Investigations into the Expressiveness of First-order Logic and Weak Path Automata on Infinite Trees

MSc Thesis *(Afstudeerscriptie)*

written by

Chase Ford
(born 1st of October, 1990 in Fort Worth, TX)

under the supervision of **Prof.dr. Yde Venema** and **Dr. Sam van Gool**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:
27th of September, 2019

Members of the Thesis Committee:
Dr. Bahareh Afshari
Dr. Alexandru Baltag
Dr. Helle Hvid Hansen
Dr. Katia Shutova (Chair)

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

In this thesis, we investigate the expressive power of first-order logic and alternating parity automata on unranked trees with no leaves. While the initial aim of this thesis was to provide a full characterization of first-order logic as a class of automata, slightly less is achieved. In particular, we introduce several closely related classes of alternating parity automata and prove that they effectively bound the expressive power of first-order logic over such structures. Inspired by automata-theoretic characterizations of first-order logic over word structures, the automata classes considered in this thesis are obtained by imposing weak acceptance conditions, antisymmetry of the reachability relation on states (also known as aperiodicity), and what we call the path condition. The essence of the latter is the semantic notion of complete additivity. In the final chapter, we investigate the bisimulation-invariant subclass of the lower bound of the first-order 'sandwich' given in this thesis using methods developed in the work of Janin and Walukiewicz.

Acknowledgements

I would like to begin by acknowledging everyone that made this thesis possible. First and foremost, I would like to thank Yde Venema and Sam van Gool for supervising this thesis; their expertise on logic and automata theory was a guiding wind in its development. When times got tough, they both exhibited nothing less than endless patience and solid advice.

Yde's expertise of alternating parity automata, their relation with MSO, and his understanding of the Janin-Walukiewicz Theorem made this thesis possible. His guidance was particularly crucial when it came to proving the closure of automata under projection. I also wish to thank Yde for his constant enthusiasm and willingness to meet with me so frequently: when things went awry, he even met with me (almost) daily for a week straight. It was truly a privilege to get to know you and to have you as my supervisor: the many lessons you taught me will surely echo through my mind for some time!

I thank Sam for traveling so often from Utrecht for our meetings. His insight and suggestions were always valuable, and several of his ideas made their way into this thesis. I want to thank him especially for teaching me about translating word automata into formulas and for the many clarifying suggestions that he made.

I thank Dick de Jongh, Nick Bezhanishvili, Benno van den Berg, Frederik Lauridsen, and Lorenzo Galeotti for their support, conversations, and the many lessons they taught me about mathematical logic. Additionally, I thank the ILLC staff–especially Tanja and Jenny–for being there when I needed them the most. A special thanks goes to the thesis committee for agreeing to read and review this thesis.

I would like to thank Brian Walter, David McAvity, Rachel Hastings, and Richard Weiss for the time they spent with me as an undergraduate. I would not be here now if it was not for their enthusiasm, patience, and support. A special thanks to Brian for introducing me to mathematical logic and for keeping in touch with me after all of these years.

Finally, I would like to thank my friends for always being in the mood for collaborating, playing a game of foosball, and grabbing a beer when the time was right. Perhaps in poor taste, I have opted to not list you all here: you know who you are! I hope that we can all stay connected for many years to come.

Contents

Chapter 1

Introduction

This thesis concerns problems arising at the interface of logic, automata theory, and theoretical computer science. We will investigate an instance of the following problem: given a logical language $\mathcal{L}$ and a class of structures $\mathcal{K}$, provide a class $\mathcal{A}$ of automata which is equivalent to $\mathcal{L}$ over $\mathcal{K}$. In particular, we will explore the relative expressive power of first-order logic with equality (FOE) and alternating parity automata over infinite *unranked* trees with no leaves. In the sequel, we shall simply refer to such structures as *trees*.

There are at least two notable motivations for such an investigation, both of which are intimately related to one another. On the one hand, several results concerning the connection between logical languages and automata have served as the key breakthrough leading to the solution of interesting decidability problems arising in mathematical logic. On the other hand, the connection between logical languages and automata has played an important role in computer science; a point that will be developed further below.

An interesting open problem in mathematical logic is the following decision problem: given as input a monadic second-order definable tree language (i.e. a class of trees) $\mathcal{T}$, decide whether $\mathcal{T}$ is first-order definable. In the context of finite word languages, a positive solution to this problem was given through the joint efforts of Büchi [4], Schützenberger [32], and McNaughton and Papert [28]. This was later extended to the context of *streams* (i.e. ω-words) languages through the work of Perrin [29] using a syntactic congruence introduced by Arnold [1]. This sequence of results is illustrative of a beautiful harmony between logic, automata theory, and algebra. For a nice overview of the principal results concerning first-order definable word languages, we refer the reader to Diekert and Gastin [13].

Decidability questions concerning logical languages are particularly relevant for computer scientists, notably in the area of *formal verification.* In this area, the structure of non-terminating processes (viz. computer programs) are modelled as trees, and logical formulas can therefore be used to encode their behaviour. Whenever a logical language is known to be decidable, one can then 'verify' that their computer programs behave as intended.

A vast sea of specification languages have been introduced to match the diversity of specification problems arising in this manner. Each of these languages come with their own flavour and qualities: ranging from temporal logics such as CTL [12] to classical logics such as first-order logic with a successor or descendant relation. With such diversity, one naturally is interested in obtaining a better understanding of how these languages relate to one another with respect to their expressiveness (i.e which properties they define) on this-or-that class of structures. Automata have proven to be a successful framework for addressing such questions. Namely, by answering questions of the general form above.

As one may expect, the identification of a suitable class of automata for a given logical language and class of structures is, in general, a non-trivial problem. For this reason, one is often interested in this task for some *rich* "yardstick" formalism: given that a logic $\mathcal{L}$ and a class $\mathcal{A}$ are effectively equivalent, there is a direct correspondence between fragments of $\mathcal{L}$ and subclasses of automata from $\mathcal{A}$. *Monadic second-order logic* (MSO), the extension of first-order logic by quantifiers ranging over sets, is a very rich framework for this purpose, and the automata theory of MSO has consequently been of historical importance.

The automata theory of MSO was first considered in the influential work of Büchi [5], where it was proven that a word language is accepted by a finite automaton if and only if it is definable in monadic second-order logic. Soon after, Büchi [4] extended this result to the case of *streams* (i.e. ω-words) by introducing automata with a Büchi acceptance condition, nowadays called the *Büchi automata*. In both cases, the reduction of monadic second-order formulas to automata was, in a sense, the key insight that was needed in order to obtain the decidability of important mathematical theories over such structures.

After the foundational work on words, results on more general structures began to attract attention. The next major breakthrough emerged in the work of Rabin [31] where it was shown that (tree) automata with a *Muller acceptance condition* [23] and MSO are effectively equivalent on the infinite binary tree. A subtle but important distinction arises whenever one passes from the setting of words to that of branching structures such as trees: branching structures can either be ranked or unranked. Roughly, a tree is ranked if it has a fixed branching degree. That is, each node has a fixed number of successors. In these terms, Rabin's work can be seen as the first breakthrough in the automata theory of MSO on ranked trees. The transition from the context of ranked trees to that of unranked trees did not happen for some time and required some new ideas.

In the 1980's, Chandra et. al [10] introduced a notion of *alternation* in the context of Turing machines which made the role of certain two-player infinite games explicit in the operational semantics of such devices. In turn, Muller and Schupp [25] adapted the notion of alternation to the setting of automata on (infinite) trees: these automata are called *alternating automata*. This was an essential breakthrough in the automata theory of MSO (and variants thereof). Shortly after, Muller, Saoudi, and Schupp [24] introduced alternating automata for *weak monadic second-order logic* (WMSO) on ranked trees, a variant of MSO in which set quantification is limited to finite sets.

In a more recent paper, Walukiewicz [36] extended Rabin's Theorem to the setting of unranked trees, where it is proven that alternating automata with a parity acceptance condition and MSO are effectively equivalent over such structures. We call these *MSO-automata*, and they will serve as an ambient class of automata from which all of the automata considered in this thesis are situated. For a comprehensive survey on tree automata and their relationship with logic, we refer the reader to [17].

Characterizations of logical languages in terms of automata over trees are in fact sufficient for many applications in computer science, due to the fact that every transition system is *bisimilar* to such a structure, called its unravelling. Bisimulation is a notion of behavioural equivalence between transitions systems (i.e. (labelled) directed graphs). Intuitively, two transitions systems are bisimilar whenever they can not be distinguished by external observers. This is an important notion in computer science where, as mentioned, (non-)terminating processes are modelled as trees. From this perspective, the relevant properties (or specifications) of transition systems are those which can not distinguish bisimilar processes.

A seminal result in the model theory of modal logic is van Benthem's Theorem [2] stating that the bisimulation-invariant fragment of first-order logic is precisely basic modal logic. In this thesis, we will be particularly interested in the work of Janin and Walukiewicz [19], where van Benthem's Theorem was extended to the setting of the modal μ-calculus[1]. In particular, they showed that the modal μ-calculus and the bisimulation-invariant fragment of MSO coincide over arbitrary transition systems.

A novelty of the proof of Janin and Walukiewicz is that they reduced the question of the bisimulation-invariant fragment of MSO to the question of the bisimulation-invariant fragment of MSO-automata, providing an automata-theoretic approach to bisimulation-invariance questions. Subsequently, this method has been fruitfully applied to obtain various bisimulation-invariance results for monadic second-order logics (i.e. variants of MSO), including the final chapter of this thesis. We proceed with a brief overview of the content of this thesis, which may be viewed as consisting of the following three components.

[1]The modal μ-calculus is a highly expressive–yet computationally well behaved– extension of modal logic by unary fixpoint operators; we refer the reader to [35] for an introduction to the modal μ-calculus, an interesting topic which lies outside the scope of this thesis.

Weak Path Automata

In Chapter 3, we introduce several classes of automata which will be studied in the remainder of the thesis, developing some of their model theory along the way. As mentioned, our chief interest lies in comparing the expressive power of first-order logic and alternating parity automata over unranked trees. The problem of characterizing first-order logic as a class of automata has been studied by Bojańczyk [3] and Potthoff [30] in the context of *finite* trees. Bojańczyk introduced the concept of a *wordsum automaton* together with a notion of *cascade product* on such automata (roughly: a composition for such automata), and proved that cascade products of wordsum automata characterize first-order logic over unranked finite trees. However, it remains an open problem to characterize first-order logic as a subclass of Walukiewicz's automata; this is the main focus of this thesis.

As noted above, Muller, Saoudi, and Schupp [24] introduced alternating tree automata with a *weak acceptance condition* and proved that such automata correspond to WMSO over ranked trees; we call these *weak automata.* Intuitively, weak automata are limited to only finitely many alternations between accepting and rejecting while processing a given structure. In fact, Mostowski [22] proved that there is an intimate link between such alternations (in automata with weak acceptance conditions) and weak quantifier alternations. More recently, a slight variation of MSO-automata with weak acceptance conditions were studied by Carreiro et al. [9] where they were shown to correspond to WMSO over unranked trees.

By combining the known relationship between weak acceptance conditions and weak quantifiers with the observation that first-order quantification may be viewed as a special type of weak quantification (i.e. singleton quantification), recognizability of a given tree language by weak automata would seem to be a necessary condition for first-order definability. Indeed, in the context of (infinite) word languages, a refinement of weakness–known in the literature as aperiodicity–is known to characterize the expressiveness of first-order logic over such structures [13]. We will call this condition *antisymmetry.*

Both weakness and antisymmetry are naturally viewed as restrictions on the graph of an automaton. For instance, antisymmetric automaton may be thought of as those automata which are based on a directed acyclic graph (DAG). Alternatively, one may introduce semantic restrictions on automata by controlling the transition structure between states. An example of such a constraint is what we will call the *(linked) path condition.*

At the heart of the path condition is the notion of complete additivity tracing back to the work of Jónsson and Tarski [20] on algebraic logic. An easy observation from their work is the close connection between this notion and singleton subsets of relational structures. More recently, complete additivity was further investigated–among many others–by Hollenberg [18] and brought into the automata-theoretic setting in the work of Fountaine & Venema [16]. More details concerning the role of complete additivity will be supplied in the sequel. Finally, combining weakness (and variants thereof) with the path condition one obtains what we we will call weak path automata (and variants thereof).

Automata and first-order logic

In this chapter, we will compare the expressive power of logical formalisms and automata via effectively defined translations. That is, for a fixed class $\mathcal{K}$ of structures, our interests will lie in providing an algorithm which transforms each formula φ of a logic $\mathcal{L}$ into an equivalent automaton $\mathbb{A}_\varphi$ over $\mathcal{K}$. In the other direction, we provide algorithms transforming each automaton $\mathbb{A}$ from a class $\mathcal{A}$ of automata into an equivalent formula $\varphi_\mathbb{A}$ on $\mathcal{K}$. We say that a logic $\mathcal{L}$ and class of automata $\mathcal{A}$ are *effectively equivalent* on $\mathcal{K}$ whenever such translations exist in *both* directions.

While we do not succeed in characterizing first-order logic as a class of automata over trees, we do prove a "sandwich theorem", effectively bounding its expressive power over such structures. Namely, we will prove the following.

Contribution 1. (Sandwich Theorem) We have the following results.

(i) For every linked antisymmetric path automaton $\mathbb{A}$ we can effectively obtain an equivalent first-order formula $\varphi_\mathbb{A}$ (over trees).

(ii) For every first-order formula φ, we can effectively obtain an equivalent linked weak path automaton $\mathbb{A}_\varphi$ (on trees).

Expressive completeness modulo bisimilarity

In the final chapter, we explore the class of linked antisymmetric path automata modulo bisimilarity. Our approach is standard and borrows many of its main ingredients from the proof of the Janin-Walukiewicz Theorem, which provided a uniform method for studying such questions for monadic second-order logics. Our main result is that their proof restricts to the class of linked antisymmetric path automata. This has been shown for the class of linked weak path automata already by Carreiro [6], where the main model-theoretic tools that we will need were additionally developed. In particular, the novelty in our work lies in showing that the relevant construction preserves antisymmetry. In the notation introduced in the thesis, the main contribution is stated as follows:

Contribution 2. $Aut^l_{sa}(\mathrm{FO}_1) \equiv Aut^l_{sa}(\mathrm{FOE}_1)/\underline{\leftrightarrow}$.

Chapter 2

Preliminaries

2.1 General conventions

We fix a countable set Prop of *propositional variables* which are denoted by small Latin letters $p, q, r \ldots$ and we typically confine our attention to a finite set $\mathrm{P} \subseteq \mathrm{Prop}$. We also fix an ambient set iVar of *individual variables* which are denoted by the letters $x, y, z, \ldots$ and we typically consider a (finite) subset $X \subseteq \mathrm{iVar}$.

We use overlined boldface letters to denote sequences. That is, we write $\overline{\mathbf{x}}$ to denote the sequence $x_1, \ldots, x_k$ of variables. We sometimes blur the distinction between the sequence $T_1, \ldots, T_k$ and the set $\{T_1, \ldots, T_k\}$, writing $\Pi \subseteq \overline{\mathbf{T}}$ to denote that the set Π is a subset of $\{T_1, \ldots, T_k\}$. Given a (non-empty) finite sequence ρ, we let $last(\rho)$ denote the final element of ρ. That is, $last(x_1, \ldots, x_k) := x_k$.

For sets X and Y, a binary relation is simply a subset $R \subseteq X \times Y$. We use the following terminology for a binary relation $R \subseteq X \times Y$. For each $x \in X$, we write $R(x)$ to denote the set $\{y \in Y \mid (x, y) \in R\}$ of *R-successors* of x. For each $y \in Y$, we write $R^{-1}(y)$ to denote the set $\{x \in X \mid (x, y) \in R\}$ of *R-predecessors* of y. We write R^+ and R^* to denote the *transitive closure* of the relation R and the *reflexive-transitive closure* of the relation R, respectively. A relation $R \subseteq X \times X$ is *well-founded* if there are no infinite R-descending sequences, i.e., no sequence $(x_0, x_1, \ldots)$ such that $(x_{i+1}, x_i) \in R$ for each $i \in \omega$.

Given sets X and Y we write $X \uplus Y$ and $X \times Y$ to denote the *disjoint union* and *cartesian product* of the sets X and Y, respectively. By $X \backslash Y$ we denote the set $\{x \in X \mid x \notin Y\}$. In an effort to cut down on the number of parentheses throughout the thesis, we will write $\wp X$ rather than $\wp(X)$. Finally, we write write both ω and $\mathbb{N}$ to denote the set $\{0, 1, 2, \ldots\}$ of natural numbers.

2.2 Transition systems and trees

Labelled Transition Systems

A *(labelled) transition system* (LTS) over a set P of propositional variables is a tuple $\mathbb{S} = (S, R, \kappa, s_I)$ consisting of a non-empty set S of *nodes* called the *carrier* of $\mathbb{S}$, an *initial node* $s_I \in S$, a binary relation $R \subseteq S \times S$, and a *marking* (or *labelling function*) $\kappa : S \to \wp\mathrm{P}$. We say that $\mathbb{S}$ is *p-free* if $p \notin \mathrm{P}$ or if $p \notin \kappa(s)$ for each $s \in S$. For each node $s \in S$, we write $\mathbb{S}, s$ to denote the transition system (S, R, κ, s) which is identical to $\mathbb{S}$ except the initial node is now s.

Markings and Valuations

Observe that we can alternatively view each marking $\kappa : S \to \wp\mathrm{P}$ as the *valuation* $V_\kappa : \mathrm{P} \to \wp S$ given by setting

$$V_\kappa(p) := \{s \in S \mid p \in \kappa(s)\}$$

for each $p \in \mathrm{P}$. Conversely, we may view each valuation $V : \mathrm{P} \to \wp S$ as the marking $m_V : S \to \wp\mathrm{P}$ defined by putting

$$m_V(s) := \{p \in \mathrm{P} \mid s \in V(p)\}$$

for each $s \in S$. We will freely and frequently bounce between a valuation (respectively marking) and its associated marking (respectively valuation).

Given a pair $V, V' : \mathrm{P} \to \wp S$ of valuations and a set $B \subseteq \mathrm{P}$, we write $V \leq_B V'$ if $V(p) \subseteq V'(p)$ for every $p \in B$. We write $V \equiv_B V'$ if $V \leq_B V'$ and $V' \leq_B V$. That is, $V \equiv_B V'$ if $V(p) = V'(p)$ for each $p \in B$.

Variants of transition systems

Let $\mathbb{S}$ be a transition system over P. Given a subset $X_p \subseteq S$, we write $\mathbb{S}[p \mapsto X_p]$ to denote the transition system $\mathbb{S} = (S, R, \kappa', s_I)$ over $\mathrm{P} \cup \{p\}$ where $\kappa' : S \to \wp(\mathrm{P} \cup \{p\})$ is the marking given by putting $\kappa'(s) := \kappa(s) \cup \{p\}$ for each $s \in X_p$ and $\kappa'(s) = \kappa(s) \setminus \{p\}$ for each $s \notin X_p$. That is, $\mathbb{S}[p \mapsto X_p]$ is the transition system obtained from $\mathbb{S}$ by first 'erasing' the colour p from each node and then colouring the nodes from X_p with p. We call a transition system $\mathbb{S}'$ a *p-variant* of the transition system $\mathbb{S}$ if $\mathbb{S}' = \mathbb{S}[p \mapsto X_p]$ for some set $X_p \subseteq S$. A p-variant of $\mathbb{S}$ is an *atomic p-variant* of $\mathbb{S}$ if X_p is a singleton.

Note that the notion of a p-variant makes sense regardless of whether or not $p \in \mathrm{P}$. In case $p \notin \mathrm{P}$, note that $\kappa'(s) = \kappa(s)$ for each $s \notin X_p$.

Paths

Let $\mathbb{S} = (S, R, V, s_I)$ be a transition system and let $0 \leq k < \omega$. A (finite) *path* through $\mathbb{S}$ is a finite (non-empty) sequence $(s_0, s_1, \ldots, s_k)$ such that $(s_i, s_{i+1}) \in R$ for each i such that $i + 1 < k$. We write $\mathit{Paths}_s(\mathbb{S})$ to denote the set of paths through $\mathbb{S}$ beginning at $s \in S$.

Trees

We will view trees as particular types of transition systems and, in this thesis, we work with unranked, serial trees. Formally, a transition system $\mathbb{S}$ over a set of propositional variables P is a (P-)*tree* if the following conditions are satisfied:

(i) for every $t \in S$, there is a unique finite path from s_I to t and

(ii) $R(t) \neq \varnothing$ for every $t \in S$.

In the context of trees, we will call the initial node the *root* of the $\mathbb{T}$. Observe that (i) implies that $R^{-1}(s_I) = \varnothing$ and $R^{-1}(t)$ is a singleton for every $t \in S$ such that $t \neq s_I$. We will treat the terms *tree language* and *class of trees* as synonyms.

Subtrees

Let $\mathbb{T} = (S, R, V, s_I)$ be a P-tree. A P-tree $\mathbb{T}' = (S', R', V', s'_I)$ is a *subtree* of $\mathbb{T}$ if $S' \subseteq S, R' = R \cap (S' \times S')$, and $V' = V \cap (S' \times \mathrm{P})$. Observe that each tree $\mathbb{T}$ and each node $s \in S$ gives rise to a subtree, called *the subtree generated by* s, with carrier $R^*(s)$ and initial node s.

Expansions and unravellings

We will now introduce several salient constructions on arbitrary transition systems. We begin by defining the *tree unravelling* of a transition system, an important tool in the model theory of modal logics.

Definition 2.2.1. Let $\mathbb{S}$ be a transition system. The *unravelling of* $\mathbb{S}$ *around* s is the transition system $\vec{\mathbb{S}} = (\mathit{Paths}_s(\mathbb{S}), \vec{R}, \vec{V}, s)$ where

$$\begin{aligned} \vec{R}(s_0, \ldots, s_k) &:= \{(s_0, s_1, \ldots, s_k, t) \mid (s_k, t) \in R\} \\ m_{\vec{V}}(s_0, \ldots, s_k) &:= m_V(s_k). \end{aligned}$$

We will now define the κ-expansion of a transition system $\mathbb{S}$. Intuitively, the κ-expansion of a transition system $\mathbb{S}$ is given as a transition system $\mathbb{S}^\kappa$ with the same initial node as $\mathbb{S}$ and κ-many copies of every node except the initial node.

Definition 2.2.2. Let $\mathbb{S}$ be a transition system and let $1 \leq \kappa \leq \omega$ be a countable cardinal. A κ-path through $\mathbb{S}$ is a finite sequence of the form $s_0 k_1 s_1 k_2 s_2 \ldots k_n s_n$ such that $0 \leq n, s_{i+1} \in R(s_i)$ for each $i < n$, and $k_i < \kappa$ for each i. We write $Paths^{\kappa}_s(\mathbb{S})$ to denote the set of κ-paths through $\mathbb{S}$ beginning at s. We define the *κ-expansion of* $\mathbb{S}$ *around* s is the transition system $\mathbb{S}^{\kappa}, s = (Paths^{\kappa}_s(\mathbb{S}), R^{\kappa}, V^{\kappa}, s)$ where

$$R^{\kappa}(sk_1s_1 \ldots k_ns_n) := \{sk_1s_1 \ldots k_ns_nkt \mid (s_n, t) \in R\}$$
$$m_{V^{\kappa}}(sk_1s_1 \ldots k_ns_n) := m_V(s_n).$$

Remark 2.2.3. Note that the 1-expansion $\mathbb{S}^1$ of a transition system $\mathbb{S}$ has nodes of the form $s_I 0 s_1 0 \ldots 0 s_k$ such that $s_{i+1} \in R(s_i)$ for each $i < k < \omega$. With this in mind, it is not hard to see that the 1-expansion of $\mathbb{S}$ can be identified with its unraveling. Note too that if $\mathbb{S}$ is serial, then so is its κ-expansion for each $1 \leq \kappa \leq \omega$. In particular, it is not hard to see that if $\mathbb{S}$ is serial, then its κ-expansion is a tree for each $1 \leq \kappa < \omega$; we leave the details to the reader.

2.3 Monadic second- and first-order logics

As mentioned in the introduction, we are broadly interested in the relative expressive power of *monadic second-order logics*: extensions of first-order logic by set quantifiers. A salient example of such a logic is *monadic second-order logic* (MSO), an extension of first-order logic by quantification over unary relations (i.e. arbitrary subsets of the domain). We will work with a slightly non-standard *one-sorted* version of MSO because it is better suited for the automata framework. That is, our syntax will only involve set variables rather than both individual variables and set variables. As observed by Walukiewicz [36], quantification over individuals can be achieved in this language by encoding each individual variable x as a singleton set variable p_x.

In this thesis, we are primarily interested in monadic first-order logic, a fragment of MSO with only quantification over individuals. We introduce a one-sorted version of first-order logic which we will call *atomic second-order logic* (AMSO)[1]. In turn, we will introduce and briefly discuss the standard two-sorted version of first-order logic which will additionally feature in this thesis.

Monadic second-order logic

Definition 2.3.1. The set $\text{MSO}(<, \text{P})$ of *(one-sorted) monadic second-order formulas* over a set P of set variables is generated by the following grammar:

$$\varphi ::= p \sqsubseteq q \mid p < q \mid \Downarrow p \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists p.\varphi$$

[1]This naming convention is inspired by the fact that there is a direct correspondence between singleton subsets and the atoms in Boolean powerset algebras.

where $p, q \in \mathrm{P}$. Whenever the set P is clear from context, we omit it from the notation and write MSO.

Definition 2.3.2. Given a transition system $\mathbb{S}$ over a set P, we define the *semantics* of $\varphi \in \mathrm{MSO}$ in $\mathbb{S}$ as follows:

$$
\begin{array}{lll}
\mathbb{S} \models p \sqsubseteq q & \text{iff} & V(p) \subseteq V(q) \\
\mathbb{S} \models p < q & \text{iff} & \text{for each } t \in V(p) \text{ there exists } u \in V(q) \text{ with } (t,u) \in R^+ \\
\mathbb{S} \models \Downarrow p & \text{iff} & V(p) = \{s_I\} \\
\mathbb{S} \models \neg\varphi & \text{iff} & \mathbb{S} \not\models \varphi \\
\mathbb{S} \models \varphi \vee \psi & \text{iff} & \mathbb{S} \models \varphi \text{ or } \mathbb{S} \models \psi \\
\mathbb{S} \models \exists p.\varphi & \text{iff} & \mathbb{S}[p \mapsto X] \models \varphi \text{ for some } X \in \wp S.
\end{array}
$$

In order to simulate first-order quantification in the language of MSO, we will crucially use the following formulas:

$$
\begin{aligned}
\mathrm{empty}(p) &:= \forall q (p \sqsubseteq q) \\
\mathrm{sing}(p) &:= \forall q (q \sqsubseteq p \to (\mathrm{empty}(q) \vee p \sqsubseteq q)).
\end{aligned}
$$

It is straightforward to see that these formulas are satisfied in a transition system $\mathbb{S}$ iff $V(p) = \varnothing$ and $V(p)$ is a singleton, respectively. As such, we may now define a 'first-order quantifier' by combining the MSO quantifier $\exists p$ with the formula $\mathrm{sing}(p)$.

Definition 2.3.3. The language $\mathrm{AMSO}(<, \mathrm{P})$ of *(one-sorted) atomic second-order logic* over a set P of set variables is given by the following grammar:

$$
\varphi ::= p \sqsubseteq q \mid p < q \mid \Downarrow p \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists p.\mathrm{sing}(p) \wedge \varphi
$$

where $p, q \in \mathrm{P}$.

That is, AMSO is the fragment of MSO which only permits quantification of the shape $\exists p.\mathrm{sing}(p) \wedge \varphi$. Note that $\mathbb{S} \models \exists p.\mathrm{sing}(p) \wedge \varphi$ iff $\mathbb{S}[p \mapsto X] \models \varphi$ for some singleton subset $X \in \wp S$.

Two-sorted monadic second-order logic

As noted above, the reader probably expected MSO to be defined as an extension of first-order logic by set quantifiers $\exists p/\forall p$ with a syntax involving both individual and set variables. This is what we call the *two-sorted language of monadic second-order logic*. The syntax and semantics of this language are completely standard, but we give an explicit formulation anyways.

Definition 2.3.4. The set $2\text{MSO}(<, \text{P}, X)$ of *two-sorted monadic second-order formulas* on a set P of set variables and a set X of individual variables, is generated by the following grammar:

$$\varphi ::= p(x) \mid x < y \mid x \approx y \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists p.\varphi$$

where $p \in \text{P}$ and $x, y \in X$. In case the sets P and X are clear from context, we will simply write $2\text{MSO}(<)$. The set $2\text{FOE}(<, \text{P}, X)$ of *two-sorted monadic first-order formulas with order* is obtained by the grammar above only now the clause allowing the set quantifier $\exists p$ is excluded. That is, $2\text{FOE}(<, \text{P}, X)$ is generated by the following grammar:

$$\varphi ::= p(x) \mid x < y \mid x \approx y \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi$$

where $p \in \text{P}$ and $x, y \in X$. We will omit the sets P and X from the notation whenever they are clear from context.

