A}, t)$. The initial parts of the two plays become

$$p(q, w)((S_d)_{d\in\mathrm{Suc}(w)}, w)(q', d) \quad \text{and} \quad \bar{p}(q, w)((\bar{S}_d)_{d\in\mathrm{Suc}(w)}, w)(q', d).$$

From this point we can repeat the argument.

Whenever $\bar{p}$ is a play in $G(\bar{\mathcal{A}}, t)$ according to the strategy described above then we have a play in $G(\mathcal{A}, t)$ such that the projections of $\bar{p}$ and p on V_1 are the same. We know that p is winning for player I in $G(\mathcal{A}, t)$ because p was played to the winning strategy f. Hence, by the definition of $\bar{\Omega}$, play $\bar{p}$ is winning for player II in $G(\bar{\mathcal{A}}, t)$.

The other direction follows by symmetry. □

Corollary 31. *The class of languages recognized by weak alternating tree automata is closed under complementation.*

The last thing to show is closure under finite projection that corresponds to the existential quantifier. Since we are dealing with alternating automata, the proof is not as easy as for nondeterministic automata. [MSS92] proved the result for WMSO on k-ary trees.

In the previous chapter, we have already defined the projection of a tree language. In the remainder of the current chapter, we prove that the classes of trees recognizable by weak alternating automata are closed under finite projection, according to the following definition.

Definition 32. Let $\Sigma \subseteq \Delta$ be finite alphabet and η be a projection map from Δ to Σ, such that $\eta = \mathrm{id}$ for every $a \in \Sigma$. Let L be a language of Δ-trees. The language $\eta_f(L)$ over the alphabet Σ is called the *finite projection* of L and is defined by

$$t' \in \eta_f(L)$$

if and only if there exists a tree

$$t \in T_\Sigma \quad \text{such that} \quad t' = \eta \circ t$$

and t has only finitely many vertices labelled from $\Delta \setminus \Sigma$.

Now we have

$$L(\exists X \varphi(X)) = \{T \in T_\Sigma \mid T \models \exists X \varphi(X)\} = \eta_f(L).$$

Lemma 33. *Let $\Sigma \subseteq \Delta$ be alphabets and let η_f be the finite projection to Σ. For every automaton $\mathcal{A}$, there exists an automaton $\mathcal{A}'$ such that*

$$L(\mathcal{A}') = \eta_f(L(\mathcal{A})).$$

Proof. Let $\mathcal{A} = (Q, \Delta, \delta, q_0, \Omega)$ be a weak alternating automaton recognizing L. The goal is to find an automaton $\mathcal{A}'$ recognizing the finite projection $\eta_f(L)$, that is constructed as follows. $\mathcal{A}'$ has two modes, the nondeterministic mode and the alternating mode. The nondeterministic mode simulates a nondeterministic automaton on the vertices labelled from $\Delta \setminus \Sigma$. It keeps track of all the possible states of $\mathcal{A}$ at a given vertex, therefore we use the power set $\mathcal{P}(Q)$ as state set of the nondeterministic mode of $\mathcal{A}'$. The alternating mode simulates the original automaton $\mathcal{A}$, running on a copy of Q that is disjoint from $\mathcal{P}(Q)$. The transition function δ' starts in $\{q_0\}$ in the nondeterministic mode and in state $S \in \mathcal{P}(Q)$ at a vertex with label $a \in \Sigma$, $\mathcal{A}'$ guesses a preimage $b \in \eta_f^{-1}(a)$ and collects every possible successor state of q arising from $\delta(q, b)$, for every $q \in S$. $\mathcal{A}'$ changes in the alternating mode if it hopes not to meet any more vertices whose preimage is in $\Delta \setminus \Sigma$ in the subtree below the current vertex. We define the automaton

$$\mathcal{A}' = (\mathcal{P}(Q) \mathbin{\dot\cup} Q, \Delta, \delta', \{q_0\}, \Omega'),$$

where

$$\delta' : (\mathcal{P}(Q) \mathbin{\dot\cup} Q) \times \Sigma \to \mathrm{WMSO}^+(\mathcal{P}(Q) \mathbin{\dot\cup} Q);$$

$$\begin{aligned} \delta'(q, a) &= \delta(q, a) \qquad \text{for} \quad q \in Q, a \in \Sigma \\ \delta'(S, a) &= \bigvee_{b \in \eta^{-1}(a)} \bigwedge_{q \in S} \delta^*(q, b) \quad \text{for} \quad S \in \mathcal{P}(Q), \end{aligned}$$

where

$$\delta^*(q, b) = \exists X[\delta(q, b)[p \vee (X \wedge \bigvee_{S : p \in S} S)/p]_{p \in Q}].$$

We set

$$\begin{aligned} \Omega'(q) &= \Omega(q) \quad \text{for } q \in Q \\ \Omega'(S) &= 1 \qquad \text{for } S \in \mathcal{P}(Q). \end{aligned}$$

It remains to show that $t \in L(\mathcal{A})$ if and only if $\eta_f(t) \subseteq L(\mathcal{A}')$. Firstly, let $t \in L(\mathcal{A})$ and $\mathcal{A}'$ running on $t' \in \eta_f(t)$. Then there exists a memoryless winning strategy f for player I in $G(\mathcal{A}, t)$. We construct a winning strategy f' for player I in $G(\mathcal{A}', t')$ by induction on the length of the play. Our induction hypothesis is that we have finite plays $(p^q)_q$ and p' satisfying the following conditions:

- Let $p' = v'_0 \dots v'_{n-1}$. If $v'_{n-1} = (S, w) \in \mathcal{P}(Q) \times \mathrm{dom}(t)$, then, for every $q \in S$, there exisits a play $p^q = v^q_1 \dots v^q_{n-1}$ that follows strategy f. If $v'_{n-1} = (q, w) \in Q \times \mathrm{dom}(t)$, there exisits a play $p^q = v^q_1 \dots v^q_{n-1}$ that follows strategy f;
- the projections on the second component of v_i and v'_i, $i \in \{1, \dots, n-1\}$ coincide;

— for every uneven $i \leq n-1$ (these are the game positions of player II) either $v'_i = v_i$ or $v_i = (p, w)$ and $v'_i = (S, w)$ for $p \in S$.

We start with the plays $p' = (\{q_0\}, \varepsilon)$ and $p^{q_0} = (q_0, \varepsilon)$. To define the strategy $f'(v'_{n-1})$, we first consider the case where $v'_{n-1} \in Q \times \mathrm{dom}(t)$. Then player I plays according to the strategy f, which is defined for all $(q, w) \in Q \times \mathrm{dom}(t)$. In this case, we set

$$v_n := v'_n = f'(v'_{n-1}) = f(v_{n-1}).$$

In the second case, $v'_{n-1} = (S, w) \in \mathcal{P}(Q) \times \mathrm{dom}(t)$. Let

$$I := \{d \in \mathrm{Suc}(w) \mid \text{there exists } v \succeq d \text{ such that } t(v) \in \Delta \setminus \Sigma\}.$$

For $d \in I$, we set

$$S'_d := \{\bigcup_{q \in S} S^q_d\},$$

and, for $d \notin I$, we set

$$S'_d = \bigcup_{q \in S} S^q_d.$$

Further, we set

$$v'_n = f'(v'_{n-1}) := ((S'_d)_{d \in \mathrm{Suc}(w)}, w).$$

In both cases, this implies that $(\mathrm{Suc}(w), (S'_d)_{d \in \mathrm{Suc}(w)}) \models \delta'(S, \eta_f(t)(w))$. For the play that follows strategy f, we define

$$v_n := f(v_{n-1}).$$

Then, it is player II's turn and he chooses $d \in \mathrm{Suc}(w)$ and some state in S'_d. If $d \in I$, then

$$v'_{n+1} := (S', d), \quad \text{if } S'_d = \{S'\},$$

and we set

$$v^q_{n+1} := (q, d), \quad \text{for every } q \in S'.$$

If $d \notin I$,

$$v'_{n+1} := (q, d), \quad \text{for some } q \in S'_d,$$

and we set

$$v^q_{n+1} := (q, d).$$

We continue this process and obtain infinite plays P and P'. The play P' of $G(\mathcal{A}', t)$ that is played according to f' is accepting, since it is the same as the infinite play P in $G(\mathcal{A}, t)$ that is played according to strategy f, except for finitely many positions. It follows that Player I wins the game $G(\mathcal{A}', t)$.

For the converse direction, let $t' \in L(\mathcal{A}')$. We show that this implies $t \in L(\mathcal{A})$ for some $t \in T_\Delta$ with $\eta_f(t) = t'$. Let f' be a winning strategy for player I in $G(\mathcal{A}', t')$. We construct t and a winning strategy f for player I in $G(\mathcal{A}, t)$ by induction.

As induction hypothesis, suppose that we have constructed a finite tree t_f of depth n such that $\eta_f(t_f) = t'_f \preccurlyeq t'$ and two finite plays p' and p satisfying the following conditions.

— $p' = v'_0 \dots v'_{n-1}$, $p = v_0 \dots v_{n-1}$ and p' follows strategy f';

— the projections on the second component of v_i and v'_i, $i \in \{1, \dots, n-1\}$ coincide;

— for every uneven $i \leq n-1$, $v_i = (q, w)$ implies $(q, w) = v'_i$ or, $v'_i = (S, w)$ for $q \in S$.

