The Power of the Weak

FACUNDO CARREIRO, University of Amsterdam
ALESSANDRO FACCHINI, IDSIA, USI-SUPSI, Switzerland
YDE VENEMA, University of Amsterdam
FABIO ZANASI, University College London

A landmark result in the study of logics for formal verification is Janin and Walukiewicz's theorem, stating that the modal μ-calculus (μML) is equivalent modulo bisimilarity to standard monadic second-order logic (here abbreviated as SMSO) over the class of labelled transition systems (LTSs for short). Our work proves two results of the same kind, one for the alternation-free or *noetherian* fragment μ_NML of μML on the modal side and one for WMSO, weak monadic second-order logic, on the second-order side. In the setting of binary trees, with explicit functions accessing the left and right successor of a node, it was known that WMSO is equivalent to the appropriate version of alternation-free μ-calculus. Our analysis shows that the picture changes radically once we consider, as Janin and Walukiewicz did, the standard modal μ-calculus, interpreted over arbitrary LTSs.

The first theorem that we prove is that, over LTSs, μ_NML is equivalent modulo bisimilarity to *noetherian* MSO (NMSO), a newly introduced variant of SMSO where second-order quantification ranges over "conversely well-founded" subsets only. Our second theorem starts from WMSO and proves it equivalent modulo bisimilarity to a fragment of μ_NML defined by a notion of continuity. Analogously to Janin and Walukiewicz's result, our proofs are automata-theoretic in nature: As another contribution, we introduce classes of parity automata characterising the expressiveness of WMSO and NMSO (on tree models) and of μ_CML and μ_NML (for all transition systems).

CCS Concepts: • **Theory of computation → Logic**; **Formal languages and automata theory**;

Additional Key Words and Phrases: Modal μ-calculus, weak monadic second-order logic, tree automata, bisimulation

ACM Reference format:
Facundo Carreiro, Alessandro Facchini, Yde Venema, and Fabio Zanasi. 2020. The Power of the Weak. *ACM Trans. Comput. Logic* 21, 2, Article 15 (January 2020), 47 pages.
https://doi.org/10.1145/3372392

We would like to thank the reviewers for their work and the editor for handling our paper. In this final version, we addressed all the minor comments that were made by the reviewers in the last round.
Authors' addresses: F. Carreiro, Google, 2930 Pearl St, Boulder, CO 80301, United States; email: fcarreiro@gmail.com; A. Facchini, Dalle Molle Institute for Artificial Intelligence USI-SUPSI, Galleria 2, Via Cantonale 2c, CH-6928 Manno, Switzerland; email: alessandro.facchini@unil.ch; Y. Venema, Institute for Logic, Language and Computation, University of Amsterdam, P.O. Box 94242, 1090 GE Amsterdam, The Netherlands; email: Y.Venema@uva.nl; F. Zanasi, Department of Computer Science, University College London, 66-72 Gower Street, 66-72 Gower Street, London WC1E 6BT, United Kingdom; email: f.zanasi@ucl.ac.uk.

1529-3785/2020/01-ART15 $15.00
https://doi.org/10.1145/3372392

1 INTRODUCTION

1.1 Expressiveness Modulo Bisimilarity

A seminal result in the theory of modal logic is van Benthem's Characterisation Theorem [van Benthem 1977], stating that, over the class of all labelled transition systems (LTSs for short), every bisimulation-invariant first-order formula is equivalent to (the standard translation of) a modal formula:

$$\mathrm{ML} \equiv \mathrm{FO}/\underline{\leftrightarrow} \qquad \text{(over the class of all LTSs).} \tag{1}$$

Over the years, a wealth of variations of the Characterisation Theorem have been obtained. For instance, van Benthem's theorem is one of the few preservation results that transfers to the setting of finite models [Rosen 1997]; for a recent, rich source of van Benthem-style characterisation results, see Dawar and Otto [2009]. The general pattern of these results takes the shape

$$M \equiv L/\underline{\leftrightarrow} \qquad \text{(over a class of models C).} \tag{2}$$

Apart from their obvious relevance to model theory, the interest in these results increases if C consists of transition structures that represent certain computational processes, as in the theory of the formal specification and verification of properties of software. In this context, one often takes the point of view that bisimilar models represent *the same* process. For this reason, only bisimulation-invariant properties are relevant. Seen in this light, Equation (2) is an *expressive completeness* result: All the relevant properties expressible in L (which is generally some rich yardstick formalism) can already be expressed in a (usually computationally more feasible) modal fragment M.

Of special interest to us is Janin and Walukiewicz [1996], which extends van Benthem's result to the setting of *second-order* logic, by proving that the bisimulation-invariant fragment of standard monadic second-order logic (SMSO) is the *modal μ-calculus* (μML), viz., the extension of basic modal logic with least- and greatest fixpoint operators:

$$\mu\mathrm{ML} \equiv \mathrm{SMSO}/\underline{\leftrightarrow} \qquad \text{(over the class of all LTSs).} \tag{3}$$

The aim of this article is to study the fine structure of such connections between second-order logics and modal μ-calculi, obtaining variations of the expressiveness completeness results of Equations (1) and (3).

Our departure point is a result from Arnold and Niwiński [1992] for the class of *binary* trees. Their setting is somewhat different from ours; in particular, since their trees have explicit functions accessing the left and right successor of a node, the notion of bisimilarity trivialises to the isomorphism relation. Nevertheless, the key observation of Arnold and Niwiński is to link the *alternation-free fragment* of a modal μ-calculus to so-called *weak* MSO, the version of monadic second-order logic where second-order quantification is restricted to finite sets. Here, the alternation-free constraint means that only trivial interactions between least and greatest fixpoint operators are permitted (more precise definitions will be provided in Section 3).