Before introducing the (standard) semantics for this language, it will be useful to first gather some basic definitions. A formula of $2\text{MSO}(<)$ is *atomic* if it has the shape $p(x), x \approx y$, or $x < y$. We denote by $At(\text{P}, X)$ the set of atomic formulas of $2\text{MSO}(<)$. We denote by $FV(\varphi)$ and $BV(\varphi)$, respectively, the set of *free (individual) variables* and the set of *bound (individual) variables* occurring in φ, defined as expected. A formula φ is a *sentence* if $FV(\varphi)$ is empty. We follow the standard convention that the sets $FV(\varphi)$ and $BV(\varphi)$ are disjoint–by renaming variables whenever it is necessary–so that no individual variable occurs both bound and free in φ.

We assume that the reader is acquainted with basic syntactic notions such as *subformula* and *substitution*. We write $sform(\varphi)$ to denote the set of *subformulas* of the formula φ. Concerning the latter, we will use the notation $\varphi[\psi/p]$ to denote the formula obtained from φ by *substituting* each occurrence of p in φ by the formula ψ.

We interpret formulas of $2\text{MSO}(<, \text{P}, X)$ in a transition system $\mathbb{S}$ over P equipped with an *interpretation (also: assignment)* $g : X \to S$ which assigns to each individual variable $x \in X$ a node $s \in S$.

Definition 2.3.5. The *semantics* of $\varphi \in 2\text{MSO}(<, \text{P}, X)$ are defined as usual for atomic formulas of the shape $p(x), \neg p(x), x \approx y$, or $x \not\approx y$ and as usual for the Boolean connectives $\vee$ and $\wedge$. The semantics of the quantifiers $\exists x/\forall x$ and $\exists p/\forall p$ and atomic formulas of the shape $x < y$ are defined as follows for each transition system $\mathbb{S}$ and each interpretation g of the individual variables from X:

$$\begin{array}{lll}
\mathbb{S}, g \models x < y & \text{iff} & g(y) \in R^+(g(x)) \\
\mathbb{S}, g \models \exists x.\varphi & \text{iff} & \mathbb{S}, g[x \mapsto s] \models \varphi \text{ for some } s \in S \\
\mathbb{S}, g \models \forall x.\varphi & \text{iff} & \mathbb{S}, g[x \mapsto s] \models \varphi \text{ for each } s \in S \\
\mathbb{S}, g \models \exists p.\varphi & \text{iff} & \mathbb{S}[p \mapsto X], g \models \varphi \text{ for some } X \subseteq S \\
\mathbb{S}, g \models \forall p.\varphi & \text{iff} & \mathbb{S}[p \mapsto X], g \models \varphi \text{ for each } X \subseteq S
\end{array}$$

Whenever $\mathbb{S}, g \models \varphi$, we say that φ is *true* in $\mathbb{S}$ with g and that $\mathbb{S}$ is a *model* of φ. Given a formula $\varphi \in 2\text{FOE}(<)$, we denote by $\text{Mod}(\varphi)$ and $\text{TMod}(\varphi)$, respectively, the class of models of φ and the class of tree models of φ. Given an assignment $g : X \to \wp S$ and a set $Y \subseteq X$, we use the notation $g[y \mapsto s_y|Y]$ to denote the assignment given by putting

$$g[y \mapsto s_y|y \in Y](x) := \begin{cases} g(x) & \text{if } x \notin Y \\ s_x & \text{if } x \in Y. \end{cases}$$

for each $x \in X$. That is, $g[y \mapsto s_y|Y]$ is the assignment which agrees with g on each node $y \notin Y$ and 'reinterprets' each $y \in Y$ as the node s_y.

Remark 2.3.6. Note that for each transition system $\mathbb{S}$ we interpret the relational symbol '$<$' in terms of the *transitive closure* of its accessibility relation R. In this sense, at least whenever $\mathbb{S}$ is a tree, the relational symbol '$<$' is best understood as a *descendant relation*.

It is well known that in the context of trees the binary relation R is first-order definable in terms of its transitive closure. That is, there is a formula $S(x, y) \in 2\text{FOE}(<)$ such that for every tree $\mathbb{T}$ and every pair of nodes $s, t \in S$ and every assignment g we have $\mathbb{S}, g \models S(s, t)$ iff $(s, t) \in R$. Indeed, we define the formula $S(x, y)$ by putting

$$S(x, y) := (x < y) \wedge \forall z(x < z \to \neg(z < y)).$$

It easily follows from this definition that $\mathbb{T}, g \models S(x, y)$ if and only if $g(x) \in R(g(y))$. In other words, for each tree $\mathbb{T}$ and each node $s \in S$ we have $R(s) = \{t \in S \mid S(s, t)\}$. That is, our first-order language is expressive enough to define properties of relations and their transitive closure. This observation will be particularly useful in Chapter 4 when we translate automata into (two-sorted) first-order formulas. On the other hand, it is well known that the transitive closure of a binary relation is not first-order definable, in general. In short, we have a gain in expressive power by interpreting '$<$' via R^+ rather than interpreting it via the relation R itself.

Observe that if $\mathbb{T}$ is a tree, then there is either none or a unique path between any two nodes. In other words, if $\pi \in Paths_t(\mathbb{T})$ leads to a node $t' \in S$, then π is the unique path with this property. Consequently, for each node $s \in S$ such that $s < t'$, there is a unique node $s^+ \in R(s)$ such that s^+ is "the next node on the path π". In fact, we can define a formula $S_{xz}(y) \in 2\text{FOE}(<)$ expressing this property. We do so by putting

$$S_{xz}(y) := S(x, y) \wedge y \leq z.$$

The following proposition follows directly from this definition and the semantics of 2FOE.

Proposition 2.3.7. *for every tree $\mathbb{T}$, each pair of nodes $t, t' \in S$ such that $t < t'$, and every interpretation g such that $g(x) = t$ and $g(z) = t'$ we have*

$$\mathbb{T}, g \models S_{xz}(y) \textit{ iff } g(y) = t^+.$$

Equivalence of the AMSO and 2FOE(<)

We begin by defining a translation of models that will be needed in order to state the equivalence of the one- and two-sorted versions of MSO. Namely, given a transition system $\mathbb{S} := (S, R, V, s_I)$ over a set P of propositional variables and an assignment $g : X \to S$ of the individual variables in X, we define the transition system $\mathbb{S}^g := (S, R, V^g, s_I)$ over the set $\mathrm{P} \uplus X$ where $V^g(p) := V(p)$ if $p \in \mathrm{P}$ and $V^g(x) := \{g(x)\}$ for each $x \in X$.

The following proposition states that the languages MSO and 2MSO(<) are effectively equivalent. We refer the reader to Venema [35] for a proof of this proposition.

Proposition 2.3.8 ([35], Proposition 9.5)**.** *The formalisms* MSO *and* 2MSO*(<) are effectively equivalent. In particular, we have the following:*

(i) There is an effective translation $(\cdot)' : \mathrm{MSO(P)} \to 2\mathrm{MSO}(<, \mathrm{P}, x)$ *such that*

$$\mathbb{S}, s \models \varphi \textit{ iff } \mathbb{S} \models \varphi'(s)$$

for every formula $\varphi \in \mathrm{MSO}(P)$*, every transition system* $\mathbb{S}$*, and every* $s \in S$*.*

(ii) There is a translation $(\cdot)^t : 2\mathrm{MSO}(<, \mathrm{P}, \mathrm{X}) \to \mathrm{MSO}(\mathrm{P} \uplus \mathrm{X})$ *such that*

$$\mathbb{S}, g \models \varphi \textit{ iff } \mathbb{S}^g \models \varphi^t$$

for every formula $\varphi \in 2\mathrm{MSO}(\mathrm{X}, \mathrm{P})$*, every transition system* $\mathbb{S}$ *and every assignment* g *of the individual variables.*

Corollary 2.3.9. *The formalisms AMSO and* 2FOE *are effectively equivalent.*

2.4 Graph games

We now introduce some basic terminology for the infinite graph games which will feature in this thesis. The games that we consider involve two players named *Éloise* ($\exists$) and *Abelard* ($\forall$), respectively. We will write Π to denote an arbitrary player from the set $\{\exists, \forall\}$. Also, given a set S, we write S^* and S^ω to denote the set of finite words and infinite ω-words (or *streams*) over S, respectively.

Graph games

A *graph game* $\mathcal{G}$ is a triple (G, E, Win) which consists of a partitioned set $G := G_\exists \uplus G_\forall$ (of *positions*) called the *graph* of $\mathcal{G}$, a binary relation $E \subseteq G \times G$ specifying the *legitimate moves*, and a *winning condition* $\mathit{Win} \subseteq G^\omega$. Each position $u \in G_\Pi$ *belongs* to the player Π.

By an *initialized graph game*, we mean a pair $(\mathcal{G}, u_I)$ such that $\mathcal{G}$ is a graph game and $u_I \in G$ is the *initial position* of game. We write $\mathcal{G}@u$ to denote the graph game $\mathcal{G}$ initialized at position $u \in G$. Given an initialized graph game $\mathcal{G}@u$, the position u either belongs to $\exists$ or it belongs to $\forall$. Whenever $u \in G_\Pi$, the player Π is supposed to begin the game $\mathcal{G}@u$.

Matches

A *match* of a graph game $\mathcal{G}@u$ is a path $(u_i)_{i<\gamma}$ through the graph (G, E) of $\mathcal{G}$ where $\gamma \leq \omega$ and $u_0 = u$. Given a finite match $\pi = (u_i)_{i\leq k}$ of $\mathcal{G}$, we write $last(\pi) := u_k$ and speak of the *current position* in the match π. If $last(\pi) \in G_\Pi$, then it is player Π's *turn* to play. Whenever it is player Π's turn, it is the task of Π to continue the match π by playing a *legitimate move* from $E(last(\pi))$. If it is player Π's turn in the match π and $u \in E(last(\pi))$, we write $\pi \cdot u$ to denote the match obtained by extending the match π by the position u. Whenever $last(\pi) \in G_\Pi$ and $E(last(\pi)) = \varnothing$, we say that the player Π *got stuck* in π. A match π is *full* if π is infinite or π is finite and either $\exists$ or $\forall$ got stuck. Otherwise, π is a *partial* match.

A full finite match π of $\mathcal{G}$ is a *winning match* for Π if Π's opponent got stuck; a full infinite match π is a winning match for $\exists$ if $\pi \in Win$ and for $\forall$ if $\pi \notin Win$. Given a graph game $\mathcal{G}$ and a player Π, we write $\mathrm{PM}_\Pi^{\mathcal{G}}$ to denote the set of partial matches of $\mathcal{G}$ such that $last(\pi) \in G_\Pi$.

Strategies

A *strategy* for Π is a function $f : \mathrm{PM}_\Pi^{\mathcal{G}} \to G$. Given a strategy f for Π in $\mathcal{G}$, we say that the match $\pi = (u_i)_{i<\gamma}$ is *f-guided* if for every $i < \gamma$ such that $u_i \in G_\Pi$ we have $u_{i+1} = f(u_0, \dots, u_i)$.

Given a position $u \in G$ and a strategy f for Π, we say that f is *surviving* if for each f-guided partial match π of the game $\mathcal{G}@u$, we have that $f(\pi)$ is legitimate whenever $last(\pi) \in G_\Pi$. Note that if Π has a surviving strategy f, then Π never gets stuck in an f-guided match. A strategy f is called a *winning strategy* for Π in $\mathcal{G}@u$ if Π wins each full f-guided match of $\mathcal{G}@u$.

We write $Win_\Pi(\mathcal{G})$ to denote the set of positions $u \in G$ such that Π has a winning strategy in the game $\mathcal{G}@u$. Given a pair of positions $u, u' \in G$, we say that the position u' is f-reachable from u if there is an f-guided partial match $\pi = (u_i)_{i<k}$ in the game $\mathcal{G}$ such that $u_0 = u$ and $last(\pi) = u'$.

Positional determinacy of parity games

In this thesis, we will be especially interested in a special class of graph games called *parity games*. A game $\mathcal{G}$ is a *parity game* if its winning condition is induced by a *priority* *function* $\Omega : G \to \mathbb{N}$ with finite range by setting

$$Win_\Omega := \{g \in G^\omega \mid \text{the maximum parity occurring infinitely often in } g \text{ is even}\}.$$

If $\mathcal{G}$ is a parity game, we emphasize this by writing $\mathcal{G} = (G, E, \Omega)$. Parity games are particularly nice because they enjoy *positional determinacy.* This means that every position is either a winning position for $\exists$ or a winning position for $\forall$ and every winning strategy can be assumed to be *positional.* A strategy f is called a *positional* strategy for Π in $\mathcal{G}$ if $f(\pi) = f(\pi')$ for every pair of partial matches $\pi, \pi' \in \mathrm{PM}_\Pi^G$ such that $last(\pi) = last(\pi')$. In the context of a parity game $\mathcal{G}$, we will therefore think about strategies for player Π as functions $f_\Pi : G_\Pi \to G$. The following theorem states the positional determinacy of parity games.

Theorem 2.4.1 ([14], [22])**.** *For every parity game $\mathcal{G}$, there exist positional strategies $f_\exists$ and $f_\forall$ for $\exists$ and $\forall$, respectively, such that for every position $u \in G$ there is a player Π such that f_Π is a winning strategy for Π in $\mathcal{G}@u$.*

2.5 One-step languages

We will now introduce the notion of a one-step language. The notion of a one-step languages was first introduced in the context of coalgebra [11],[**GY04**]. For our purposes, one-step languages will serve as the type of the transition map of alternating parity automata. Automata based on one-step languages were first introduced by Venema [34]; subsequently, they have been fruitfully applied to obtain various automata-theoretic characterizations of logical languages. See Carreiro [6] for a rich source of examples.

After introducing the general notion of a one-step language, we will turn our attention towards the concrete one-step languages which will feature in this thesis. A *one-step language* is a map $\mathcal{L}$ which assigns to each finite set A of *labels* a set $\mathcal{L}(A)$ of *one-step formulas* over the set A. One-step languages are meant to be interpreted with a *truth relation* $\models_1$ between one-step formulas and *one-step models.*

Definition 2.5.1. A one-step model is a pair $\mathbb{D} = (D, V)$ consisting of a set D, called the *domain* of $\mathbb{D}$, and a valuation $V : A \to \wp D$ assigning to each label $a \in A$ a set $V(a) \in \wp D$ of *nodes* from D.

Given a one-step language $\mathcal{L}$ and a truth-relation $\models_1$ between $\mathcal{L}$-formulas and one-step models, we call the pair $(\mathcal{L}, \models_1)$ a *one-step logic.* Whenever $\mathbb{D} \models_1 \alpha$, we say that α is *true* in $\mathbb{D}$; if $\mathbb{D} \not\models_1 \alpha$, then we say that α is *false* in $\mathbb{D}$.

Definition 2.5.2. Let $\mathcal{L}_0$ and $\mathcal{L}_1$ be a one-step languages and let A be a set. For each pair of formulas $(\alpha, \beta) \in \mathcal{L}_0(A) \times \mathcal{L}_1(A)$, we define $\alpha \equiv \beta$ if for every one-step model $(D, V : A \to \wp D)$ we have $(D, V) \models_1 \alpha$ iff $(D, V) \models_1 \beta$.

We assume that each one-step $\mathcal{L}$ has a fragment $\mathcal{L}^+$, which we will call the *positive fragment of* $\mathcal{L}$ characterizing the *monotone* formulas from $\mathcal{L}$.

Definition 2.5.3. Let $\mathcal{L}$ be a one-step language and let A be a set. A one-step formula $\alpha \in \mathcal{L}(A)$ is *monotone* in $B \subseteq A$ if for every one-step model (D, V) such that $(D, V) \models_1 \alpha$ we have $(D, U) \models_1 \alpha$ for every valuation $U : A \to \wp D$ such that $V \leq_B U$. We say that $\alpha \in \mathcal{L}(A)$ is *monotone* if α is monotone in each A.

Remark 2.5.4. Instead of the definition above, we could have said that $\alpha \in \mathcal{L}(A)$ is monotone if it is monotone in each $a \in A$. That is, we could have defined the notion of monotonicity by demanding that α is monotone in $\{a\}$ for each $a \in A$. These two definitions are easily seen to be equivalent.

One-step languages will prove to be one of the most important ingredients of this thesis because they determine the type of the transition map of alternating parity automata (cf. Section 2.6). Because of this, the model theory of one-step logics is a useful tool in the theory of alternating parity automata. For example, in [21] it was shown that the closure of a class of tree languages under complementation is related to the semantic notion of *Boolean duals*; we will return to this point shortly. First, we introduce the concrete one-step languages FOE_1 and FO_1 of one-step first order formulas with and without equality, respectively.

First-order one-step languages

We will define one-step languages FOE_1 and FO_1 as special fragments of the language 2FOE(<). As an ambient language, we define the set FOE(A) of *two-sorted monadic first-order formulas* over a set A of monadic predicates and a set X of individual variables as follows:

$$\alpha ::= a(x) \mid \neg a(x) \mid x \approx y \mid x \not\approx y \mid \alpha \vee \alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha \mid \forall x.\alpha$$

where $a \in A$ and $x, y \in X$.

Note that FOE is obtained by a grammar that is closely related to that of 2FOE(<). The main differences are this: we have now excluded the clause permitting atomic formulas of the shape $x < y$ and we will always assume that each formula of FOE is in *negation normal form*. That is, negations may only occur at the level of monadic predicates. It is well known that we may do so without loss of generality because every formula from 2FOE (i.e. the language obtained from 2FOE by omitting the clause for <) is equivalent to a formula from 2FOE in negation normal form. In particular, we import the semantics of FOE directly from the semantics of 2FOE(<).

Remark 2.5.5. As a convention, we will use the Greek characters $\alpha, \beta, \ldots$ to denote formulas from FOE and we will use the characters $\varphi, \psi, \ldots$ to denote formulas from 2FOE(<).

The *equality-free fragment* FO of FOE is obtained from the grammar above by excluding the clauses for (in)equality. That is, the set $\mathrm{FO}(A, X)$ of *equality-free monadic first-order formulas* over a sets A of labels and a set X of individual variables is generated by the following grammar:

$$\alpha ::= a(x) \mid \neg a(x) \mid \alpha \vee \alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha \mid \forall x.\alpha$$

where $a \in A$ and $x, y \in X$.

For each language $\mathcal{L} \in \{\mathrm{FOE}, \mathrm{FO}\}$, we define the positive fragment $\mathcal{L}^+$ of $\mathcal{L}$ as the language generated by the grammar as $\mathcal{L}$, only now the clause allowing formulas of the shape $\neg a(x)$ is excluded. That is, we define the *positive fragment of monadic first-order logic* to be the set $\mathrm{FOE}^+(A)$ of FOE-formulas generated by the grammar

$$\alpha ::= a(x) \mid x \approx y \mid x \not\approx y \mid \alpha \vee \alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha \mid \forall x.\alpha$$

where $a \in A$ and $x, y \in \mathrm{iVar}$. As before, we obtain the equality-free fragment, FO^+, of FOE^+ by omitting the clauses for equality: "$x \approx y$" and "$x \not\approx y$". The following theorem states that, indeed, these sets characterize the monotone fragments of monadic first-order logic with and without equality, respectively. For a detailed proof of this fact, we refer the reader to [[6], Lemma 5.1.25] and [[6], Lemma 5.1.29].

Theorem 2.5.6. *Let $\mathcal{L} \in \{\mathrm{FOE}, \mathrm{FO}\}$. For each set A and each formula $\alpha \in \mathcal{L}(A)$, we have that α is monotone in every $a \in A$ if and only if α is equivalent to a formula of $\mathcal{L}^+(A)$.*

Definition 2.5.7. The one-step languages FOE_1 and FO_1 of one-step first-order formulas with and without equality consist of the positive sentences from FOE^+ and FO^+, respectively.

For example, among the monadic first-order formulas $\exists x.\neg b(x)$ and $b(x) \vee \forall y.a(y)$, only the latter belongs to the positive fragment FOE^+ because the former has a negative occurrence of the atomic formula $b(x)$. Neither of these formulas belong to the one-step language FOE_1: the former because it is not monotone and the latter because it is not a sentence. A typical example of a formula from FOE_1 is $\exists x.b(x) \vee \forall y.a(y)$.

We will explicitly use the notion of a *subformula* of a monadic first-order formula in order to define a graph structure on alternating parity automata. For this reason, we give an explicit definition even though it is completely standard.

Definition 2.5.8. Let A be a finite set of monadic predicates. For each formula $\alpha \in \mathrm{FOE}^+(A)$, we define the set $\mathit{sform}(\alpha)$ of subformulas of α according to the following induction on the structure of α. If α is atomic, then $\mathit{sform}(\alpha) := \{\alpha\}$. Inductively, we define $\mathit{sform}(\alpha_0 \star \alpha_1) := \mathit{sform}(\alpha_0) \cup \mathit{sform}(\alpha_1) \cup \{\alpha_0 \star \alpha_1\}$ for each $\star \in \{\vee, \wedge\}$ and $\mathit{sform}(\mathcal{Q}x.\beta) := \mathit{sform}(\beta) \cup \mathit{sform}(\mathcal{Q}.\beta)$ for each $\mathcal{Q} \in \{\exists, \forall\}$.

We will now develop some of the model theory of monadic first-order logic with(out) equality. In particular, we will first provide normal form theorems for the one-step languages FOE_1 and FO_1 which will be very useful in our automata-theoretic investigations. In turn, we will define the *dual fragment* of FOE.

Normal Forms for FO_1

We will now present normal form results for the one-step languages $\mathrm{FO(E)}_1$. As far as we can tell, these results are folklore. For a nice overview and proofs of these results, we refer to [6] where some of the model theory of these languages is developed.

Definition 2.5.9. Given a set A of monadic predicates and a set $B \subseteq A$, the *A-type associated with* B is the formula defined by putting

$$\tau_B(x) := \bigwedge_{a \in B} a(x) \wedge \bigwedge_{a \notin B} \neg a(x).$$

The *positive A-type* associated with $B \subseteq A$ is the formula

$$\tau_B^+(x) := \bigwedge_{a \in B} a(x)$$

expressing only the positive information of the A-type τ_B. If $B = \varnothing$, then we define $\tau_B^+(x)$ to be the formula $x \approx x$.

Remark 2.5.10. Let $\mathbb{D}$ be a one-step model with marking $m : D \to \wp A$, let $d \in D$, and let $B \subseteq A$. The meaning of the *full* A-type τ_B and the *positive* A-type τ_B^+ in $\mathbb{D}$ is expressed as follows:

- $\mathbb{D} \models \tau_B(d)$ iff $m(d) = B$.
- $\mathbb{D} \models \tau_B^+(d)$, iff $B \subseteq m(d)$.

Definition 2.5.11. A formula $\alpha \in \mathrm{FO}(A)$, is in *positive basic form* if α has the shape

$$\nabla^+_{\mathrm{FO}}(\overline{\mathbf{T}}, \Pi) = \exists x_1 \dots x_k \bigwedge_{i \leq k} \exists x.\tau^+_{T_i}(x_i) \wedge \forall x \bigvee_{B \in \Pi} \tau^+_B(x)$$

for some set $\overline{\mathbf{T}} \subseteq \wp A$ and some set $\Pi \subseteq \overline{\mathbf{T}}$.

We have the following normal form theorem for positive sentences of monadic first-order logic. Its proof is an application of the theory of Ehrenfeucht-Fraïsse games. We refer the reader to [[6], Corollary 5.1.26] for more detail.

Theorem 2.5.12. *For each set A of propositional variables, there is an effective procedure transforming each sentence of* $\mathrm{FO}^+(A)$ *into an equivalent disjunction of sentences in positive basic form.*

Normal forms for FOE_1

As the reader might expect, the addition of the equality symbol into the language of first-order logic yields a richer basic form due to its ability to 'count types'. In order to simplify notation, we first introduce an auxiliary formula $\pi(x_1, \ldots, x_n, \Pi) \in \mathrm{FOE}(A)$ where $x_1, \ldots, x_n$ are (individual) variables and $\Pi \subseteq \wp A$. To this end, we define

$$\pi(x_1, \ldots, x_n, \Pi) := \forall z(\mathrm{diff}(x_1, \ldots, x_n, z) \to \bigvee_{B \in \Pi} \tau_B^+(z))$$

where $\mathrm{diff}(x_1, \ldots, x_m)$ is the formula

$$\mathrm{diff}(x_1, \ldots, x_m) := \bigwedge_{1 \leq j < j' \leq m} x_j \not\approx x_{j'}$$

expressing that the variables $x_1, \ldots, x_m$ denote distinct nodes from the domain. Note that $\pi(\overline{\mathbf{x}}, \Pi)$ expresses that every node which is distinct from the interpretations of the variables $x_1, \ldots, x_m$ has one of the types from the set Π. We can now provide the analogue of Definition 2.5.11 for the language FOE^+.

Definition 2.5.13. A formula $\alpha \in \mathrm{FOE}(A)$, is in *positive basic form* if α has the form

$$\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi) = \exists x_1 \ldots x_k(\mathrm{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_{i \leq k} \tau^+_{T_i}(x_i) \wedge \pi(\overline{\mathbf{x}}, \Pi))$$

for some some sequence $\overline{\mathbf{T}} = T_1, \ldots, T_k$ of A-types and some set $\Pi \subseteq \overline{\mathbf{T}}$.

Note that the formula $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$ expresses that there are distinct nodes $d_1, \ldots, d_k$ of the domain realizing each of the positive A-types from the set $\{T_1, \ldots, T_k\}$ and every other node realizes one of the positive A-types from the set $\Pi \subseteq \overline{\mathbf{T}}$. That is, these formulas partition the set into two parts: one part consisting of distinct witnesses for each of the types in $\overline{\mathbf{T}}$ and another part consisting of the remaining nodes, each labelled by some type from the set Π.

The following theorem is a normal form result for positive sentences from monadic first-order logic with equality. While this result seems to be folklore, a detailed proof can be found in [[6], Theorem 5.1.12].

Theorem 2.5.14. *For each set A of propositional variables, there is an effective procedure transforming each sentence of* $\mathrm{FOE}^+(A)$ *into an equivalent disjunction of sentences in positive basic form.*

Boolean duals

In this section we introduce the notion of a *Boolean dual*. Among other things, we will use the closure of the one-step language FOE_1 to prove complementation theorems for several classes of automata.

Definition 2.5.15. Let A be a set. The formulas $\alpha, \beta \in \mathrm{FOE}(A)$ are each other's *Boolean dual* if for every one-step model (D, V) and every assignment g of the individual variables we have

$$(D, V), g \models \alpha \text{ if and only if } (D, V^c), g \not\models \beta$$

where the valuation V^c is given by setting $V^c(a) := D \backslash V(a)$ for every $a \in A$. We say that a one-step language $\mathcal{L}$ is *closed under Boolean duals* if, for each set A, every formula $\varphi \in \mathcal{L}(A)$ has a Boolean dual $\alpha^\delta \in \mathcal{L}(A)$.

We will now define a *dualization operator* $(\cdot)^\delta$ providing, for each set A, a Boolean dual α^δ for each formula $\alpha \in \mathrm{FOE}(A)$.

Definition 2.5.16. The *Boolean dual* α^δ of $\alpha \in \mathrm{FOE}(A)$ is defined by the following induction on formulas from FOE :

$$\begin{aligned} (a(x))^\delta &:= a(x) & (\neg a(x))^\delta &:= \neg a(x) \\ (x \approx y)^\delta &:= x \not\approx y & (x \not\approx y)^\delta &:= x \approx y \\ (\varphi \vee \psi)^\delta &:= \varphi^\delta \wedge \psi^\delta & (\varphi \wedge \psi)^\delta &:= \varphi^\delta \vee \psi^\delta \\ (\exists x.\psi)^\delta &:= \forall x.\psi^\delta & (\forall x.\psi)^\delta &:= \exists x.\psi^\delta \end{aligned}$$

Observe that for each set A and each language $\mathcal{L} \in \{\mathrm{FO}, \mathrm{FOE}\}$, we have that $\alpha^\delta \in \mathcal{L}(A)$ if $\alpha \in \mathcal{L}(A)$. Moreover, $(\cdot)^\delta$ preserves positivity of each monadic predicate. That is, the positive fragment $\mathcal{L}^+(A)$ of $\mathcal{L}$ is closed under $(\cdot)^\delta$. We leave the proof of the following proposition stating the formulas α and α^δ are each others Boolean duals to the reader.

Proposition 2.5.17. *Let $\mathcal{L} \in \{\mathrm{FOE}, \mathrm{FO}\}$. For each set A and each formula $\alpha \in L(A)$, the formulas α and α^δ are each others Boolean duals. In particular, $\mathcal{L}$ is closed under Boolean duals.*

Proof. By a routine induction on the structure of $\alpha \in \mathcal{L}(A)$. □

2.6 Alternating parity automata

Alternating parity automata are simple finite-state devices that operate on (possibly) infinite structures, such as directed graphs. In this section, we introduce parity automata in a general setting; we will introduce various concrete classes of parity automata in Chapter 3.