We start with the plays $p = (q_0, \varepsilon)$ and $p' = (\{q_0\}, \varepsilon)$. For the inductive step, let $v_{n-1} = (q, w)$ be the last position of p. To find the next move $f(q, w)$ for player I, we consider $f'(v'_{n-1})$. The last position v'_{n-1} of p' can be of two kinds. The first one is that $v_{n-1} = (q, w) \in Q \times \mathrm{dom}(t)$. Then we set

$$v_n = f(q, w) := f'(q, w) = v'_n$$

and we choose

$$t(w) := t'(w).$$

If the last position of p' contains a set state such that $v'_{n-1} = (S', w)$, we consider $f'(S', w)$ which is of the form

$$f'(S', w) = ((S'_d)_{d \in \mathrm{Suc}(w)}, w).$$

For every $d \in \mathrm{Suc}(w)$, S'_d contains those set states that are possible successor states of every $q \in S'$. Since

$$(\mathrm{Suc}(w), (P'_{q'})_{q' \in Q'}) \models \delta'(S', t'(w)) = \bigvee_{b \in \eta^{-1}(t(w))} \bigwedge_{q \in S'} \delta^*(q, b),$$

where $P'_{q'} := \{d \in \mathrm{Suc}(w) \mid q' \in S'_d\}$, there exists $b \in \eta^{-1}(t'(w))$ such that, for every $q \in S'$,

$$(\mathrm{Suc}(w), (P'_{q'})_{q' \in Q'}) \models \delta^*(q, b).$$

But this implies that

$$(\mathrm{Suc}(w), (P_q)_{q \in Q}) \models \delta(q, b),$$

for every $q \in S'$, where $P_q := \bigcup\{P'_{S''} \mid q \in S''\} \cup P'_q$. We guess the preimage $t(w) := b$ and choose $(S^q_d)_{d \in \mathrm{Suc}(w)}$ such that

$$S^q_d := \{p \in Q \mid d \in P_p\}$$

for some $q \in S'$ and set

$$v_n = f(q, w) := ((S^q_d)_{d \in \mathrm{Suc}(w)}, w) \text{ and } v'_n := f(v'_{n-1}).$$

Now it is player II's turn and he chooses

$$v_{n+1} := (q, d),$$

where $q \in S_d^q$. We assume that the choice of player II in $\mathcal{G}(\mathcal{A}', t')$ is

$$\begin{aligned} v'_{n+1} &:= (S, d) \quad \text{if } S'_d = \{S\} \subseteq \mathcal{P}(Q), \\ v'_{n+1} &:= (q, d) \quad \text{if } S'_d \subseteq Q. \end{aligned}$$

We continue this procedure to infinity and obtain infinite plays P' in $G(\mathcal{A}', t')$, P in $G(\mathcal{A}, t)$ and a tree $t : \mathrm{dom}(t) \to \Delta$ with finitely many vertices labelled from $\Delta \setminus \Sigma$. The play P satisfies the parity condition of $\mathcal{A}$ since the projection $\eta_f(t)$ is finite and the infinite play P' in $G(\mathcal{A}', \eta_f(t))$ that follows f is winning. This means that game positions that contain set states occur only finitely many times in P' and the remaining positions are of the form (q, w_n) for infinitely many $n < \omega$. But P coincides with P' up to finitely many exceptions and thus satisfies the parity condition, too.

Hence, we showed that $t \in L(\mathcal{A})$ if and only if $\eta_f(t) \in L(\mathcal{A}')$. $\square$

Corollary 34. *The class of languages recognized by weak alternating tree automata is closed under finite projection.*

4 Limits of WMSO

In the previous section we showed that for every WMSO-formula there is a weak alternating automaton that recognizes the correponding tree language. Here we present a counterexample showing that the converse statement is false. The example is based on the fact that WMSO cannot distinguish between infinitely branching trees with and without an infinite branch. Since weak alternating automata can distinguish between such trees, it follows that they are strictly stronger than WMSO on infinitely branching trees.

4.1 T_ω versus T_n

Let us introduce the trees our counterexample is based on. The result of the whole section is shown in two steps: first, we prove the m-equivalence of a tree of finite depth (which depends on m) and a tree of depth ω. Another proof for this can be found in [Kus08].

Definition 35. For $n \in \mathbb{N}$ define the tree

$$T_n = \{((a_0, i_0) \dots (a_k, i_k)) | a_j, i_j \in \mathbb{N}, n > a_0 > a_1 > \dots > a_k \text{ and arbitrary } i_0, \dots i_k\}.$$

The tree $T_\omega = \bigcup_{n \in \mathbb{N}} T_n$ is the union of all T_n.

Lemma 36. *Fix $m \in \mathbb{N}$. There exists an $n_0 \in \mathbb{N}$ such that, for every $n \in \mathbb{N}$, there is an index $k < n_0$ such that $T_n \equiv_m T_k$.*

Proof. Consider the sequence $\mathrm{Th}_m(T_0), \mathrm{Th}_m(T_1), \dots$. Since the number of m-theories is finite, choose n_0 such that every theory has occured at least once in $(\mathrm{Th}_m(T_n))_{n<n_0}$. □

Lemma 37. *(i) $T_n \cong T_k \oplus T_n$ for all $k, n \in \mathbb{N} \cup \{\omega\}$, $k \leq n$.*

(ii) For a finite subset $P \subseteq T_\omega$, we can find $n \in \mathbb{N}$ and a subset $P' \subseteq T_n$ such that

$$(T_\omega, P) \cong (T_n, P') \oplus (T_\omega, \emptyset).$$

Proof. by definiton of T_n and T_ω. □

Lemma 38. *For every natural number m there exists a natural number n_0 such that*

$$T_n \equiv_m T_\omega \quad \textit{for all } n \geq n_0.$$

Proof. We prove the claim by induction on m. For $m = 0$ the claim is trivial.

For the inductive step assume $T_n \equiv_m T_\omega$ for $n \geq n_1$. We claim that $T_n \equiv_{m+1} T_\omega$ for $n \geq \max\{n_0, n_1\}$ where n_0 is the constant from Lemma 36. We use Lemma 12 to prove the claim. Let $P \subseteq T_n$ be finite. Then

$$(T_n, P) \cong (T_n, \emptyset) \oplus (T_n, P) \equiv_m^{\text{WMSO}} (T_\omega, \emptyset) \oplus (T_n, P) \cong (T_\omega, P).$$

To prove the other direction, select a finite set $Q \subseteq T_\omega$. By Lemma 37(b), there exists $l \in \mathbb{N}$ and $Q' \subseteq T_l$ such that $(T_\omega, Q) \cong (T_\omega, \emptyset) \oplus (T_l, Q')$. If $l < n_0$, we set $(T_k, P) := (T_l, Q')$. If $l \geq n_0$, by Lemma 36 there exists $k < n_0$ such that T_l has the same $(m+1)$-theory as T_k. By Lemma 12, $\{\text{Th}_m(T_l, P) \mid P \subseteq T_l\} = \{\text{Th}_m(T_k, Q') \mid Q' \subseteq T_k\}$. Hence, select $P \subseteq T_k$ with $\text{Th}_m(T_l, Q') = \text{Th}_m(T_k, P)$. Now we have

$$(T_\omega, \emptyset) \equiv_m (T_n, \emptyset)$$

by induction hypothesis; and also $(T_l, Q') \equiv_m (T_k, P)$. Then by Lemma 21 and Lemma 37(a)

$$(T_\omega, Q) \cong (T_\omega, \emptyset) \oplus (T_l, Q') \equiv_m (T_n, \emptyset) \oplus (T_k, P) \cong (T_n, P).$$

□

4.2 T_∞ versus T_n

Definition 39. Let $\infty \notin \mathbb{N}$ be a new symbol. Then

$$T_\infty := \{\infty^k w \mid k \in \mathbb{N}, w \in T_\omega\}$$

is a tree with an infinite path.

By defnition of $\oplus'_x$, we get

Observation 40. *Let $x = \infty^k$ be a vertex on the infinite path of T_∞. Then*

$$T_\infty \cong \langle T^+, x\rangle \oplus'_x T_\infty,$$

where $T^+ = \{y \in T_\infty \mid \infty^{k+1} \not\preceq y\}$. We also have

$$T^+ \cong T_\omega.$$

Now we can prove that

Lemma 41. *For every $m \in \mathbb{N}$, we have $T_\infty \equiv_m T_\omega$.*

Proof. We prove the claim by induction on m. For $m = 0$ there is nothing to show. For the inductive step assume $T_\infty \equiv_m T_\omega$. First, let P be a finite subset of T_ω. Then there exists $P' \subseteq T_\infty$ such that $(T_\infty, P') \cong (T_\infty, \emptyset) \oplus (T_\omega, P)$. We have by induction hypothesis and Lemma 21

$$(T_\infty, P') \cong (T_\infty, \emptyset) \oplus (T_\omega, P) \equiv_m (T_\omega, \emptyset) \oplus (T_\omega, P) \cong (T_\omega, P).$$

Conversely, let $P \subseteq T_\infty$. Let l be the maximal distance from the root to an element of P. Now choose the vertex $v := \infty^l$. We divide T_∞ here and obtain by Observation 40 a tree T^+ such that

$$(T_\infty, P) \cong (T^+, v, P) \oplus'_v (T_\infty, \emptyset).$$

By Observation 40, there exists $x' \in T_\omega$ such that

$$(T_\omega, x') \cong (T^+, v),$$

which implies that there exists $Q \subseteq T_\omega$ such that

$$(T_\omega, x', Q) \cong (T^+, v, P).$$

Together with Lemma 23 and the induction hypothesis we obtain

$$(T_\infty, P) \cong (T^+, v, P) \oplus'_v (T_\infty, \emptyset) \equiv_m (T_\omega, x', Q) \oplus'_{x'} (T_\omega, \emptyset) \cong (T_\omega, Q).$$

Lemma 12 gives us now $T_\infty \equiv_{m+1} T_\omega$.

□

Now put Lemma 38 and Lemma 41 together and obtain the desired result:

Theorem 42. *For every $m \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that*

$$T_\infty \equiv_m T_n, \text{ for all } n > N.$$

In terms of automata this yields the following result.

Theorem 43. *On infinitely branching trees, weak alternating automata are strictly stronger than weak monadic second order logic.*

Proof. For any alphabet Σ and $Q = \{q\}$, let $\mathcal{A}_{\text{fin}} := (Q, \Sigma, q, \Omega)$ be a a tree automaton with transition function

$$\delta(q, a) = \forall x P_q(x)$$

and parity condition $\Omega(q) = 1$.

This automaton recognizes only trees with finite branches, since every infinite run produces a sequence of states that violates the parity condition. □

5 Conversion into Büchi Automata

In this chapter we show how to convert weak alternating automata on infinitely branching trees into Büchi automata.

In [MSS92] the same result was shown, but for weak alternating automata on infinite k-ary trees ($k \in \mathbb{N}$). Earlier, Rabin had shown in [Rab70] that a language L of binary infinte trees is definable in WMSO if and only if L and its complement $\overline{L}$ are recognised by Büchi-automata. It follows that WMSO is expressively equivalent to weak alternating automata on k-ary infinite trees.

Recall that an automaton $\mathcal{A}$ is a Büchi automaton, if the acceptance condition uses only priorities $0, 1$ and a co-Büchi automaton uses priorities $1, 2$. Note that we will not add $+1$ every time we take the complement of a Co-Büchi automaton; if $\bar{\Omega}(Q) \in \{1, 2\}$, then $\bar{\bar{\Omega}}(Q) \in \{0, 1\}$. A Büchi-automaton $\mathcal{A}$ accepts a tree t, if there exists a run $r : \mathrm{dom}(t) \to \mathcal{P}(Q \times Q)$ such that for every path $(w_n)_{n<\omega}$ of t and every sequence $(q_n)_{n<\omega} \in Q^\omega$ such that $q_0 = q_I$ and $(q_n, q_{n+1}) \in r(w')$, the sequence $(\Omega(q_n))_{n<\omega}$ meets 0 infinitely often. Formally, for every $i < \omega$, there exists $j < \omega$ such that $j \geq i$ and $\Omega(q_j)$ is even. Co-Büchi automata employ the dual condition that $(\Omega(q_n))_{n<\omega}$ meets 2 finitely many times. That is, for every $j < \omega$, there exists $i < \omega$ such that $j \geq i$ and $\Omega(q_j)$ is not even. This proves the following lemma.

Lemma 44. *The complement of a Büchi automaton is a Co-Büchi automaton and vice versa.*

Languages accepted by Büchi Automata are not closed under complement in general, cf. [Tho97, Example 6.2.].