On the basis of the results by Janin and Walukiewicz and by Arnold and Niwiński it is natural to conjecture that

$$\mu_N\mathrm{ML} \equiv \mathrm{WMSO}/\underline{\leftrightarrow} \qquad \text{(over the class of finitely branching trees),} \tag{4}$$

where a tree is said to be *finitely branching* if each node has only finitely many immediate successors, and the logic μ_NML is the alternation-free fragment of μML, or, as we shall explain futher on, the "noetherian" version of the modal μ-calculus. Note that this logic μ_NML is a very natural fragment of the full μ-calculus; in particular, since the alternation depth of a fixpoint formula is one of the key parameters that determine the complexity of model checking algorithms for μML [Calude et al. 2017], the alternation-free fragment is of special interest for applications. Similarly, *weak* MSO, the logic featuring on the right-hand side of Equation (4), is a well-known variation of standard MSO, and it also has been studied in the setting of applications in software verification (see, e.g., Grädel et al. [2002, Ch. 3]).

In other words, Equation (4) is an interesting expressive completeness statement, linking two well-known logical systems. Nevertheless, while we will show that Equation (4) holds, our investigations take a wider scope. Note that Equation (4) only offers a comparison of the logics μ_NML and WMSO on finitely branching trees, whereas Equations (1) and (3) work at the level of arbitrary models. In fact, it turns out that the picture in the more general setting is far more subtle. First, we know that already at the level of arbitrary trees, the equation $\mu_N\mathrm{ML} \equiv \mathrm{WMSO}/\underline{\leftrightarrow}$ is *false*, since the class of conversely well-founded trees, definable by the formula $\mu x.\Box x$ of μ_NML, is not WMSO-definable. Moreover, whereas WMSO is a fragment of SMSO on finitely branching trees, as soon as we allow for infinite branching the two logics turn out to have *incomparable* expressive power—we will discuss the details in Section 3.

One of the main questions of this work, then, is to clarify the relation between $\mathrm{WMSO}/\underline{\leftrightarrow}$ and μ_NML on arbitrary LTSs. We shall prove that, in this more general setting, Equation (4) "splits" into the following two results, which refer to a relatively unknown fragment μ_CML of the modal μ-calculus and introduce a new second-order logic NMSO.

Theorem 1.1.

$$\mu_C\mathrm{ML} \equiv \mathrm{WMSO}/\underline{\leftrightarrow} \qquad \textit{(over the class of all LTSs)}, \tag{5}$$

$$\mu_N\mathrm{ML} \equiv \mathrm{NMSO}/\underline{\leftrightarrow} \qquad \textit{(over the class of all LTSs)}. \tag{6}$$

For the first result, Equation (5), our strategy is to start from WMSO and seek a suitable modal fixpoint logic characterising its bisimulation-invariant fragment. Second-order quantification $\exists p.\varphi$ in WMSO requires p to be interpreted over a finite subset of an LTS. We identify a notion of *continuity* as the modal counterpart of this constraint, and call the resulting logic μ_CML, the *continuous* μ-calculus. This fragment of μML, which was introduced in van Benthem [2006] under the name of "ω-μ-calculus," can defined by the same grammar as the full μML,

$$\varphi \ ::= q \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \mu p.\varphi',$$

with the difference that φ' does not just need to be positive in p but also continuous in p. This terminology refers to the fact that φ' is interpreted by a function that is continuous with respect to the Scott topology; as we shall see in Section 3, p-continuity can be given a *syntactic* characterisation, as a certain fragment of μML, which will be used to define the logic μ_CML.

For our second result, Equation (6), we move in the opposite direction. That is, we look for a natural second-order logic of which μ_NML is the bisimulation-invariant fragment. Symmetrically to the case of Equation (5) of WMSO and continuity, a crucial aspect is to identify which constraint on second-order quantification corresponds to the constraint on fixpoint alternation expressed by μ_NML. Our analysis stems from the observation that, when a formula $\mu p.\varphi$ of μ_NML is satisfied in a tree model $\mathbb{T}$, the interpretation of p must be a subset of a *conversely well-founded* subtree of $\mathbb{T}$, because alternation-freedom prevents p from occurring in a ν-subformula of φ. We introduce the concept of a *noetherian* subset as a generalisation of this property from trees to arbitrary LTSs: Intuitively, a subset of a LTS $\mathbb{S}$ is called noetherian if it is a subset of a bundle of paths that does not contain any infinite ascending chain. (Precise definitions will be supplied in Section 3.) The logic NMSO appearing in Equation (6), which we call *noetherian* second-order logic, is the variation of MSO restricting second-order quantification to noetherian subsets.

A unifying perspective over these results can be given through the lens of König's lemma, saying that a subset of a tree $\mathbb{T}$ is finite precisely when it is included in a subtree of $\mathbb{T}$, which is both finitely branching and conversely well founded. In other words, finiteness on trees has two components, a *horizontal* (finite branching) and a *vertical* (well-foundedness) dimension. The bound imposed by NMSO-quantification acts only on the *vertical* dimension, whereas WMSO-quantification acts on both. It then comes as no surprise that Equations (5) and (6) collapse to Equation (4) on finitely branching trees. The restriction to (unbounded) finitely branching models nullifies the difference between noetherian and finite, equating WMSO and NMSO (and thus also μ_NML and μ_CML).