Formally, an *(alternating) parity automaton* based on the *one-step language* $\mathcal{L}$ and alphabet C is a quadruple (A, Θ, Ω, a_I) consisting of a finite set A of *states*, a distinguished state $a_I \in A$ called the *initial state*, a *priority map* $\Omega : A \to \mathbb{N}$, and a *transition function*

$$\Theta : A \times C \to \mathcal{L}^+(A).$$

In practice, the alphabet will usually be a set $C := \wp\mathrm{P}$ where P is a set of propositional variables and we will call $c \in C$ a *colour*. We write $Aut(\mathcal{L})$ to denote the class of parity automata based on the one-step language $\mathcal{L}$. Given an automaton $\mathbb{A}$ and a state $a \in A$, we write $\mathbb{A}.a$ to denote the automaton (A, Θ, Ω, a) which is identical to $\mathbb{A}$ except now the initial state is a. We call a state $a \in A$ a *μ-state* (respectively *ν-state*) if $\Omega(a)$ is odd (respectively even). And, for $\sigma \in \{\mu, \nu\}$, we write A^σ to denote the set of σ-states from A.

Note that if $\mathbb{A}$ is a parity automaton based on the one-step language $\mathcal{L}$, then the codomain of the transition map Θ of $\mathbb{A}$ is the positive fragment $\mathcal{L}^+(A)$ of $\mathcal{L}$ over the set of states from $\mathbb{A}$. In other words, states from $\mathbb{A}$ lead a double life as monadic predicates which may occur positively in the transition formulas of $\mathbb{A}$. Throughout this thesis, we will always work with alternating parity automata $\mathbb{A}$ such that for each pair $(a, c) \in A \times C$ the formula $\Theta(a, c)$ is a sentence.

Remark 2.6.1. Walukiewicz [36] introduced a class of alternating parity automata for MSO over trees. As mentioned in the introduction, these automata form an ambient class for the automata that we will investigate in the sequel. For this reason, we will always assume that each automaton $\mathbb{A}$ is from this class unless mentioned otherwise.

Definition 2.6.2. An MSO-*automaton* is an a automaton from the class $Aut(\mathrm{FOE}_1)$.

We will now fix some basic terminology for alternating parity automata. We will begin by defining the graph of an automaton and, in turn, we define their operational semantics via an infinite two-player parity game. Finally, we will develop some background regarding the relationship between complementation of tree languages and the closure of one-step languages under Boolean duals.

The occurrence graph

We now introduce a natural graph structure on alternating parity automata which is useful tool for better understanding their 'dynamics'. We will first fix some basic terminology.

Let $\mathbb{A} = (A, \Theta, \Omega, a_I) \in Aut(\mathrm{FOE}_1)$ be an MSO-automaton on alphabet C. For each pair of states $a, b \in A$, we say that b *occurs* in a iff $b(x) \in \mathit{sform}(\Theta(a, c))$ for some $x \in iVar$ and some colour $c \in C$. We write $\lhd_{\mathbb{A}}$ to denote the transitive closure of the occurrence relation. Whenever $b \lhd_{\mathbb{A}} a$ we say that b is *active* in a. Note that b is active in a if there is a finite sequence $a = a_0, a_1, \ldots, a_k = b$ of states such that $1 \leq k < \omega$ and a_{i+1} occurs in a_i for every $i < k$. Whenever the automaton $\mathbb{A}$ is clear from context, we suppress the subscript and simply write "$\lhd$" instead of "$\lhd_{\mathbb{A}}$". With this terminology now in place, we can now define the graph of an MSO-automaton.

Definition 2.6.3. Let $\mathbb{A}$ be an MSO-automaton. The *(occurrence) graph of* $\mathbb{A}$ is the directed graph $G(\mathbb{A}) = (A, E)$ where $(a, b) \in E$ iff b occurs in a. Observe that the adjacency relation E is nothing more than the converse of the occurrence relation.

Observe that an arbitrary MSO-automaton $\mathbb{A}$ may contain states $a, b \in A$ which are distinct and active in one another. Intuitively, this situation corresponds to a 'loop' in the graph of $\mathbb{A}$. In Chapter 3, we will be most interested in the 'maximal loops' contained in the graph of an automaton. We formalize this with the notion of a *cluster*.

Definition 2.6.4. Let $\mathbb{A}$ be an MSO-automaton. For each pair of states $a, b \in A$ we write $a \bowtie b$ iff $a \lhd b$ and $b \lhd a$. A *cluster* of $\mathbb{A}$ is an equivalence class of the smallest equivalence relation containing $\bowtie$. A cluster is called *degenerate* if it is of the form $\{a\}$ for some $a \in A$ such that $a \not\bowtie a$. For each $a \in A$, we write C_a to denote the unique cluster containing the state a.

Note that every MSO-automaton $\mathbb{A}$ only has finitely many clusters. We define the relation $\sqsubset$ on clusters of an MSO-automaton by putting $C_0 \sqsubset C_1$ iff $C_0 \neq C_1$ and there exists $a_0 \in C_0$ and $a_1 \in C_1$ such that $a_0 \lhd a_1$. We say that C_0 is *above* C_1 whenever $C_0 \sqsubset C_1$. For each MSO-automaton $\mathbb{A}$, we define the *cluster graph of* $\mathbb{A}$ to be the directed graph $\mathit{Clust}(\mathbb{A}) = (V, E)$ where V is the set of clusters of $\mathbb{A}$ and $(C_0, C_1) \in \sqsubset$ iff C_0 is above C_1.

In the sequel, it will be useful to fix the following terminology. For each MSO-automaton $\mathbb{A}$ and each $a \in A$, we define the *downset* and *upset* of a to be the sets $\downarrow a := \{b \in A \mid b \lhd a\}$ and $\uparrow a := \{b \in A \mid a \lhd b\}$, respectively. We will sometimes write $\Uparrow a$ (respectively $\Downarrow a$) as a shorthand for the set $\uparrow a \cup \{a\}$ (respectively $\downarrow a \cup \{a\}$). Note that $\Downarrow a$ coincides with $\downarrow a$ whenever a is active in itself. Intuitively, $\downarrow a$ is the set of nodes that 'sit below' a in the occurrence graph $G(\mathbb{A})$. Similarly, the set $\uparrow a$ is the set of states 'above' a in $G(\mathbb{A})$.

Acceptance game

The operational semantics of alternating parity automata are formalized via the following infinite two-player parity game, presented as table.

Definition 2.6.5. Let $\mathbb{S} = (S, R, V, s_I)$ be a transition system and let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a parity automaton. The *acceptance game* $\mathcal{A}(\mathbb{A}, \mathbb{S})$ associated with $\mathbb{A}$ and $\mathbb{S}$ is the parity game given in the following table.

Position	Player	Admissible moves	Priority
$(a, s) \in A \times S$	$\exists$	$\{V : A \to \wp R(s) \mid (R(s), V) \models \Theta(a, \kappa(s))\}$	$\Omega(a)$
$V : A \to \wp R(s)$	$\forall$	$\{(b, t) \mid t \in V(b)\}$	0

We say that the automaton $\mathbb{A}$ *accepts* the transition system $\mathbb{S}$ if the pair (a_I, s_I) is a winning position for $\exists$ in the game $\mathcal{A}(\mathbb{A}, \mathbb{S})$. That is, if she has a winning strategy in the game $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, s_I)$. Otherwise, we say that $\mathbb{A}$ *rejects* $\mathbb{S}$. We call a positions of the type $(a, s) \in A \times S$ are called *basic* positions.

Definition 2.6.6. Let $\mathbb{A}$ and $\mathbb{A}'$ be alternating parity automata on the same alphabet $\wp$P. We say that $\mathbb{A}$ and $\mathbb{A}'$ are *equivalent* and write $\mathbb{A} \equiv \mathbb{A}'$ if for every transition system $\mathbb{S}$ over P we have that $\mathbb{A}$ accepts $\mathbb{S}$ iff $\mathbb{A}'$ accepts $\mathbb{S}$.

A match of the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{T})$ for a parity automaton $\mathbb{A}$ and a transition system $\mathbb{S}$ proceeds in rounds in which the players inspect a local area of $\mathbb{S}$ via one-step formulas. In each round, it is the job of $\exists$ to demonstrate that the automaton $\mathbb{A}$ correctly describes the current window of $\mathbb{S}$ from the perspective of the current state of $\mathbb{A}$. We will sometimes informally refer to a full match of the acceptance game as a *run* of the automaton $\mathbb{A}$ on $\mathbb{S}$.

More specifically, at a basic position $(a, s) \in A \times S$, it is $\exists$'s turn to play. Her task is to produce a valuation (or, equivalently, a marking) $V_{a,s}$ satisfying the one-step formula $\Theta(a, \kappa(s))$ in the set $R(s)$ of successors of s. After she has chosen such a valuation, it is $\forall$'s turn to select the next basic position from the set $\{(b, t) \in A \times S \mid t \in V(b)\}$.

Note that the formula $\Theta(a, \kappa(s))$ may be unsatisfiable in the set $R(s)$. For example, if $\Theta(a, \kappa(s))$ is the formula $\exists x \exists y. x \not\approx y \wedge b(x) \wedge a(y)$ and $|R(s)| = 1$, then clearly $\exists$ will be stuck. On the other hand, if $\exists$ can play the empty marking (e.g. if $\Theta(a, c) = \forall x. x \approx x$), then she wins immediately because $\forall$ will have an empty set of available moves. In short, both players can get stuck in the acceptance game so an arbitrary match may be finite or infinite.

Remark 2.6.7. A typical match π in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a, s)$ initialized at the basic position (a, s) follows the "move pattern" $\exists\forall$. In other words, π has the shape

$$(a, s)V_1(a_1, s_1)V_2(a_2, s_2)\ldots$$

where, for each $i < \omega$, the basic position (a_{i+1}, s_{i+1}) is such that $s_{i+1} \in R(s_i)$ and $s_{i+1} \in V_{a_i, s_i}(a_{i+1})$. We will identify the match π with the sequence $(a, s)(a_1, s_1)\ldots$ of basic positions which occur in it. Observe that this sequence contains all of the relevant information for determining whether π is an accepting or rejecting 'run' of $\mathbb{A}$ on $\mathbb{S}$.

Finally, observe that if π is finite, then the sequence $(s, s_1, \ldots, s_k)$ is nothing more than a finite path through $\mathbb{S}$ whereas, if π is infinite, then $(s, s_1, \ldots)$ is a branch of $\mathbb{S}$. In this sense, we may think of each run of $\mathbb{A}$ on $\mathbb{S}$ as an instance of $\mathbb{A}$ 'processing' some branch–i.e. an ω-path–or some path through $\mathbb{S}$.

Proposition 2.6.8. *Let $\mathbb{A} = (A, \Theta, \Omega, a_i)$ and $\mathbb{A}' = (A, \Theta', \Omega, a_I)$ be parity automaton from $Aut(\mathcal{L})$. Assume that $\Theta(a, c) \equiv \Theta'(a, c)$ for each pair $(a, c) \in A \times \wp P$. Then $\mathbb{A} \equiv \mathbb{A}'$.*

Proof. It is straightforward to check that any winning strategy for $\exists$ in the game $\mathcal{A}(\mathbb{A}, \mathbb{S})$ is also a winning strategy for her in the game $\mathcal{A}(\mathbb{A}', \mathbb{S})$, and vice versa. We leave the details to the reader. □

Remark 2.6.9. Let A be a set and let $\alpha \in \mathcal{L}(A)$ for some one-step language $\mathcal{L}$. Note that whenever $b \in A$ does not occur in the one-step formula α, the meaning of α is *independent* of the name b in the sense that for every domain D and every pair of valuations $V, V' : A \to \wp D$ such that $V \equiv_{A \setminus \{b\}} V'$, we have $(D, V) \models \alpha$ iff $(D, V') \models \alpha$. Consequently, for each parity automaton $\mathbb{A}$ and each pair of states $a, b \in A$, if b does not occur in a, then for each valuation $V : A \to \wp D$ and each colour $c \in C$, we have $(D, V[b \mapsto \varnothing]) \models \Theta(a, c)$ if $(D, V) \models \Theta(a, c)$.

One consequence of the observation above is that an arbitrary strategy for $\exists$ in the acceptance game may give $\forall$ a larger pool of available moves than is strictly necessary. Obviously, we may always assume that $\exists$ plays according to a *minimal strategy.*

Definition 2.6.10. A strategy f for $\exists$ in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$ is *minimal* if, for every basic position $(a, s) \in A \times S$, the valuation $V_{a,s}$ suggested by f at (a, s) is such that

$$(R(s), V) \not\models \Theta(a, \kappa(s)) \text{ for every valuation } V \text{ such that } V <_A V_{a,s}.$$

That is, a strategy is minimal if it only suggests valuations $V_{a,s}$ which can not be "shrunk" without compromising the truth of the one-step formula $\Theta(a, \kappa(s))$ in the set $R(s)$. Due to the following fact, we will always assume that $\exists$ plays according to a minimal strategy.

Fact 2.6.11. If $\exists$ has a winning strategy in the game $\mathcal{A}(\mathbb{A}, \mathbb{S})$, then she has a minimal winning strategy as well.

We will further develop the theory of minimal strategies throughout this thesis. Namely, the shape of her minimal strategies varies significantly depending on the structure of automata from a class $\mathcal{A}$. The existence and structure of her minimal strategies will be particularly useful in Chapter 4 when we translate automata into formulas.

Closure under complementation

As mentioned before, many interesting questions in the theory of alternating parity automata can be recast as questions about one-step language $\mathcal{L}$. In this section, we will see a concrete example of the interplay between alternating parity automata and one-step model theory.

Based on ideas stemming from [25] and [21], complementation theorems for the class $Aut(\mathcal{L})$ of alternating parity automata can be proven through a combination of Boolean duals and a "role swap" between the players $\exists$ and $\forall$ in the acceptance game. We will now develop the tools that we will need in order to apply this idea to the concrete classes introduced in Chapter 3.

Definition 2.6.12. Let $\mathcal{L}$ be a one-step language and assume that there is a map $(\cdot)^\delta : \mathcal{L} \to \mathcal{L}$ which provides, for each set A, a Boolean dual $\alpha^\delta \in \mathcal{L}(A)$ for each $\alpha \in \mathcal{L}(A)$. Given a parity $\mathcal{L}$-automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$, we define its *complement* $\mathbb{A}^\delta$ as the $\mathcal{L}$-automaton $(A, \Theta^\delta, \Omega^\delta, a_I)$ where $\Theta^\delta(a, c) = (\Theta(a, c))^\delta$ and $\Omega^\delta(a) = \Omega(a) + 1$ for every $a \in A$ and every $c \in \wp\mathrm{P}$.

Proposition 2.6.13. *Let $\mathcal{L}$ and $(\cdot)^\delta$ be as in the previous definition. For every automaton $\mathbb{A} \in Aut(\mathcal{L})$ and every transition system $\mathbb{S}$ we have that*

$$\mathbb{A}^\delta \textit{ accepts } \mathbb{S} \textit{ if and only if } \mathbb{A} \textit{ rejects } \mathbb{S}.$$

Proof. We refer to [21] for a proof. □

As a consequence of Proposition 2.6.13, the class $Aut(\mathcal{L})$ is closed under complementation if the one-step language $\mathcal{L}$ is closed under Boolean duals. For example, the language FOE^+ is closed under Boolean duals (cf. Section 2.5) hence the class of tree languages recognized by MSO-automata is closed under complementation. In Chapter 3, we will show that the same is true for special subclasses of MSO-automata.

2.7 Bisimulation

An important notion of behavioural equivalence between transition systems is that of *bisimulation.* The aim of this section is to define and provide the relevant background for this concept.

Definition 2.7.1. Let $\mathbb{S} = (S, R, \kappa, s_I)$ and $\mathbb{S}' = (S', R', \kappa', s'_I)$ be transition systems over the same set P. A *bisimulation* is a relation $Z \subseteq S \times S'$ satisfying the following conditions for all $(t, t') \in Z$:

(**atom**) $p \in \kappa(t)$ iff $p \in \kappa'(t')$ for each $p \in \mathrm{P}$;

(**back**) for each $s' \in R'(t')$, there exists $s \in R(t)$ such that $(s, s') \in Z$;

(**forth**) for each $s \in R(t)$, there exists $s' \in R(t')$ such that $(s, s') \in Z$.

We say that $\mathbb{S}$ and $\mathbb{S}'$ are *bisimilar* (notation: $\mathbb{S} \underline{\leftrightarrow} \mathbb{S}'$) if there is a bisimulation $Z \subseteq S \times S'$ such that $(s_I, s'_I) \in Z$.

Definition 2.7.2. Let $\mathcal{A}$ be a class of alternating parity automata. An automaton $\mathbb{A} \in \mathcal{A}$ is *bisimulation-invariant* if $\mathbb{S} \underline{\leftrightarrow} \mathbb{S}'$ implies that $\mathbb{A}$ accepts $\mathbb{S}$ iff $\mathbb{A}$ accepts $\mathbb{S}'$, for each pair $\mathbb{S}, \mathbb{S}'$ of transition systems. A class $\mathcal{A}$ of automata is bisimulation-invariant if each automaton $\mathbb{A}$ is bisimulation-invariant. We denote by $\mathcal{A}/\underline{\leftrightarrow}$ the class consisting of those $\mathbb{A} \in \mathcal{A}$ which are bisimulation-invariant.

In Chapter 5, we will investigate the *bisimulation-invariant* fragments of various classes of automata. An important tool in this investigation is the following fact, which is fundamental in the model theory of modal logics.

Fact 2.7.3. For each transition system $\mathbb{S}$ and every $1 \leq \kappa \leq \omega$, we have that $\mathbb{S}$ and its κ-expansion $\mathbb{S}^\kappa$ are bisimilar.

Chapter 3

Weak path automata

In this chapter, we introduce a variety of subclasses of alternating parity automata. Namely, we will be interested in subclasses of what we will call *weak path automata*. Weak path automata are obtained from a class $Aut(\mathcal{L})$ of alternating parity automata by combining a constraint on the parity map (weakness) with a constraint on the one-step language $\mathcal{L}$ itself. That is, the path condition requires that transition formulas come from a special fragment of $\mathcal{L}$.

From now on–even though some of our definitions will be given for arbitrary transition systems–we limit our discussion to trees because they are easier to visualize and build intuition. We view trees as having essentially two 'dimensions': a *horizontal dimension* and a *vertical dimension*. For example, a typical property of the horizontal dimension of trees is "every node has exactly k successors", whereas "along some path there is a node labelled with p and q" expresses a property of the vertical dimension of trees. In this light, weakness is best viewed as a restriction on the expressiveness of automata when it comes to properties of the vertical dimension, while the path condition limits the expressivity of automata in the horizontal dimension.

3.1 Weak automata

We will begin by introducing the notion of a *weak alternating parity automata*. The notion of a weak acceptance condition goes back to the work of Rabin. In the context of alternating automata on trees, weak acceptance conditions were introduced by Muller, Saoudi, and Schupp [24]. For more details on weak automata, we refer to [17].

Definition 3.1.1. Let $\mathcal{L}$ be a one-step language. An automaton $\mathbb{A} \in Aut(\mathcal{L})$ is *weak* if for each pair of states $a, b \in A$ the following condition is satisfied:

(weakness) if $a \triangleleft b$ and $b \triangleleft a$, then $\Omega(a) = \Omega(b)$.

We write $Aut_w(\mathcal{L})$ to denote the class of *weak alternating parity automata*.

In words, weak automata are those automata such that every pair of states belonging to the *same cluster* have the *same priority.* That is, we may associate a unique priority to each cluster of a weak automaton. For this reason, we will take the liberty of speaking of the priority of a cluster–in addition to the priority of a state–in the context of weak automata. Namely, the priority of a cluster C of a weak automaton $\mathbb{A}$ is the unique priority of the states contained in C.

In order to understand the intuition behind the weakness condition, we will first make the following observation about arbitrary MSO-automata. Let f be a minimal strategy for $\exists$ in the acceptance game $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{T})$ for some MSO-automaton $\mathbb{A}$ and some tree $\mathbb{T}$. Recall that f always suggests valuations $V_{a,s}$ such that $V_{a,s}(b) = \varnothing$ for each $b \in A$ that does not occur in a (cf. Remark 2.6.9). Consequently, for each full f-guided match $\pi = (a_0, s_0)(a_1, s_1)\ldots$ we have that a_{i+1} occurs in a_i for each i.

Now, let a be a state that occurs infinitely often and let i be the least index such that $a_i = a$. Note that for each $j \geq i$, we have that $a_j \lhd a$. Furthermore, as a occurs infinitely often, there exists $k > j$ such that $a_k = a$ hence also $a \lhd a_k$. That is, $a \bowtie a_j$ so, in particular, we have that $a_j \in C_a$. Say that a cluster C appears infinitely often in the match π if some state $a \in C$ appears infinitely often in π. In these terms, we have just shown that a cluster appears infinitely often iff it appears cofinally. In other words, each full f-guided match stabilizes in some final cluster C_f. In the setting of weak automata, this means that exactly one priority is seen infinitely often. Thus, the only thing that really matters about the priority of a cluster is its parity.

Fact 3.1.2 ([27])**.** Every weak automaton $\mathbb{A} \in Aut_w(\mathcal{L})$ is equivalent to a weak automaton $\mathbb{A}' = (A, \Theta, \Omega', a_I)$ where $\Omega' : A \to \{0, 1\}$.

Proof. Define the map $\Omega' : A \to \{0, 1\}$ by setting $\Omega'(a) = \Omega(a) \mod 2$. It is straightforward to see that any winning strategy for $\exists$ in the game $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{S})$ is also a winning strategy for her in the game $\mathcal{G}' = \mathcal{A}(\mathbb{A}', \mathbb{S})$, and vice versa. □

3.2 Weak path automata

We will now introduce and briefly discuss the class of *alternating weak path parity automata.* As mentioned above, this class is a refinement of the class of weak automata obtained by imposing an additional constraint on the transition map corresponding to a limitations on their expressivity about the 'horizontal' dimension of trees. We begin by developing some of the basic model theory and notation surrounding the semantic notion of *complete additivity.*

One-step complete additivity

Recall that for a set $B \subseteq A$ and a pair $V, V' : A \to \wp D$ of valuations, we write $V \equiv_B V'$ if $V(b) = V'(b)$ for each $b \in B$.

Definition 3.2.1. We say that a formula $\alpha \in \mathcal{L}(A)$ is *completely additive* in $a \in A$ if α is monotone in a and, for every one-step model (D, V) such that $(D, V) \models \alpha$, there exists a valuation $V' \equiv_{A\setminus\{a\}} V$ such that $(D, V') \models \varphi$ and either $V'(a) = \varnothing$ or $V'(a)$ is a singleton subset of $V(a)$. We say that φ is *completely multiplicative* in $a \in A$ if its Boolean dual is completely additive in a.

In words, a formula α is completely additive in the propositional variable a if each valuation making the formula α true can be 'shrunk' to a valuation assigning a to either none or a unique node from the domain. That is, a formula is completely additive in a if its meaning depends on at most one node being coloured with a. We lift the notion of complete additivity *in a propositional variable* to that of being complete additivity *in a set of propositional variables* as follows: $\alpha \in \mathcal{L}(A)$ is completely additive in $B \subseteq A$ if α is monotone in each $b \in B$ and, for each valuation $V : A \to \wp D$ such that $(D, V) \models \alpha$, either

(i) $(D, V[b \mapsto \varnothing \mid b \in B]) \models \alpha$ or

(ii) $(D, V[b^* \mapsto \{t\}, b \mapsto \varnothing \mid b \in B\setminus\{b^*\}]) \models \alpha$ for some $b^* \in B$ and $t \in V(b^*)$.

Note that if a formula α is completely additive in a set B of colours, then the meaning of α depends on at most one node being coloured with at most one colour from B. For example, the formula $\exists x \exists y(x \not\approx y \wedge a(x) \wedge b(y))$ is completely additive in a (respectively b) because its meaning only depends on one node being labelled with a. However, it is not completely additive in $\{a, b\}$ because its meaning depends on labelling two distinct nodes with a and b, respectively. We will now define the completely additive fragments of FOE^+ and FO^+.

Definition 3.2.2. Let A be a set of monadic predicates and let $A' \subseteq A$. The set $\mathrm{ADD}_{A'}\mathrm{FOE}^+(A)$ of monadic first-order formulas which are completely additive in $A' \subseteq A$ is generated by the following grammar:

$$\alpha ::= \beta \mid a(x) \mid \exists x.\alpha \mid \alpha \vee \alpha \mid \alpha \wedge \beta$$

where $a \in A'$ and $\beta \in \mathrm{FOE}^+(A\setminus A')$. Note that the equality is included in β. The set $\mathrm{ADD}_{A'}\mathrm{FO}^+$ of monadic first-order formulas without equality which are completely additive in $A' \subseteq A$ is generated by the same grammar except now $\beta \in \mathrm{FO}^+(A\setminus A')$. We denote the dual fragments (i.e. the completely multiplicative fragments) by $\mathrm{MUL}_{A'}\mathrm{FOE}^+(A)$ and $\mathrm{MUL}_{A'}\mathrm{FOE}^+(A)$

Theorem 3.2.3. *Let A be a set and let $\mathcal{L} \in \{\mathrm{FO}^+, \mathrm{FOE}^+\}$. For each subset $A' \subseteq A$ and each formula $\alpha \in \mathcal{L}(A)$, we have that α is completely additive in A' iff it is equivalent to a formula from $\mathrm{ADD}_{A'}\mathcal{L}(A)$.*

Proof. We refer the reader to [[6], Theorem 5.1.45 and Theorem 5.1.49] for details. □

Weak path automata

We will now define the class of *alternating weak path parity automata.*

Definition 3.2.4. The class $Aut_{wa}(\mathcal{L})$ of *weak path automata* is given by the weak parity automata $\mathbb{A} = \langle A, \Theta, \Omega, a_I \rangle$ from $Aut_w(\mathcal{L})$ such that every cluster C of $\mathbb{A}$ satisfies the following condition:

(**path**) Either $\Theta(a, c) \in \mathrm{ADD}_C\mathrm{FOE}^+(A)$ for every every pair $(a, c) \in C \times \wp\mathrm{P}$ or $\Theta(a, c) \in \mathrm{MUL}_C\mathrm{FOE}^+(A)$ for every pair $(a, c) \in C \times \wp\mathrm{P}$.

While weakness imposes that states from the same cluster have a uniform priority, the path condition requires that each cluster is either uniformly completely additive in itself *or* uniformly completely multiplicative in itself. Note that this gives rise to four types of clusters (respectively states). We will additionally be interested in the following subclass of weak path automata obtained by *linking* the semantic notions of complete additivity and complete multiplicativity with the parity of states.

Definition 3.2.5. The class $Aut^l_{wa}(\mathcal{L})$ of *linked weak path automata* is given by the weak path parity automata $\mathbb{A} = \langle A, \Theta, \Omega, a_I \rangle$ from $Aut_{wa}(\mathcal{L})$ that satisfy the following condition:

(linked path) For every state $a \in A$:

If $a \in A^\mu$, then $\Theta(a, c)$ is completely additive in C_a, for each $c \in \wp\mathrm{P}$

If $a \in A^\nu$, then $\Theta(a, c)$ is completely multiplicative in C_a, for each $c \in \wp\mathrm{P}$.

Remark 3.2.6. Linked weak path automata were investigated in [6] (under the name "additive-weak automata") where it was shown that a tree language is recognized by a linked weak path automaton based on FOE_1 iff it is definable in *weak chain logic*, a version of monadic second-order logic which quantifies over finite chains. See [[6], Theorem 7.3.1] for details.

3.3 Antisymmetric path automata

We will now introduce the class of *(alternating) antisymmetric path parity automata.* These automata are obtained from the class of weak path automata by requiring that each cluster is a singleton.

Definition 3.3.1. The class $Aut_{sa}(\mathcal{L})$ of *antisymmetric path automata* is given by the parity automata $\mathbb{A} = \langle A, \Theta, \Omega, a_I \rangle$ from $Aut_{wa}(\mathcal{L})$ such that the relation $\lhd_{\mathbb{A}}$ is antisymmetric (i.e. clusters are singletons).

Recall that clusters are essentially 'maximal loops' in the occurence graph of a parity automaton. Intuitively, whenever an automaton $\mathbb{A}$ is antisymmetric, its occurrence graph is 'loop free'. That is, $\mathbb{A}$ is based on a directed acyclic graph (DAG). The following proposition gives another useful way of thinking about antisymmetric automata, and will be utilized when we translate linked antisymmetric automata into first-order formulas in Chapter 4.