Theorem 45. *For every weak alternating automaton $\mathcal{A}$, there exists a non-deterministic Büchi automaton $\mathcal{A}'$ such that $L(\mathcal{A}) = L(\mathcal{A}')$.*

Proof. Our proof follows that of [MSS92]. Let $\mathcal{A} = (Q, \Sigma, \delta, q_I, \Omega)$ be a weak alternating automaton. A non-deterministic Büchi automaton $\mathcal{A}' = (Q', \Sigma, \delta', q_0', \Omega')$ recognizing the same language as $\mathcal{A}$ can be constructed as follows.

Let $\bar{F} = (Q_1, \ldots, Q_r)$ be the list of all strongly connected components in the transition graph of $\mathcal{A}$ that contain states of uneven priority. The states of $\mathcal{A}'$ are 3-tuples (S, i, T) where

— the set $S \in \mathcal{P}(Q)$ keeps track of the copies of $\mathcal{A}$ running at a time;

— the index $i \in \{0, \dots, r-1\}$ indicates the strongly connected component of $\mathcal{A}$ in which $\mathcal{A}'$ is looking for rejecting subsets meeting the current state S;

— the set T contains the copies of $\mathcal{A}'$ in Q_i.

The initial state is $q_0' = (\{q_I\}, 0, \emptyset)$. Let $\mathrm{Suc}(w)$ be the successor set of a vertex w. To keep track of all copies of $\mathcal{A}$ running at a time, we have to collect the parallel successor states of every $p \in S$ in a set P_d^p such that, for every $p \in S$ and $a \in \Sigma$,

$$\mathfrak{D}_w((P_d^p)_{d \in \mathrm{Suc}(w)}) \models \delta(p, a).$$

Then we have

$$S_d := \bigcup_{p \in S} P_d^p$$

and

$$T_d := \bigcup_{p \in T} P_d^p \cap Q_i$$

for every successor $d \in \mathrm{Suc}(w)$ and index i. T_d only uses states which were already in T and still belong to S_d. Thus, the transition function of $\mathcal{A}'$ is

$$\delta' : (\mathcal{P}(Q) \times \mathbb{Z}_r \times \mathcal{P}(Q)) \times \Sigma \to \mathrm{WMSO}^+(\mathcal{P}(Q) \times \mathbb{Z}_r \times \mathcal{P}(Q))$$

such that

$$\delta'((S, i, T), a) = \bigwedge_{q \in S} \delta^*(q, a),$$

where

$$\delta^*(q,a) = \begin{cases} \delta(q,a)[\theta_0(p)/p]_{p \in Q} & \text{if } q \notin T \wedge T \neq \emptyset \\ \delta(q,a)[\theta_1(p)/p]_{p \in Q} & \text{if } T = \emptyset \\ \delta(q,a)[\theta_2(p)/p]_{p \in Q} & \text{if } q \in T, \end{cases}$$

$$\theta_0(p) = \bigvee_{\substack{S' : p \in S' \\ T' \subseteq Q_i}} P_{(S', i, T')}(x),$$

$$\theta_1(p) = \begin{cases} \displaystyle\bigvee_{\substack{S' : p \in S' \\ T' \subseteq Q_{i+1}}} P_{(S', i+1, T')}(x) & \text{if } p \notin Q_{i+1} \\ \displaystyle\bigvee_{\substack{S' : p \in S' \\ T' : p \in T'}} P_{(S', i+1, T')}(x) & \text{if } p \in Q_{i+1}, \end{cases}$$

$$\theta_2(p) = \begin{cases} \displaystyle\bigvee_{\substack{S' : p \in S' \\ T' \subseteq Q_i}} P_{(S', i, T')}(x) \text{ if } p \notin Q_i \\ \displaystyle\bigvee_{\substack{S' : p \in S' \\ T' : p \in T'}} P_{(S', i, T')}(x) \text{ if } p \in Q_i. \end{cases}$$

The priority function $\Omega' : Q' \to \{0,1\}$ is defined by

$$\Omega'(S,i,T) = \begin{cases} 0 & \text{if and only if} \quad T = \emptyset \\ 1 & \text{else.} \end{cases}$$

It remains to show that $\mathcal{A}'$ recognizes the same language as $\mathcal{A}$ does. First, we prove $L(\mathcal{A}) \subseteq L(\mathcal{A}')$. Let $r : \mathrm{dom}(t) \to \mathcal{P}(Q \times Q)$ be an accepting run of $\mathcal{A}$ on t. We construct a run $r' : \mathrm{dom}(t) \to Q'$ of $\mathcal{A}'$ on t from r by induction. Let $r'(\varepsilon) = (\{q_0\}, 0, \emptyset)$. Suppose as induction hypothesis that r' is defined up to vertex w and let $r'(w) = (S,i,T)$. For every successor $w' \in \mathrm{Suc}(w)$, we construct $r'(w') = (S', i', T')$ from $r(w)$. For every $q \in S$ we define

$$S'_q := \{q' \mid (q,q') \in r(w')\}$$

and set

$$S' := \bigcup_{q \in S} S'_q;$$

$$i' = \begin{cases} i+1 & \text{if } T = \emptyset \\ i & \text{else;} \end{cases}$$

$$T' := \{q' \mid (q,q') \in r(w'), q \in T, q' \in Q_{i'}\}.$$

If we continue this process to infinity, we obtain a run of $\mathcal{A}'$ on t. Suppose that there exists a branch $\pi \subseteq t$ such that $r'(\pi)$ does not satisfy the parity condition. This means that there exists a sequence $(S_n, i_n, T_n)_{n<\omega}$ with $T_n \neq \emptyset$, for almost every n. Then there exists n_0 and i^* such that $i_n = i^*$ for every $n \geq n_0$ and a sequence $(q_n)_{n<\omega}$ such that $(q_{n-1}, q_n) \in r(w)$ and $q_n \in Q_{i^*}$ for $n \leq n_0$. Therefore, $(q_n)_{n<\omega}$ violates the parity condition, which contradicts our assumption that r is a successful run on t. Hence, $\mathcal{A}'$ accepts t and we have shown that $L(\mathcal{A}) \subseteq L(\mathcal{A}')$.

Now we show $L(\mathcal{A}') \subseteq L(\mathcal{A})$ by constructing a run of $\mathcal{A}$ on t from a given run of $\mathcal{A}'$ on t. Let $r' : \mathrm{dom}(t) \to Q$ be an accepting run of $\mathcal{A}'$ on t. The run of $\mathcal{A}$ starts in $r(\varepsilon) = \{(q_0, q_0)\}$. Assume as induction hypothesis that the run r is already defined on w and furthermore that

$$r(w) \subseteq \{(q,q') \mid q' \in S\},$$

where $(S,i,T) = r'(w)$. Let w' be a successor of w and assume that $r'(w') = (S', i', T')$. We set

$$r(w') := \{(q', q'') \mid q' \in S, q'' \in S' \text{ and } q'' \text{ is needed by } \delta(q', t(w)) \text{ at } w'\},$$

where we can say that q'' is needed by δ at w' if it satisfies the following condition. Since

$$(\mathrm{Suc}(w), (P'_{q'})_{q' \in Q'}) \models \delta'((S,i,T), t(w)),$$

where

$$P'_{q'} = \{d \in \mathrm{Suc}(w) \mid q' \in S_d\},$$

it follows that, for every $q' \in S$,

$$(\mathrm{Suc}(w), (P'_{q'})_{q' \in Q'}) \models \delta^*(q', t(w)).$$

But this implies

$$(\mathrm{Suc}(w), (P_q)_{q \in Q}) \models \delta(q', t(w)), \text{ where } P_q = \{d \in \mathrm{Suc}(w) \mid d \in P'_{(S,i,T)} \text{ for } q \in S\}.$$

If $w' \in P_{q''}$, we say that q'' is needed by $\delta(q', (t(w)))$ at vertex w'.

If we continue according to this constuction, $\mathfrak{D}^r_{w,q} \models \delta(q, t(w))$, for every $q \in Q$ and $w \in t$ which confirms that r is a run of $\mathcal{A}$ on t. Since r' is an accepting run on t, we know that for every infinite branch $\pi \in t$, $r(\pi)$ satisfies the parity condition Ω', because $r'(\pi)$ contains no infinite sequence of states of uneven parity. This implies that for every $\pi \in t$, $r(\pi)$ also satisfies the parity condition ω. Thus, $L(\mathcal{A}') \subseteq L(\mathcal{A})$. □

From the above proof follows that

Corollary 46. *The class of languages of infinitely branching trees recognized by weak alternating automata is contained in the class of languages of infinitely branching trees recognized by Büchi and Co-Büchi automata.*

From Lemma 45 and Lemma 24 we obtain

Corollary 47. *For every formula* $\varphi \in$ WMSO *there exists a Büchi automaton and a Co-Büchi automaton recognizing* $L(\varphi)$.

Proof. Let $\varphi \in$ WMSO define the language of infinitely branching trees L_φ. Then $\neg\varphi$ defines $\overline{L_\varphi}$. By Lemma 44 and Theorem 45, both L_φ and $\overline{L_\varphi}$ are recognizable by a Büchi automaton. □

6 Translating Automata to WMSO-Formulae on Finitely Branching Trees

In [Rab70], Michael O. Rabin published the proof of the equivalence between weak monadic second-order logic over binary trees and finite non-deterministic Büchi automata. In this chapter we present the proof of the direction from automata to WMSO for arbitray finitely branching trees. The construction considers infinite trees as a limit of finite prefix trees, which is of course not possible for infinitely branching trees.

In this chapter, we use only trees where every node has a finite number of successors.

Definition 48. Let t be a finitely branching labelled tree. A *frontier* in t is a set $G \subseteq \mathrm{dom}(t)$ such that $|G \cap \pi| = 1$, for every branch $\pi \subseteq \mathrm{dom}(t)$. Obviously, if G is a frontier in t, then G is finite. For two frontiers $G_1, G_2 \subseteq \mathrm{dom}(t)$, we say that G_2 *is bigger than* G_1 ($G_2 > G_1$) if for every $y \in G_2$ there exists $x \in G_1$ such that $x \prec y$.

A *prefix* E of t is a set $E = \{x | x \preccurlyeq y \text{ for some } y \in G\}$ where G is a fixed frontier in t. We write $E \sqsubseteq t$ if E is a prefix of t. For E as above, G is called the frontier *of* E and denoted by $\mathrm{Ft}(E)$.

For a tree t, let $t_x \subseteq t$ be the subtree with root x.

Definition 49. Let $\mathcal{A}$ be a non-deterministic Büchi automaton. The set of states with priority 0 is denoted by $F = \{q \in Q \mid \Omega(q) = 0\}$ and is also called the set of *accepting states*.