Another interesting observation concerns the relative expressive power of WMSO with respect to standard MSO. As mentioned above, WMSO is *not* strictly weaker than SMSO on arbitrary LTSs. Nonetheless, putting together Equations (3) and (5) reveals that WMSO collapses within the boundaries of SMSO-expressiveness when it comes to bisimulation-invariant formulas, because μ_CML is strictly weaker than μML. In fact, modulo bisimilarity, WMSO turns out to be even weaker than NMSO, as μ_CML is also a fragment of μ_NML. In a sense, this new landscape of results tells us that the feature distinguishing WMSO from SMSO/NMSO, *viz.* the ability of expressing cardinality properties of the horizontal dimension of models, disappears once we focus on the bisimulation-invariant part and thus is not computationally relevant.

1.2 Automata-theoretic Characterisations

Janin and Walukiewicz's proof of Equation (3) passes through a characterisation of the two logics involved in terms of *parity automata*. In a nutshell, a parity automaton $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ processes LTSs as inputs, according to a transition function Δ defined in terms of a so-called *one-step logic* $L_1(A)$, where the states A of $\mathbb{A}$ may occur as unary predicates. The map $\Omega : A \to \mathbb{N}$ assigns to each state a *priority*; if the least priority value occurring infinitely often during the computation is even, then the input is accepted. Both SMSO and μML are characterised by classes of parity automata: What changes is just the one-step logic, which is, respectively, first-order logic with (FOE_1) and without (FO_1) equality,

$$\mathrm{SMSO} \equiv Aut(\mathrm{FOE}_1) \qquad \text{(over the class of all trees)}, \tag{7}$$

$$\mu\mathrm{ML} \equiv Aut(\mathrm{FO}_1) \qquad \text{(over the class of all LTSs)}. \tag{8}$$

This kind of automata-theoretic characterisation, which we believe is of independent interest, also underpins our two correspondence results. As the second main contribution of this article, we introduce new classes of parity automata that exactly capture the expressive power of the second-order languages WMSO and NMSO (over tree models) and of the modal languages μ_CML and μ_NML (over arbitrary models).

Let us start from the simpler case, that is, NMSO and μ_NML. As mentioned above, the leading intuition for these logics is that they are constrained in what can be expressed about the *vertical* dimension of models. In automata-theoretic terms, we translate this constraint into the requirement that runs of an automaton can see at most one parity infinitely often: This yields the class of so-called *weak* parity automata [Muller et al. 1992], which we write as $Aut_w(L_1)$ for a given one-step logic L_1.[1] We shall show the following:

Theorem 1.2.

$$\mathrm{NMSO} \equiv Aut_w(\mathrm{FOE}_1) \qquad \textit{(over the class of all trees)}, \tag{9}$$

$$\mu_N\mathrm{ML} \equiv Aut_w(\mathrm{FO}_1) \qquad \textit{(over the class of all LTSs)}. \tag{10}$$

It is worthwhile to zoom in on our main point of departure from Janin and Walukiewicz' proofs of Equations (7) and (8). In the characterisation (7), due to Walukiewicz [1996], a key step is to show

[1]Interestingly, Muller et al. [1992] introduces the class $Aut_w(\mathrm{FOE}_1)$ to show that it characterises WMSO on finitely branching trees, whence the name of *weak* automata. As discussed above, this correspondence is an "optical illusion," due to the restricted class of models that are considered, on which NMSO = WMSO.

that each automaton in $Aut(\mathrm{FOE}_1)$ can be simulated by an equivalent *non-deterministic* automaton of the same class. This is instrumental in the projection construction, allowing us to build an automaton equivalent to $\exists p.\varphi \in$ MSO starting from an automaton for φ. Our counterpart, Equation (9), is also based on a simulation theorem. However, we cannot proceed in the same manner, as the class $Aut_w(\mathrm{FOE}_1)$, unlike $Aut(\mathrm{FOE}_1)$, is *not* closed under non-deterministic simulation. Thus we devise a different construction, which starting from a weak automaton $\mathbb{A}$ creates an equivalent automaton $\mathbb{A}'$, which acts non-deterministically only on a *conversely well-founded* part of each accepted tree. It turns out that the class $Aut_w(\mathrm{FOE}_1)$ is closed under this variation of the simulation theorem; moreover, the property of $\mathbb{A}'$ is precisely what is needed to make a projection construction that mirrors NMSO-quantification.

We now consider the automata-theoretic characterisation of WMSO and μ_CML. Whereas in Equations (9) and (10) the focus was on the vertical dimension of a given model, the constraint that we now need to translate into automata-theoretic terms concerns both *vertical* and *horizontal* dimensions. Our revision of Equations (7) and (8) thus moves on two different axes. The constraint on the vertical dimension is handled analogously to the cases of Equations (9) and (10), by switching from standard to *weak* parity automata. The constraint on the horizontal dimension requires more work. The first problem lies in finding the right one-step logic, which should be able to express cardinality properties, as WMSO is able to do. An obvious candidate would be weak monadic second-order logic itself, or, more precisely, its variation WMSO_1, over the signature of unary predicates (corresponding to the automata states). A very helpful observation from Väänänen [1977] is that we can actually work with an equivalent formalism that is better tailored to our aims. Indeed, $\mathrm{WMSO}_1 \equiv \mathrm{FOE}_1^\infty$, where FOE_1^∞ is the extension of FOE_1 with the generalised quantifier $\exists^\infty$, with $\exists^\infty x.\varphi$ stating the existence of *infinitely* many objects satisfying φ.