Proposition 3.3.2. *An alternating parity automaton* $\mathbb{A}$ *is antisymmetric iff* $\lhd_{\mathbb{A}}$ *is well-founded.*

Proof. We leave the details to the reader. □

As before, we will also be interested in the *linked* version of antisymmetric automata, defined as before: the class $Aut^{l}_{sa}(\text{FOE}_1)$ of *linked antisymmetric path automata* consists of those automata $\mathbb{A}$ from the class $Aut_{sa}(\text{FOE}_1)$ of antisymmetric path automata which satisfy the linked path condition.

Remark 3.3.3. We will now investigate the effect of antisymmetry and the path condition on minimal winning strategies for $\exists$ in the acceptance game. We focus in particular on a linked antisymmetric path automaton $\mathbb{A}$ and a state $a \in A^{\mu}$. To this end, let $\mathbb{T}$ be a tree and let f be a minimal winning strategy for $\exists$ in the game $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{T})$. Finally, let $(a, s) \in Win_{\exists}(\mathcal{G})$.

Because $\mathbb{A}$ satisfies the linked path condition, it follows that $\Theta(a, c)$ is completely additive in a for every colour $c \in C$. Because of this, for each one-step model (D, V) such that $(D, V) \models \Theta(a, c)$, we can shrink the valuation V to a valuation V' such that $(D, V') \models \Theta(a, c)$ and such that $V'(a)$ is empty or a singleton. Now, applying this observation to the valuation $V_{a,s}$ suggested by her strategy f, it follows that $V_{a,s}(a)$ is either a singleton or empty by minimality.

Recall that $b \in {\uparrow}a$ if $a \lhd b$. Now, if $b \in {\uparrow}a$ and $b \neq a$, then antisymmetry ensures that b does not occur in a. That is, for each one-step model (D, V) such that $(D, V) \models \Theta(a, c)$, we have that $(D, V[b \mapsto \varnothing]) \models \Theta(a, c)$ as well. Hence $V_{a,s}(b) = \varnothing$ for each $b \in {\uparrow}a$ by minimality of her strategy f. In other words, antisymmetry ensures whenever a is currently 'activated', only states which occur below a in the occurrence graph can be 'activated' in the future.

Based on these observations, we will now prove one of the main technical tools that we will need concerning linked antisymmetric path automata. Following the observations above, the idea is this: whenever $\exists$ plays according to a minimal strategy in the acceptance game $\mathcal{A}(\mathbb{A},\mathbb{T})@(a,s)$ for a linked antisymmetric path automaton $\mathbb{A}$, a tree $\mathbb{T}$, and a winning position $(a,s) \in A^{\mu} \times S$, her strategy guides her along a finite path. Antisymmetry adds to this by ensuring that once the state a is 'deactivated', it will never be activated again.

Proposition 3.3.4. *Let $\mathbb{A} \in Aut^{l}_{sa}(\mathrm{FOE}_1)$ and let $\mathbb{T}$ be a tree. For each $a \in A^{\mu}$ and each $s \in S$ such that $(a,s) \in Win_{\exists}(\mathcal{A}(\mathbb{A},\mathbb{T}))$, there exists a winning strategy f_{π} for $\exists$ in the game $\mathcal{G} = \mathcal{A}(\mathbb{A},\mathbb{T})$ such that there is a unique finite path $\pi \in Paths_s(\mathbb{T})$ satisfying the following properties.*

(i) For each $(b,t) \in Win_{\exists}(\mathcal{G})$ such that $b \lhd a$, the valuation $V_{b,t}$ suggested by f_{π} at (b,t) is such that $V_{b,t}(a') = \varnothing$ for each $a' \in \Uparrow a$.

(ii) For every $i < length(\pi)$, the valuation V_{a,s_i} suggested by f_{π} at position (a,s_i) is such that $V_{a,s_i}(a) = \{s_{i+1}\}$, and the valuation $V_{a,last(\pi)}$ is such that $V_{a,last(\pi)}(a) = \varnothing$.

Proof. Let $\mathbb{A} \in Aut^{l}_{sa}(\mathrm{FOE}_1)$ and let $\mathbb{T}$ be a tree with labelling function $\kappa : S \to \wp\mathrm{P}$. Suppose that $(a,s) \in A^{\mu} \times S$ is a winning position for $\exists$ in the game $\mathcal{G} := \mathcal{A}(\mathbb{A},\mathbb{T})$, and let f be a *minimal* positional winning strategy for $\exists$ in $\mathcal{G}$. For each position $(b,t) \in Win_{\exists}(\mathcal{G})$, we write $V_{b,t}$ to denote the valuation suggested by f at (b,t). We will show that f itself satisfies (i) and (ii). In order to see that (i) holds, assume that $(b,t) \in Win_{\exists}(\mathcal{G})$ is such that $b \lhd a$. As $\mathbb{A}$ is antisymmetric, it follows that a does *not* occur in b. In particular, if $(R(t), V_{b,t}) \models \Theta(b, \kappa(s))$, then $(R(t), V_{b,t}[a \mapsto \varnothing]) \models \Theta(b, \kappa(t))$ as well. Then $V_{b,t}(a) = \varnothing$ because f is minimal. Hence f satisfies (i).

We now show that f satisfies (ii). To this end, note that $\Theta(a,c)$ is completely additive in a for every colour $c \in \wp\mathrm{P}$ because $a \in A^{\mu}$. By Remark 3.3.3, it then follows that for each $t \in S$ either

(1) $V_{a,t}(a) = \varnothing$ or

(2) $V_{a,t} = \{t^{+}\}$ for some $t^{+} \in R(t)$.

In other words, for each node $t \in S$ there is either none or a unique node $t^{+} \in R(t)$ such that $(a, t^{+}) \in V_{a,t}$. On these grounds, we inductively define a sequence $\rho := (a,s_0),(a,s_1),\ldots$ of basic positions by putting $(a,s_0) := (a,s)$ and $(a,s_{i+1}) := (a,s_i^{+})$ if V_{a,s_i} satisfies (1) and undefined otherwise. Note that ρ must be finite because otherwise ρ is an infinite f-guided match in which every basic position has odd parity. That is, because otherwise $\exists$ loses the f-guided match ρ in the game $\mathcal{G}@(a,s)$ in contradiction with our assumption that $(a,s) \in Win_{\exists}(\mathcal{G})$. So, indeed, ρ is finite.

It easily follows from the definition of ρ that the sequence $\pi := s_0, s_1, \ldots$ is a finite sequence satisfying (1). Moreover, π is uniquely determined by the strategy f and position (a,s) because, by (2), for each $i \in \omega$ the position (a, s_{i+1}) is uniquely determined by (a, s_i). Thus f satisfies (i) and (ii), as desired. ☐

Corollary 3.3.5. *Let $\mathbb{A} \in Aut^{-}_{sa}(\mathrm{FOE}_1)$ and let $\mathbb{T}$ be a tree. For every winning position $(a, s) \in A^{\mu} \times S$ in the game $\mathcal{G} := \mathcal{A}(\mathbb{A}, \mathbb{T})$, the strategy f_{π} given by Proposition 3.3.4 for (a, s) satisfies the following for every $t \in S$: the position (a, t) is f_{π}-reachable in $\mathcal{G}@(a, s)$ iff $t = s_i$ for some $i \leq length(\pi)$.*

We shall now show that the class of linked antisymmetric path automata is closed under complementation.

Proposition 3.3.6. *If $\mathbb{A} \in Aut^{l}_{sa}(FOE_1)$, then $\mathbb{A}^{\delta} \in Aut^{l}_{sa}(FOE_1)$.*

Proof. We begin by checking the path condition. To this end, note that if $a \in (A^{\delta})^{\mu}$, then $a \in A^{\nu}$ because $\Omega^{\delta}(a) = \Omega(a) + 1$. Thus $\Theta(a, c)$ is completely multiplicative in a for each colour $c \in \wp\mathrm{P}$. In particular, $\Theta^{\delta}(a, c)$ is completely additive in a for each colour $c \in \wp\mathrm{P}$. Similarly, if $a \in (A^{\delta})^{\nu}$, then $\Theta^{\delta}(a, c)$ is completely multiplicative in a for each $c \in \wp\mathrm{P}$. It remains to be seen that if $\mathbb{A}$ is antisymmetric, then so is $\mathbb{A}^{\delta}$. It is straightforward to syntactically check that, for each $a \in A$ and each one-step formula $\alpha \in \mathrm{FOE}_1(A)$, we have that a occurs in α if and only if a occurs in α^{δ}. In particular, $\mathbb{A}^{\delta}$ is antisymmetric if $\mathbb{A}$ is, as desired. $\square$

Chapter 4

Automata and first-order logic

In this Chapter, we provide effective bounds on the expressive power of first-order logic. Namely, we prove the following 'sandwich theorem' for the expressiveness of FOE on trees.

Theorem 4.0.1 (Sandwich Theorem)**.** *We have the following effective bounds on the expressive power of first-order logic with equality on trees:*

$$Aut^l_{sa}(\mathrm{FOE}_1) \leq \mathrm{FOE}(<) \leq Aut^l_{wa}(\mathrm{FOE}_1).$$

To start, we will transform each linked antisymmetric path automata $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ into an equivalent sentence $\xi_{\mathbb{A}}$ from 2FOE, showing that first-order logic is at least as expressive as these automata. The underlying idea is to provide a sentence of two-sorted first-order logic which successfully describes the behaviour of such automata over trees. At the heart of the matter is Proposition 3.3.4 where the structure of minimal winning strategies for such automata were clarified.

In turn, we translate each formula $\varphi \in$ AMSO into an equivalent linked weak path automata $\mathbb{A}_\varphi \in Aut^l_{wa}(\mathrm{FOE}_1)$. We do so by induction on the structure of formulas from AMSO; recall that this is the one-sorted version of first-order logic.

As is typical for alternating parity automata, proving that the class of tree languages recognizable by linked weak path automat is closed under complementation and union– the automata-theoretic analogues of negation and disjunction, respectively–is somewhat straightforward. Most of our work will be devoted to closing that linked weak path automata are closed under *atomic projection*, the automata analogue of first-order quantification.

4.1 From antisymmetric path automata to 2FOE formulas

In this section, we will transform automata from $Aut^l_{sa}(\mathrm{FOE}_1)$ into equivalent two-sorted first-order formulas.

Theorem 4.1.1. *Given an automaton* $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ *on* P, *we can effectively construct an equivalent sentence* $\xi_{\mathbb{A}} \in 2\mathrm{FOE}(\mathrm{P})$.

We will prove Theorem 4.1.1 by induction on the well-founded relation $\lhd_{\mathbb{A}}$. The crux of the proof lies in obtaining a formula $\chi_{\mathbb{A},a}(x) \in \mathrm{FOE}(\mathrm{P} \uplus {\downarrow}a)$ (in one free variable x) which describes the 'behaviour' of $\mathbb{A}.a$ and such that only states occurring strictly below a in the occurrence graph of $\mathbb{A}$ may occur as monadic predicates. That is, we crucially obtain a formula $\chi_{\mathbb{A},a}(x)$ which satisfies the following conditions:

(i) Each of the propositional variables occurring in $\chi_{\mathbb{A},a}(x)$ are from $\mathrm{P} \uplus {\downarrow}a$;

(ii) $(a, s) \in Win_{\exists}(\mathcal{A}(\mathbb{A}, \mathbb{T}))$ iff $\mathbb{T}[b \mapsto [\![\mathbb{A}.b]\!] \mid b \in {\downarrow}a] \models \chi_{\mathbb{A},a}(s)$.

In fact, this is the key step required in order to handle the base case of the induction. Indeed, it follows directly from (i) and (ii) that the formula $\chi_{\mathbb{A},a_I}(x)$ itself contains only propositional variables P and is such that the following holds on trees:

$$\mathbb{T} \models \chi_{\mathbb{A},a_I}(s_I) \text{ if and only if } \mathbb{A} \text{ accepts } \mathbb{T}. \tag{$*$}$$

That is, $\chi_{\mathbb{A},a_I}(x)$ is equivalent to $\mathbb{A}$ over trees, at least whenever the variable x is interpreted as the root of the tree. In order to obtain a sentence, the only additional insight that we will require is that 2FOE is expressive enough to define the root of a tree. This is indeed the case: we define the formula

$$root(x) := \forall z(z \leq x \rightarrow z = x)$$

expressing that the (interpretation of the) variable x is the root. On the basis of this definition and the semantics of 2FOE, it is straightforward to prove the following proposition.

Proposition 4.1.2. *For every tree* $\mathbb{T}$ *and every interpretation* g *of the individual variables from* iVar*, we have the following equivalence:*

$$\mathbb{T}, g \models root(x) \textit{ if and only if } g(x) = s_I.$$

Furthermore, obtaining the formula $\chi_{\mathbb{A},a}(x)$ will additionally serve as the key step in the inductive step of the proof, only now the additional 'insight' that will be needed is a suitable substitution of the sentences given by the inductive hypothesis. We defer these details to later and turn our attention towards crafting the formula $\chi_{\mathbb{A},a}(x)$.

Recall that for each automaton $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ and each pair $(a, c) \in A \times \wp\mathrm{P}$ either $a \in A^{\mu}$ and $\Theta(a, c)$ is completely additive in a or $a \in A^{\nu}$ and $\Theta(a, c)$ is completely multiplicative in a. Due to the different behaviours exhibited by this variety of states, the exact shape of the formula $\chi_{\mathbb{A},a}(x)$ will differ depending on whether a is μ- or a ν-state. For now, we restrict our attention to constructing the formula $\chi_{\mathbb{A},a}(x)$ for each

$\mathbb{A} \in Aut^l_{sa}(\text{FOE}_1)$ and each $a \in A^\mu$. In fact, this will be sufficient because of the dual nature of these state types; we return to this in due time.

Linked antisymmetric path automata have a very special structure reflected in Proposition 3.3.4. In fact, this proposition is the key insight that we will need in order to provide the formula $\chi_{\mathbb{A},a}(x)$. Namely, it states that whenever $(a, s) \in Win_\exists(\mathcal{A}(\mathbb{A}, \mathbb{T}))$ for some tree $\mathbb{T}$, her minimal winning strategy guides her to a 'final' position (a, s^*) where the state a will be needed by Proposition 3.3.4. Moreover, the automaton $\mathbb{A}$ never gets 'stuck' along the unique path π leading to s^* because otherwise $\exists$ would lose. That is, each of the requirements set by the transition formulas along π are met.

Due to these observations, most of our efforts lie in formalizing the following statement for a winning position $(a, s) \in A^\mu \times S$: there is a node $s^* \in R^+(s)$ satisfying the following conditions:

(1) the state a is never needed while $\mathbb{A}$ processes the subtree $\mathbb{T}.s^*$;

(2) for each node t such that $s \leq t \leq s^*$, each of the 'requirements' specified by the transition formula $\Theta(a, \kappa(t))$ are satisfiable in the set $R(t)$.

We will view the node s^* as an 'exit' from the state a. With this viewpoint in place, one may interpret (2) as stating that the automaton 'traces a (safe) path towards its exit from a'. That is, as the automaton scans the tree $\mathbb{T}$ it never encounters a moment where it gets stuck. We proceed to identifies nodes that serve as an escape from the state $a \in A$. In turn, we will define a substitution on formulas which detect the next node on the path towards an exit by using the formula $S_{xz}(y)$ defined in Chapter 2. Finally, we will put these pieces together to form the formula $\chi_{\mathbb{A},a}(x)$.

As a basis for both of these substitutions, we first a notion of *relativization* transforming the 'local' perspective of one-step first-order logic into the 'global' perspective of 2FOE, reminiscent of the standard translation of basic modal logic into first-order logic.

Relativization

Recall that for a set A (of names) and a set X of individual variables, the language $\text{FOE}_1(A, X)$ is the set of one-step formulas of first-order logic (with equality) in which the names from A may occur positively as monadic predicates and only individual variables from X may occur, free or bound. Also, recall that the set $At(A, X)$ consists of the atomic formulas over the sets A and X.

Definition 4.1.3. For each set A (of names), each set $X \subseteq iVar$, and each fresh individual variable x, we define the *relativization* $\rho_x : \text{FOE}_1(A, X) \to 2\text{FOE}(A, X \uplus \{x\})$ by induction on the structure of $\alpha \in \text{FOE}_1(A, X)$ as follows:

- $\rho_x(\alpha) := \alpha$ for each $\alpha \in At(A, X)$
- $\rho_x(\alpha \heartsuit \beta) := \rho_x(\alpha) \heartsuit \rho_x(\beta)$ for $\heartsuit \in \{\wedge, \vee\}$
- $\rho_x(\neg\alpha) := \neg\rho_x(\alpha)$
- $\rho_x(\exists x.\alpha) := \exists x(S(z, x) \wedge \rho_x(\alpha))$
- $\rho_z(\forall x.\varphi) := \forall x(S(z, x) \to \rho_x(\varphi))$.

Given an automaton $\mathbb{A}$ on alphabet $\wp$P and a state $a \in A$, the monadic predicates occurring (if any) in the formula $\Theta(a, c)$ are always of the form $b(x)$ where $b \in {\downarrow}a$. In order to compare the 'local' meaning of the formula $\Theta(a, c)$, Thus, in order to compare the meaning of one-step formula $\Theta(a, c)$–which is evaluated 'locally' (i.e. at a node) in a P-tree $\mathbb{T}$–and its relativization, we will also 'relativize' the tree $\mathbb{T}$ by colouring a set $S_b \subseteq S$ with b for each $b \in A$. With this in mind, the next proposition states that the semantic relationship between a one-step formula and its relativization is as expected.

Proposition 4.1.4. *Let $\mathbb{T}$ be a tree. Assume that we are given a subset $S_a \subseteq S$ for each $a \in A$. Then, for each node $s \in S$ and each assignment $g : iVar \to R(s)$, the following are equivalent for every formula $\alpha \in \mathrm{FOE}_1(A, X \backslash \{z\})$:*

(i) $(R(s), V), g \models \alpha$ *for the valuation* $V : A \to \wp R(s)$ *given by* $V(a) = S_a \cap R(s)$ *for each* $a \in A$.

(ii) $\mathbb{T}[b \mapsto S_b \mid b \in A], g[z \mapsto s] \models \rho_z(\alpha)$.

Proof. Let $\mathbb{T}'$ denote the $(\mathrm{P} \uplus A)$-tree $\mathbb{T}[b \mapsto S_b \mid b \in A]$ with labelling function $\kappa' : S \to \wp(\mathrm{P} \uplus A)$. We proceed by induction on the complexity of α. It is not difficult to see that the equivalence holds if $\varphi \in \{x \approx y : x, y \in iVar\}$ because α is z-free by assumption.

Suppose that α has shape $b(x)$ for some $b \in A$. Assume (i) so that we have a valuation $V : A \to \wp R(s)$ such that $(R(s), V), g \models b(x)$ and $V(a) = S_a \cap R(s)$ for every $a \in A$. The former implies that $g(x) \in V(b)$ hence also $g(x) \in S_b$ by the latter. Thus $\mathbb{T}', g \models b(x)$. As $b(x) \in \mathrm{FOE}_1(A, X \backslash \{z\})$, we have that $x \neq z$ whence $\mathbb{T}', g[z \mapsto s] \models b(x)$.

Now assume that $\mathbb{T}', g[z \mapsto s] \models b(x)$. Then we have that $g(x) \in S_b$. As the interpretation g has type $g : \mathrm{iVar} \to R(s)$, we also have $g(x) \in R(s)$. Hence $g(x) \in S_b \cap R(s)$. Define the required valuation $V : A \to \wp R(s)$ by setting $V(a) = S_a \cap R(s)$ for every $a \in A$. Then $(R(s), V) \models b(x)$ since $g(x) \in V(b)$, as required.

As the Boolean cases in which α is a disjunction or conjunction follow immediately from the inductive hypothesis, we immediately focus on the case in which α is an existential formula. Assume that $(R(s), V), g \models \exists x.\beta$ for some valuation V as in (i). Then there exists $d \in R(s)$ such that $(R(s), V), g[x \mapsto d] \models \beta$. By the inductive hypothesis, it follows that $\mathbb{T}', g[x \mapsto d] \models \beta$. As $d \in R(s)$, we also have $\mathbb{T}', g[x \mapsto d] \models S(s, d)$. Hence

$\mathbb{T}', g[x \mapsto d] \models S(s,d) \wedge \beta$. But this just means that $\mathbb{T}', g \models \exists x(S(s,x) \wedge \beta)$. In particular, $\mathbb{T}', g[z \mapsto s] \models \exists x(S(z,x) \wedge \beta) = \rho_z(\beta)$. This concludes our proof from (*i*) to (*ii*). The direction from (*ii*) to (*i*) is handled completely analogously. □

Identifying an exit

We will now introduce a substitution $(\cdot)[\bot/a] : 2\mathrm{FOE}(\mathrm{P} \uplus \{a\}) \to 2\mathrm{FOE}(\mathrm{P})$ on first-order formulas. For an automaton $\mathbb{A}$ and a node s^* in a tree $\mathbb{T}$, we shall want that $\mathbb{T} \models \rho_x(\Theta(a, \kappa(s^*)))[\bot/a]$ exactly whenever "in the subtree $\mathbb{T}.s^*$, the state a is no longer needed".

Definition 4.1.5. For each formula $\varphi \in 2\mathrm{FOE}(A, X)$ and each $a \in A$, we define the formula

$$\varphi[\bot/a] := \varphi[a(x) \mapsto x \not\approx x \mid x \in X].$$

That is, the formula $\varphi[\bot/a]$ is the formula obtained from φ by substituting each occurrence of $a(x)$ by the formula $x \not\approx x$.

The main result that we will need regarding this substitution is the following proposition, which we view as the formal counterpart of the intuitive description given in (1).

Proposition 4.1.6. *Let $\mathbb{T}$ be a tree and let A be a set. Assume that for each $a \in A$ we are given a set $S_a \subseteq S$. Then, for every node $s \in S$, every assignment $g : X \to R(s)$, and every $a \in A$, the following are equivalent for every one-step formula $\alpha \in \mathrm{FOE}_1(A, X \backslash \{z\})$:*

(i) $(R(s), V), g \models \varphi$ *for the valuation* $V : A \to \wp R(s)$ *such that* $V(b) = \varnothing$ *for each* $b \in {\Uparrow}a$ *and* $V(b) = S_b \cap R(s)$ *for every* $b \in {\downarrow}a$

(ii) $\mathbb{T}[b \mapsto S_b \mid b \in {\downarrow}a], g[z \mapsto s] \models \rho_z(\varphi)[\bot/a]$.

Proof. By a routine induction on the complexity of $\alpha \in \mathrm{MFOE}^+(A)$. □

Tracing a path to an exit

We now turn our attention towards item (2) of the informal description. The situation is this: we are in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{T})@(a, s)$ and the current position has the shape (a, t) for some $t < s^*$. Because we have not yet arrived at the special node s^* where a is no longer needed, the one-step formula $\Theta(a, \kappa(t))$ requires that a be labelled at exactly one node in the set $R(t)$.

Now, $\exists$'s minimal strategy recommends a valuation $V_{a,t}$ such that $V_{a,t}(a) = \{t^+\}$ where t^+ is the unique node from $R(t)$ on the path from t to s^*. Recall that we introduced the formula $S_{xz}(y)$ which, on trees, expresses that y is the unique successor on the path from x to z (cf. Proposition 2.3.7). That is, it identifies where her minimal strategy will use the colour a. This motivates the following definition.

Definition 4.1.7. Let P be a set of propositional variables and let X be a set of individual variables. For each $a \in \mathrm{P}$, each pair $x, z \in X$, and each formula $\varphi \in \mathrm{FOE}(\mathrm{P}, X)$, the formula $\varphi[S_{xz}/a]$ is the formula obtained from φ by replacing each occurrence of $a(y)$ by $S_{xz}(y)$, for every $y \in X$.

For example, the substitution $\exists y \exists y'(y \not\approx y' \wedge a(y) \wedge b(y'))[S_{xz}/a]$ results in the formula $\exists y \exists y'(y \not\approx y' \wedge S_{xz}(y) \wedge b(y'))$. The following proposition is the main result that we will need concerning the substitution $(\cdot)[S_{xz}/a]$, stating that the meaning of such substitutions is as intended.

Proposition 4.1.8. *Let $\mathbb{T}$ be a tree. Assume that for each $a \in A$ we are given a subset $S_a \subseteq S$. Then for each $a \in A$, each pair of nodes $s, s^* \in S$ such that $s^* \in R^+(s)$, and each assignment $g : iVar \to R(s)$, the following are equivalent for each formula $\alpha \in \mathrm{FOE}_1(A, X \backslash \{x, z\})$:*

(i) $(R(s), V) \models \alpha$ *for the valuation* $V : A \to \wp R(s)$ *such that* $V(a) = \{s^+\}$ *and* $V(b) = S_b \cap R(s)$ *for each* $b \in A \backslash \{a\}$

(ii) $\mathbb{T}[b \mapsto S_b \mid b \in \downarrow a], g[x \mapsto s, z \mapsto s^*] \models \rho_x(\alpha)[S_{xz}/a]$.

Proof. Let $\mathbb{T}$ be a P-tree with labelling function $\kappa : S \to \wp\mathrm{P}$, let $\mathbb{T}'$ denote the $\downarrow a$-variant $\mathbb{T}[b \mapsto S_b | b \in \downarrow a]$ of $\mathbb{T}$, and let $g' := g[x \mapsto s, z \mapsto s^*]$. Let $s, s^* \in S$ be given such that $s^* \in R^+(s)$, let $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$, and let $a \in A$. We proceed by induction on the structure of $\alpha \in \mathrm{MFOE}^+(A)$. First, we show that if α has the shape $b(y)$ for some $b \in A$, then the equivalence holds. We will distinguish two cases on the basis of whether or not $b = a$.

(1) If $b \neq a$, then the substitution $\rho_x(b(y))[S_{xz}/a]$ results in the formula $b(y)$ itself because relativization acts as the identity on atomic formulas and a does not occur in $b(y)$. Observe that $(R(s), V) \models b(y)$ iff $g(y) \in V(b) = S_b$ iff $\mathbb{T}' \models b(y)$ because $V_\kappa(b) = S_b$. Hence the equivalence holds when $b \neq a$.

(2) If $b = a$, then the substitution $\rho_x(b(y))[S_{xz}/a]$ results in the formula $S_{xz}(y)$. We begin by showing the implication from (i) to (ii); our goal is to show that $\mathbb{T}', g' \models S_{xz}(y)$. As $V(a) = \{s^+\}$, we have $g(y) = s^+$. In particular, $g(y) = g(x)^+$ whence $\mathbb{T}', g' \models S_{xz}(y)$, as required. Conversely, assume that (ii) holds so that $\mathbb{T}', g' \models S_{xz}(y)$. Then $g(y) = s^+$ so that the valuation $V' : A \to \wp D$ defined by $V(a) = \{s^+\}$ and $V(b) = \varnothing$ for every $b \in A \backslash \{a\}$ is such that $(D, V') \models a(y)$. By monotonicity, it follows that $(D, V) \models a(y)$ as well. Thus the equivalence holds whenever $b = a$.

The remaining atomic propositions either have the shape $y_1 \approx y_2$ or $y_1 \not\approx y_2$. Both cases are handled in a manner completely analogous to the proof of (1) because relativization acts as the identity on atomic formulas and because a does not occur in the formulas $y_1 \approx y_2$ and $y_1 \not\approx y_2$ (hence also the substitution acts as the identity). The Boolean cases in which α has the shape $\alpha_0 \vee \alpha_1$ or $\alpha_0 \wedge \alpha_1$ follow immediately from the induction hypothesis because relativization and the substitution $(\cdot)[S_{xz}/a]$ respect Booleans.