Let $r \in \mathrm{Run}(\mathcal{A}, t)$. If we restrict the domain of r to a prefix $E \subseteq \mathrm{dom}(t)$, we obtain a *partial run* $r \restriction E$. The set of partial runs of $\mathcal{A}$ on t is denoted by $\mathrm{pRun}(\mathcal{A}, t)$.

Lemma 50. *For every finitely branching tree t that is recognized by a non-deterministic Büchi automaton, there exists a sequence of prefixes $E_n \sqsubseteq t$ such that $\mathrm{Ft}(E_n) \subseteq F$ and $\mathrm{Ft}(E_n) < \mathrm{Ft}(E_{n+1})$ for every $n < \omega$.*

Proof. Let $\mathcal{A}$ be a non-deterministic automaton and $t \in L(\mathcal{A})$. Fix an accepting run $r \in \mathrm{Run}(\mathcal{A}, t)$. Then for every $\pi \subseteq \mathrm{dom}(t)$, $(r(w))_{w \in \pi}$ visits states of priority 0 infinitely often. Let

$$G_n := \{x \in t \mid \text{there exists a path such that } x \text{ is the } n\text{-th vertex with } \Omega(x) = 0\}.$$

Let E_n be the prefix of t with frontier G_n. □

We want to show that a tree language L is definable in WMSO provided that both L and its complement $\bar{L}$ are recognized by non-deterministic Büchi automata. To get closer to our result, we need to study the question of when $L(\mathcal{A}) \cap L(\mathcal{B}) \neq \emptyset$, for non-deterministic Büchi automata $\mathcal{A}$ and $\mathcal{B}$.

Let $\mathcal{A} = (Q, \Sigma, \delta, q_I, \Omega)$, $\mathcal{B} = (Q', \Sigma, \delta', q'_I, \Omega')$. If $t \in L(\mathcal{A}) \cap L(\mathcal{B})$, then there are two accepting runs $r \in \mathrm{Run}(\mathcal{A}, t)$ and $r' \in \mathrm{Run}(\mathcal{B}, t)$. Hence there exists a finite prefix $E \subseteq \mathrm{dom}(t)$ and two frontiers in t, G and G', such that $G, G' < \mathrm{Ft}(E), r(G) \subseteq F$ and $r'(G') \subseteq F'$, where F and F', respectively, are the sets of accepting states. For every $x \in \mathrm{Ft}(E)$ there exists a finite subtree $E_1 \subseteq t_x$, frontiers G_1 and G'_1 of t_x such that $G_1, G'_1 < \mathrm{Ft}(E_1)$, $r(G_1) \subseteq F$ and $r'(G'_1) \subseteq F'$. And so on, for the nodes $x \in \mathrm{Ft}(E_1)$.

These considerations motivate the following construction of a sequence of subsets of $Q \times Q'$.

— Define $H_0 = Q \times Q'$.

— Define H_{i+1} inductively on i by $(q, q') \in H_{i+1}$ if and only if $(q, q') \in H_i$ and there exists a finite Σ-tree $e : E \to \Sigma$, where $\mathrm{dom}(E) \neq \{\varepsilon\}$, frontiers $G, G' < \mathrm{Ft}(E)$, and partial runs $r \in \mathrm{pRun}(\mathcal{A}, e)$ and $r' \in \mathrm{pRun}(\mathcal{B}, e)$ such that :

  (i) $r(\varepsilon) = q$, $r'(\varepsilon) = q'$;

  (ii) $r(G) \subseteq F$, $r'(G') \subseteq F'$;

  (iii) for every $x \in \mathrm{Ft}(E)$ we have $(r(x), r(x')) \in H_i$.

Also, if $H_i = H_{i+1}$, then $H_i = H_{i+k}$ for every $k < \omega$. Since $H_{i+1} \subseteq H_i$, it follows that if $|Q| \cdot |Q'| = m$, then certainly $H_m = H_{m+k}$ for $k < \omega$. With the above notations we have the following.

Lemma 51. *Let t be a Σ-tree, $\mathcal{A}$ and $\mathcal{B}$ nondeterministic Büchi automata and runs $r \in \mathrm{Run}(\mathcal{A}, t)$ and $r' \in \mathrm{Run}(\mathcal{B}, t)$. If there exists a strictly increasing sequence $(E_i)_{i \leq m}$ of finite prefixes of t, where $E_0 = \{\varepsilon\}$ and, for each $i < m$, there are two frontiers (of t) G_i, G'_i satisfying*

$$\mathrm{Ft}(E_i) \leq G_i < \mathrm{Ft}(E_{i+1}),\ r(G_i) \subseteq F$$

and

$$\mathrm{Ft}(E_i) \leq G'_i < \mathrm{Ft}(E_{i+1}),\ r'(G'_i) \subseteq F',$$

then $(q_I, q'_I) \in H_m$.

Proof. We claim that $(r(x), r'(x)) \in H_k$ for every $x \in \mathrm{Ft}(E_i)$ with $i \leq m - k$. We prove this by induction on $k \leq m$.

The base case is trivial, since $H_0 = Q \times Q'$, $(r(x), r'(x)) \in H_0$ for every $x \in t$.

As induction hypothesis, assume $k \leq m-1$ and $(r(x), r'(x)) \in H_k$, for every $x \in Ft(E_i)$, $i \leq m - k$.

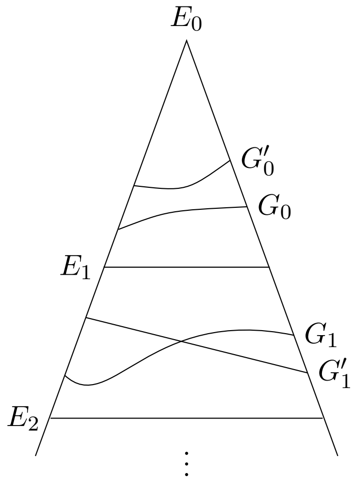


Figure 6.1: The construction of finte prefixes with accepting frontiers of Lemma 51.

Now let $k+1 \leq m$ and therefore $i \leq m-(k+1)$ and $x \in Ft(E_i)$. Consider the finite tree $E = t_x \cap E_{i+1}$. Note that $\mathrm{Ft}(E) = t_x \cap \mathrm{Ft}(E_{i+1})$ and that $G_i \cap t_x$ and $G'_i \cap t_x$ are frontiers in E (and in t_x). Now $r \restriction E$ and $r' \restriction E$ are, respectively, partial $\mathcal{A}$- and $\mathcal{B}$-runs on t. Also, $r(G_i \cap t_x) \subseteq F$ and $r'(G'_i \cap t_x) \subseteq F'$. By induction hypothesis, $r((y), r'(y)) \in H_k$ for every $x \in \mathrm{Ft}(E_i)$, $i \leq k$. Thus, $(r(x), r'(x))$ satifies the criteria to be in H_{k+1}. Since $\varepsilon \in \mathrm{Ft}(E_0)$ we have, in particular, $(q_I, q'_I) = (r(\varepsilon), r'(\varepsilon)) \in H_m$. □

Theorem 52. *$L(\mathcal{A}) \cap L(\mathcal{B}) \neq \emptyset$ if and only if $(q_I, q'_I) \in H_m$, for $m = |Q| \cdot |Q'|$.*

Proof. ($\Rightarrow$) Assume $t \in L(\mathcal{A}) \cap L(\mathcal{B})$. There exist runs $r \in \mathrm{Run}(\mathcal{A}, t)$ and $r' \in \mathrm{Run}(\mathcal{B}, t)$ such that $r(\varepsilon) = q_I$, $r'(\varepsilon) = q'_I$ and for every branch $\pi \in \mathrm{dom}(t)$, $r(\pi)$ and $r'(\pi)$ satisfy the parity condition. By Lemma 50, this implies the existence of two strictly increasing sequences $(G_i)_{i \leq m}$ and $(G'_i)_{i \leq m}$ of frontiers in t such that $r(G_i) \subseteq F$ and $r'(G'_i) \subseteq F'$. Define $\mu : \mathbb{N} \to \mathbb{N}$ increasing and $E_i \sqsubseteq t$ as follows. Let $E_0 = \emptyset$ and if E_n is already defined, choose $\mu(n)$ minimal such that $\mathrm{Ft}(E_n) < G_{\mu(n)}, G_{\mu(n)}$ and E_{n+1} such that $G_{\mu(n)}, G'_{\mu(n)} < \mathrm{Ft}(E_{n+1})$. The construction satisfies the requirements of Lemma 51. Hence, we have $(q_I, q'_I) \in H_m$.

($\Leftarrow$) To prove the converse assertion, let $(q_I, q'_I) \in H_m$. Because $H_m = H_{m+1}$, we have for every $(q, q') \in H_m$ a finite Σ-tree $e(q, q')$, where $\mathrm{dom}(e) = E(q, q') \neq \{\varepsilon\}$, partial runs $r(q, q') \in \mathrm{pRun}(\mathcal{A}, e(q, q'))$ and $r'(q, q') \in \mathrm{pRun}(\mathcal{B}, e(q, q'))$ and frontiers $G, G' < \mathrm{Ft}(E(q, q'))$ satisfying

(i) $r(q, q')(\varepsilon) = q, r'(q, q')(\varepsilon) = q'$;

(ii) $r(q, q')(G) \subseteq F$, $r'(q, q')(G') \subseteq F'$;

(iii) $x \in \mathrm{Ft}(E(q,q'))$ implies $(r(q,q')(x), r'(q,q')(x)) \in H_m$.

To get accepting runs of $\mathcal{A}$ and $\mathcal{B}$ on an infinite tree, we construct a sequence of fused finite trees and runs as follows. Start with $e(q_I, q_I')$ and define a partial $\mathcal{A}$-run $r_1(x) = r(q_I, q_I')(x)$, $x \in E(q_I, q_I)$. Note that for $x \in \mathrm{Ft}(E(q_I, q_I')), (r_1(x), r_1'(x)) \in H_m$.