At this stage, our candidate automata class for WMSO could be $Aut_w(\mathrm{FOE}_1^\infty)$. However, this fails because FOE_1^∞ bears too much expressive power: Since it extends FOE_1, we would find that, over tree models, $Aut_w(\mathrm{FOE}_1^\infty)$ extends $Aut_w(\mathrm{FOE}_1)$, whereas we already saw that $Aut_w(\mathrm{FOE}_1) \equiv$ NMSO is incomparable to WMSO. It is here that we crucially involve the notion of *continuity*. For a class $Aut_w(L_1)$ of weak parity automata, we call *continuous-weak* parity automata, forming a class $Aut_{wc}(L_1)$, those satisfying the following additional constraint:

— for every state a with even priority $\Omega(a)$, every one-step formula $\varphi \in L_1(A)$ defining the transitions from a has to be continuous in all states a' lying in a cycle with a; dually, if $\Omega(a)$ is odd, then every such φ has to be a'-cocontinuous.[2]

We can now formulate our characterisation result as follows.

Theorem 1.3.

$$\mathrm{WMSO} \equiv Aut_{wc}(\mathrm{FOE}_1^\infty) \qquad \textit{(over the class of trees)}, \tag{11}$$

$$\mu_C\mathrm{ML} \equiv Aut_{wc}(\mathrm{FO}_1) \qquad \textit{(over the class of all LTSs)}. \tag{12}$$

Thus automata for WMSO deviate from SMSO-automata $Aut(\mathrm{FOE}_1)$ on two different levels: at the global level of the automaton run, because of the weakness and continuity constraint, and at the level of the one-step logic defining a single transition step. Another interesting point stems from pairing Equations (11) and (12) with the expressive completeness result of Equation (5): Although automata for WMSO are based on a more powerful one-step logic (FOE_1^∞) than those for μ_CML

[2]It is important to stress that, even though continuity is a semantic condition, we have a *syntactic* characterisation of FOE_1^∞-formulas satisfying it (see Carreiro et al. [2018]), meaning that $Aut_{wc}(\mathrm{FOE}_1^\infty)$ is definable independently of the structures taken as input.

(FO_1), modulo bisimilarity they characterise the same expressiveness. This connects to our previous observation that the ability of WMSO to express cardinality properties on the horizontal dimension vanishes in a bisimulation-invariant context.

1.3 Outline

It is useful to conclude this introduction with a roadmap of how the various results are achieved. In a nutshell, the two expressive completeness theorems (5) and (6) will be based respectively on the following two chains of equivalences:

$$\mu_N\mathrm{ML} \equiv \mu_N\mathrm{FO}_1 \equiv Aut_w(\mathrm{FO}_1) \equiv Aut_w(\mathrm{FOE}_1)/{\underline{\leftrightarrow}} \equiv \mathrm{NMSO}/{\underline{\leftrightarrow}} \text{ (over LTSs)}, \tag{13}$$

$$\mu_C\mathrm{ML} \equiv \mu_C\mathrm{FO}_1 \equiv Aut_{wc}(\mathrm{FO}_1) \equiv Aut_{wc}(\mathrm{FOE}_1^\infty)/{\underline{\leftrightarrow}} \equiv \mathrm{WMSO}/{\underline{\leftrightarrow}} \text{ (over LTSs)}. \tag{14}$$

After giving a precise definition of the necessary preliminaries in Sections 2 and 3, we proceed as follows. First, Section 4 introduces parity automata parametrised over a one-step language L_1, in the standard ($Aut(L_1)$), weak ($Aut_w(L_1)$), and continuous-weak ($Aut_{wc}(L_1)$) forms. With Theorems 4.33 and 4.34, we show that

$$\mu_N L_1 \equiv Aut_w(L_1) \qquad \mu_C L_1 \equiv Aut_{wc}(L_1) \qquad \text{(over LTSs)}, \tag{15}$$

where $\mu_N L_1$ and $\mu_C L_1$ are extensions of L_1 with fixpoint operators subject to a "noetherianess" and a "continuity" constraint, respectively. Instantiating Equation (15) yields the second equivalence both in Equations (13) and (14):

$$\mu_N\mathrm{FO}_1 \equiv Aut_w(\mathrm{FO}_1) \qquad \mu_C\mathrm{FO}_1 \equiv Aut_{wc}(\mathrm{FO}_1) \qquad \text{(over LTSs)}.$$

Next, in Section 5, Theorem 5.2, we show how to construct from a WMSO-formula an equivalent automaton of the class $Aut_{wc}(\mathrm{FOE}_1^\infty)$. In Section 6, Theorem 6.2, we show the analogous characterisation for NMSO and $Aut_w(\mathrm{FOE}_1)$. These two sections yield part of the last equivalence in Equation (14) and in Equation (13), respectively,

$$Aut_w(\mathrm{FOE}_1) \geq \mathrm{NMSO} \qquad Aut_{wc}(\mathrm{FOE}_1^\infty) \geq \mathrm{WMSO} \qquad \text{(over trees)}. \tag{16}$$

Notice that, differently from all the other proof pieces, Equation (16) only holds on trees, because the projection construction for automata relies on the input LTSs being tree shaped.