In order to conclude the proof, we now show that the equivalence holds for formulas of the shape $\exists y.\beta$. Recall that the relativization of $\exists y.\beta$ is the formula $\exists y.(S(x,y) \wedge \rho_x(\beta))$. Thus we wish to show that $(R(s), V) \models \exists x.\beta$ iff $\mathbb{T}', g' \models \exists y.(S(x,y) \wedge \rho_x(\beta)[S_{xz}/a])$. Observe that we have the following chain of equivalences:

$$\begin{aligned}
(R(s), V), g \models \exists y.\beta &\iff (R(s), V), g[y \mapsto d] \models \beta \text{ for some } d \in R(s) \\
&\iff \mathbb{T}', g'[y \mapsto d] \models \rho_x(\beta)[S_{xz}/a] && \text{(Ind. Hyp.)} \\
&\iff \mathbb{T}', g'[y \mapsto d] \models S(x,y) \wedge \rho_x(\beta)[S_{xz}/a] && (d \in R(s)) \\
&\iff \mathbb{T}', g' \models \exists y.S(x,y) \wedge \rho_x\beta[S_{xz}/a].
\end{aligned}$$

Thus, by induction, the equivalence holds for each formula $\alpha \in \mathrm{MFOE}^+(A)$. Hence also the equivalence holds for each one-step formula $\alpha \in \mathrm{FOE}_1(A)$, as desired. □

Putting the pieces together

Recall that for a set A of monadic predicates and a subset $B \subseteq A$, we write $\tau_B(x)$ to denote the formula $\bigwedge_{b\in B} b(x) \wedge \bigwedge_{b \notin B} \neg b(x)$. As an auxiliary formula, we define, for each automaton $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ on alphabet $\wp\mathrm{P}$ and each $a \in A$, the formula

$$\mathbf{exit}_{\mathbb{A},a}(z) := \bigvee_{c \in \wp\mathrm{P}} (\tau_c(z) \wedge \rho_z(\Theta(a,c))[\bot/a]).$$

Finally, we define $\chi_{\mathbb{A},a}(x)$ to be the formula

$$\exists z(x \leq z \wedge \mathbf{exit}_{\mathbb{A},a}(z) \wedge \forall y(x \leq y < z \rightarrow \bigvee_{c\in\wp\mathrm{P}} (\tau_c(y) \wedge \rho_y(\Theta(a,c))[S_{yz}/a]))).$$

Proposition 4.1.9. *For each automaton* $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ *and each* $a \in A^\mu$*, the following are equivalent for every* P*-tree* $\mathbb{T}$ *and every node* s *in* $\mathbb{T}$ *:*

(i) $(a, s) \in Win_\exists(\mathcal{A}(\mathbb{A}, \mathbb{T}))$

(ii) $\mathbb{T}[b \mapsto [\![\mathbb{A}.b]\!]^{\mathbb{T}} \mid b \in {\downarrow}a] \models \chi_{\mathbb{A},a}(s)$

Proof. Let $\mathbb{T}$ be a P-tree with labelling function $\kappa : S \to \wp\mathrm{P}$ and let $\mathbb{T}'$ denote the $(\mathrm{P} \uplus {\downarrow}a)$-tree $\mathbb{T}[b \mapsto [\![\mathbb{A}.b]\!]^{\mathbb{T}} \mid b \in {\downarrow}a]$ with labelling function $\kappa' : S \to \wp(\mathrm{P} \uplus {\downarrow}a)$. We will begin by showing the implication from (i) to (ii). To this end, suppose that $s \in S$ is such that $\mathbb{T}.s \models \mathbb{A}.a$. Our goal is to show that $\mathbb{T}' \models \chi_{\mathbb{A},a}(s)$. By inspection of the structure of $\chi_{\mathbb{A},a}(x)$, we wish to show that there is a node $s^* \in R^+(s)$ such that

(1) $\mathbb{T}', g[z \mapsto s^*] \models \rho_z(\Theta(a, \kappa(s^*)))[\bot/a]$ and

(2) for each $s \leq t < s^*$ we have $\mathbb{T}', g[y \mapsto t, z \mapsto s^*] \models \rho_z(\Theta(a, \kappa(t)))[S_{yz}/a]$.

Since $a \in A^\mu$ and $(a, s) \in \mathit{Win}_\exists(\mathcal{G})$ by assumption, there is a strategy f_π for $\exists$ in the game $\mathcal{G}$ as in Proposition 3.3.4 for the position (a, s). We will prove that (1) and (2) are satisfied by $s^* := \mathit{last}(\pi)$. In order to see that (1) holds, note that (a, s^*) is a winning position because it is f_π-reachable from the winning position (a, s). Hence $(R(s^*), V_{a,s^*}) \models \Theta(a, \kappa(s^*))$. Recall that the valuation V_{a,s^*} is such that $V_{a,s^*}(b) = \varnothing$ for each $b \in \Uparrow a$ by Proposition 3.3.4(i) and Proposition 3.3.4(ii). Furthermore, $V_{a,s^*}(b) \subseteq [\![\mathbb{A}.b]\!]^{\mathbb{T}} \cap R(s^*)$ for each $b \in \downarrow a$ because f_π is a winning strategy for $\exists$. By monotonicity, it follows that $(R(s^*), V) \models \Theta(a, \kappa(s^*))$ where $V : A \to \wp R(s^*)$ is the valuation described in Proposition 4.1.6 for the sets $S_b := [\![\mathbb{A}.b]\!]^{\mathbb{T}}$. Thus $\mathbb{T}', g[z \mapsto s^*] \models \rho_z(\Theta(a, \kappa(s^*)))[\bot/a]$.

We now proceed to show that (2) holds. We proceed just as before, only now we will blend Proposition 3.3.4 together with Proposition 4.1.8. To this end, let $t \in S$ be such that $s \leq t < s^*$. As the path from s to s^* is unique, it follows that $t = s_i$ for some $i < k$. Hence (a, t) is f-reachable from position (a, s) by Corollary 3.3.5. In particular, $(a, t) \in \mathit{Win}_\exists(\mathcal{G})$ so that $(R(t), V_{a,t}) \models \Theta(a, \kappa(t))$.By Proposition 3.3.4(ii), we have that $V_{a,t}(a) = \{t^+\}$. As before, $V_{a,t}(b) \subseteq [\![\mathbb{A}.b]\!]^{\mathbb{T}} \cap R(s)$ for each $b \in \downarrow a$ since f_π is a winning strategy for $\exists$. By monotonicity, it follows that $(R(t), V) \models \Theta(a, \kappa(t))$ for the valuation V defined in Proposition 4.1.8(i). Hence (2) holds and we have now completed the proof from (i) to (ii).

We now show the implication from (ii) to (i). To this end, assume that $\mathbb{T}' \models \chi_{\mathbb{A},a}(s)$. Then there exists a node $s^* \in S$ such that $s^* \in R^+(s)$ which satisfies (1) and (2). Let $(s_0, \ldots, s_k)$ be the unique finite path from $s = s_0$ to $s^* = s_k$ in $\mathbb{T}$. Our goal is to provide a winning strategy f for $\exists$ in the game $\mathcal{G}@(a, s)$. Let $f_\exists$ be a fixed positional winning strategy for $\exists$ in the game $\mathcal{G}$ guaranteed by the positional determinacy of parity games. In order to define the strategy f, we distinguish the following cases for basic positions $(q, t) \in A \times S$.

(**A**) If $q = a$ and $t = s_i$ for some $i \leq k$, then we distinguish the following cases on the basis of whether or not $i = k$:

(**a**) If $i < k$, let f suggest the valuation $V_{q,t} : A \to \wp R(t)$ defined by setting $V(a) = \{t^+\}$ and $V(b) = [\![\mathbb{A}.b]\!]^{\mathbb{T}} \cap R(t)$ for each $b \neq a$. Note that f suggests the valuation defined in Proposition 4.1.8(i) at (q, t).

(**b**) If $i = k$, we let f suggest the valuation $V_{q,t} : A \to \wp R(t)$ defined by setting $V(b) = \varnothing$ for each $b \in \Uparrow a$ and $V(b) = [\![\mathbb{A}.b]\!]^{\mathbb{T}} \cap R(t)$ for each $b \notin \Uparrow a$. Note that f suggests the valuation defined in Proposition 4.1.6(i) at (q, t).

(**B**) If $q \in \downarrow a$ and $(q, t) \in Win_\exists(\mathcal{G})$, we let f suggest the same valuation $V_{q,t}$ that her positional winning strategy $f_\exists$ suggests.

(**C**) At all other positions, $\exists$ plays randomly.

The legitimacy of the valuations suggested by f at position described in $\mathbf{A}(a)$ and $\mathbf{A}(b)$ follows immediately by (1) together with Proposition 4.1.8 and (2) together with Proposition 4.1.6, respectively. Furthermore, the moves defined in **B** are legitimate for f because they are legitimate for her winning strategy $f_\exists$. In order to show that f is in fact a winning strategy for $\exists$ in $\mathcal{G}@(a, s)$, it suffices to show that any partial f-guided match leads to a winning position (q, t) such that $q \in \downarrow a$.

Indeed, let π be an f-guided match. Due to the legitimacy of her moves, the strategy f is surviving for $\exists$ so she wins every finite match. Now, assume that $\pi = (a, s)(a_1, s_1)\cdots$ is an infinite f-guided match. By the claim below, there is a winning position (a_i, t_i) such that $a_i \in \downarrow a$. Now, observe that $\pi' = (a_i, t_i)(a_{i+1}, t_{i+1})\cdots$ is an infinite $f_\exists$-guided match in the game $\mathcal{G}@(a_i, t_i)$ by definition of the strategy f. As $f_\exists$ is a winning strategy, it follows that π' is a winning match for $\exists$. Hence also π is a winning match for $\exists$ because the parity of the two matches disagree for only a finite initial segment of π. So, indeed, the following claim concludes the proof.

Claim. *For each $n < \omega$ and each partial f-guided match $\pi = (a_1, t_1)\cdots(a_n, t_n)$ in the game $\mathcal{G}@(a, s)$, either*

(I) $a_i = a$ and $t_i = s_i$ for each $i \leq n$ or

(II) there exists $j \leq n$ such that $(a_j, t_j) \in Win_\exists(\mathcal{G})$, $a_j \in \downarrow a$, and $t_j \neq s_i$ for each $i \leq j$.

Proof. We proceed by induction on $n \in \omega$. If $n = 1$, then π is the initialized match consisting of the single position (a, s). Hence (II) is satisfied if $n = 1$. Now, inductively assume that for each $1 \leq n \leq m$ we have that every partial f-guided match $(a_1, t_1)\cdots(a_n, t_n)$ satisfies either (I) or (II). Let $\pi = (a_1, t_1)\cdots(a_{m+1}, t_{m+1})$ be a partial

f-guided match in $\mathcal{G}@(a,s)$; we aim to show that (I) or (II) obtains. Define π_I to be the partial f-guided match $(a_1,t_1)\cdots(a_m,t_m)$ consisting of the first m rounds of π. By the inductive hypothesis, the match π_I satisfies either (I) or (II); we distinguish cases on this basis.

If (II) is satisfied by π_I, then it immediately follows that (II) is satisfied by π as well because π_I is an initial segment of π. Otherwise, the match π_I satisfies (I). We further distinguish two cases on the basis of whether $m = k$ or $m < k$.

- If $m = k$, then $(a_m, t_m) = (a, s^*)$ by assumption. At position (a, s^*), the strategy f suggests the valuation V_{a,s^*} defined in $\mathbf{A}(b)$. By definition, we have $V(b) \neq \varnothing$ only if $b \in {\downarrow}a$. Moreover, for each $b \in {\downarrow}a$ we have $V_{a,s^*}(b) \subseteq [\![\mathbb{A}.b]\!]^{\mathbb{T}}$. In particular, the position (a_{m+1}, t_{m+1}) is a winning position and $a_{m+1} \in {\downarrow}a$. Finally note that $t_{m+1} \neq s_i$ for any $i \leq k$ because $t_{m+1} > s_k$. Hence (II) is satisfied.
- If $m < k$, then the valuation suggested by f at $(a_m, t_m) = (a, s_m)$ was defined in such a manner that if $t \in V_{a,s_m}(b)$, then either $b = a$ and $t = s_{m+1}$ or $b \in {\downarrow}a$ and $t \in [\![\mathbb{A}.b]\!]^{\mathbb{T}}$. Note that this simply states that the position (a_{m+1}, t_{m+1}) either satisfies (I) or (II), as desired.

Hence, by induction, every f-guided partial match satisfies either (I) or (II), as claimed. □

□

As a corollary, we obtain the analogue of this proposition for ν-states. The key observation is that each ν-state a in an automaton $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ is a μ-state in the complement $\mathbb{A}^\delta$.

Corollary 4.1.10. *For each automaton $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ and each $a \in A$, the following are equivalent for every* P*-tree $\mathbb{T}$ and every node s in $\mathbb{T}$:*

$$(a,s) \in Win_\exists(\mathcal{A}(\mathbb{A},\mathbb{T})) \text{ if and only if } \mathbb{T}[b \mapsto [\![\mathbb{A}.b]\!]^{\mathbb{T}} | b \in {\downarrow}a] \models \chi_{\mathbb{A},a}(s).$$

Proof. As mentioned, for each automaton $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ and each $a \in A^\nu$, we have that a is a μ-state in the complement automaton $\mathbb{A}^\delta$ by construction. By Proposition 3.3.6, it follows that $\mathbb{A}^\delta \in Aut^l_{sa}(\mathrm{FOE}_1)$ as well. Thus, by Proposition 4.1.9, there is a formula $\chi_{\mathbb{A}^\delta,a}(x) \in \mathrm{FOE}(\mathrm{P}{\uplus}{\downarrow}a)$ such that the following chain of equivalences holds for each tree $\mathbb{T}$ and each node s in $\mathbb{T}$:

$$\begin{aligned}
\mathbb{T}.s \models \mathbb{A}.a &\iff \mathbb{T}.s \not\models \mathbb{A}^\delta.a && \text{(Theorem 2.6.13)}\\
&\iff \mathbb{T}[b \mapsto [\![\mathbb{A}^\delta.b]\!]^{\mathbb{T}} | b \in {\downarrow}a] \not\models \chi_{\mathbb{A}^\delta,a}(s) && \text{(Proposition 4.1.9)}\\
&\iff \mathbb{T}[b \mapsto [\![\mathbb{A}^\delta.b]\!]^{\mathbb{T}} | b \in {\downarrow}a] \models \neg\chi_{\mathbb{A}^\delta,a}(s). && \text{(semantics of FOE)}
\end{aligned}$$

Hence, for each $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ and each $a \in A^\nu$, there exists a formula $\chi_{\mathbb{A},a}(x) \in \mathrm{FOE}(\mathrm{P}{\uplus}{\downarrow}a)$ satisfying the equivalence in Proposition 4.1.9. □

The following simultaneous substitution of formulas will be utilized in the inductive step of the proof of Proposition 4.1.1.

Definition 4.1.11. Let P be a set of propositional variables such that $\{a_1, \ldots, a_n\} \subseteq \mathrm{P}$ and let $\{\psi_1, \ldots, \psi_n\} \subseteq \mathrm{2FOE}(<, \mathrm{P})$ be a set of formulas. For each formula $\varphi \in \mathrm{2FOE}(<, \mathrm{P})$, we write $\varphi[\psi_i/a_i | i \leq n]$ to denote the formula obtained from φ by replacing each occurrence of $a_i(x)$ by the formula ψ_i for each $i \leq n$.

Lemma 4.1.12. *Let $A = \{a_1, \ldots, a_k\}$ be a set of monadic predicates and let $\{\psi_1, \ldots, \psi_k\} \subseteq \mathrm{2FOE}(<, \mathrm{P})$ be a set of formulas. For every formula $\varphi \in \mathrm{2FOE}(<, \mathrm{P} \uplus A)$, we have $\varphi[\psi_i/a_i \mid i \leq n] \in \mathrm{2FOE}(<, P)$.*

Proof. By induction on the complexity of the formula $\varphi \in \mathrm{2FOE}(\mathrm{P} \uplus A)$. □

We have now gathered all of the ingredients to prove Theorem 4.1.1. Before we do so, recall that $\mathbb{A}.b$ is the automaton (A, Θ, Ω, b) which is identical to $\mathbb{A}$ only now the initial state is $b \in A$.

Proof of Theorem 4.1.1. For each automaton $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$ on alphabet $\wp\mathrm{P}$, we provide an equivalent sentence $\xi_{\mathbb{A}} \in \mathrm{2FOE}(<, \mathrm{P})$. To this end, we proceed by induction on the well-founded relation $\lhd_{\mathbb{A}}$.

If $\downarrow a_I = \varnothing$, then $\mathbb{A}$ consists of the single state a_I. By Corollary 4.1.10, there exists a formula $\chi_{\mathbb{A},a_I} \in \mathrm{2FOE}(\mathrm{P} \uplus \downarrow a_I)$ such that for every tree $\mathbb{T}$ and every node s in $\mathbb{T}$ we have

$$(a_I, s) \in Win_\exists(\mathcal{A}(\mathbb{A}, \mathbb{T})) \text{ if and only if } \mathbb{T}[b \mapsto [\![\mathbb{A}.b]\!]^{\mathbb{T}} \mid b \in \downarrow a] \models \chi_{\mathbb{A},a_I}(s). \qquad (*)$$

As $\downarrow a_I = \varnothing$, we have that $\mathbb{T}[b \mapsto [\![\mathbb{A}.b]\!]^{\mathbb{T}} \mid b \in \downarrow a] = \mathbb{T}$. Note that by taking the node s to be the root s_I of $\mathbb{T}$ in $(*)$, we have

$$\mathbb{T} \models \mathbb{A} \text{ if and only if } \mathbb{T} \models \chi_{\mathbb{A},a_I}(s_I).$$

We define the sentence $\xi_{\mathbb{A}} := \exists x.root(x) \wedge \chi_{\mathbb{A},a_I}(x)$. As $\downarrow a_I = \varnothing$, it follows that $\chi_{\mathbb{A},a_I}(x) \in \mathrm{2FOE}(\mathrm{P})$ hence also $\xi_{\mathbb{A}} \in \mathrm{2FOE}(\mathrm{P})$. That is, $\xi_{\mathbb{A}}$ has the right shape. Moreover, by combining the semantics of $root(x)$ with $(*)$, we have that $\mathbb{T} \models \xi_{\mathbb{A}}$ iff $\mathbb{A}$ accepts $\mathbb{T}$, as required.

Now, inductively assume that for each $b \in \downarrow a$ we are given a sentence $\xi_{\mathbb{A}.b} \in \mathrm{2FOE}(\mathrm{P})$ which is equivalent to the automaton $\mathbb{A}.b$. Again, let $\chi_{\mathbb{A},a_I}(x) \in \mathrm{2FOE}(\mathrm{P} \uplus \downarrow a_I)$ be the formula guaranteed by Proposition 4.1.9. We define the formula

$$\xi_{\mathbb{A}} := \exists x.root(x) \wedge \psi_{\mathbb{A},a_I}(x)[b \mapsto \xi_{\mathbb{A}.b} | b \in \downarrow a].$$

As $\chi_{\mathbb{A},a_I}(x)$ contains only monadic predicates from $\mathrm{P} \uplus \downarrow a_I$ and each of the sentences $\xi_{\mathbb{A}.b}$ is A-free, it follows from Lemma 4.1.12 that $\xi_{\mathbb{A}} \in \mathrm{2FOE}(\mathrm{P})$. It follows immediately by Proposition 4.1.9 that for every tree $\mathbb{T}$ we have $\mathbb{A}$ accepts $\mathbb{T}$ if and only if $\mathbb{T} \models \xi_{\mathbb{A}}$. □

4.2 From AMSO formulas to linked weak path automata

We will now provide an effective translation from AMSO formulas into parity-linked weak-path automata.

Proposition 4.2.1. *For every formula* φ *of* AMSO(P) *we can effectively construct an equivalent automaton* $\mathbb{A}_\varphi \in Aut^l_{wa}(\mathrm{FOE}_1)$ *on alphabet* $\wp\mathrm{P}$.

We will prove this proposition by induction on the complexity of formulas. As mentioned in the introduction to this chapter, the most challenging part of this induction lies in the inductive step corresponding to existential quantification. In order to simulate first-order quantification (i.e. quantification over *singletons*), we will provide a construction $\exists p.(-) : Aut^l_{wa}(\mathrm{FOE}_1) \to Aut^l_{wa}(\mathrm{FOE}_1)$ such that for every automaton $\mathbb{A}$ on alphabet $\wp(\mathrm{P} \uplus \{p\})$ and every P-tree $\mathbb{T}$, the following equivalence holds:

$$\mathbb{A} \text{ accepts } \mathbb{T} \text{ iff } \exists p.\mathbb{A} \text{ accepts } \mathbb{T}[p \mapsto \{t\}] \text{ for some } t \in S.$$

We first recast this property as the closure under *atomic projection* of the class of tree languages recognized by weak-path automata.

Closure of weak-path automata under atomic projection

Definition 4.2.2. Fix a set P of propositional letters. Let $\mathcal{T}$ be a tree language of $(\mathrm{P} \uplus \{p\})$-trees. The *atomic projection* of $\mathcal{T}$ over p is the tree language $\exists p.\mathcal{T}$ of P-trees defined as

$$\exists p.\mathcal{T} := \{\mathbb{T} \mid \mathbb{T}[p \mapsto \{t\}] \in \mathcal{T} \text{ for some } t \in S\}.$$

A class $\mathcal{A}$ of automata is *closed under atomic projection* if for each $\mathbb{A} \in \mathcal{A}$, there exists $\mathbb{A}' \in \mathcal{A}$ such that $\mathrm{TMod}(\mathbb{A}') = \exists p.\mathbb{A}'$.

In words, the atomic projection of a tree language $\mathcal{T}$ over a propositional variable p is the tree language containing precisely those trees $\mathbb{T}$ which admit some atomic p-variant $\mathbb{T}' \in \mathcal{T}$. We will now proceed to show that the class $Aut^l_{wa}(\mathrm{FOE}_1)$ is closed under atomic projection. To be precise, we will prove the following proposition.

Proposition 4.2.3. *For each automaton* $\mathbb{A} \in Aut^l_{wa}(\mathrm{FOE}_1)$ *on alphabet* $\wp(\mathrm{P} \uplus \{p\})$ *we can effectively obtain an automaton* $\exists p.\mathbb{A} \in Aut^l_{wa}(\mathrm{FOE}_1)$ *on alphabet* $\wp\mathrm{P}$ *such that the following are equivalent for every tree* $\mathbb{T}$*:*

(i) $\exists p.\mathbb{A}$ *accepts* $\mathbb{T}$

(ii) $\mathbb{A}$ *accepts an atomic* p*-variant of* $\mathbb{T}$.

In other words, $\mathrm{TMod}(\exists p.\mathbb{A}) = \exists p.(\mathrm{TMod}(\mathbb{A}))$.

That is, for each automaton $\mathbb{A}$ on alphabet P, we will provide an automaton $\exists p.\mathbb{A}$ on $\mathrm{P}\backslash\{p\}$ for each $p \in \mathrm{P}$ and, for each tree $\mathbb{T}$, the automaton $\exists p.\mathbb{A}$ accepts $\mathbb{T}$ iff $\mathbb{T}$ admits an atomic p-variant $\mathbb{T}_p$ which is accepted by $\mathbb{A}$. In contrast with our previous work, note that we now compare automata over different alphabets.

Intuitively, given a $\mathrm{P}\backslash\{p\}$-tree $\mathbb{T}$, the automaton $\exists p.\mathbb{A}$ should behave as follows: initially, $\exists p.\mathbb{A}$ will search for a node t_p to colour with p–or, equivalently, a node t_p which it will treat as if it is already labelled with p. As soon as it chooses such a node, it 'switches' into a final mode which corresponds to an acceptance game for the automaton $\mathbb{A}$ and the subtree $\mathbb{T}.t_p$. In short, we will want to partition each match π of the acceptance game $\mathcal{A}(\exists p.\mathbb{A}, \mathbb{T})$ into two parts, corresponding to a 'search for p' mode and an 'alternating mode'.

Based on ideas introduced by Facchini, Venema, and Zanasi ([37], [15]), a natural first step towards such a division of 'modes' is to divide the carrier of $\exists p.\mathbb{A}$ into two 'sorts', which we will call macro-states (from $\wp A$) and states (from A), respectively. That is, we will take that carrier of $\exists p.\mathbb{A}$ to be the set $\wp A \uplus A$ where A is the carrier of the automaton $\mathbb{A}$. In order to simulate the intuitive behaviour of the automaton $\exists p.\mathbb{A}$ sketched above, we will craft the transition map of the automaton $\exists p.\mathbb{A}$ in such a manner that the following are satisfied:

- transitions 'across' sorts only occur from macro-states to states;
- for each pair of states $a, b \in A$, we have $b \lhd_{\mathbb{A}} a$ iff $b \lhd_{\exists p.\mathbb{A}} a$.

That is, the graph structure of $\mathbb{A}$ is preserved at the level of states in the automaton $\exists p.\mathbb{A}$. Combining these properties with an appropriate priority map will prove to be enough to capture that the alternating mode occurs cofinally in each match of the acceptance game.

However, we will require something slightly more about the 'search mode': the acceptance game $\mathcal{A}(\exists p.\mathbb{A}, \mathbb{T})$ admits numerous concurrent matches depending on the choices $\forall$ makes. In particular, distinct matches may suggest distinct atomic p-variants of a given tree $\mathbb{T}$, and these choices may be crucial. The point is this: we require a uniform choice of which node to colour p across all matches. Fortunately, this is possible through the notion of *non-determinism.*

Muller and Schupp [26] introduced an interesting approach to proving that a class of automata is closed under such-and-such projection, called a *simulation theorem.* The key observation is that, for special classes of automata, each automaton $\mathbb{A}$ can be effectively transformed into an equivalent non-deterministic automaton $\exists p.\mathbb{A}$. Intuitively, an automaton $\mathbb{A}$ is *non-deterministic* if every winning strategy for $\exists$ in the game $\mathcal{A}(\mathbb{A}, \mathbb{T})$ may be reduced to a winning strategy which is *functional* in A.

Definition 4.2.4. Given a parity automaton $\mathbb{A}$ and a transition system $\mathbb{S}$, a strategy f for $\exists$ in the initialized acceptance game $\mathcal{A}(\mathbb{A},\mathbb{S})@(a,t)$ is *functional* in $B \subseteq A$ if for every $s \in S$ there is at most one $b \in B$ such that (b,s) is a reachable position in an f-guided match starting at (a,t).

In other words, an automaton $\mathbb{A}$ *non-deterministic* if, whenever $\forall$ has the choice between playing the positions (a,s) and (b,s), we have that $a{=}b$. Zanasi ([37], Remark 3.5) observed that this construction does not preserve the weakness condition. The good news is this: we only need the automaton $\exists p.\mathbb{A}$ to be non-deterministic while it is in 'search mode' (i.e. $\wp A$). Keeping functionality in mind, we will now introduce a transformation on one-step formulas that will be needed in order to define the transition map. In essence, these formulas will 'control' the transitions between macro-states.

Additive liftings

In order to define the non-deterministic part of the automaton $\exists p.\mathbb{A}$, we will now define a translation $\widehat{(-)} : \mathrm{MFOE}^+(A) \to \mathrm{MFOE}^+(\wp A \cup A)$ which *lifts* each formula $\alpha \in \mathrm{MFOE}^+(A)$ to a formula $\widehat{\alpha} \in \mathrm{MFOE}^+(\wp A \cup A)$. Before we do so, we will briefly discuss some properties that this translation ought to exhibit.

First, as we want the automaton $\exists p.\mathbb{A}$ to behave non-deterministically in $\wp A$, we will want the lifting of a formula to be defined in such a manner that whenever the lifting can be made true in some one-step model, it can be made true by assigning either none or a unique macro-state to every node from the domain. We express this property of formulas through the notion of *separability.*

Definition 4.2.5. Let A be a set with $B \subseteq A$. A valuation $V : A \to \wp D$ is B*-separating* if $|m_V(d) \cap B| \leq 1$ for every $d \in D$. We say that a formula $\alpha \in \mathrm{FOE}_1(A)$ is B*-separating* if α is monotone in B and for each one-step model (D,V) such that $(D,V) \models \alpha$, there exists a B-separating valuation $V' : A \to \wp D$ such that $V' \leqslant_B V$ and $(D,V') \models \alpha$.

Now, we additionally want the automaton $\exists p.\mathbb{A}$ to stay in its non-deterministic mode for only a finite initial part of each run. As such, we will assign each macro-state an odd parity. In order to make sure that $\exists p.\mathbb{A}$ lands in the right class of automata, this means that the formula $\Theta^{\exists}(B,c)$ should be completely additive in $\wp A$. Recall that, informally, α is completely additive in a set A' whenever any valuation making α true in some domain can be shrunk to valuation assigning either none or a unique state some colour from A'. With all of this being said, we now define the notion of the *additive lifting* of a one-step formula, drawing inspiration from those developed by Carreiro et al. ([6], [7], [8]).

Definition 4.2.6. For each formula $\alpha = \nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}},\Pi) \in \mathrm{FOE}_1(A)$ in positive basic form where $\overline{\mathbf{T}} = \{T_1,\ldots,T_k\}$, the *additive lifting* of α is the formula $\widehat{\alpha} \in \mathrm{FOE}_1(\wp A \uplus A)$

defined as follows:

$$\exists x_1 \dots x_k(\text{diff}(\overline{\mathbf{x}}) \wedge \pi(\overline{\mathbf{x}}, \Pi) \wedge \bigvee_{i \leq k} (T_i(x_i) \wedge \bigwedge_{j \neq i} \tau^+_{T_j}(x_j)))$$
$$\vee\ \exists \overline{\mathbf{x}} x_{k+1}(\text{diff}(\overline{\mathbf{x}}, x_{k+1}) \wedge \pi(\overline{\mathbf{x}}, x_{k+1}, \Pi) \wedge \bigwedge_{i \leq k} \tau^+_{T_i}(x_i) \wedge \bigvee_{B \in \Pi} B(x_{k+1})).$$

The following proposition states that additive liftings have the essence explained above.