Denote $e(q_I, q_I') = e_1$. Now fuse E_1 and $E(r_1(x), r_1'(x))$ at every $x \in \mathrm{Ft}(E_1)$ to obtain

$$E_2 := E_1 \cup \bigcup_{x \in \mathrm{Ft}(E_1)} xE(r_1(x), r_1'(x)).$$

Let $e_2 : E_2 \to \Sigma$, where $e_2(x) = e_1(x)$ for $x \in E_1$ and $e_2(z) = e(q,q')(y)$ for $y = xz$, where $x \in \mathrm{Ft}(E_1), z \in E(q,q')$ and $q = r_1(x)$ and $q' = r_1'(x)$. Since $e_2(x) = e_1(x)$ for $x \in \mathrm{Ft}(E_1)$, e_2 extends e_1. Extend r_1 and r_1' to partial runs $r_2 \in \mathrm{pRun}(\mathcal{A}, E_2)$ and $r_2' \in \mathrm{pRun}(\mathcal{B}, E_2)$ as follows. If $y \in (E_2 - E_1) \cup \mathrm{Ft}(E_1)$ then there is a unique $x \in \mathrm{Ft}(E_1)$ such that $y \in xE(q,q')$, where $r_1(x) = q$ and $r_1'(x) = q'$. Assume $y = xz, z \in E(q,q')$ and set $r_2(y) = r(q,q')(z)$, $r_2'(y) = r'(q,q')(z)$. Note that for $y \in \mathrm{Ft}(E_1)$, we have $r_1(y) = r_2(y)$, $r_1'(y) = r_2'(y)$. Thus r_2 and r_2' indeed extend r_1 and r_1', respectively. Also, for $x \in \mathrm{Ft}(E_2)$ we have $(r_2(x), r_2'(x)) \in H_m$. Thus this process can be continued to infinity which results in a sequence of partial Σ-trees $(e_i)_{i<\omega}$ and of runs $(r_i)_{i<\omega}$ and $(r_i')_{i<\omega}$. Let $t = \lim_{i\to\infty} e_i$; $r = \lim_{i\to\infty} r_i$; $r' = \lim_{i\to\infty} r_i'$. Then $r \in \mathrm{Run}(\mathcal{A}, t)$, $r' \in \mathrm{Run}(\mathcal{B}, t)$, $r(\varepsilon) = q_I$, $r'(\varepsilon) = q_I'$. Our construction implies that for every $x \in \mathrm{Ft}(E_i)$, $r \restriction (\mathrm{dom}(t_x) \cap E_{i+1})$ and $r' \restriction (\mathrm{dom}(t_x) \cap E_{i+1})$ coincide with some $r(q,q')$ and $r'(q,q')$, respectively. This entails the existence of two frontiers G_i, G_i' of t with $\mathrm{Ft}(E_i) \leq G_i < \mathrm{Ft}(E_{i+1})$, $\mathrm{Ft}(E_i) \leq G_i < \mathrm{Ft}(E_{i+1})$ such that $r(G_i) \subseteq F, r'(G_i') \subseteq F'$. Thus r is a successful $\mathcal{A}$-run for t and r' is a successful $\mathcal{B}$-run for t. Hence $t \in L(\mathcal{A}) \cap L(\mathcal{B})$. $\square$

Let t be a labelled tree accepted by $\mathcal{A}$. We want to express this fact by certain statements that are WMSO-definable in the structure $\mathfrak{t} = (\mathrm{dom}(t), \preccurlyeq, \mathrm{Root}, \bar{P})$. On t there exists a run $r \in \mathrm{Run}(\mathcal{A}, t)$ and an infinite sequence of prefixes $(G_i)_{i<\omega}$ such that $r(\varepsilon) = q_I$, $r(\mathrm{Ft}(G_i)) \subseteq F$, $i < \omega$. This implies that for every finite subtree $E \subseteq \mathrm{dom}(t)$ there exists a subtree $g : G \to \Sigma$ such that $E \subseteq G \subseteq \mathrm{dom}(t)$ and a partial run $r' \in \mathrm{Run}(\mathcal{A}, g)$ such that $r'(\varepsilon) = q_I$, $r'(\mathrm{Ft}(G)) \subseteq F$. Namely, $G = G_i$ for an appropriate i, and $r' = r \restriction G$. This r' has the property that for every $x \in \mathrm{Ft}(G)$, there exists a finite tree $G' \subseteq \mathrm{dom}(t_x)$ and a run $r'' \in \mathrm{Run}(\mathcal{A}, g')$ for a labelled tree $g' : G' \to \Sigma$ such that $r''(\varepsilon) = q_I$, $r''(\mathrm{Ft}(G')) \subseteq F$. Namely, $G' = G_j \cap \mathrm{dom}(t_x)$ for an appropriate $j > i$ and $r'' = r \restriction G'$. And so on. These facts can now be formalized by an inductive definition. Let $t : \mathrm{dom}(t) \to \Sigma$ be a Σ-tree.

- Define $K_0(t) = Q \times \mathrm{dom}(t)$.
- Let $q \in Q$, $x \in \mathrm{dom}(t)$, t a Σ-tree. $(q, x) \in K_{i+1}(t)$ if and only if
  (i) for every finite subtree $E \subseteq \mathrm{dom}(t_x)$, there exists a finite subtree G such that $E \sqsubseteq G \sqsubseteq t$;

(ii) for $g : G \to \Sigma$ there exists a run $r \in \mathrm{pRun}(\mathcal{A}, t)$ such that $r(x) = q$ and $r(\mathrm{Ft}(G)) \subseteq F$;

(iii) for all $y \in \mathrm{Ft}(G)$, $(r(y), y) \in K_i(t)$.

For every fixed $i < \omega$ and fixed $q \in Q$ let $K_i^q(t) := \{x \in \mathrm{dom}(t) \mid (q, x) \in K_i(t)\}$.

Lemma 53. *$K_i^q(t)$ is* WMSO*-definable, i.e. there exists $\varphi \in$* WMSO *such that*

$$\mathfrak{t} \models \varphi(x) \text{ iff } x \in K_i^q(t),$$

where $\mathfrak{t}$ is the tree structure encoding t.

Proof. We check definability in WMSO for each point of the above definition of K_i^q.

Ad (i). The finite trees G, E are represented by finite sets. The tree t_x can be defined by

$$\vartheta(x, v) := x \preccurlyeq v$$

and the prefix relation is defined by

$$A \sqsubseteq B := \forall x \forall y (A(y) \wedge x \preccurlyeq y \to A(x)) \wedge \forall x (A(x) \to B(x)).$$

The formula φ_1 says that, if E is a finite subtree of t_x, we have $E \subset G \sqsubseteq t_x$;

$$\varphi_1(E, G) := E \sqsubseteq t_x \to E \subseteq G \sqsubseteq t_x.$$

Ad (ii). Let $r : \mathrm{dom}(t) \to Q$ be a run of a Büchi automaton on t. A run r is represented via finite sets $(S_q)_{q \in Q}$, where $S_q = \{x \in \mathrm{dom}(t) \mid r(x) = q\}$. We are able to interpret $(D_w, (P_q)_{q \in Q})$ in T. Let the definition scheme be

$$\Phi_w := (\phi_w, (\psi_{P_q})_{q \in Q}),$$

where

$$\phi_w(x) := \mathrm{Suc}(w, x)$$

and

$$\psi_{P_q}(x) := S_q(x).$$

By theorem 18, we have that

$$T \models \Phi_w^\sharp(\delta'_{q,t(w)}) \quad \text{if and only if} \quad \Phi_w^*(T) \models \delta'_{q,t(w)}$$

for every $w \in \mathrm{dom}(t)$ and $q \in Q$.

F is a frontier of G if and only if

$$\forall x \exists y (G(x) \to F(y) \wedge x \preccurlyeq y) \wedge \forall x (F(x) \to G(x)) \wedge \neg \exists x \exists y (F(x) \wedge G(y) \wedge x \prec y).$$

The formula $\varphi_2(S,G)$ says "$(S_q)_q$ encodes a partial run r such that $r(x) = q$ and $r(\mathrm{Ft}(G)) \subseteq F$":

$$\varphi_2(S,G) := \bigwedge_{q\in Q} \forall x(G(x) \wedge S_q(x) \to \Phi_x^\sharp(\delta'(q,t(x))) \wedge S_q(x) \wedge \forall z[\mathrm{Ft}(G)(z) \to \bigvee_{q\in F} S_q(z)].$$

Ad (iii). We define $\varphi_3(G,\bar{S})$ saying that, for all $y \in \mathrm{Ft}(G)$, $(r(y),y) \in K_{i-1}(t)$,

$$\varphi_3(G,\bar{S}) := \forall y(\mathrm{Ft}(G)(y) \to \bigwedge_{p\in Q}[S_p(y) \to K_{i-1}^p(y)]),$$

where we asssume that we have already defined a formula for K_{i-1}^p by induction. Thus we obtain the formula

$$\varphi(x) := \forall E\exists G\exists\bar{S}(\varphi_1(E,G) \wedge \varphi_2(G,\bar{S}) \wedge \varphi_3(G,\bar{S})).$$

Hence, $x \in K_i^q(t)$ if and only if $\mathfrak{t} \models \varphi$. □

Theorem 54. *Let L be a language of finitely branching trees. If L and its complement $\bar{L}$ are recognizable by non-deterministic Büchi automata, then L is* WMSO*-definable.*

Proof. Let L be recognized by $\mathcal{A} = (Q,\Sigma,\delta,q_I,\Omega)$ and let $\mathcal{B} = (Q',\Sigma,\delta',q_I',\Omega')$ be the automaton recognizing $\bar{L}$. Let $|Q|\cdot|Q'| = m$. Thus $L(\mathcal{A}) \cap L(\mathcal{B}) = \emptyset$. We claim that $t \in L(\mathcal{A})$, if and only if $(q_I,\varepsilon) \in K_m(t)$. Since this last relation is definable in WMSO, this will prove our theorem.

($\Rightarrow$) If $t \in L(\mathcal{A})$ and r an accepting run, then $(r(x),x) \in K_i(t)$ for every $i < \omega$, $x \in \mathrm{dom}(t)$, by the remarks preceding the definition of K_i. Hence, $(q_I,\varepsilon) \in K_m(t)$.

($\Leftarrow$) Assume by way of contradiction that $(q_I,\varepsilon) \in K_m(t)$ for $t : \mathrm{dom}(t) \to \Sigma$ but $t \notin L(\mathcal{A})$, i.e. $t \in L(\mathcal{B})$. Let $r' \in \mathrm{Run}(\mathcal{B},t)$ be an accepting run of $\mathcal{B}$ on t. We show by induction on $0 \leq k \leq m-1$ that there exists a sequence of trees and runs of $\mathcal{A}$ that satisfy the conditions of Lemma 51. Define $E_0 = \{\varepsilon\}$ and let G_0' be a frontier of t such that $r'(G_0) \subseteq F'$. Since $(q_I,\varepsilon) \in K_m(t)$, there exsits a finite tree $\bar{G}_0$ and an $\mathcal{A}$-run $r_0 : \bar{G}_0 \to Q$ on $\bar{g}_0 : \bar{G}_0 \to \Sigma$ such that $r_0(\varepsilon) = q_I$ and, for $G_0 = \mathrm{Ft}(\bar{G}_0)$,

(1) $r_0(G_0) \subseteq F$ and

(2) $(r_0(x),x) \in K_{m-1}(t)$ for all $x \in G_0$.

For some $1 \leq k \leq m-2$, assume as induction hypothesis that there exists finite trees E_k and $\bar{G}_k$ such that $E_k \subseteq \mathrm{dom}(t)$ and $E_k \subseteq \bar{G}_k$, the labelled tree $\bar{g}_k : G_k \to \Sigma$ and an $\mathcal{A}$-run r_k on $\bar{G}_k$ such that, for $G_k = \mathrm{Ft}(\bar{G}_k)$

(1) $r_k(G_k) \subseteq F$ and

(2) $(r_k(x),x) \in K_{m-(k+1)}(t)$ for all $x \in G_k$.