Section 7 yields the remaining bit of the automata characterisations. Theorem 7.1 shows

$$\mu_N\mathrm{FOE}_1 \leq \mathrm{NMSO} \qquad \mu_C\mathrm{FOE}_1^\infty \leq \mathrm{WMSO} \qquad \text{(over LTSs)},$$

which, paired with Equation (15), yields

$$Aut_w(\mathrm{FOE}_1) \equiv \mu_N\mathrm{FOE}_1 \leq \mathrm{NMSO} \quad Aut_{wc}(\mathrm{FOE}_1^\infty) \equiv \mu_C\mathrm{FOE}_1^\infty \leq \mathrm{WMSO} \text{ (over LTSs)}.$$

Putting the last equation and Equation (16) together, we have our automata characterisations,

$$Aut_w(\mathrm{FOE}_1) \equiv \mathrm{NMSO} \qquad Aut_{wc}(\mathrm{FOE}_1^\infty) \equiv \mathrm{WMSO} \qquad \text{(over trees)},$$

which also yield the rightmost equivalence in Equation (14) and in Equation (13), because any LTS is bisimilar to its tree unraveling,

$$Aut_w(\mathrm{FOE}_1)/{\underline{\leftrightarrow}} \equiv \mathrm{NMSO}/{\underline{\leftrightarrow}} \qquad Aut_{wc}(\mathrm{FOE}_1^\infty)/{\underline{\leftrightarrow}} \equiv \mathrm{WMSO}/{\underline{\leftrightarrow}} \qquad \text{(over LTSs)}.$$

At last, Section 8 is split into two parts. First, Theorem 8.1 extends the results in Section 4 to complete the following chains of equivalences, yielding the first block in Equation (13) and in Equation (14),

$$\mu_N\mathrm{ML} \equiv \mu_N\mathrm{FO}_1 \equiv Aut_w(\mathrm{FO}_1) \qquad \mu_C\mathrm{ML} \equiv \mu_C\mathrm{FO}_1 \equiv Aut_{wc}(\mathrm{FO}_1) \qquad \text{(over LTSs)}.$$

As a final step, Subsection 8.2 fills the last gap in Equations (13) and (14) by showing

$$Aut_w(\mathrm{FO}_1) \equiv Aut_w(\mathrm{FOE}_1)/\underline{\leftrightarrow} \qquad Aut_{wc}(\mathrm{FO}_1) \equiv Aut_{wc}(\mathrm{FOE}_1^\infty)/\underline{\leftrightarrow} \qquad \text{(over LTSs).}$$

1.4 Conference Versions and Companion Paper

This article is based on two conference papers [Facchini et al. 2013; Carreiro et al. 2014], which were based in turn on a Master's thesis [Zanasi 2012] and a Ph.D. dissertation [Carreiro 2015]. Each of the two conference papers focussed on a single expressive completeness theorem between Equations (5) and (6): Presenting both results in a mostly uniform way has required an extensive overhaul, involving the development of new pieces of theory, as in particular in Sections 4, 7, and 8. All missing proofs of the conference papers are included, and the simulation theorem for NMSO- and WMSO-automata is simplified, as it is now based on macro-states that are sets instead of relations. Moreover, we amended two technical issues with the characterisation $\mu_N\mathrm{ML} \equiv \mathrm{NMSO}/\underline{\leftrightarrow}$ presented in Facchini et al. [2013]. First, the definition of noetherian subset in NMSO has been made more precise to prevent potential misunderstandings arising with the formulation in Facchini et al. [2013]. Second, as stated in Facchini et al. [2013], the expressive completeness result was only valid on trees. In this version, we extend it to arbitrary LTSs, thanks to the new material in Section 7.

Finally, our approach depends on model-theoretic results on the three main one-step logics featuring in this article: FO_1, FOE_1, and FOE_1^∞. We believe these results to be of independent interest, and to save some space here, we decided to restrict our discussion of the model theory of these monadic predicate logics in this article to a summary. Full details can be found in the companion paper [Carreiro et al. 2018].

2 PRELIMINARIES

We assume the reader to be familiar with the syntax and (game-theoretic) semantics of the modal μ-calculus and with the automata-theoretic perspective on this logic. For background reading, we refer to Grädel et al. [2002] and Venema [2012]; the purpose of this section is to fix some notation and terminology.

2.1 Transition Systems and Trees

Throughout this article we fix a set Prop of elements that will be called *proposition letters* and denoted with small Latin letters $p, q, \ldots$. We will often focus on a finite subset $\mathrm{P} \subseteq_\omega \mathrm{Prop}$ and denote with C the set $\wp(\mathrm{P})$ of *labels* on P; it will be convenient to think of C as an *alphabet*. Given a binary relation $R \subseteq X \times Y$ for any element $x \in X$, we indicate with $R[x]$ the set $\{y \in Y \mid (x, y) \in R\}$, while R^+ and R^* are defined, respectively, as the transitive closure of R and the reflexive and transitive closure of R. The set $\mathrm{Ran}(R)$ is defined as $\bigcup_{x \in X} R[x]$.

A P-*labeled transition system* (LTS) is a tuple $\mathbb{S} = \langle T, R, \kappa, s_I\rangle$, where T is the universe or domain of $\mathbb{S}$, $\kappa : T \to \wp(\mathrm{P})$ is a colouring (or marking), $R \subseteq T^2$ is the accessibility relation, and $s_I \in T$ is a distinguished node. We call $\kappa(s)$ the colour, or type, of node $s \in T$. Observe that the colouring $\kappa : T \to \wp(\mathrm{P})$ can be seen as a valuation $\kappa^\natural : \mathrm{P} \to \wp(T)$ given by $\kappa^\natural(p) := \{s \in T \mid p \in \kappa(s)\}$. A P-*tree* is a P-labeled LTS in which every node can be reached from s_I, and every node except s_I has a unique predecessor; the distinguished node s_I is called the *root* of $\mathbb{S}$. Each node $s \in T$ uniquely defines a subtree of $\mathbb{S}$ with carrier $R^*[s]$ and root s. We denote this subtree by $\mathbb{S}.s$.