Proposition 4.2.7. *Let $\alpha \in \text{FOE}_1(A)$ be a sentence in positive basic form. For every one-step model $\mathbb{D} = (D, V{:}\wp A \uplus A \to \wp D)$ such that $\mathbb{D} \models \widehat{\alpha}$, there exists a valuation $V^s : \wp A \uplus A \to \wp D$ such that $V^s \leqslant_{\wp A} V$ and such that*

(i) $(D, V^s) \models \widehat{\alpha}$

(ii) there exists $B \in \wp A$ such that $V'(B)$ is a singleton and $V'(B') = \varnothing$ for each $B' \in \wp A$ such that $B \neq B'$.

In particular, $\widehat{\alpha}$ is completely additive in $\wp A$ and $\wp A$-separating.

Proof. Suppose that $\mathbb{D} \models \widehat{\alpha}$. We distinguish cases on the basis of which disjunct of $\widehat{\alpha}$ is true in $\mathbb{D}$. We shall restrict our attention to showing that if

$$\mathbb{D} \models \exists x_1 \dots x_k(\text{diff}(\overline{\mathbf{x}}) \wedge \pi(\overline{\mathbf{x}}, \Pi) \wedge \bigvee_{i \leq k} (T_i(x_i) \wedge \bigwedge_{j \neq i} \tau^+_{T_j}(x_j))), \qquad (*)$$

then there is a valuation $V^s \leqslant_{\wp A} V$ with the properties (i) and (ii); the remaining case in which the other disjunct of $\widehat{\alpha}$ is true in $\mathbb{D}$ is a straightforward adaptation of the following reasoning. Unwinding the meaning of $(*)$, it follows that there exist distinct nodes $d_1, \dots, d_k \in D$ such that, for some $i \leq k$, the following properties are satisfied:

(1) $T_i \in m_V(d_i)$ for some $i \leq k$;

(2) for every $a \in T_j$ we have that $a \in m_V(d_j)$ for each $j \leq k$ such that $j \neq i$;

(3) $\mathbb{D} \models \pi(\overline{\mathbf{d}}, \Pi)$.

We define the valuation $V^s : \wp A \cup A \to \wp(D)$ in terms of its associated marking m_{V^s} by setting

$$m_{V^s}(d) := \begin{cases} m_V(d) \cap A & \text{if } d \neq d_i \\ \{T_i\} & \text{if } d = d_i. \end{cases}$$

It is straightforward to check that the 'shrinking' valuation V^s preserves (1), (2), and (3) hence also $(D, V^s) \models \widehat{\alpha}$. It remains to be seen that (ii) is satisfied by V^s. Simply note that if $d \neq d_i$, then $m_{V^s}(d) \subseteq A$ whence, for every $B \in \wp A$, we have that $d \notin V^s(B)$. Also note that $d_i \in V^s(B)$ iff $B = T_i$. Hence $V^s(T_i)$ is a singleton and $V^s(B) = \varnothing$ for each $B \in \wp A$ such that $B \neq T_i$. In short, V^s satisfies (i) and (ii). Now, in order to see that $\widehat{\alpha}$ is completely additive in $\wp A$ and $\wp A$-separating, simply note that the valuation V^s is such that $|V^s(B)| \leq 1$ for every $B \in \wp A$. □

We will now prove the following proposition stating that we may transform each valuation making α true into a valuation making its additive lifting $\widehat{\alpha}$ true in the same set D.

Proposition 4.2.8. *Let $\alpha \in \mathrm{FOE}_1(A)$ be in positive basic form and let $(D, V : A \to \wp D)$ be a one-step model such that $(D, V) \models \alpha$. Then, for each node $d \in D$, there exists a valuation $V_d^{\Uparrow} : (\wp A \uplus A) \to \wp D$ such that*

(i) $(D, V_d^{\Uparrow}) \models \widehat{\alpha}$;

(ii) for some set $B_d \subseteq m_V(d)$, we have $V_d^{\Uparrow}(B_d) = \{d\}$ and for each $B \in \wp A$ such that $B \neq B_d$, we have $V_d^{\Uparrow}(B) = \varnothing$;

(iii) $V_d^{\Uparrow}(a) \subseteq V(a)$ *for each $a \in A$.*

Proof. Suppose that α is a sentence in positive basic form and let $(D, V : A \to \wp D)$ be a one-step model such that $(D, V) \models \alpha$. Then there exist distinct nodes $d_1, \ldots, d_k \in D$ such that

(1) for each $i \leq k$, we have that $T_i \subseteq m_V(d_i)$ and

(2) $(D, V) \models \pi(\overline{\mathbf{d}}, \Pi)$.

For each $d \in D$ such that $d \neq d_i$, let $T_d \in \Pi$ be such that $T_d \subseteq m_V(d)$, and for each d_i define $T_{d_i} = T_i$. As a first step, we define a 'shrinking' $U : A \to \wp D$ of the valuation V in terms of its associated marking: for each $d \in D$, we define $m_U(d) = m_V(d) \cap T_d$. Note that, in fact, $m_U(d) = T_d$ for each $d \in D$. Clearly, the one-step model (D, U) also satisfies both (1) and (2) hence also $(D, U) \models \alpha$. Now, let $d \in D$ be fixed but arbitrary. We define $V_d^{\Uparrow} : A^{\exists} \to \wp D$ in terms of its associated marking as follows:

$$m_{V_d^{\Uparrow}}(d) := \begin{cases} m_U(d') & \text{if } d' \neq d \\ \{m_U(d')\} & \text{if } d' = d. \end{cases}$$

Items (ii) and (iii) both follow directly from the definition of $V_d^{\Uparrow}$. We proceed to show that $\mathbb{D}_d^{\Uparrow} := (D, V_d^{\Uparrow}) \models \widehat{\alpha}$. In this direction, we distinguish cases on the basis of whether or not $d = d_i$ for some $i \leq k$.

If $d = d_i$ for some $i \leq k$, then $T_i = m_V(d)$ by (1). Hence $T_i \in m_{\widehat{V}_d}(d) = \{m_U(d)\} = \{T_i\}$. Furthermore, for each $j \leq k$ such that $i \neq j$ we have $T_j = m_U(d_j) = m_{\widehat{V}}(d_j)$ by definition of the valuation $\widehat{V}_d$. Combining these observations, we have

$$\mathbb{D}_d^{\Uparrow} \models T_i(d_i) \wedge \bigwedge_{i \neq j} \tau^+_{T_j}(d_j).$$

Similarly, it follows from (2) and the definition of $\widehat{V}_d$ that $\mathbb{D}_d^{\Uparrow} \models \pi(\overline{\mathbf{d}}, z, \Pi)$. Hence $\mathbb{D}_d^{\Uparrow} \models \widehat{\varphi}$, as required.

Now suppose that $d \in D\backslash\{d_1, \ldots, d_k\}$. Then $\mathbb{D}_d^{\Uparrow} \models \text{diff}(\overline{\mathbf{d}}, d)$. By definition of the valuation U, we have $T_d = m_U(d)$ whence $T_d \in m_{\widehat{V}_d}(d)$ by the definition of $\widehat{V}_d$. Furthermore, for each $i \leq k$, we have that $T_i \subseteq m_U(d_i) = m_{\widehat{V}_d}(d_i)$ by (1) and the definition of U. Hence

$$\mathbb{D}_d^{\Uparrow} \models \bigvee_{B \in \Pi} B(d) \wedge \bigwedge_{i \leq k} \tau^+_{T_i}(d_i).$$

Just as before, (2) implies $\mathbb{D}_d^{\Uparrow} \models \pi(\overline{\mathbf{d}}, d, \Pi)$. In short, $\mathbb{D}_d^{\Uparrow} \models \widehat{\varphi}$, as desired. □

The atomic projection of an MSO-automaton

For each alternating parity automaton $\mathbb{A}$ and each set $B \subseteq A$ of states, we define

$$\Theta(B, c) := \bigwedge_{a \in B} \Theta(a, c).$$

We now have all of the information that will be needed to define the construction $\exists p.(-)$.

Definition 4.2.9. Let $\mathbb{A} = \langle A, \Theta, \Omega, a_I \rangle$ be an MSO-automaton. Fix a colour $c \in C$ and a set $B \subseteq A$. By Theorem 2.5.14 there exists a sentence $\psi_{B,c} \in \text{FOE}_1(A)$ of the form $\bigvee \alpha_i$ where each disjunct α_i is of the form $\nabla^+_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$ for some $\overline{\mathbf{T}} \in \wp(A)^k$ and some $\Pi \subseteq \overline{\mathbf{T}}$, and such that $\psi_{B,c} \equiv \Theta(B, c)$. We define $\widehat{\Theta}(B, c) := \bigvee \widehat{\alpha}_i$, where $\widehat{\alpha}_i$ is the additive lifting of α_i as defined in Definition 4.2.6.

Note that $\widehat{\Theta}(B, c) \in \text{FOE}_1(\wp A \cup A)$. We now define the atomic projection of an MSO-automaton.

Definition 4.2.10. Let $\mathbb{A} = \langle A, \Theta, \Omega, a_I \rangle$ be an MSO-automaton on $\text{P} \uplus \{p\}$. The *atomic projection* of $\mathbb{A}$ over p is the MSO-automaton $\exists p.\mathbb{A} := \langle A^\exists, \Theta^\exists, \Omega^\exists, a_I^\exists \rangle$ on P whose components are given as follows:

$$\begin{array}{lll} A^\exists := \wp A \uplus A & \Omega^\exists(a) := \Omega(a) & \Theta^\exists(a, c) := \Theta(a, c) \\ a_I^\exists := \{a_I\} & \Omega^\exists(B) := 1 & \Theta^\exists(B, c) := \widehat{\Theta}(B, c) \vee \Theta(B, c \cup \{p\}) \end{array}$$

where $a \in A$ and $B \in \wp A$.

Proposition 4.2.11. *If* $\mathbb{A} \in Aut^l_{wa}(\text{FOE}_1)$, *then* $\exists p.\mathbb{A} \in Aut^l_{wa}(\text{FOE}_1)$.

Proof. Suppose that $\mathbb{A} \in Aut^l_{wa}(\text{FOE}_1)$ and let $\exists p.\mathbb{A}$ be its atomic projection over the propositional variable p. Our goal is to show that $\exists p.\mathbb{A}$ satisfies weakness and the parity-linked path condition. We begin by showing that $\exists p.(\mathbb{A})$ is weak.

To this end, first observe, for each state $a \in A$ and each macro-state $B \in \wp A$, the relation $B \triangleleft_{\exists p.\mathbb{A}} a$ does not hold because $\Theta(a, c) \in \text{FOE}_1(A)$ for each colour c. This means that for each cluster C of $\exists p.\mathbb{A}$, either $C \subseteq A$ or $C \subseteq \wp A$. Moreover, every cluster $C \subseteq A$ is also a cluster of $\mathbb{A}$ because $\Theta^{\exists}(a, c) = \Theta(a, c)$ for every $a \in A$ and every colour c. Hence $\Omega^{\exists}(a) = \Omega^{\exists}(b)$ for every cluster $C \subseteq A$ of $\exists p.\mathbb{A}$ and each pair of states $a, b \in C$ because $\Omega^{\exists}(a) = \Omega(a)$ for each $a \in A$ and $\mathbb{A}$ is weak. For each cluster $C \subseteq \wp A$, the weakness condition is trivially satisfied because $\Omega^{\exists}(B) = 1$ for every $B \in \wp A$. In short, $\exists p.\mathbb{A}$ is weak.

We now show that $\exists p.\mathbb{A}$ satisfies the path condition. It is entirely straightforward to see that for each state $a \in A$, the formula $\Theta^{\exists}(a, c) = \Theta(a, c)$ is completely additive in a whenever $a \in (A^{\exists})^{\mu}$ and completely multiplicative in a whenever $a \in (A^{\exists})^{\nu}$ because $\mathbb{A}$ is a path automaton and $\Omega^{\exists}(a) = \Omega(a)$. As $B \in (A^{\exists})^{\mu}$ for every macro-state B, it now suffices to show that $\Omega^{\exists}(B)$ is completely additive in B for each macro-state B. Simply note that $\Theta(B, c)$ is completely additive in B because it is $\wp A$-free by construction and $\widehat{\Theta}(B, c)$ is completely additive in B by Proposition 4.2.12. Hence $\Theta^{\exists}(B, c)$ is completely additive in B by the syntactic characterization given in Proposition 3.2.3, as desired. □

Functionality

We will now show that the automaton $\mathbb{A}$ is in fact non-deterministic in $\wp A$. That is, we will show that every winning strategy for $\exists$ in the acceptance game $\mathcal{A}(\exists p.\mathbb{A}, \mathbb{T})$ can be transformed into a winning strategy which is *functional* in $\wp A$. Actually, we will prove something a bit stronger than functionality in $\wp A$. In particular, we show that $\exists$ has a winning strategy which "traces a finite $\wp A$-path" through $\mathbb{T}$, as in the proposition below. Note that this proposition states that the automaton $\exists p.\mathbb{A}$ has the essence described in the introduction to this section.

Lemma 4.2.12. *Let* $\mathbb{A} \in Aut^l_{wa}(\text{FOE}_1)$, *let* $\mathbb{T}$ *be a tree, and let* $\mathcal{G} = \mathcal{A}(\exists p.\mathbb{A}, \mathbb{T})$ *where* $\exists p.\mathbb{A}$ *is the atomic projection of* $\mathbb{A}$. *For every* $B \in \wp A$ *and every node* $s \in S$ *such that* $(B, s) \in Win_{\exists}(\mathcal{G})$, *there is a winning strategy* f_{π} *for* $\exists$ *in the game* $\mathcal{G}@(B, s)$ *such that there exists a unique finite path* $\pi \in Paths_s(\mathbb{T})$ *with the following properties.*

(i) For each winning position $(a, t) \in A \times S$, *the valuation* $V_{a,t}$ *suggested by* f_{π} *from position* (a, t) *is such that* $V_{a,t}(Q) = \varnothing$ *for each* $Q \in \wp A$.

(ii) *For each $i < length(\pi)$ and each $B \in \wp A$ such that $(B, s_i) \in Win_{\exists}(\mathcal{G})$, the valuation V_{B,s_i} suggested by f_π at the position $(B, s_i) \in \wp A \times S$ is such that $V_{B,s_i}(B^+)$ is a singleton for some $B^+ \in \wp A$ and $V_{B,s_i}(Q) = \varnothing$ for every $Q \in \wp A$ such that $Q \neq B^+$. If $s = last(\pi)$, and $(B, s) \in Win_{\exists}(\mathcal{G})$, then $V_{B,s_i}(Q) = \varnothing$ for every $Q \in \wp A$.*

Proof. Suppose that $(B, s) \in \wp A \times S$ is a winning position and let f be a positional winning strategy for $\exists$ in the game $\mathcal{G}$. Without loss of generality, we assume that f is in fact a minimal strategy. Our goal is to provide a winning strategy f_π for $\exists$ in the game $\mathcal{G}@(B, s)$ which is functional in $\wp A$. As usual, we write $V_{q,t}$ to denote the valuation suggested by f at $(q, t) \in A^{\exists} \times S$. We define the strategy f_π according to the following case distinction on basic positions $(q, t) \in A^{\exists} \times S$:

(1) If $q \in A$ and $(q, t) \in Win_{\exists}(\mathcal{G})$, we let f_π suggest the valuation $V'_{q,t} : A^{\exists} \to \wp R(t)$ defined as $V_{q,t}[B \mapsto \varnothing | B \in \wp A]$ at position (q, t).

(2) If $q \in \wp A$ and $(q, t) \in Win_{\exists}(\mathcal{G})$, the valuation $V_{q,t}$ is a legitimate move for $\exists$ hence also $(R(t), V_{q,t}) \models \Theta^{\exists}(q, \kappa(t))$. Recall that the formula $\Theta^{\exists}(q, \kappa(t))$ is defined to be the disjunction $\widehat{\Theta}(q, \kappa(t)) \vee \Theta(q, \kappa(t) \cup \{p\})$. We distinguish two additional cases on the basis of whether or not $(R(t), V_{q,t}) \models \Theta(q, \kappa(t) \cup \{p\})$:

   (a) If $(R(t), V_{q,t}) \models \Theta(q, \kappa(t) \cup \{p\})$, we let f_π suggest the valuation $V'_{q,t}$ defined in (1).

   (b) If $(R(t), V_{q,t}) \not\models \Theta(q, \kappa(t) \cup \{p\})$, then $(R(t), V_{q,t}) \models \widehat{\Theta}(q, c)$ since $(R(t), V_{q,t}) \models \Theta^{\exists}(q, \kappa(t))$. Thus $(R(t), V_{B,t}) \models \widehat{\varphi}$ for some disjunct $\widehat{\varphi}$ of $\widehat{\Theta}(q, \kappa(t))$. We let f_π suggest the $\wp A$-separating valuation $V^s_{q,t}$ guaranteed by Proposition 4.2.7.

(3) In all other cases, $\exists$ plays randomly.

Note that the valuations suggested by f_π in clause (1) and clause (2)(a) are legitimate because the formulas described in those clauses are $\wp A$-free; the valuation $V^s_{q,t}$ is legitimate at each position (q, t) as described in (2)(b) by Proposition 4.2.7(i). To see that f_π is in fact a winning strategy for $\exists$, simply note that $f_\pi(q, t) \subseteq f(q, t)$ for each winning position $(q, t) \in A^{\exists} \times S$. In particular, this implies that every f_π-guided match in the game $\mathcal{G}@(B, s)$ is also a f-guided match hence also each f_π-guided match is winning for $\exists$ because (B, s) is a winning position and f is a winning strategy.

We now show that f_π has the properties (i) and (ii). As (i) is built directly into (1) in the definition of f_π, we immediately proceed to show that f_π enjoys (ii). To this observe that at each winning position $(Q, t) \in \wp A \times S$, her winning strategy f suggests only legitimate moves so that either $(R(t), V_{Q,t}) \models \Theta(Q, \kappa(t) \cup \{p\})$ or $(R(t), V_{Q,t}) \models \widehat{\Theta}(Q, \kappa(t))$. As f is minimal, the former implies that the valuation $V_{Q,t}$ is such that $V_{Q,t}(Q') = \varnothing$ for each $Q' \in \wp A$ because $\Theta(B, \kappa(t) \cup \{p\})$ is $\wp A$-free. If the latter holds, note that by Proposition 4.2.7 we have that $V_{Q,t}(Q^+)$ is a singleton for some $Q^+ \in \wp A$ and $V_{Q,t}(Q') = \varnothing$ for each $Q' \in \wp A$ such that $Q' \neq Q^+$. In short, for each winning position $(Q,t) \in \wp A \times S$, there is at most one pair $(Q^+, t^+) \in \wp A \times R(t)$ such that $(Q^+, t^+) \in V_{Q,t}$.

From this, we may inductively define a sequence $\rho := (B_0, s_0), (B_1, s_1), \ldots$ of basic positions by putting $(B_0, s_0) = (B, s), (B_i, s_i)$ and $(B_{i+1}, s_{i+1}) := (B_i^+, s_i^+)$ if $(B_i^+, s_i^+) \in V_{B_i, s_i}$ for some $B_i^+ \in \wp A$ and undefined otherwise. It follows immediately from this definition that if (B_{i+1}, s_{i+1}) is defined, then $s_{i+1} \in V_{B_i,s_i}(B_{i+1})$.

Consequently, ρ must be finite because otherwise ρ is an infinite f_π-guided match of in the game $\mathcal{G}@(B,s)$ in which every basic position has odd parity; a contradiction with our assumption that $(B,s) \in \mathit{Win}_\exists(\mathcal{G})$. As ρ is finite, it follows by construction that $\mathit{last}(\rho)$ is a basic position satisfying (1). That is, $V_{\mathit{last}(\rho)}(Q) = \varnothing$ for every $Q \in \wp A$.

It is now straightforward to see that the sequence $\pi := s_0, s_1, \ldots$ is a finite sequence satisfying (ii). To see that π is unique, observe that (2) expresses that for each $i \in \omega$ the position (B_{i+1}, s_{i+1}) is uniquely determined by (B_i, s_i) hence also ρ is unique given (B_0, s_0).

□

Corollary 4.2.13. *Let $(B,s) \in \mathit{Win}_\exists(\mathcal{A}(\exists p.\mathbb{A}, \mathbb{T}))$ and let f_π be the strategy given in Lemma 4.2.12. Then for every basic position $(Q,t) \in \wp A \times S$ is f_π-reachable from (B,s) iff $(Q,t) = (B_i, s_i)$ for some i. In particular, f_π is functional in $\wp A$.*

Proof. Follows immediately from the uniqueness of the sequence ρ. □

Proof of the main result

We are now well on our way to proving Proposition 4.2.3. Before we prove the proposition, we will first prove a sequence of lemmas stating how to transform legitimate moves for the atomic projection construction into legitimate moves for the original automaton and vice versa.

Lemma 4.2.14. *Let $\mathbb{A} \in \mathit{Aut}^l_{wa}(\mathrm{FOE}_1)$ be on alphabet $\wp(\mathrm{P} \uplus \{p\})$, let $\exists p.\mathbb{A}$ be its atomic projection, let $B \in \wp A$ be a macro-state, and let $c \in \wp P$ be a colour. Assume that $(D, V_{B,c}) \models \widehat{\Theta}(B,c)$ for some set D and some valuation $V_{B,c} : A^\exists \to \wp D$. Then there is a valuation $V^\Downarrow_{B,c} : A \to \wp D$ such that*

(i) $(D, V^\Downarrow_{B,c}) \models \Theta(b,c)$ for each $b \in B$

(ii) for each node $d \in D$ and each state $a \in A$, if $d \in V^\Downarrow_{B,c}(a)$ then either $d \in V_{B,c}(a)$ or $d \in V_{B,c}(Q)$ for some $Q \in \wp A$ such that $a \in Q$.

Proof. Suppose that $(D, V_{B,c}) \models \widehat{\Theta}(B, c)$ for some valuation $V_{B,c} : \wp A \uplus A \to \wp D$. We define $V^{\Downarrow}_{B,c} : A \to \wp D$ by setting

$$V^{\Downarrow}_{B,c}(a) := V_{B,c}(a) \cup \bigcup_{a \in Q} V_{B,c}(Q)$$

for each $a \in A$. Note that item (ii) follows immediately from the definition of $V^{\Downarrow}_{B,c}$; we proceed to show that (i) obtains as well. Our goal is to show that $(D, V^*) \models \Theta(b, c)$ for every $b \in B$. To this end, recall that the formula $\widehat{\Theta}(B, c)$ is defined to be the additive lifting of the formula $\psi_{B,c}$ and $\psi_{B,c}$ is a disjunction $\bigvee_i \widehat{\alpha_i}$ of sentences in positive basic form which is equivalent to $\bigwedge_{b \in B} \Theta(b, c)$. Due to the equivalence of $\bigwedge_{b \in B} \Theta(b, c)$ and $\bigvee_i \alpha_i$, it suffices to show that $(D, V^*) \models \alpha_i$ for some disjunct α_i of $\psi_{B,c}$. Let $\widehat{\alpha_i}$ be a disjunct of $\widehat{\Theta}(B, c)$ such that $(D, V_{B,c}) \models \widehat{\varphi}$. We proceed to show that $(D, V^{\Downarrow}_{B,c}) \models \alpha$. To this end, we distinguish cases on the basis of whether or not

$$(D, V_{B,c}) \models \exists x_1 \ldots x_k (\text{diff}(\overline{\mathbf{x}}) \wedge \pi(\overline{\mathbf{x}}, \Pi) \wedge \bigvee_{i \leq k} (T_i(x_i) \wedge \bigwedge_{j \neq i} \tau^+_{T_j}(x_j))) \qquad (*)$$

holds. If $(*)$ holds, there exist distinct nodes $d_1, \ldots, d_k \in D$ with the following properties.

(1) For some $i \leq k$ we have $d_i \in V_{B,c}(T_i)$.

(2) For each $j \leq k$ such that $j \neq i$, we have $d_j \in V_{B,c}(a)$ for each $a \in T_j$.

(3) $(D, V_{B,c}) \models \pi(\overline{\mathbf{d}}, \Pi)$.

It follows directly from the observation that, for each $a \in A, V_{B,c}(a) \subseteq V^{\Downarrow}_{B,c}(a)$ that (2) and (3) are preserved by replacing each instance of "$V_{B,c}$" by "$V^{\Downarrow}_{B,c}$". In order to complete the proof, it suffices to show that for $d_i \in V^{\Downarrow}_{B,c}(a)$ for each $a \in T_i$. This is also immediate by the definition of $V^{\Downarrow}_{B,c}$: for each $a \in T_i$, observe that $d_i \in V_{B,c}(T_i) \subseteq V^{\Downarrow}_{B,c}(a)$. Hence $(D, V^{\Downarrow}_{B,c}) \models \varphi$ if $(*)$ holds. Now suppose that $(*)$ is not satisfied. Then, as $(D, V_{B,c}) \models \widehat{\varphi}$,

$$(D, V_{B,c}) \models \exists \bar{x} x_{k+1} (\text{diff}(\bar{x}, x_{k+1}) \wedge \pi(\bar{x}, x_{k+1}, \Pi) \wedge \bigwedge_{i \leq k} \tau^+_{T_i}(x_i) \wedge \bigvee_{B \in \Pi} B(x_{k+1})) \qquad (**)$$

Unwinding the meaning of $(**)$, we have that there exists distinct nodes $d_1, \ldots, d_{k+1} \in D$ with the following properties.

(1') For each $i \leq k$, we have that $d_i \in V_{B,c}(a)$ for each $a \in T_i$.

(2') The node d_{k+1} is such that $d_{k+1} \in V_{B,c}(B^*)$ for some set $B^* \in \Pi$.

(3') $(D, V_{B,c}) \models \pi(\overline{\mathbf{d}}, \Pi)$.

Just as before, it follows immediately from the definition of $V^{\Downarrow}_{B,c}$ that both (2′) and (3′) are preserved by $V^{\Downarrow}_{B,c}$. This time, we want to show that $d_{k+1} \in V^{\Downarrow}_{B,c}(a)$ for each $a \in B^*$. This follows from the same reasoning as in the previous case. That is, $(D, V^{\Downarrow}_{B,c}) \models \varphi$, as desired. □

Lemma 4.2.15. *Let $\mathbb{A} \in Aut^{l}_{wa}(\mathrm{FOE}_1)$ be on alphabet $\wp\mathrm{P}$ and let $B \subseteq A$ be a set of states. Fix a colour $c \in \wp P$ and assume that we are given a family $\{V_{b,c} : A \to \wp D \mid b \in B\}$ of valuations such that $(D, V_{b,c}) \models \Theta(b, c)$ for each $b \in B$. Then for each node $s \in D$ there is a valuation $V^{s}_{B,c} : A^{\exists} \to \wp D$ such that*

(i) $(D, V_{B,c}) \models \widehat{\Theta}(B, c)$

(ii) *if $s \in V^{s}_{B,c}(q)$, then $q \in \wp A$;*

(iii) *for each $d \in D$, if $d \in V^{s}_{B,c}(a)$ for some $a \in A$, then $d \in V_{b,c}(a)$ for some $b \in B$*

(iv) *for each $d \in D$, if $d \in V^{s}_{B,c}(Q)$ for some $Q \in \wp A$, then for every $a \in Q$ there exists $b \in B$ such that $d \in V_{b,c}(a)$.*

Proof. We begin by defining a valuation $U : A \to \wp D$ which collects the information contained in the family $\{V_{b,c} |\ b \in B\}$ of valuations into a single map defined by setting

$$U(a) = \bigcup_{b \in B} V_{b,c}(a).$$

for each $a \in A$. Our goal is to define, for each $s \in D$, a valuation $V^{s}_{B,c} : A^{\exists} \to \wp D$ such that $(D, V^{s}_{B,c}) \models \widehat{\Theta}(B, c)$. To this end, recall that the formula $\psi_{B,c}$ is defined to be a disjunction $\bigvee \alpha_i$ of sentences in positive basic form which is equivalent to $\bigwedge_{b\in B} \Theta(b, c)$ and $\widehat{\Theta}(B, c)$ is the disjunction $\bigvee_i \widehat{\varphi}_i$ of their additive liftings. Thus, in order to satisfy $\widehat{\Theta}(B, c)$, it suffices to satisfy one of its disjuncts.

Now, as $(D, V_{b,c}) \models \Theta(b, c)$ for each $b \in B$ and $V_{b,c} \subseteq U$ for each $b \in B$, it follows that $(D, U) \models \bigwedge_{b\in B} \Theta(b, c)$ by monotonicity of the formulas $\Theta(b, c)$. Hence also $(D, U) \models \psi_{B,c}$. Let α be a disjunct of $\psi_{B,c}$ such that $(D, U) \models \alpha$. We will tailor the valuation $V_{B,c}$ to make the additive lifting $\widehat{\alpha}$ true in the set D. We define $V^{d}_{B,c} := U^{\Uparrow}_{d}$ where $U^{\Uparrow}_{d}$ is the valuation given by Proposition 4.2.8 for the valuation U and node d.