Now let $E_{k+1} \subseteq \mathrm{dom}(t)$ a finite tree such that $G_k, G'_k < \mathrm{Ft}(E_{k+1})$. There exists a frontier G'_{k+1} of t such that $\mathrm{Ft}(E_{k+1}) \subseteq F'$. Applying the above statement (2) to each $x \in G_k$ and the finite subtree $\mathrm{dom}(t_x) \cap E_{k+1}$ of t_x, we get the existence of a finite tree $E_{k+1} \subseteq \bar{G}_{k+1}$ and an extension $r_{k+1} : \bar{G}_{k+1} \to Q$ of r_k such that r_{k+1} is an $\mathcal{A}$-run on $\bar{g}_{k+1} : \bar{G}_{k+1} \to \Sigma$ and, for $G_{k+1} = \mathrm{Ft}(\bar{G}_{k+1})$,

(1) $r_{k+1}(G_{k+1}) \subseteq F$ and

(2) $(r_k(x), x) \in K_{m-k+2}(t)$ for all $x \in G_{k+1}$.

Extend the run $r_{m-1} : \bar{G}_{m-1} \to Q$ in some way to a run $r \in \mathrm{Run}(\mathcal{A}, t)$. Then $(E_i)_{i<m}$, r and r' satisfy the conditions of Corollary 51. Hence $L(\mathcal{A}) \cap L(\mathcal{B}) \neq \emptyset$, a contradiction. Thus $(q_I, \varepsilon) \in K_m(t)$ if and only if $t \in L(\mathcal{A})$. $\square$

7 Equivalence of MSO and WMSO on Trees of Finite Cantor-Bendixon Rank

In this chapter we consider the question for which kind of trees can we define MSO-definable tree languages also in WMSO. To this end we need a topological characterisation of infintely branching trees for which we can define the notion of a rank that depends on the number of nested infinite branches in a tree. It was shown in [BIS13] that MSO-definable languages of finitely branching thin trees are WMSO-definable if and only if the rank is bouded by a natural number. First, we show that this result does not hold for infinitely branching trees. Then we present a proof for finitely branching trees without algebraic methods.

The notion of Cantor-Bendixon rank stems from a topological setting, cf. [Kec95]. Since it is possible to think of a tree as the topological space of its infinite branches (consult [PP04] for details), we have the following definition. For a given tree, delete those nodes x whose subtree T_x has only finitely many infinite branches. Repeat the procedure until a tree without infinite branches is left. The least number of repetitions is the Cantor Bendixon rank of the tree. Formally,

Definition 55. For a tree T define

$$T' := T \setminus \{x \in T : T_x \text{ has finitely many infinite branches}\}.$$

For $\alpha \in \mathrm{ORD}$, let

$$T^0 := T,$$

$$T^{\alpha+1} := (T^\alpha)',$$

$$T^\delta := \bigcap_{\alpha<\delta} T^\alpha, \text{ if } \delta \text{ is limit.}$$

The *Cantor-Bendixon rank* is defined by

$$\mathrm{CB}(T) = \min\{\alpha \mid T^\alpha \text{ has no infinite branches}\}.$$

In a tree T, we say that *node x is of rank n* if $\mathrm{CB}(T_{\restriction x}) = n$ for the subtree $T_{\restriction x} \subseteq T$.

7.1 Counterexample for Infinitely Branching Trees

The first lemma presents a couterexample to the equivalence of MSO and WMSO on trees of Cantor Bendixon rank 1, which have finitely many infinite paths.

Lemma 56. *There exist trees T and T' such that* $\mathrm{CB}(T) = \mathrm{CB}(T') = 1$ *and*

$$T \equiv_{\mathrm{WMSO}} T', \ \textit{but } T \not\equiv_{\mathrm{MSO}} T'.$$

Proof. We consider two trees with one infinite branch together with a unary predicate P. In one of the trees the infinite branch is colored by P and in the other tree it is not. More precisely, we define

$$T := \langle T_\omega, \emptyset \rangle \oplus \langle T_\infty, T_\infty \rangle \cong \langle T_\infty, P \rangle \quad \text{and} \quad T' := \langle T_\omega, T_\omega \rangle \oplus \langle T_\infty, \emptyset \rangle \cong \langle T_\infty, P' \rangle,$$

where T_ω and T_∞ are the trees defined in Chapter 4.

To prove that $T \not\equiv_{\mathrm{MSO}} T'$, we can directly write an MSO-formula that distinguishes the trees. Define

$$\mathrm{Branch}(X) := (\forall x \in X)(\forall y \in X)(x \preccurlyeq y \vee y \preccurlyeq x) \wedge \neg \exists z((\forall x \in X)(x \preccurlyeq z)).$$

Then,

$$T \models \exists X(\mathrm{Branch}(X) \wedge X \subseteq P))$$

and

$$T' \not\models \exists X(\mathrm{Branch}(X) \wedge X \subseteq P)).$$

It remains to prove that $T \equiv_{\mathrm{WMSO}} T'$. By Lemma 41, we have

$$T_\omega \equiv_{\mathrm{WMSO}} T_\infty,$$

which implies that

$$\langle T_\omega, \emptyset \rangle \equiv_{\mathrm{WMSO}} \langle T_\infty, \emptyset \rangle$$

and

$$\langle T_\omega, T_\omega \rangle \equiv_{\mathrm{WMSO}} \langle T_\infty, T_\infty \rangle.$$

Thus, we obtain

$$\langle T_\omega, \emptyset \rangle \oplus \langle T_\infty, T_\infty \rangle \equiv_{\mathrm{WMSO}} \langle T_\omega, T_\omega \rangle \oplus \langle T_\infty, \emptyset \rangle$$

by using the preservation of $\equiv_{\mathrm{WMSO}}$ under $\oplus$ (Lemma 21).

Hence, $T \equiv_{\mathrm{WMSO}} T'$. □

7.2 A Proof for Finitely Branching Trees

The goal of the remainder of the chapter is to prove that for a finitely branching tree T of Cantor Bendixon rank $n < \omega$ and a given MSO_m-theory, we are able to define in WMSO that $\mathrm{Th}_m^{\mathrm{MSO}}(T) = \theta$ and $\mathrm{CB}(T) = n$. The fact is stated in the following theorem.

Theorem 57. *Let θ be an MSO_m-theory and $n < \omega$. There exists a* WMSO*-formula $\varphi_\theta^n(x)$ such that $T \models \varphi_\theta^n(x)$ if and only if $\mathrm{Th}_m(T_{\restriction x}) = \theta$, for every finitely branching tree T of rank n and every $x \in T$.*

The first step is to show that $\mathrm{CB}(T) = n$ is definable in WMSO.

Lemma 58. *There exists a formula $\varphi_n(x) \in$ WMSO such that $T \models \varphi_n(x)$ if and only if $\mathrm{CB}(T_{\restriction x}) \leq n$, for every finitely branching tree T with vertex $x \in T$.*

Proof. We prove the claim by induction on n.

Let $n = 0$. Finitely branching trees with $\mathrm{CB}(T) = 0$ are finite. Then

$$\varphi_0(x) := \exists X \forall y[(x \preccurlyeq y) \to (y \in X)].$$

If $T \models \varphi_0(x)$, then every node below x is contained in a finite set, hence $T_{\restriction x}$ is finite.

If $T_{\restriction x} \subseteq T$ is finite then there exists a finite set that contains every node of $T_{\restriction x}$, thus $T \models \varphi_0(x)$.

Assume that we have already defined the formula φ_n. A WMSO-formula for $n+1$ has to say that there exists a finite antichain C such that

— every node of rank $> n$ is comparable with C

— below every node of C there is only one path whose nodes have rank $> n$.

This can be formalized in WMSO. We use the abbreviation for "C is a finite antichain"

$$\mathrm{Antichain}(C) := (\forall x \in C)(\forall y \in C)((x \preccurlyeq y \vee y \preccurlyeq x) \to x = y)$$

to obtain the formula

$$\begin{aligned}\varphi_{n+1}(x) := &\exists C[\mathrm{Antichain}(C) \wedge (\forall c \in C)(x \preccurlyeq c)\\ &\wedge \forall z[x \preccurlyeq z \wedge \neg\varphi_n(z) \to (\exists y \in C)(z \preccurlyeq y \vee y \preccurlyeq z)]\\ &\wedge (\forall x \in C)(\forall y \forall z)[x \preccurlyeq y \wedge x \preccurlyeq z \wedge \neg\varphi_n(y) \wedge \neg\varphi_n(z) \to (y \preccurlyeq z \vee z \preccurlyeq y)]].\end{aligned}$$

Assume $\mathrm{CB}(T_{\restriction x}) = n+1$ for $T_{\restriction x} \subseteq T$. By definition, the Cantor Bendixon derivative $(T_{\restriction x})^n$ consists of finitely many infinite branches, where every node is of rank 1. For every infinite branch β fix a vertex $x \in \beta$ such that no other infinite branch contains a vertex $y \succeq x$. Let C be the set of those vertices x. Then $T \models \varphi_{n+1}(x)$.

Now assume $T \models \varphi_{n+1}(x)$. Then there exists a finite antichain C such that for every node y of rank $> n$ there exists a path that contains y and a node of C. Any of these paths do not branch anymore below C. Thus we have finitely many infinite branches that contain only nodes of rank $> n$. Assume that these nodes were of rank $> n+1$. Hence, $(T_{\restriction x})^n$ has only finitely many infinite branches and $(T_{\restriction x})^{n-1} = \emptyset$. Thus $\mathrm{CB}(T_{\restriction x}) = n+1$. $\square$

To conclude the proof of Theorem 57, we need a few more steps. First, we show the claim for trees that consist only of one branch of Cantor Bendixon rank n.

Assume that T has exactly one branch π whose nodes are of CB–rank n. Let

$$\mathrm{S}(x) := \{y \in T : \mathrm{Suc}(x, y) \wedge \neg\varphi_{n-1}(y)\}$$

be the successor of x that is also in π. We define

$$A(x) = T_{\restriction x} \setminus T_{\restriction \mathrm{S(x)}}$$

and obtain the decomposition

$$T := \sum_{x \in \pi} A(x).$$

Now we state a composition lemma that shows the conditions to preserve the MSO-theory of a branch of T from the MSO-theory in the interpreted labelled linear ordering.