A *path* through an LTS $\mathbb{S} = \langle T, R, \kappa, s_I\rangle$ is a sequence $(s_i)_{i<\alpha}$, where α is finite but non-zero, or $\alpha = \omega$, and $(s_i, s_{i+1}) \in R$ whenever $i + 1 < \alpha$. In particular, we allow paths of the form s for any $s \in S$. A tree is called *conversely well founded* if it does not contain any infinite path.

The *tree unravelling* of an LTS $\mathbb{S}$ is given by $\hat{\mathbb{S}} := \langle T_P, R_P, \kappa', s_I\rangle$, where T_P is the set of finite paths in $\mathbb{S}$ stemming from s_I, $R_P(t, t')$ iff t' is a one-step extension of t and the colour of a path $t \in T_P$

is given by the colour of its last node in T. The *ω-unravelling* $\mathbb{S}^\omega$ of $\mathbb{S}$ is defined similarly, now taking as nodes all "generalised" paths of the form $(s_I, n_1, s_1, \ldots, n_k, s_k)$, where $n_i \in \omega$ for each i.

A *p-variant* of a transition system $\mathbb{S} = \langle T, R, \kappa, s_I \rangle$ is a $\mathsf{P} \cup \{p\}$-transition system $\langle T, R, \kappa', s_I \rangle$ such that $\kappa'(s) \setminus \{p\} = \kappa(s) \setminus \{p\}$ for all $s \in T$. Given a set $S \subseteq T$, we let $\mathbb{S}[p \mapsto S]$ denote the p-variant, where $p \in \kappa'(s)$ iff $s \in S$.

Let $\varphi \in L$ be a formula of some logic L, we use $\mathsf{Mod}_L(\varphi) = \{\mathbb{S} \mid \mathbb{S} \models \varphi\}$ to denote the class of transition systems that make φ true. The subscript L will be omitted when L is clear from context. A class C of transition systems is said to be *L-definable* if there is a formula $\varphi \in L$ such that $\mathsf{Mod}_L(\varphi) = \mathsf{C}$. We use the notation $\varphi \equiv \psi$ to mean that $\mathsf{Mod}_L(\varphi) = \mathsf{Mod}_L(\psi)$, and given two logics L, L' we use $L \equiv L'$ when the L-definable and L'-definable classes of models coincide.

2.2 Games

We introduce some terminology and background on infinite games. All the games that we consider involve two players called *Éloise* ($\exists$) and *Abelard* ($\forall$). In some contexts, we refer to a player Π to specify a a generic player in $\{\exists, \forall\}$. Given a set A, by A^* and A^ω we denote respectively the set of words (finite sequences) and streams (or infinite words) over A.

A *board game* $\mathcal{G}$ is a tuple $(G_\exists, G_\forall, E, \mathit{Win})$, where $G_\exists$ and $G_\forall$ are disjoint sets whose union $G = G_\exists \cup G_\forall$ is called the *board* of $\mathcal{G}$, $E \subseteq G \times G$ is a binary relation encoding the *admissible moves*, and $\mathit{Win} \subseteq G^\omega$ is a *winning condition*. An *initialized board game* $\mathcal{G}@u_I$ is a tuple $(G_\exists, G_\forall, u_I, E, \mathit{Win})$, where $u_I \in G$ is the *initial position* of the game. In a *parity game*, the set Win is given by a *parity function*, that is, a map $\Omega : G \to \omega$ of finite range, in the sense that a sequence $(a_i)i < \omega$ belongs to Win iff the maximal value n that is reached as $n = \Omega(a_i)$ for infinitely many i is even.

Given a board game $\mathcal{G}$, a *match* of $\mathcal{G}$ is simply a path through the graph (G, E), that is, a sequence $\pi = (u_i)_{i<\alpha}$ of elements of G, where α is either ω or a natural number, and $(u_i, u_{i+1}) \in E$ for all i with $i + 1 < \alpha$. A match of $\mathcal{G}@u_I$ is supposed to start at u_I. Given a finite match $\pi = (u_i)_{i<k}$ for some $k < \omega$, we call $\mathit{last}(\pi) := u_{k-1}$ the *last position* of the match; the player Π such that $\mathit{last}(\pi) \in G_\Pi$ is supposed to move at this position, and if $E[\mathit{last}(\pi)] = \emptyset$, then we say that Π *got stuck* in π. A match π is called *total* if it is either finite, with one of the two players getting stuck, or infinite. Matches that are not total are called *partial*. Any total match π is *won* by one of the players: If π is finite, then it is won by the opponent of the player who gets stuck. Otherwise, if π is infinite, then the winner is $\exists$ if $\pi \in \mathit{Win}$, and $\forall$ if $\pi \notin \mathit{Win}$.

Given a board game $\mathcal{G}$ and a player Π, let $\mathsf{PM}^{\mathcal{G}}_\Pi$ denote the set of partial matches of $\mathcal{G}$ whose last position belongs to player Π. A *strategy for* Π is a function $f : \mathsf{PM}^{\mathcal{G}}_\Pi \to G$. A match $\pi = (u_i)_{i<\alpha}$ of $\mathcal{G}$ is *f-guided* if for each $i < \alpha$ such that $u_i \in G_\Pi$ we have that $u_{i+1} = f(u_0, \ldots, u_i)$. Let $u \in G$ and a f be a strategy for Π. We say that f is a *surviving strategy* for Π in $\mathcal{G}@u$ if for each f-guided partial match π of $\mathcal{G}@u$, if $\mathit{last}(\pi)$ is in G_Π, then $f(\pi)$ is legitimate, that is, $(\mathit{last}(\pi), f(\pi)) \in E$. We say that f is a *winning strategy* for Π in $\mathcal{G}@u$ if, additionally, Π wins each f-guided total match of $\mathcal{G}@u$. If Π has a winning strategy for $\mathcal{G}@u$, then u is called a *winning position* for Π in $\mathcal{G}$. The set of positions of $\mathcal{G}$ that are winning for Π is denoted by $\mathit{Win}_\Pi(\mathcal{G})$.