Indeed, note that (i) and (ii) follow directly from Proposition 4.2.8(i) and (ii), respectively. To see that (iii) holds, simply note that if $d \in U^{\Uparrow}_{s}(a)$ for some $a \in A$, then $d \in U(a)$ by Proposition 4.2.8(ii). Hence, by definition of U, we have that $d \in V_{b,c}$ for some $b \in B$. Finally, to see that (iv) holds, assume $d \in D$ is such that $d \in U^{\Uparrow}_{s}(Q)$ for some $Q \in \wp A$. This can be so only if $d = s$ by Proposition 4.2.8(ii). In fact, we also have that $Q \subseteq m_U(d)$ by the same proposition. In other words, we have that $d \in U(a)$ for each $a \in Q$. By definition of U, it follows immediately that for each $a \in Q$ we have $d \in V_{b,c}(a)$ for some $b \in B$, as desired. □

We have now gathered all of the ingredients that we will need in order to prove Proposition 4.2.3.

Proof of Proposition 4.2.3. Our goal is to show that for each automaton $\mathbb{A} \in Aut^l_{wa}(\text{FOE}_1)$, the following are equivalent for every tree $\mathbb{T}$:

(i) $\mathbb{T} \models \exists p.\mathbb{A}$

(ii) $\mathbb{T}[p \mapsto \{t\}] \models \mathbb{A}$ for some $t \in S$.

We shall begin by proving the implication from (i) to (ii): let $\mathbb{A} \in Aut^l_{wa}(\text{FOE}_1)$ be an automaton on alphabet $\wp(\mathrm{P} \uplus \{p\})$ and let $\mathbb{T}$ be a P-tree such that $\mathbb{T} \models \exists p.\mathbb{A}$. We begin by defining an atomic p-variant $\mathbb{T}_p$ of $\mathbb{T}$.

To this end, let $f_\exists$ be a positional winning strategy for $\exists$ in the game $\mathcal{G}_\exists := \mathcal{A}(\exists p.\mathbb{A}, \mathbb{T})$. As $(a^\exists_I, s_I) \in Win_\exists(\mathcal{G}_\exists)$, we may transform $f_\exists$ into a positional winning strategy f_ρ for $\exists$ as in Lemma 4.2.12; for the position $(a^\exists_I, s_I)$, we have that there is a unique finite path $\rho = (s_I{=}s_0, \ldots, s_k{=}s^*)$ satisfying the properties laid out in that lemma. With this in mind, we define

$$\mathbb{T}_p := \mathbb{T}[p \mapsto \{s^*\}]$$

and we write $\kappa' : S \to \wp(\mathrm{P}\uplus\{p\})$ to denote its labelling function. Note that $\kappa'(t) = \kappa(t)$ if $t \neq s^*$ and $\kappa'(s^*) = \kappa(s^*) \uplus \{p\}$. Clearly $\mathbb{T}_p$ is an atomic p-variant of $\mathbb{T}$. We will now show that $\mathbb{A}$ accepts $\mathbb{T}_p$. First, recall that by Corollary 4.2.13, for each node s_i on the path ρ, there is a unique macro-state B_i such that (B_i, s_i) is f_ρ-reachable. Moreover, for every node t such that $t \neq s_i$ for every i, there is no such macro-state.

We define a strategy f for $\exists$ in the game $\mathcal{G}$ according to the following case distinction on basic positions $(a, s) \in A \times S$.

(1) If $(a, s) \in Win_\exists(\mathcal{G}_\exists)$, then we make a further case distinction:

  (a) if s such that $s \neq s_i$ for every $i \leq k$, then we let f suggest the same valuation $V_{a,s}$ as her winning strategy f_ρ.

  (b) if $s = s_k$, let $V_{Q_s,s}$ be the valuation suggested at the unique position $(Q_s, s) \in \wp A \times S$ (guaranteed by Corollary 4.2.13) which is f_ρ-reachable from $(a^\exists_I, s_I)$ in the game $\mathcal{G}_\exists$. We let f suggest the valuation $V_{Q_s,s}$ at position (a, s).

  (c) if $s = s_i$ for some $i < k$, let V_{B_i,s_i} be the valuation suggested by f_ρ at position (B_i, s_i). We let f suggest the valuation $V^{\Downarrow}_{B_i,s_i}$ at (a, s).

(2) At all other positions, $\exists$ plays randomly.

We will show that each of the moves suggested by f are legitimate in the game $\mathcal{G}@(a_I, s_I)$ in stages, while playing an f-guided match π in the game $\mathcal{G}@(a_I, s_I)$. Along with the match π, we will maintain an f_ρ-guided shadow match $\pi_\exists$, carrying the following inductive hypothesis from one round to the next.

(†) Where the current position of the match π is $(a, s) \in A \times S$, one of the following obtains:

(†.1) if $s = s_i$ for some $i \leq k$, then the current position in the match $\pi_\exists$ of the game $\mathcal{G}_\exists$ has the shape (B, s) for some $B \in \wp A$ such that $a \in B$.

(†.2) if $s \neq s_i$ for every $i \leq k$, then the current position in the match $\pi_\exists$ is the same position (a, s) as the current position in the match π.

The matches π and $\pi_\exists$ initialize in position (a_I, s_I) and $(a_I^\exists, s_I)$, respectively. As $a_I^\exists = \{a_I\}$, we have that $a_I \in a_I^\exists$. Furthermore, the node s_I is on the path ρ. Hence (†.1) is maintained in the initial round. Now, inductively assume that we have maintained the relation (†) between the partial f_ρ-guided matches π and $\pi_\exists$ in the games $\mathcal{G}@(a_I, s_I)$ and $\mathcal{G}_\exists@(a_I^\exists, s_I)$, respectively. In order to show that we can carry (†) into the next round, we make a case distinction on the basis of whether (†.1) or (†.2) currently holds.

If (†.2) holds, then the current position in both π and $\pi_\exists$ is the same position $(a, s) \in A \times S$ for some s such that $s \neq s_i$ for every $i \leq k$. Observe that this implies s' is not on the path π for every $s' \in R(s)$ as well. In other words, we aim to show that (†.2) can be carried into the next round. This follows immediately from Proposition 4.2.12(i): we have that $V_{a,s}(Q) = \varnothing$ for every $Q \in \wp A$. Hence, if $\forall$ plays a position $(q, t) \in V_{a,s}$, then $q \in A$ and this move can be matched in the shadow match $\pi_\exists$ because the strategy f agrees with f_ρ on positions of the shape (a, s).

Otherwise, (†.1) is currently satisfied and the current position of the matches π and $\pi_\exists$ are of the form (a, s_i) and (Q, s_i) for some $Q \in \wp A$ such that $a \in Q$ and some $i \leq k$, respectively. As f_ρ is winning for $\exists$ and (Q, s_i) is f_ρ-reachable, it follows that her strategy suggests a legitimate move $V_{Q,s}$ from this position. That is, $(R(s_i), V_{Q,s}) \models \Theta^\exists(Q, \kappa(s_i))$. We distinguish cases on the basis of whether $i = k$ or $i < k$.

(i) If $i < k$, then $p \notin \kappa'(s)$ so that $\kappa(s) = \kappa'(s)$. Moreover, it follows from Lemma 4.2.12 that $(R(t), V_{Q,t}) \models \widehat{\Theta}(Q, \kappa(t))$. Then, by Lemma 4.2.14, the valuation $V^{\Downarrow}_{Q,t} : A \to \wp R(t)$ is such that $(R(t), V^{\Downarrow}_{Q,t}) \models \Theta(a, \kappa(t))$ since $a \in Q$. Hence $V^{\Downarrow}_{Q,t}$ is a legitimate move for $\exists$ at position (a, t); we let g suggest $V^{\Downarrow}_{Q,t}$ at (a, t). This valuation has the property (Lemma 4.2.14) that for any $d \in R(t)$ we have the following for every $b \in A$: $d \in V^{\Downarrow}_{Q,t}(a)$ if and only if (a) $d \in V_{Q,t}(a)$ or (b) $d \in V_{Q,t}(Q')$ for some $Q' \in \wp A$ such that $a \in Q'$. We shall now show that (†) can be carried into the next round. To this end, assume that $\forall$ plays the position (a', s') in the next round. There are two possibilities: either $s' = s_{i+1}$ or s' is not on the path ρ. In either case, it is straightforward to show that any move made by $\forall$ in the match π can be mirrored in the match $\pi_\exists$ using Proposition 4.2.12(ii).

(ii) If $i = k$, then $V_{Q,t}$ is such that $V_{Q,t}(Q') = \varnothing$ for every $Q' \in \wp A$ by Proposition 4.2.12(ii). This implies that $(R(t), V_{Q,t}) \models \Theta(Q, \kappa(t) \cup \{p\})$. As $a \in Q$, this implies that $(R(t), V_{Q,t}) \models \Theta(a, \kappa(t) \cup \{p\})$; since $t = s_k$, we have $\kappa'(t) = \kappa(t) \cup \{p\}$. Hence the valuation $V_{Q,t}$ is a legitimate move for $\exists$ in π. We let g suggest the valuation $V_{Q,t}$ in the match π as well. It is easy to see that any move made by $\forall$ in π can be mirrored by $\exists$ in the shadow match $\pi_\exists$ in a manner that maintains (†).

With this special relationship between f and f_ρ now given, we proceed to show that f is in fact a winning strategy for $\exists$ in $\mathcal{G}$. To this end, suppose that π is a full f-guided match of the game $\mathcal{G}$. We have just shown that there exists an f_ρ-guided (shadow) match $\pi_\exists$ in the game $\mathcal{G}_\exists$ satisfying (†). The key observation is that a position of the type $\wp A \times S$ may occur in at most the first k rounds of the shadow match $\pi_\exists$. Indeed, this follows directly from Corollary **??**. Then, in every round k' such that $k \leq k'$, it follows that the current position in round k' in the match π is identical to the current position in round k' in $\pi_\exists$. As $\Omega^\exists(a) = \Omega(a)$ for every state $a \in A$ and the match $\pi_\exists$ is guided by the winning strategy f_π, it follows that π is a winning match for $\exists$. Hence f is a winning strategy for $\exists$ in the game $\mathcal{G}@(a_I, s_I)$, as desired.

We will now show the implication from (ii) to (i). To this end, let $\mathbb{T}_p := \mathbb{T}[p \mapsto \{t\}]$ be an atomic p-variant of $\mathbb{T}$ such that $\mathbb{T}_p \models \mathbb{A}$. Let ρ denote the unique finite path from the root s_I to the node t and, just as before, we denote the labelling functions of the trees $\mathbb{T}$ and $\mathbb{T}_p$ by κ and κ', respectively. Let f be a positional winning strategy for $\exists$ in the game $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{T}_p)$. We now provide $\exists$ with a winning strategy $f_\exists$ in the game $\mathcal{G}_\exists := \mathcal{A}(\exists p.\mathbb{A}, \mathbb{T})$ (initiated at $(a_I^\exists, s_I)$) according to the following case distinction on basic positions $(q, s) \in A^\exists \times S$:

(i) If $(q, s) = (a, s) \in A \times S$ is a winning position for $\exists$ in $\mathcal{G}$ and $\mathbb{T}_p.s$ is p-free, we let $f_\exists$ suggest the same valuation $V_{a,s}$ that her winning strategy f suggests at (a, s).

(ii) If $(q, s) = (Q, s) \in \wp A \times S$ is such that s is on the path ρ and $(a, s) \in \mathit{Win}_\exists(\mathcal{G})$ for every $a \in Q$, we further distinguish the following two cases:

 (a) If $s < t$, then there is a unique node $s^+ \in R(s)$ such that s^+ is on the path ρ. In this case, we let $f_\exists$ suggest the valuation $V_{Q,s}$ given by Proposition 4.2.15 for the node s^+ and the set $\{V_{a,s} | a \in Q\}$ containing, for each $a \in Q$, the valuation $V_{a,s}$ suggested by f at (a, s).

 (b) If $s = t$, we let $f_\exists$ suggest the valuation $V_{Q,t} := \bigcup_{a \in Q} V_{a,s}$.

(iii) At all other positions, $\exists$ plays randomly.

On the basis of Proposition 4.2.15 and the assumptions made in our case distinction, it is straightforward to see that each of the moves suggested by $f_\exists$ in (ii) are legitimate.

We will now show that for every $f_\exists$-guided partial match $\pi_\exists$ there is a 'special' set $\mathfrak{B}$ of f-guided partial matches in $\mathcal{G}$ by induction on the number of rounds. The inductive hypothesis that we wish to carry from round to round is the following statement (†) :

> If $(q, s) \in A^\exists \times S$ is the current position in $\pi_\exists$, one of the following obtains:
>
> $\dagger_1$ $q = a \in A$ and the current bundle consists of a single f-guided partial match π whose current position is also (a, s). Moreover, the subtree $\mathbb{T}_p.s$ is p-free.
>
> $\dagger_2$ $q = Q \in \wp A$ and, for every $a \in Q$, there is an f-guided partial match π_a such that the current position in π_a is (a, s). Moreover, we have $s \leq t$.

As the matches $\pi_\exists$ and π initialize, respectively, in the positions $(\{a_I\}, s_I)$ and (a_I, s_I), it follows immediately that ($\dagger_2$) is satisfied in the initial round. Now, inductively assume that for every $f_\exists$-guided partial match with at most n rounds we have maintained a bundle of f-guided partial (shadow) matches satisfying (†). In order to show that we can carry (†) into the next round, we distinguish cases on the basis of whether $\dagger_1$ or $\dagger_2$ was met in the current round of $\pi_\exists$.

If the current position in $\pi_\exists$ has the shape $(a, s) \in A \times S$, then the current bundle consists of a unique f-guided match π whose current position is also (a, s) and the subtree $\mathbb{T}_p.s$ is p-free. As $f_\exists$ suggests the same valuation at (a, s) as the strategy f, it follows that any move made by $\forall$ in the game $\mathcal{G}_\exists$ can be mirrored in the game $\mathcal{G}$. That is, if the next position in the match $\pi_\exists$ is the position $(a', s') \in A \times S$, we take the bundle consisting of the match π extended by the position (a', s') into the next round. Moreover, the subtree $\mathbb{T}.s'$ is p-free because the subtree $\mathbb{T}.s$ is p-free. Hence (†1) is satisfied in this case.

Otherwise, the current position of the match $\pi_\exists$ has the form $(Q, s) \in \wp A \times S$ for some $s \leq t$ and, for each $a \in Q$, the current set $\mathfrak{B}$ contains an f-guided shadow match π_a such that the current position of π_a is (a, s). We distinguish cases on the basis of whether $s < t$ or $s = t$ holds:

(i) If $s = t$, then $f_\exists$ suggests the valuation $V_{Q,s} = \bigcup_{a \in Q} V_{a,t}$. Observe that $V_{Q,s}(Q') = \varnothing$ for each $Q' \in \wp A$ so that each admissible move for $\forall$ at position $V_{Q,s}$ has the type $A \times S$. That is, in order to carry (†) into the next round we must satisfy ($\dagger_1$). To this end, note that if $\forall$ picks the position (a', s') from $V_{Q,s}$, then $s' \in V_{a,s}(a')$ for some $a \in Q$. In this event, we may take the set $\{\pi_a \cdot (a', s')\}$ where $\pi_a \cdot (a', s')$ is the f-guided match π_a extended by the position (a', s'). Also, as $t < s'$ it follows that $\mathbb{T}_p.s'$ is p-free. Hence ($\dagger_1$) is satisfied.

(ii) If $s < t$, then $f_\exists$ suggests the valuation $V_{Q,s}$ given by Proposition 4.2.15. In order to maintain (†), we distinguish cases based on the type of $\forall$'s next move.

(a) If $\forall$ picks $(a', s') \in A \times S$, then this is legitimate so that $s' \in V_{Q,\kappa(s)}(a)$. Then, by Lemma 4.2.15(ii), we have that $s' \in V_{q,\kappa(s)}$ for some $q \in Q$. In this event, we carry the set $\{\pi_a \cdot (a', s')\}$ where $\pi_a \cdot (a', s')$ is the (partial) f-guided match π_a extended by the position (a', s'). By Lemma 4.2.15(iii), we have that $s' \neq s^+$ because $a \notin \wp A$ and $s' \in V_{Q,s}(a)$. In other words, the node s' is not contained in the subtree $\mathbb{T}_p.t$ whence $\mathbb{T}_p.s'$ is p-free. In short, $(\dagger_1)$ is preserved.

(b) Now suppose that $\forall$ picks $(Q', s') \in \wp A \times S$. Then, in order to preserve $(\dagger)$, we will provide an f-guided match $\pi_{a'}$ such that the current position in $\pi_{a'}$ is (a', s') for each $a' \in Q'$. As (Q', s') is an admissible move for $\forall$, we have that $s' \in V_{Q,s}(Q')$. As $Q' \in \wp A$, for each $b \in Q'$ there exists $a \in Q$ such that $s' \in V_{a,s}(b)$ by Lemma 4.2.15(iv). That is, for each $b \in Q$, there is a state $a_b \in Q$ such that (b, s') is an admissible move for $\forall$ in the match π_{a_b} contained in the current set of shadow matches. With this on hand, we define the next set of matches to be $\{\pi_{a_b} \cdot (b, s') \mid b \in Q'\}$. Then $(\dagger_2)$ is preserved.

Thus for every partial $f_\exists$-guided match $\pi_\exists$ there is a set of matches as in $(\dagger)$. We will now use this to show that $f_\exists$ is in fact a winning strategy for $\exists$ in $\mathcal{G}_\exists$. To this end, le $\pi_\exists$ be a full $f_\exists$-guided match in the game $\mathcal{G}_\exists$. As $V_{\kappa'}(p)$ is a singleton, it follows that a position as described in $(\dagger_1)$ occurs after finitely many rounds in the match $\pi_\exists$. Furthermore, every position thereafter also is described in $(\dagger_1)$. But this just means that there is an infinite final segment of $\pi_\exists$ which follows her winning strategy f whence $\pi_\exists$ is winning for $\exists$, as desired. $\square$

From Formulas to Automata

In this section, we give an effective transformation of formulas from AMSO to automata from $Aut^l_{wa}(\mathrm{FOE}_1)$.

Proposition 4.2.16. *For each formula φ of* AMSO *we can effectively construct an equivalent automaton $\mathbb{A}_\varphi \in Aut^l_{w.a.}(\mathrm{FOE}_1)$.*

We will prove Proposition 4.2.16 by induction on $\varphi \in$ AMSO. In order to simulate first-order quantification by means of automata, we will use the atomic projection construction introduced in Definition 4.2.10. The Boolean cases in which φ is a disjunction or a negated formula correspond, respectively, to the closure of the class of tree languages recognized by automata from $Aut^l_{w.a.}(\mathrm{FOE}_1)$ under union and complementation; we address these matters now.

Closure under Boolean operations

We will first show that the class of tree languages accepted by automata from $Aut^l_{w.a.}(\mathrm{FOE}_1)$ is closed under union. Recall that $\mathrm{TMod}(\mathbb{A})$ is the tree language of $\mathbb{A}$ and it consists of those trees $\mathbb{T}$ such that $\mathbb{A}$ accepts $\mathbb{T}$.

Proposition 4.2.17. *For each pair of automata* $\mathbb{A}_0, \mathbb{A}_1 \in Aut^l_{w.a.}(\mathrm{FOE}_1)$, *there is an automaton* $\mathbb{A} \in Aut^l_{w.a.}(\mathrm{FOE}_1)$ *such that* $\mathrm{TMod}(\mathbb{A}) = \mathrm{TMod}(\mathbb{A}_0) \cup \mathrm{TMod}(\mathbb{A}_1)$.

Proof. Let $\mathbb{A}_0 = (A_0, \Theta_0, \Omega_0, a_{i,0}), \mathbb{A}_1 = (A_1, \Theta_1, \Omega_1, a_{i,1}) \in Aut^l_{w.a.}(\mathrm{FOE}_1)$ on the alphabets $\wp\mathrm{P}_0$ and $\wp\mathrm{P}_1$, respectively. We define $\mathbb{A}$ to be the automaton (A, Θ, Ω, a_i) where $A = A_0 \uplus A_1 \uplus \{a_i\}$ and the functions $\Omega : A \to \mathbb{N}$ and $\Theta : A \times \wp(\mathrm{P}_0 \cup \mathrm{P}_1) \to \mathrm{FOE}_1(A)$ are defined as follows, for every $a \in A$ and every colour $c \in \wp(\mathrm{P}_0 \cup \mathrm{P}_1)$:

$$\Omega(a) := \begin{cases} 1 & \text{if } a = a_i \\ \Omega_0(a) & \text{if } a \in A_0 \\ \Omega_1(a) & \text{if } a \in A_1 \end{cases} \quad \Theta(a, c) := \begin{cases} \Theta_0(a_{i,0}, c \cap \mathrm{P}_0) \vee \Theta_1(a_{i,1}, c \cap \mathrm{P}_1) & \text{if } a = a_i \\ \Theta_0(a, c \cap \mathrm{P}_0) & \text{if } a \in A_0 \\ \Theta_1(a, c \cap \mathrm{P}_1) & \text{if } a \in A_1. \end{cases}$$

It is straightforward to check that for every tree $\mathbb{T}$ we have $\mathbb{A}$ accepts $\mathbb{T}$ if and only if $\mathbb{A}_0$ accepts $\mathbb{T}$ or $\mathbb{A}_1$ accepts $\mathbb{T}$. To conclude the proof, we will show that $\mathbb{A}$ sits in the class $Aut^l_{w.a.}(\mathrm{FOE}_1)$. To this end, first note that every cluster C of $\mathbb{A}$ is such that either $C = \{a_i\}$ or C is a cluster of $\mathbb{A}_i$ for some $j \in \{0, 1\}$. On the basis of this observation, we have that $\mathbb{A}$ satisfies the weakness condition because $C = \{a_i\}$ is a singleton and if C is a cluster of $\mathbb{A}_j$ for some $j \in \{0, 1\}$, then for each pair $a, b \in C$ we have $\Omega(a) = \Omega_j(a) = \Omega_j(b) = \Omega(b)$ because $\mathbb{A}_j$ is weak.

It remains to be seen that $\mathbb{A}$ is a path automaton. To this end, observe that if C is a μ-cluster of $\mathbb{A}_j$, then for each $a \in C$ and each colour $c \in \wp(\mathrm{P}_0 \cup \mathrm{P}_1)$, we have $\Theta(a, c) = \Theta_j(a, c \cap \mathrm{P}_j)$. Hence $\Theta(a, c)$ is completely additive in C for every colour c because $\mathbb{A}_j$ is a path automaton; the case of ν-clusters follows similarly. Also, note that $\Theta(a_i, c)$ is $\{a_i\}$-free whence $\Theta(a_i, c)$ is completely additive in a_i. In short, $\mathbb{A}$ is a path automaton, as desired. □

We will now show that the class of tree languages recognized by automata from $Aut^l_{w.a.}(\mathrm{FOE}_1)$ is closed under complementation. To this end, we will show that the general results of Section 2.6 restrict to the class $Aut^l_{w.a.}(\mathrm{FOE}_1)$.

Proposition 4.2.18. *For each automaton* $\mathbb{A} \in Aut^l_{w.a.}(\mathrm{FOE}_1)$, *there is an automaton* $\overline{\mathbb{A}} \in Aut^l_{w.a.}(\mathrm{FOE}_1)$ *such that for each tree* $\mathbb{T}$ *we have* $\overline{\mathbb{A}}$ *accepts* $\mathbb{T}$ *iff* $\mathbb{A}$ *rejects* $\mathbb{T}$.

Proof. By Proposition 2.6.13, it suffices to show that the automaton $\mathbb{A}^\delta$ is a weak-path automaton whenever $\mathbb{A}$ is.[1] To this end, recall that the dualization operator $(\cdot)^\delta$ (on formulas) acts as the identity on atomic formulas of the form $a(x)$. Hence, for each $a \in A$ and each $\varphi \in \mathrm{FOE}_1(A)$, we have that a occurs in φ if and only if a occurs in φ^δ. In other words, for each pair of states $a, b \in A$ and each colour $c \in \wp\mathrm{P}$, we have that b occurs in $\Theta(a, c)$ iff b occurs in $\Theta(a, c)^\delta$. Thus C is a cluster of $\mathbb{A}$ iff C is a cluster of $\mathbb{A}^\delta$. But then it is straightforward to see that $\mathbb{A}^\delta$ is a weak-path automaton due to the dual nature

[1] Recall that for each $a \in A$ and each colour $c \in C$, we defined $\Theta^\delta(a, c) = \Theta(a, c)^\delta$ and $\Omega^\delta(a) = \Omega(a)+1$

of the notions of complete additivity and complete multiplicativity, together with the parity 'shift' induced by the priority map Ω^δ. □

Proof of Main Result

Proof of Proposition 4.2.16. The proof is by induction on the structure of $\varphi \in$ AMSO. For the base case, we provide automata corresponding to the atomic formulas $\Downarrow p, p \sqsubseteq q$, and $p < q$:

- We define $\mathbb{A}_{\Downarrow p}$ to be the automaton $(\{a_i, a\}, \Theta, \Omega, a_I)$ where $\Omega(a_i) = 0 = \Omega(a)$ and $\Theta : \{a\} \times C \to \mathrm{FOE}_1(A)$ is defined by setting
$$\Theta(a_i, c) := \begin{cases} \forall x.a(x) & \text{if } p \in c \\ \bot & \text{if } p \notin c \end{cases} \quad \Theta(a, c) := \begin{cases} \forall x.a(x) & \text{if } p \notin c \\ \bot & \text{if } p \in c. \end{cases}$$
- We define $\mathbb{A}_{p \sqsubseteq q}$ to be the automaton $(\{a\}, \Theta, \Omega, a)$ where $\Omega(a) = 0$ and $\Theta : \{a\} \times C \to \mathrm{FOE}_1(A)$ is defined by setting
$$\Theta(a, c) := \begin{cases} \bot & \text{if } \ p \in c \ \text{ and } \ q \notin c \\ \forall x.a(x) & \text{otherwise.} \end{cases}$$
- We define $\mathbb{A}_{p<q} := (\{a_i, a\}, \Theta, \Omega, a_i)$ where $\Omega(a_i) = 0, \Omega(a) = 1$, and the transition map $\Theta : \{a_i, a\} \times C \to \mathrm{FOE}_1(A)$ is defined by setting
$$\Theta(a_i, c) := \begin{cases} \forall x.a_i(x) & \text{if } p \notin c \\ \forall x.(a_i(x) \wedge a(x)) & \text{if } p \in c \end{cases} \quad \Theta(a, c) := \begin{cases} \exists x.a(x) & \text{if } q \notin c \\ \top & \text{if } q \in c. \end{cases}$$

If $\varphi = \psi_0 \vee \psi_1$, there exists automata $\mathbb{A}_{\psi_0}$ and $\mathbb{A}_{\psi_1}$ equivalent to ψ_0 and ψ_1, respectively. Let $\mathbb{A}_{\psi_0 \vee \psi_1}$ be the automaton obtained from Proposition 4.2.17 applied to the automata $\mathbb{A}_{\psi_0}$ and $\mathbb{A}_{\psi_1}$. We have the following chain of equivalences showing that $\mathbb{A}_{\varphi_0 \vee \varphi_1} \equiv \varphi_0 \vee \varphi_1$ (on trees):

$$\begin{aligned} \mathbb{A}_{\varphi_0 \vee \varphi_1} \text{ accepts } \mathbb{T} &\iff \mathbb{A}_{\varphi_0} \text{ accepts } \mathbb{T} \text{ or } \mathbb{A}_{\psi_1} \text{ accepts } \mathbb{T} && \text{(Proposition 4.2.17)} \\ &\iff \mathbb{T} \models \psi_0 \text{ or } \mathbb{T} \models \psi_1 && \text{(Inductive Hypothesis)} \\ &\iff \mathbb{T} \models \psi_0 \vee \psi_1. && \text{(Semantics of AMSO)} \end{aligned}$$

It is straightforward to check that these have the correct shape using the syntactic characterization given in Theorem 3.2.3 and by noting that transitions only occur from a_i to a in the first and last case (weakness is trivial for the third case). The remaining Boolean case in which $\varphi = \neg\psi$ follows similarly by combining the induction hypothesis with Proposition 4.2.18. Finally, if $\varphi = \exists p.sing(p) \wedge \psi$, there exists an automaton $\mathbb{A}_\psi \in Aut^l_{w.a.}(\mathrm{FOE}_1)$ which is equivalent to ψ (on trees) by the induction hypothesis.