Lemma 59. *There exists a function f with the following property. Let $T = \sum_{n<\omega} T_n$ and $T' = \sum_{n<\omega} T'_n$ be trees and $m < \omega$. For every MSO_m-theory τ, we define*

$$H_\tau := \{n \in \omega : \mathrm{Th}_m(T_n) = \tau\} \quad \textit{and} \quad H'_\tau := \{n \in \omega : \mathrm{Th}_m(T'_n) = \tau\}.$$

Then

$$\langle \omega, \leq, (H_\tau)_\tau \rangle \equiv^{\mathrm{MSO}}_{f(m)} \langle \omega, \leq, (H'_\tau)_\tau \rangle$$

implies

$$\sum_{n<\omega} T_n \equiv^{\mathrm{MSO}}_m \sum_{n<\omega} T'_n.$$

Proof. Define the function f inductively by

$$f(0) = 0, \quad f(m+1) = f(m) + k,$$

where k is number of MSO_{m+1}-theories. We apply Lemma 12 saying that

$$\sum_{n<\omega} T_n \equiv^{\mathrm{MSO}}_m \sum_{n<\omega} T'_n$$

if and only if

— for every P there exists P' such that $(\sum_{n<\omega} T_n, P) \equiv^{\mathrm{MSO}}_{m-1} (\sum_{n<\omega} T'_n, P')$

— and for every P' there exists P such that $(\sum_{n<\omega} T_n, P) \equiv^{\mathrm{MSO}}_{m-1} (\sum_{n<\omega} T'_n, P')$.

By symmetry, it is sufficient to show that for every $P \subseteq \sum_{n<\omega} T_n$, we find $P' \subseteq \sum_{n<\omega} T'_n$ such that

$$(\sum_{n<\omega} T_n, P) \equiv^{\mathrm{MSO}}_{m-1} (\sum_{n<\omega} T'_n, P').$$

Assume $\langle \omega, \leq, (H_\tau)_\tau \rangle \equiv_{f(m)} \langle \omega, \leq, (H'_\tau)_\tau \rangle$. Adding P to $\sum_{n<\omega} T_n$ we obtain a new labelling $(I_\sigma)_\sigma$ in $\langle \omega, \leq, (H_\tau)_\tau \rangle$ where

$$I_\sigma = \{ n \in \omega \mid \mathrm{Th}^{\mathrm{MSO}}_{m-1}(T_n, P_n) = \sigma \},$$

with $P_n = T_n \cap P$, for each MSO_{m-1}-theory σ. Since $\langle \omega, \leq, (H_\tau)_\tau \rangle \equiv_{f(m)} \langle \omega, \leq, (H'_\tau)_\tau \rangle$, we find for every $I_\sigma \subseteq \omega$ a set $I'_\sigma \subseteq \omega$ such that

$$(\omega, \leq, (H_\tau)_\tau, (I_\sigma)_\sigma) \equiv_{f(m)-k} (\omega, \leq, (H'_\tau)_\tau, (I'_\sigma)_\sigma),$$

where k is the number of MSO_{m-1}-theories σ. Given $n < \omega$, fix σ and τ such that $n \in H_\tau$ and $n \in I_\sigma$. Then there exists a fomula $\varphi_\sigma \in \tau$ saying that there exists a predicate P such that $\mathrm{Th}^{\mathrm{MSO}}_{m-1}(T_n, P_n) = \sigma$. For every $n < \omega$, choose $P'_l \subseteq T'_l$ such that $\mathrm{Th}^{\mathrm{MSO}}_{m-1}(T'_l, P'_l) = \sigma_n$, where σ_n is the MSO_{m-1}-theory such that $n \in I'_{\sigma_n}$. Then

$$P' := \bigcup_{l<\omega} P'_l$$

has the desired properties.

□

Corollary 60. *For every* MSO*-formula* φ*, there exists an* MSO*-formula* φ' *such that*

$$\sum_{n<\omega} T_n \models \varphi, \quad \textit{if and only if} \quad \langle \omega, \leq, (H_\tau)_\tau \rangle \models \varphi'.$$

Proof. By Lemma 59, there exists a function h mapping

$$\mathrm{Th}^{\mathrm{MSO}}_{f(m)}(\langle \omega, \leq, (H_\tau)_\tau \rangle) \quad \text{to} \quad \mathrm{Th}^{\mathrm{MSO}}_{m}(\sum_{n<\omega} T_n).$$

Thus, we have

$$\begin{aligned}
\sum_{n<\omega} T_n \models \varphi \quad & \text{iff} \quad \varphi \in h(\mathrm{Th}_{f(m)}(\langle \omega, \leq, (H_\tau)_\tau \rangle)) \\
& \text{iff} \quad \mathrm{Th}_{f(m)}(\langle \omega, \leq, (H_\tau)_\tau \rangle) \in h^{-1}(\{ \theta \mid \theta \ \mathrm{MSO}_m\text{-theory}, \ \varphi \in \theta \}) \\
& \text{iff} \quad \langle \omega, \leq, (H_\tau)_\tau \rangle \models \bigvee \{ \bigwedge \eta \mid \eta \, \mathrm{MSO}_{f(m)}\text{-theory}, \ \varphi \in h(\eta) \} := \varphi'.
\end{aligned}$$

□

The same idea of composing trees and describing their theory through a labelling of the natural numbers is applied in the following lemma. This time we compose the trees at their root and do not need an ordering on $[n]$. The definition of the function f as well as the proof method is the same as in the proof of Lemma 60.

Lemma 61. *Let $n < \omega$. For each* MSO_m*-theory η, we define unary predicates*

$$H_\eta := \{i \in [n] \mid T_i \models \eta\} \text{ and } H'_\eta := \{i \in [n] \mid T'_i \models \eta\}.$$

There exists a function f such that

$$\langle [n], (H_\eta)_\eta \rangle \equiv^{\mathrm{MSO}}_{f(m)} \langle [n], (H'_\eta)_\eta \rangle$$

implies

$$\bigoplus_{i \leq n} T_i \equiv^{\mathrm{MSO}}_m \bigoplus_{i \leq n} T'_i.$$

Corollary 62. *For every* MSO*-formula φ, there exists an* MSO*-formula φ' such that*

$$\bigoplus_{i<n} T_n \models \varphi, \quad \text{if and only if} \quad \langle [n], (H_\eta)_\eta \rangle \models \varphi'.$$

To transform the result from Lemma 59 to Corollary 62 to WMSO, we need a theorem of McNaughton that was first stated in terms of automata, but is equivalent to the following formulation. Consider [PP04] for details.

Theorem 63 (McNaughton)**.** *For every $\varphi \in$ MSO there exists a $\varphi' \in$ WMSO such that*

$$\langle \omega, \leq, \overline{P} \rangle \models \varphi \quad \text{if and only if} \quad \langle \omega, \leq, \overline{P} \rangle \models \varphi'.$$

Lemma 64. *Let θ be an* MSO_m*-theory and $n < \omega$. For every tree T of rank n that is of the form*

$$T_{\restriction x} := \sum_{\substack{y \in \pi \\ x \leq y}} A(y) \text{ with } \mathrm{CB}(A(y)) < n,$$

for some path π, there exists a WMSO*-formula $\varphi^n_\theta(x)$ such that*

$$T \models \varphi^n_\theta(x) \quad \text{if and only if} \quad \mathrm{Th}_m(T_{\restriction x}) = \theta.$$

Proof. To prove the above lemma, we need to interpret the path π in the tree structure T. Such a structure is given by $\langle \omega, \leq, (H_\tau)_\tau \rangle$, where τ stands for an MSO_m-theory and

$$H_\tau := \{x \in \pi : \mathrm{Th}_m(A(x)) = \tau\}.$$

For an MSO_m-theory τ we set

$$\varphi^n_\tau(x) := \varphi_n(x) \wedge \bigwedge \tau.$$

We define the interpretation

$$\mathcal{I} : T \to \langle \omega, \leq, (H_\tau)_\tau \rangle$$

by the definition scheme

$$\langle \phi_\omega, \psi_\leq, (\psi_{H_\tau})_\tau \rangle,$$

where

$$\begin{aligned} \phi_\omega(x) &:= \neg\varphi_{n-1}(x), \\ \psi_\leq(x, y) &:= x \preccurlyeq y \wedge \neg\varphi_{n-1}(x) \wedge \neg\varphi_{n-1}(y), \\ \psi_{H_\tau}(x) &:= (\varphi_\tau^{n-1})^{(A(x))}(x), \end{aligned}$$

where $A(x, y) := x \preccurlyeq y \wedge \forall z[x \prec z \preccurlyeq y \implies \neg\varphi_{n-1}(z)]$.

Now we can create the formula that is needed to prove Lemma 64. We want $\varphi_\theta^n(x) \in$ WMSO to say that

— $\mathrm{CB}(T_{\restriction x}) = n$ and

— $T_{\restriction x} \models \theta_x$.

By Lemma 58, the first formula is already in WMSO. To define the second part, we know that

$$T_{\restriction x} = \sum_{\substack{y \in \pi \\ x \leq y}} A(y).$$

We use Corollary 60 and McNaughton's Theorem and obtain a formula $\theta_x'' \in$ WMSO such that

$$\langle \omega, \leq, (H_\tau)_\tau \rangle \models \theta_x'' \quad \text{if and only if} \quad \sum_{\substack{y \in \pi \\ x \leq y}} A(y) \models \theta''$$

and use the interpretation $\mathcal{I}$ to get that

$$T_{\restriction x} \models \mathcal{I}(\theta'') \quad \text{if and only if} \quad \langle \omega, \leq, (H_\tau)_\tau \rangle \models \theta_x''.$$

Thus we obtain the desired formula $\varphi_\theta^n(x) := \varphi_n(x) \wedge \mathcal{I}(\theta_x'')$. □

To prove Theorem 57, we introduce composition operations $\oplus_l$ and $\oplus'$. $\oplus_l$ and $\oplus'$ are almost the same as $\oplus_l'$ and $\oplus$ (that were presented in Chapter 2.3.2), they compose two trees once at the root and once at a leaf l, but this time with a new edge between the trees instead of fusing them.