A strategy f is called *positional* if $f(\pi) = f(\pi')$ for each $\pi, \pi' \in \mathsf{Dom}(f)$ with $\mathit{last}(\pi) = \mathit{last}(\pi')$. A board game $\mathcal{G}$ with board G is *determined* if $G = \mathit{Win}_\exists(\mathcal{G}) \cup \mathit{Win}_\forall(\mathcal{G})$, that is, each $u \in G$ is a winning position for one of the two players. The next result states that parity games are positionally determined.

Fact 2.1 ([Emerson and Jutla 1991; Mostowski 1991]). *For each parity game $\mathcal{G}$, there are positional strategies $f_\exists$ and $f_\forall$ respectively for player $\exists$ and $\forall$, such that for every position $u \in G$ there is a player Π such that f_Π is a winning strategy for Π in $\mathcal{G}@u$.*

In the sequel, we will often assume, without notification, that strategies in parity games are positional. Moreover, we think of a positional strategy f_Π for player Π as a function $f_\Pi : G_\Pi \to G$.

2.3 The Modal μ-Calculus

The language of the modal μ-calculus (μML) is given by the following grammar:

$$\varphi ::= q \mid \neg q \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu p.\varphi \mid \nu p.\varphi,$$

where $p, q \in \mathrm{Prop}$ and p is positive in φ (i.e., p is not negated). We will freely use standard syntactic concepts and notations related to this language, such as the sets $FV(\varphi)$ and $BV(\varphi)$ of *free* and *bound* variables of φ, and the collection $Sfor(\varphi)$ of subformulas of φ. We use the standard convention that no variable is both free and bound in a formula and that every bound variable is fresh. We let $\mu\mathrm{ML}(\mathsf{P})$ denote the collection of formulas φ with $FV(\varphi) \subseteq \mathsf{P}$. Sometimes we write $\psi \trianglelefteq \varphi$ to denote that ψ is a subformula of φ. For a bound variable p occurring in some formula $\varphi \in \mu\mathrm{ML}$, we use δ_p to denote the binding definition of p, that is, the unique formula such that either $\mu p.\delta_p$ or $\nu p.\delta_p$ is a subformula of φ.

We need some notation for the notion of *substitution*. Let φ and $\{\psi_z \mid z \in Z\}$ be modal fixpoint formulas, where $Z \cap BV(\varphi) = \varnothing$. Then we let $\varphi[\psi_z/z \mid z \in Z]$ denote the formula obtained from φ by simultaneously substituting each formula ψ_z for z in φ (with the usual understanding that no free variable in any of the ψ_z will get bound by doing so). In case Z is a singleton z, we will simply write $\varphi[\psi_z/z]$ or $\varphi[\psi]$ if z is clear from context.

The semantics of this language is completely standard. Let $\mathbb{S} = \langle T, R, \kappa, s_I\rangle$ be a transition system and $\varphi \in \mu\mathrm{ML}$. We inductively define the *meaning* $[\![\varphi]\!]^{\mathbb{S}}$, which includes the following clauses for the least (μ) and greatest (ν) fixpoint operators:

$$[\![\mu p.\psi]\!]^{\mathbb{S}} := \bigcap \left\{X \subseteq T \mid X \supseteq \psi_p^{\mathbb{S}}(X)\right\}$$
$$[\![\nu p.\psi]\!]^{\mathbb{S}} := \bigcup \left\{X \subseteq T \mid X \subseteq \psi_p^{\mathbb{S}}(X)\right\},$$

where the map $\psi_p^{\mathbb{S}} : \wp(T) \to \wp(T)$ represents how the meaning of ψ depends on that of p:

$$\psi_p^{\mathbb{S}}(X) := [\![\psi]\!]^{\mathbb{S}[p\mapsto X]}.$$

We say that φ is *true* in $\mathbb{S}$ (notation $\mathbb{S} \Vdash \varphi$) iff $s_I \in [\![\varphi]\!]^{\mathbb{S}}$.

We will now describe the semantics defined above in game-theoretic terms. That is, we will define the evaluation game $\mathcal{E}(\varphi, \mathbb{S})$ associated with a formula $\varphi \in \mu\mathrm{ML}$ and a transition system $\mathbb{S}$. This game is played by two players ($\exists$ and $\forall$) moving through positions (ξ, s), where $\xi \trianglelefteq \varphi$ and $s \in T$. In an arbitrary position (ξ, s) it is useful to think of $\exists$ trying to show that ξ is true at s and of $\forall$ of trying to convince her that ξ is false at s. The rules of the evaluation game are given in the following table.