By Proposition 4.2.3, there is an automaton $\exists p.\mathbb{A}_\psi$ such that, for every tree $\mathbb{T}$, $\exists p.\mathbb{A}_\Psi$ accepts $\mathbb{T}$ iff $\mathbb{A}_\Psi$ accepts $\mathbb{T}[p \mapsto \{t\}]$ for some node t in $\mathbb{T}$. In short, we have the following chain of equivalences for each tree $\mathbb{T}$:

$$\begin{aligned}
\mathbb{T} \models \exists p.\mathbb{A}_\psi &\iff \mathbb{T}[p \mapsto \{t\}] \models \mathbb{A}_\Psi \text{ for some node } t \text{ in } \mathbb{T} && \text{(Proposition 4.2.3)}\\
&\iff \mathbb{T}[p \mapsto \{t\}] \models \psi \text{ for some node } t \text{ in } \mathbb{T} && \text{(Induction Hypothesis)}\\
&\iff \mathbb{T} \models \exists p.sing(p) \wedge \psi. && \text{(Semantics of AMSO)}
\end{aligned}$$

□

Chapter 5

Expressive completeness modulo bisimilarity

In this chapter, we explore the expressiveness of $Aut^l_{sa}(\text{FOE}_1)$ and $Aut_{sa}(\text{FOE}_1)$ modulo bisimulation. Until now, we have focused on alternating parity automata based on the one-step language FOE_1. However, the class $Aut(\text{FO}_1)$ is quite interesting in itself: while the class of MSO-automata is a nice framework for studying monadic second-order logics, the class $Aut(\text{FO}_1)$ is rather a framework for analyzing fragments of modal fixpoint logics. Indeed, Janin and Walukiewicz [19] gave this a precise sense. They showed the following equivalence:

$$Aut(\text{FO}_1) \equiv \mu\text{ML} \tag{$*$}$$

where μML denotes the modal μ-calculus, an expressive specification language extending that of basic modal logic by least (and greatest) fixpoint operators. This equivalence is effective and holds over arbitrary transition systems. We call automata from $Aut(\text{FO}_1)$ *modal automata*. The equivalence ($*$) was a crucial step in Janin and Walukiewicz celebrated theorem, stating that every bisimulation-invariant formula of monadic second-order logic is effectively equivalent to a formula of the modal μ-calculus (over arbitrary transition systems):

$$\mu\text{ML} \equiv \text{MSO}/\underline{\leftrightarrow}\ .$$

As a consequence, every modal automaton is bisimulation-invariant. One novelty of their proof was that it supplied a systematic way of studying bisimulation-invariance problems at the level of automata. The downside is of course that one must first obtain automata-theoretic characterizations of logical languages; a non-trivial problems. As mentioned, we will study the bisimulation-invariant fragment of (linked) antisymmetric path automata based on FOE_1. In particular, we will prove the following theorem.

Theorem 5.0.1. *We have the following expressive completeness results:* $Aut^l_{sa}(\text{FO}_1) \equiv Aut^l_{sa}(\text{FOE}_1)/\underline{\leftrightarrow}$

5.1 Linked antisymmetric path automata modulo bisimilarity

In this section, we investigate the class $Aut^l_{sa}(\mathrm{FOE}_1)$ modulo bisimilarity. In particular, we will prove the following equivalence:

$$Aut^l_{sa}(\mathrm{FO}_1) \equiv Aut^l_{sa}(\mathrm{FOE}_1)/\underline{\leftrightarrow}\ . \tag{$*$}$$

Moreover, we show that this equivalence is effective: we give translations in both directions. The inclusion from left to right in $(*)$ is in fact quite straightforward: by the Janin-Walukiewicz Theorem, every automaton $\mathbb{A} \in Aut^l_{sa}(\mathrm{FO}_1)$ is bisimulation-invariant and $Aut^l_{sa}(\mathrm{FO}_1) \subseteq Aut^l_{sa}(\mathrm{FOE}_1)$. In particular, we may take the translation from left to right to be the identity map. We just proved the following proposition.

Proposition 5.1.1. *There is an effective translation* $(\cdot)' : Aut^l_{sa}(\mathrm{FO}_1) \to Aut^l_{sa}(\mathrm{FOE}_1)$.

The interesting direction of this equivalence lies in showing that every bisimulation-invariant automaton from $Aut^l_{sa}(\mathrm{FOE}_1)$ is equivalent to an automaton from $Aut^l_{sa}(\mathrm{FO}_1)$. That is, the inclusion from right to left in $(*)$. An important step in the proof of Janin and Walukiewicz [19] is to define a construction $(-)^{\blacktriangledown} : Aut(\mathrm{FOE}_1) \to Aut(\mathrm{FO}_1)$ such that, for every automaton $\mathbb{A}$, the equivalence

$$\mathbb{A}^{\blacktriangledown} \text{ accepts } \mathbb{S} \text{ iff } \mathbb{A} \text{ accepts } \mathbb{S}^{\omega}.$$

holds for every transition system $\mathbb{S}$ and its ω-expansion $\mathbb{S}^{\omega}$. From this, $(*)$ follows more or less immediately due to the fact that every transition system is bisimulation to its ω-expansion $\mathbb{S}^{\omega}$. An interesting observation made by Venema [9] is that the construction $(\cdot)^{\blacktriangledown}$ on automata is completely determined by a translation $(\cdot)^{\blacktriangledown} : \mathrm{FOE}_1 \to \mathrm{FO}_1$ at the one-step level.

To be precise, we will need the following analogues of the normal form theorems (cf. Theorem 2.5.14) for the one-step languages FOE_1 and FO_1, respectively. Recall that for a set A of monadic predicates and a subset $A' \subseteq A$, we write $\mathrm{ADD}_{A'}\mathrm{FOE}^+(A)$ to denote the set of monadic first-order formulas that are completely additive in A'.

Proposition 5.1.2. *Let A be set of monadic predicates and let $A' \subseteq A$. Then, for each $\mathcal{L} \in \{\mathrm{FOE}^+, \mathrm{FO}^+\}$, we have the following: for every formula $\alpha \in \mathrm{ADD}_{A'}\mathcal{L}^+(A)$ we can effectively obtain an equivalent formula α^c in the positive basic form $\bigvee_i \nabla^+_{\mathcal{L}}(\overline{\mathbf{T}}, \Pi)$ where $\overline{\mathbf{T}} \in (\wp A)^k$ for some $k \in \omega, \Pi \subseteq \overline{\mathbf{T}}$, and the following holds for each disjunct: Π is A'-free and there is at most one element of A' contained in the concatenation of the lists $T_1 \cdots T_k$.*

Proof. See [[6], Corollary 5.1.48(i), Corollary 5.1.54(i)] for details. □

Observe that whenever $\mathcal{L} = \mathrm{FO}^+$, we view $\overline{\mathbf{T}}$ as a set rather than as a list. We will now define the translation $(\cdot)^\blacktriangledown$ described above.

Definition 5.1.3. Fix a set A of propositional variables. For each sentence $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi) \in \mathrm{FOE}_1(A)$ in positive basic form we define

$$(\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi))^\blacktriangledown := \nabla^+_{\mathrm{FO}}(\overline{\mathbf{T}}, \Pi).$$

We lift this translation to disjunctions of positive basic forms by simply setting

$$(\bigvee_i \alpha_i)^\blacktriangledown := \bigvee_i \alpha_i^\blacktriangledown.$$

By Proposition 2.5.14, we can extend this definition to a translation $(-)^\blacktriangledown : \mathrm{FOE}_1(A) \to \mathrm{FO}_1(A)$. We call $\varphi^\blacktriangledown$ the *associate* of φ.

We now have all of the ingredients that are needed in order to define the construction $(-)^\blacktriangledown$ on MSO-automata.

Definition 5.1.4. Let $\mathbb{A}$ be an MSO-automaton. By Proposition 5.1.2, we may obtain a formula $\psi_{a,c} \equiv \Theta(a, c)$ for each pair $(a, c) \in A \times \wp \mathrm{P}$ which is a disjunction of positive basic forms. We define the automaton $\mathbb{A}^\blacktriangledown := (A, \Theta^\blacktriangledown, \Omega, a_I)$ by setting

$$\Theta^\blacktriangledown(a, c) := \psi^\blacktriangledown_{a,c}$$

for each $(a, c) \in A \times \wp \mathrm{P}$. Note that $\mathbb{A}^\blacktriangledown \in Aut(\mathrm{FO}_1)$ for each automaton $\mathbb{A}$.

We will now prove that for each linked antisymmetric path automaton $\mathbb{A}$, the automaton $\mathbb{A}^\blacktriangledown$ lands in the right fragment. That is, $\mathbb{A}^\blacktriangledown$ is also a linked antisymmetric path automata.

Proposition 5.1.5. *If* $\mathbb{A} \in Aut^l_{sa}(\mathrm{FOE}_1)$, *then* $\mathbb{A}^\blacktriangledown \in Aut^l_{sa}(\mathrm{FO}_1)$.

Proof. Let $\mathbb{A}$ be an automaton from $Aut^l_{sa}(\mathrm{FOE}_1)$. It follows directly from the definition of $\Theta^\blacktriangledown$ that $\mathbb{A}^\blacktriangledown \in Aut(\mathrm{FO}_1)$; we proceed to show that $\mathbb{A}^\blacktriangledown$ is antisymmetric and satisfies the linked path condition. We start by showing that $(-)^\blacktriangledown$ preserves antisymmetry. To this end, we will show that if $b \in A$ occurs in $\Theta^\blacktriangledown(a, c)$, then b occurs in $\Theta(a, c)$ as well. Indeed, if b occurs in $\Theta^\blacktriangledown(a, c)$, then b occurs in some disjunct $(\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi))^\blacktriangledown$ of $\psi^\blacktriangledown_{a,c}$. From this it easily follows that b occurs in $\Theta(a, c)$, as claimed. Thus $\mathbb{A}^\blacktriangledown$ is antisymmetric if $\mathbb{A}$ is.

We will now show that $\mathbb{A}^\blacktriangledown$ satisfies the linked path condition. As the construction $(-)^\blacktriangledown$ preserves the parity of each state $a \in A$, we wish to show that if $\Theta(a, c)$ is completely additive (respectively multiplicative) in a, then $\Theta^\blacktriangledown(a, c)$ is completely additive (respectively multiplicative) in a as well. To this end, note that if $\Theta(a, c)$ is completely additive in a, then $\psi_{a,c}$ has the form described in Theorem 5.1.2 (for $\mathcal{L} = \mathrm{FOE}_1$). From this, it easily follows that $\psi^\blacktriangledown_{a,c}$ has the form described in Theorem 5.1.2. Hence $\psi^\blacktriangledown_{a,c} = \Theta^\blacktriangledown(a, c)$ is completely additive in a, as required. □

In other words, the translation $(-)^{\blacktriangledown} : Aut(\mathrm{FOE}_1) \to Aut(\mathrm{FO})$ restricts to a translation $(-)^{\blacktriangledown} : Aut^l_{sa}(\mathrm{FOE}_1) \to Aut^l_{sa}(\mathrm{FO})$. This translation is also effective because it relies only on the effective translation, given in Proposition 5.1.2 assigning each sentence in $\mathrm{FOE}_1(A)$ to an equivalent disjunction of sentences in positive basic form, and the translation on basic forms given in Definition 5.1.3. The following proposition states a well known model-theoretic relationship–tracing back to the work of Janin and Walukiewicz [19]–between the FOE basic form φ and its associate $\varphi^{\blacktriangledown}$. In essence, it provides the key link between these formulas that is needed in order to associate matches in two related acceptance games arising in the standard proof of the Proposition 5.1.7 stated below.

Proposition 5.1.6. *For each one-step formula $\alpha \in \mathrm{FOE}_1(A)$ in positive basic form and each one-step model (D, V) we have the following:*

(i) If $(D, V) \models \alpha^{\blacktriangledown}$, then there is a valuation $V_\pi : A \to \wp(D \times \omega)$ such that

(a) $(D \times \omega, V_\pi) \models \alpha$, and

(b) $(d, i) \in V_\pi(a)$ implies $d \in V(a)$.

(ii) If $(D \times \omega, U) \models \alpha$, then there is a valuation $U^\pi : A \to \wp D$ such that

(a) $(D, U^\pi) \models \alpha^{\blacktriangledown}$.

(b) $d \in U^\pi(a)$ implies $(d, i) \in U(a)$ for some $i \in \omega$.

Proof. We begin by proving (i). To this end, let (D, V) be a one-step model such that $(D, V) \models \alpha^{\blacktriangledown}$. Define the valuation V_π by setting

$$V_\pi(a) := \{(d, i) \in D \times \omega \mid d \in V(a)\}$$

for each $a \in A$. As (i)(b) follows immediately from this definition, we turn our attention towards (i)(a). By definition, the formula $\alpha^{\blacktriangledown}$ has the shape $\nabla^+_{\mathrm{FO}}(\overline{\mathbf{T}}, \Pi)$. Let $d_1, \ldots, d_k \in D$ be nodes satisfying the A-types $T_1, \ldots, T_k$, respectively. From the universal part of the formula, it follows that for each $d \in D$ there exists $T_d \in \overline{\mathbf{T}}$ such that $d \in \bigcap_{a \in T_d} V(a)$.

We will now use this information to the end of showing (i)(a). That is, we wish to show that $(D, V_\pi) \models \nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$. First, observe that $(d_1, 1), \ldots, (d_k, k)$ is a sequence of k distinct nodes from $D \times \omega$. It follows immediately from the definition of V_π that, for each $i \leq k$, we have that (d_i, i) satisfies the A-type T_i because, for each $i \leq k$, we have $d_i \in V(a)$ for each $a \in T_i$. Moreover, for each node $(d, j) \in D \times \omega$ which is not among $(d_1, 1), \ldots, (d_k, k)$, we have that d satisfies one of the types from Π hence also (d, j) satisfies some type from Π by definition of V_π by the same reasoning. Hence $(D, V_\pi) \models \alpha$, as desired.

We will now show that (ii) holds. To this end, suppose that $(D \times \omega, V) \models \alpha$. Define the valuation $U^\pi : A \to D$ by setting

$$U^\pi(a) := \{d \in D \mid (d, i) \in V(a) \text{ for some } i \in \omega\}.$$

Again, item (ii)(b) follows directly from this definition so we turn our attention towards (ii)(a). That is, we proceed to show that $(D, U^\pi) \models \nabla^+_{\mathrm{FO}}(\overline{\mathbf{T}}, \Pi)$ if $(D \times \omega, U) \models \nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$. Observe that if $(d_1, n_1), \ldots, (d_k, n_k) \in D \times \omega$ are the distinct witnesses of the A-types $T_1, \ldots, T_k$ given by the existential part of $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$, then we have that $(d_i, n_i) \in \bigcap_{a \in T_i} U(a)$ for each $i \leq k$. That is, for each $i \leq k$ and each $a \in T_i$, we have that $(d_i, n) \in U(a)$ for some $n \in \omega$ hence also $d_i \in \bigcap_{a \in T_i} U^\pi(a)$.

Now, from the universal part of $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$, we have that each node $(d, i) \in D \times \omega$ witnesses some type $T_{d,i} \in \Pi$. Just as above, it follows that d witnesses $T_{d,i}$ in (D, U^π). In short, $(D, U^\pi) \models \nabla^+_{\mathrm{FO}}(\overline{\mathbf{T}}, \Pi)$, as desired. □

The following proposition states an important model-theoretic relationship between a tree and its ω-expansion, and its statement is a key ingredient in the proof of Janin and Walukiewicz's celebrated theorem. Its statement is therefore well known and so we opt to omit a proof and refer the interested reader to Venema [35] for the details.

Proposition 5.1.7. *For every automaton $\mathbb{A} \in Aut_{sa}(\mathrm{FOE}_1)$ and every tree $\mathbb{T}$ we have the following equivalence:*

$$\mathbb{A}^\blacktriangledown \textit{ accepts } \mathbb{T} \textit{ iff } \mathbb{A} \textit{ accepts } \mathbb{T}^\omega.$$

We have now gathered all of the ingredients needed to prove the remaining inclusion in the equivalence $(*)$:

Proof of $(*)$. Suppose that $\mathbb{A} \in Aut_{sa}(\mathrm{FOE}_1)/\underline{\leftrightarrow}$. Then, for every transition system $\mathbb{S}$ we have the following chain of equivalences:

$$\begin{aligned} \mathbb{A} \text{ accepts } \mathbb{S} &\text{ iff } \mathbb{A} \text{ accepts } \hat{\mathbb{S}}^\omega && (\mathbb{A} \text{ is bisimulation-invariant}) \\ &\text{ iff } \mathbb{A}^\blacktriangledown \text{ accepts } \hat{\mathbb{S}} && (\text{Proposition 5.1.7}) \\ &\text{ iff } \mathbb{A}^\blacktriangledown \text{ accepts } \mathbb{S}. && (\mathbb{A}^\blacktriangledown \text{ is bisimulation-invariant}) \end{aligned}$$

□

Chapter 6

Conclusion

Let us briefly reflect on what was and what was not accomplished in this thesis.

Reflections on Automata and First-order Logic

In the second chapter, we introduced several classes of automata and paid particular attention to the theory of linked antisymmetric path automata. We showed that the combination of antisymmetry and path conditions are in fact related to finite paths through trees: we showed that each winning position of a given acceptance game whose first component is a μ-state may be identified with a unique finite path through the tree.

This essence was a key ingredient in the translation from automata from $Aut^{l}_{sa}(\mathrm{FOE}_1)$ into first-order formulas. Moreover, attempts to find a first-order definable tree language which is not recognized by linked antisymmetric path automata have been in vain. For this reason, I believe that this class remains a good candidate for capturing first-order definability of tree languages. That is, we believe that the class of linked antisymmetric path automata and first-order logic could in fact be effectively equivalent over trees. However, several attempts at translating first-order formulas into such automata were made in the writing of this thesis, each of which had their own issues.

Let me begin by discussing with the non-issues in these attempts. Just as in the case of linked *weak* path automata, the closure of linked antisymmetric path recognizable languages under complementation and union follow in a straightforward manner. In fact, the astute reader might have noticed that the automata translations of atomic formulas, disjunctions, and negations are in linked antisymmetric path automata themselves. The main issue arises when one wishes to show that this class is closed under atomic projection. Two attempts were made in this direction.

First, as the two-sorted construction preserves the weakness condition, one may naturally wonder whether this construction additionally preserves antisymmetry. Unfortunately, this is not the case. A counter-example can be given by an automaton with two states. Indeed, let $\mathbb{A}$ be the automaton $(\{a, b\}, \Theta, \Omega, a)$ where a is active in itself (say, $\Theta(a, c) = \exists x.a(x)$ whenever $p \in c$ and $\Theta(a, c) = \exists x.b(x)$ otherwise) and where the transitions of b are given by putting $\Theta(b, c) = \forall x.x \approx x$ for every colour $c \in \{p, q\}$; for concreteness, the priority map may be given by $\Omega(a) = 0 = \Omega(b)$. Observe that $a \lhd a$ and $b \lhd a$ give a complete description of the relation $\lhd$. It is now straightforward to check that

(i) $\mathbb{A}$ is a linked antisymmetric path automaton;

(ii) the macro-states $B_0 = \{a, b\}$ and $B_1 = \{a\}$ form a cluster in the atomic projection construction $\exists p.\mathbb{A}$.

In short, the two-sorted construction considered in this thesis does not preserve antisymmetry. The second attempt failed for essentially the same reason. I think that the moral is that antisymmetry is a difficult condition to preserve, and one would benefit from a better understanding of exactly why the two-sorted construction and weakness condition work so nicely together.

With all of this being said, I am very interested in whether or not the equivalence $Aut^l_{sa}(\text{FOE}) \equiv \text{FOE}$ holds on trees. Given the work in this thesis, a proof of this equivalence boils down to providing an appropriate projection construction. It seems that the nature of such a construction would require a very different flavour than the two-sorted construction considered in this thesis. On the other hand, a counter-example to this equivalence would also be satisfying. This would raise at least two interesting questions.

(1) Which fragment of FOE does the class $Aut^l_{sa}(\text{FOE})$ correspond to on trees?

(2) Which class of alternating parity automata correspond to FOE on trees?

We will now shift our focus to reflections on the results related to linked weak path automata based on FOE_1. Notably, we proved that for every first-order formula φ we can effectively obtain an equivalent linked weak path automaton $\mathbb{A}_\varphi$ over trees. A natural question following such a result is this: can we provide a translation in the other direction? The author strongly believes that the answer to this question is 'no'. This is because one can provide a linked weak path automaton $\mathbb{A}$ such that $\mathbb{A}$ accepts $\mathbb{T}$ iff there is a node at an even level of $\mathbb{T}$ coloured by p. A node t occurs at an even level if the unique finite path from the root to t has odd length (e.g. the root occurs at an even level). We explain only the essence of this construction: an automaton $\mathbb{A}$ with a single cluster consisting of two nodes can 'count' modulo 2. There is a well known proof (using Ehrenfeucht-Fraïssé games) that this is not a first-order definable property on ω-streams. As ω-streams are a particular example of a tree, this argument applies to trees as well.

Reflections on Bisimulation-invariance

In the final chapter, we explored the bisimulation-invariant 'fragment' of $Aut^l_{sa}(\mathrm{FOE}_1)$, using the framework set up in the proof of the Janin-Walukiewicz Theorem and the observation, this question boiled down to a question at the one-step level. We showed that $Aut^l_{sa}(\mathrm{FO}_1) \equiv Aut^l_{sa}(\mathrm{FOE}_1)/\underline{\leftrightarrow}$.

Recall that we briefly discussed the relationship between the modal μ-calculus and the class $Aut(\mathrm{FO}_1)$. In particular, we discussed these automata as a fruitful ambient class for studying fragments of modal fixpoint logics. One example of such a logic is the so-called *computation tree logic* (CTL): an extension of modal logic by path quantifiers $\exists \mathrm{U}\varphi\psi$ and $\forall \mathrm{U}\varphi\psi$ expressing that there is a (respectively for every) finite path, beginning at the current node, such that ψ is true *until* φ is true . While the quantifier $\exists\mathrm{U}$ is first-order definable, it is known that the quantifier $\forall\mathrm{U}$ is not.

An automata-theoretic characterization of the expressive power of CTL was given in [33] using automata with a Büchi acceptance condition. That is, parity games were not used in the operational semantics of these automata. In order to capture precisely CTL definability, two constraints were imposed on the automata there. First, the graph structure of such automata was constrained to have only singleton clusters; this is clearly what we call antisymmetry. Second, a notion of *dominance* (on clusters) was introduced. We strongly believe that dominance is essentially what we call the path condition. It would therefore be interesting to obtain effective translations between these automata. Such translations would give a new perspective on CTL and raise interesting questions on matters of size and complexity.

References

[1] André Arnold. “A Syntactic Congruence for Rational ω-languages”. In: *Theoretical Computer Science* (1985), pp. 333–335.

[2] Johan van Benthem. “Modal Correspondence Theory”. PhD Thesis. University of Amsterdam, 1977.

[3] Mikołaj Bojańczyk. “Decidable Properties of Tree Languages”. PhD thesis. Warsaw University, 2004.

[4] Julius Büchi. “On a Decision Method in Restricted Second-order Arithmetic”. In: *The Collected Works of J. Richard Büchi*. Ed. by Saunders Mac Lane and Dirk Siefkes. Springer New York, 1962, pp. 425–435.

[5] Julius Büchi. “Weak Second-Order Arithmetic and Finite Automata”. In: *Mathematical Logic Quarterly* (1960), pp. 66–92.

[6] Facundo Carreiro. “Fragments of Fixpoint Logics: Automata and Expressiveness”. PhD thesis. Institute for Logic, Language, and Computation, University of Amsterdam, 2015.

[7] Facundo Carreiro, Alessandro Facchini, Yde Venema, and Fabio Zanasi. “Model Theory of Monadic Predicate Logic with the Infinity Quantifier”. In: *CoRR abs/1809.03262* (2018a).

[8] Facundo Carreiro, Alessandro Facchini, Yde Venema, and Fabio Zanasi. “The Power of the Weak”. In: *CoRR abs/1809.03896* (2018b).

[9] Facundo Carreiro, Alessandro Facchini, Yde Venema, and Fabio Zanasi. “Weak MSO: Automata and Expressiveness Modulo Bisimilarity”. In: *Proceedings of the Joint Meeting of the 23rd Annual Conference on Computer Science in Logic and the 29th Annual Symposium on Logic in Computer Science (CSL-LiCS ‘14)* (2014).

[10] Ashok Chandra, Dexter Kozen, and Larry Stockmeyer. “Alternation”. In: *Journal of ACM* (1981), pp. 114–133.

[11] Corina Cîrstea and Dirk Pattinson. “Modular Construction of Modal Logics”. In: *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR ‘04)*. 2004, pp. 258–275.

[12] Edmund Clarke and E. Allen Emerson. "Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic". In: *Proceedings of the Workshop on the Logic of Programs*. 1981, pp. 52–71.

[13] Volker Diekert and Paul Gastin. "First-order Definable Languages". In: *Logic and Automata: History and Perspectives*. 2008, pp. 261–306.

[14] E. Allen Emerson and Charanjit Jutla. "Tree Automata, Mu-Calculus and Determinacy". In: Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (SFCS '91). 1991, pp. 368–377.

[15] Alessandro Facchini, Yde Venema, and Fabio Zanasi. "A Characterization Theorem for the Alternation-Free Fragment of the Modal μ-Calculus". In: *Proceedings of the 28th Annual Symposium on Logic in Computer Science (LICS '13)* (2013), pp. 478–487.

[16] Gaëlle Fontaine and Yde Venema. "Some Model Theory for the Modal μ-calculus: Syntactic Characterisations of Semantic Properties". In: *Logical Methods in Computer Science* 14(1) (2018).

[17] Eric Grädel, Wolfgang Thomas, and Thomas Wilke. *Automata Logics, and Infinite Games: A Guide to Current Research*. Springer, 2002.

[18] Marco Hollenberg. "Logic and Bisimulation". PhD Thesis. Zeno Institute of Philosophy, Utrecht University, 1998.

[19] David Janin and Igor Walukiewicz. "On the Expressive Completeness of the Propositional mu-Calculus with Respect to Monadic Second-order Logic". In: *Proceedings of the 7th Annual International Conference on Concurrency Theory (CONCUR '96)*. 1996, pp. 263–277.

[20] Bjarni Jonsson and Alfred Tarski. "Boolean Algebras with Operators. Part I". In: *American Journal of Mathematics* (1951), pp. 891–939.

[21] Christian Kissig and Yde Venema. "Complementation of Coalgebra Automata". In: *Proceedings of the 3rd Conference on Algebra and Coalgebra (CALCO '09)*. 2009, pp. 81–96.

[22] Andrzej Mostowski. "Hierarchies of Weak Automata and Weak Monadic Formulas". In: *Theoretical Computer Science* (1991), pp. 323–335.

[23] David Muller. "Infinite Sequences and Finite Machines". In: *Proceedings of the Fourth Annual Symposium on Switching Circuit Theory and Logical Design (SWCT '63)*. 1963, pp. 3–16.

[24] David Muller, Ahmed Saoudi, and Paul Schupp. "Alternating Automata, The Weak Monadic Theory of the Tree, and its Complexity". In: *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP '86)*. 1986.

[25] David Muller and Paul Schupp. "Alternating Automata on Infinite Trees". In: *Theoretical Computer Science* (1987), pp. 267–276.

[26] David Muller and Paul Schupp. "Simulating Alternating Tree Automata by Nondeterministic Automata: New results and New Proofs of the Theorems of Rabin, McNaughton and Safra". In: *Theoretical Computer Science* (1995), pp. 69–107.

[27] Jakub Neumann, Andrzej Szepietowski, and Igor Walukiewicz. "Complexity of Weak Acceptance Conditions in Tree Automata". In: *Information Processing Letters* (2002), pp. 181–187.

[28] Robert McNaughton Seymour Papert. *Counter-Free Automata (M.I.T. Research Monograph No. 65)*. The MIT Press, 1971.

[29] Dominique Perrin. "Recent Results on Automata and Infinite Words". In: *Proceedings of the 11th Annual Symposium on the Mathematical Foundations of Computer Science (MFCS '84)*. 1984, pp. 134–148.

[30] Andreas Potthoff. "First-Order Logic on Finite Trees". In: *Proceedings of the 6th Annual International Joint Conference on the Theory and Practice of Software Development (TAPSOFT '95)*. 1995, pp. 123–139.

[31] Michael Rabin. "Decidability of Second-order Theories and Automata on Infinite Trees". In: *Transactions of the American Mathematical Society* 141 (1969), pp. 1–35.

[32] Marcel Schützenberger. "On Finite Monoids Having Only Trivial Subgroups". In: *Information and Control* (1965), pp. 190–194.

[33] Sander in 't Veld. "Temporal Logics, Automata, and the Modal μ-calculus". MSc Thesis. Institute for Logic, Language, and Computation, University of Amsterdam, 2016.

[34] Yde Venema. "Automata and Fixed Point Logic: A Coalgebraic Perspective". In: *Information and Computation* 204 (2006), pp. 637–678.

[35] Yde Venema. *Lectures on the Modal μ-calculus*. Lecture Notes. Dec. 2018.

[36] Igor Walukiewicz. "Monadic Second Order Logic on Tree-Like Structures". In: *Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS '96)*. 1996, pp. 401–413.

[37] Fabio Zanasi. "Expressiveness of Monadic Second-order Logics on Infinite Trees of Arbitrary Branching Degree". MSc Thesis. Institute for Logic, Language, and Computation, University of Amsterdam, 2012.