Definition 65. Let S and T be trees. We define the composed structure $S \oplus' T$ by

$$S \oplus' T := \langle V_{S \oplus' T}, \mathrm{Root}^{S \oplus' T}, \preccurlyeq^{S \oplus' T} \rangle,$$

where

$$\begin{aligned} V_{S\oplus' T} &:= V_S \dot\cup V_T \dot\cup \{r\} \\ \mathrm{Root}^{S\oplus' T} &:= \{r\} \\ \preccurlyeq^{S\oplus' T} &:= \preccurlyeq^S \dot\cup \preccurlyeq^T \cup \{(r,x) \mid x \in V_T \dot\cup V_S\}. \end{aligned}$$

The structure $S \oplus_l T$ is defined as

$$S \oplus_l T := \langle V_{S\oplus_l T}, \mathrm{Root}^{S\oplus_l T}, \preccurlyeq^{S\oplus_l T}\rangle,$$

where

$$\begin{aligned} V_{S\oplus_l T} &:= V_S \dot\cup V_T, \\ \mathrm{Root}^{S\oplus_l T} &:= \mathrm{Root}^T, \\ \preccurlyeq^{S\oplus_l T} &:= \preccurlyeq^S \dot\cup \preccurlyeq^T \cup \{(v,w) \mid v \preccurlyeq l, w \in S\} \end{aligned}$$

Lemma 66. *The* MSO_m*-theory of* $S \oplus' T$ *is uniquely determined by the* MSO_m*-theories of* S *and* T *and the theory of* $S \oplus_l T$ *is uniquely determined by the* MSO_m*-theories of* S *and* $\langle T, l\rangle$*.*

Proof. Let U be the singleton tree. We have $S \oplus' T = \Phi^*_{\oplus'}(S \dot\cup T \dot\cup U)$, where the definition scheme $\Phi_{\oplus'}$ is

$$\begin{aligned} \phi(x) &= \top; \\ \psi_{\mathrm{Root}}(x) &= U(x); \\ \psi_{\preccurlyeq}(x,y) &= x \preccurlyeq y \vee U(x). \end{aligned}$$

Further, $S \oplus_l T = \Phi^*_{\oplus_l}(S \dot\cup T)$, where the definition scheme $\Phi_{\oplus_l}$ is defined through

$$\begin{aligned} \phi'(x) &= \top; \\ \psi'_{\mathrm{Root}}(x) &= \mathrm{Root}(x) \wedge T(x); \\ \psi'_{\preccurlyeq}(x,y) &= x \preccurlyeq y \vee (x \preccurlyeq l \wedge S(y)). \end{aligned}$$

Like in the proof of Lemma 23 the claim follows from Lemma 19 and Lemma 15. □

Definition 67. Let $* \in \{\oplus, \oplus', \oplus_l, \oplus'_l\}$. We define the *composition of theories*

$$\eta * \zeta := \mathrm{Th}^{\mathrm{MSO}}_m(\mathfrak{A} * \mathfrak{B})$$

for some structures $\mathfrak{A}, \mathfrak{B}$ with $\mathrm{Th}^{\mathrm{MSO}}_m(\mathfrak{A}) = \eta$ and $\mathrm{Th}^{\mathrm{MSO}}_m(\mathfrak{B}) = \zeta$.

Proof of Theorem 57. We prove the claim by induction on the Cantor-Bendixon rank of T.

For $\mathrm{CB}(T) = 0$, the tree is finite and the claim is trivial since for finite trees, we have $\mathrm{Th}_m^{\mathrm{MSO}}(T) = \mathrm{Th}_m^{\mathrm{WMSO}}(T)$. The induction hypothesis is that for an MSO_m-theory θ and a fixed number $n \in \mathbb{N}$, there exists a WMSO-formula $\varphi_\theta^n(x)$ such that $T \models \varphi_\theta^n(x)$ if and only if $\mathrm{Th}_m(T_{\restriction x}) = \theta$ and $\mathrm{CB}(T_{\restriction x}) = n$, for every finitely branching tree T of rank n.

To prove the claim for trees of rank $n+1$, we need to define those vertices in T where branches of rank $n+1$ are branching into further paths of rank $n+1$. We call those vertices branching points and define them by the formula $\vartheta_{n+1}(x)$ saying "x is a branching point of rank $n+1$".

$$\begin{aligned}\vartheta_{n+1}(x) :=& \exists y \exists z[\neg\varphi_n(x) \wedge \neg\varphi_n(y) \wedge \neg\varphi_n(z) \wedge ((\mathrm{Suc}(x,y) \wedge \mathrm{Suc}(x,z) \wedge y \neq z) \\ & \vee (\mathrm{Suc}(y,x) \wedge \mathrm{Suc}(y,z) \wedge y \neq x))] \vee \mathrm{Root}(x).\end{aligned}$$

Of course it happens that in such a branching point there are also successors of rank $< n+1$. To define those points y in relation to the respective branching point x, we have the formula

$$\psi_{n+1}(x,y) := x \preccurlyeq y \wedge \varphi_n(y) \wedge \neg\exists z[x \prec z \prec y \wedge \neg\varphi_n(z)].$$

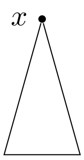
To describe the theory of T at the root, we guess a theory ξ_x for every branching point x, that is composed of the theories of the subtrees of $T_{\restriction x}$. We start this composition of theories at the biggest branching points with respect to $\preccurlyeq$ and proceed up to the root of T.

The WMSO-formula guesses the predicates

$$P_\xi := \{x \in T \mid \mathrm{Th}_m^{\mathrm{MSO}}(T_{\restriction x}) = \xi\}.$$

During this process we encounter three diffrent types of branching points with different kinds of composed subtrees.

(i) The first one is the easiest case, where the branching point x is on a unique branch of T^n. By Lemma 64, the theory ξ_x of $T_{\restriction x}$ is defined by $\varphi_\xi^n(x)$.

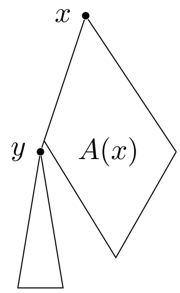


Thus the formula defining this situation is

$$\varphi^\dagger(x) := \neg\exists y[x \prec y \wedge \vartheta(y)] \wedge \bigwedge_\xi (\varphi_\xi^n(x) \leftrightarrow P_\xi(x)).$$

(ii) The branching point x is of the second type, if x is followed by a single path of rank $< n+1$ that is followed by another branching point y. Let

$$A(x) := T_{\restriction x} \setminus T_{\restriction y}.$$



Then we have

$$T_{\restriction x} := T_{\restriction y} \oplus_l A(x),$$

where l is the vertex such that $T_{\restriction x} \models \mathrm{Suc}(l, y)$. This is defined in WMSO by

$$\alpha(x) := \vartheta_{n+1}(x) \wedge \exists y (x \prec y \wedge \forall z (x \prec z \wedge \vartheta_{n+1}(z) \rightarrow y \preccurlyeq z) \wedge \vartheta_{n+1}(y)).$$

The WMSO-formula defining $\mathrm{Th}_m^{\mathrm{MSO}}(T_{\restriction x})$ is

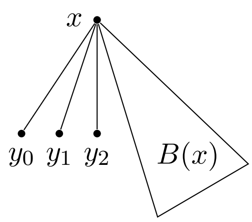
$$\varphi^\sharp(x) = \alpha(x) \wedge \bigvee_{(\eta,\zeta)} [P_{\eta \oplus_l \zeta}(x) \wedge P_\eta(y) \wedge (\varphi_\zeta^n)^{(A(x))}].$$

(iii) The branching point x can also be of a third type, where x has more than one successor $y_0, \dots, y_k$ of CB-rank $n+1$. Then

$$T_{\restriction x} = (T_{\restriction y_0} \oplus \dots \oplus T_{\restriction y_k}) \oplus_x B(x),$$

where

$$B(x) := T_{\restriction x} \setminus \bigcup \{T_{\restriction y} \mid \mathrm{Suc}(x, y) \wedge \neg \varphi_n(y)\}.$$



This is defined by the formula

$$\beta(x) := \exists y_1 \exists y_2 [y_1 \neq y_2 \wedge \mathrm{Suc}(x, y_1) \wedge \mathrm{Suc}(x, y_2) \wedge \neg\varphi_n(y_1) \wedge \neg\varphi_n(y_2)].$$

Let $T_1 = T_{\restriction y_0} \oplus \ldots \oplus T_{\restriction y_{k-1}}$. We define the interpretation

$$\mathcal{I} : T_1 \to \langle [n], (H_\eta)_\eta \rangle$$

by the definition scheme

$$\langle \phi_{[n]}, (\psi_{H_\eta})_\eta \rangle,$$

where

$$\begin{aligned} \phi_{[n]}(x, y) &:= \mathrm{Suc}(x, y) \wedge \neg\varphi_n(y), \\ \psi_{H_\eta}(y) &:= P_\eta(y). \end{aligned}$$

Let $\chi_\zeta := \bigwedge \zeta$. By Corollary 62 we know that for χ_ζ, there exists $\chi'_\zeta \in$ MSO such that

$$\bigoplus_{i<n} T_i \models \chi_\zeta \quad \text{if and only if} \quad \langle [n], (H_\eta)_\eta \rangle \models \chi'_\zeta.$$

Since $[n]$ is finite, χ'_ζ is also in WMSO. By the interpretation $\mathcal{I}$ we obtain that

$$\langle [n], (H_\eta)_\eta \rangle \models \chi'_\zeta \quad \text{if and only if} \quad T_1 \models \mathcal{I}(\chi'_\zeta).$$

The formula to define the m-theory of $T_{\restriction x}$, if $\beta(x)$ holds, is then

$$\varphi^*(x) := \beta(x) \wedge \bigvee_{(\eta,\zeta)} [P_{\eta \oplus_l \zeta}(x) \wedge \mathcal{I}(\chi'_\zeta) \wedge (\varphi^n_\eta)^{(B(x))}(x)].$$

Now, the WMSO-formula to describe the MSO$_m$-theory at every branching point $x \in T$ is

$$\varphi^{n+1}_\theta(x) := \exists (P_\xi)_\xi [\forall y (\vartheta_{n+1}(y) \to \varphi^\dagger(y) \vee \varphi^*(y) \vee \varphi^\sharp(y)) \wedge P_\theta x]^{(T_{\restriction x})}.$$

□

Bibliography

[BB01] Dietmar Berwanger and Achim Blumensath. The monadic theory of tree-like structures. In *Automata, Logics, and Infinite Games*, pages 285–302, 2001.

[BIS13] Mikołaj Bojańczyk, Tomasz Idziaszek, and Michał Skrzypczak. Regular languages of thin trees. *STACS*, 2013.

[Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik und Grundl. Math.*, 6:66–92, 1960.

[EF95] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory.* Perspectives in Mathematical Logic. Springer, 1995.

[Kec95] A.S. Kechris. *Classical Descriptive Set Theory.* Graduate Texts in Mathematics. Springer, 1995.

[Kus08] Dietrich Kuske. Compatibility of shelah and stupp's and muchnik's iteration with fragments of monadic second order logic. In *STACS*, pages 467–478, 2008.

[Mak04] J. A. Makowsky. Algorithmic uses of the feferman-vaught theorem. *Annals of Pure and Applied Logic*, 126:159–213, 2004.

[McN66] Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966.

[MSS92] David E. Muller, Ahmed Saoudi, and Paul E. Schupp. Alternating automata, the weak monadic theory of trees and its complexity. *Theor. Comput. Sci.*, 97(2):233–244, 1992.

[PP04] D. Perrin and J.É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games.* Number Bd. 141 in Pure and Applied Mathematics. Elsevier, 2004.

[Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Soc.*, 141:1–35, 1969.

[Rab70] Michael O. Rabin. Weakly definable relations and special automata. In Yehoshua Bar-Hillel, editor, *Proceedings of the Symposium on Mathematical Logic and Foundations of Set Theory (SMLFST'70)*, pages 1–23. North-Holland, 1970.

[Rab72] Michael O. Rabin. *Automata on infinite objects and Church's problem.* CBMS Regional Conference Ser. in Mathematics Series. Conference Board of the Mathematical Sciences, 1972.

[Tho97] Wolfgang Thomas. Languages, automata and logics. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer-Verlag, 1997.

[Wal02] Igor Walukiewicz. Monadic second-order logic on tree-like structures. *Theor. Comput. Sci.*, 275(1-2):311–346, 2002.