Position	Player	Admissible moves
$(\psi_1 \vee \psi_2, s)$	$\exists$	$\{(\psi_1, s), (\psi_2, s)\}$
$(\psi_1 \wedge \psi_2, s)$	$\forall$	$\{(\psi_1, s), (\psi_2, s)\}$
$(\Diamond\varphi, s)$	$\exists$	$\{(\varphi, t) \mid t \in R[s]\}$
$(\Box\varphi, s)$	$\forall$	$\{(\varphi, t) \mid t \in R[s]\}$
$(\mu p.\varphi, s)$	$-$	$\{(\varphi, s)\}$
$(\nu p.\varphi, s)$	$-$	$\{(\varphi, s)\}$
(p, s) with $p \in BV(\varphi)$	$-$	$\{(\delta_p, s)\}$
$(\neg q, s)$ with $q \in FV(\varphi)$ and $q \notin \kappa(s)$	$\forall$	$\emptyset$
$(\neg q, s)$ with $q \in FV(\varphi)$ and $q \in \kappa(s)$	$\exists$	$\emptyset$
(q, s) with $q \in FV(\varphi)$ and $q \in \kappa(s)$	$\forall$	$\emptyset$
(q, s) with $q \in FV(\varphi)$ and $q \notin \kappa(s)$	$\exists$	$\emptyset$

Every finite match of this game is lost by the player that got stuck. To give a winning condition for an infinite match let p be, of the bound variables of φ that get unravelled infinitely often, the one such that δ_p the highest subformula in the syntactic tree of φ. The winner of the match is $\forall$ if p is a μ-variable and $\exists$ if p is a ν-variable. We say that φ is true in $\mathbb{S}$ iff $\exists$ has a winning strategy in $\mathcal{E}(\varphi, \mathbb{S})$.

Proposition 2.2 (Adequacy Theorem). *Let $\varphi = \varphi(p)$ be a formula of μML in which all occurrences of p are positive, $\mathbb{S}$ be a LTS and $s \in T$. Then:*

$$s \in [\![\mu p.\varphi]\!]^{\mathbb{S}} \iff (\mu p.\varphi, s) \in \mathrm{Win}_{\exists}(\mathcal{E}(\mu p.\varphi, \mathbb{S})). \tag{17}$$

2.4 Bisimulation

Bisimulation is a notion of behavioral equivalence between processes. For the case of transition systems, it is formally defined as follows.

Definition 2.3. Let $\mathbb{S} = \langle T, R, \kappa, s_I \rangle$ and $\mathbb{S}' = \langle T', R', \kappa', s'_I \rangle$ be P-labeled transition systems. A *bisimulation* is a relation $Z \subseteq T \times T'$ such that for all $(t, t') \in Z$ the following holds:

(atom). $\kappa(t) = \kappa'(t')$;
(forth). for all $s \in R[t]$ there is $s' \in R'[t']$ such that $(s, s') \in Z$;
(back). for all $s' \in R'[t']$ there is $s \in R[t]$ such that $(s, s') \in Z$.

Two pointed transition systems $\mathbb{S}$ and $\mathbb{S}'$ are *bisimilar* (denoted $\mathbb{S} \underline{\leftrightarrow} \mathbb{S}'$) if there is a bisimulation $Z \subseteq T \times T'$ containing (s_I, s'_I).

The following observation about tree unravellings is the key to understand the importance of tree models in the setting of invariance modulo bismilarity results.

Fact 2.4. *$\mathbb{S}$, $\hat{\mathbb{S}}$, and $\mathbb{S}^{\omega}$ are bisimilar for every transition system $\mathbb{S}$.*

A class C of transition systems is *bisimulation closed* if $\mathbb{S} \underline{\leftrightarrow} \mathbb{S}'$ implies that $\mathbb{S} \in \mathrm{C}$ iff $\mathbb{S}' \in \mathrm{C}$ for all $\mathbb{S}$ and $\mathbb{S}'$. A formula $\varphi \in L$ is *bisimulation-invariant* if $\mathbb{S} \underline{\leftrightarrow} \mathbb{S}'$ implies that $\mathbb{S} \Vdash \varphi$ iff $\mathbb{S}' \Vdash \varphi$ for all $\mathbb{S}$ and $\mathbb{S}'$.

Fact 2.5. *Each μML-definable class of transition systems is bisimulation closed.*

3 MONADIC SECOND-ORDER LOGICS AND MODAL μ-CALCULI

In this section, we introduce the main logics of our narrative, i.e., the weak and noetherian versions of monadic second-order logic on the one hand and the continuous and noetherian fragments of the modal μ-calculus on the other. We also briefly discuss some model-theoretic properties and results related to these logics.

3.1 Monadic Second-order Logics

Three variations of monadic second-order logic feature in our work: *standard*, *weak*, and *noetherian* monadic second-order logic; and for each of these three variations, we consider a one-sorted and a two-sorted version. As we will see later, the one-sorted version fits better in the automata-theoretic framework, whereas it is more convenient to use the two-sorted approach when translating μ-calculi to second-order languages. In both the one-sorted and the two-sorted versions, the syntax of the three languages is the same, the difference lying in the semantics, more specifically in the type of subsets over which the second-order quantifiers range. In the case of standard and weak monadic second-order logic, these quantifiers range over all, respectively, finite subsets of the model. In the case of NMSO, we need the concept of a *noetherian* subset of an LTS.

Definition 3.1. Let $\mathbb{S} = \langle T, R, \kappa, s_I \rangle$ be an LTS, and let B be a set of finite paths that all share the same starting point s; we call B a *bundle rooted at* s, or simply an s-*bundle*, if B does not contain an infinite ascending chain $\pi_0 \sqsubset \pi_1 \sqsubset \cdots$, where $\sqsubset$ denotes the (strict) initial-segment relation on paths. A *bundle* is simply an s-bundle for some $s \in T$. Finally, a subset X of T is called *noetherian* if there is a bundle B such that each $t \in X$ lies on some path in B.

Example 3.2. Let $\mathbb{S} = \langle T, R, \kappa, s_I \rangle$ be a labelled transition system.

(1) Since the empty bundle is a bundle, the empty set is a noetherian set in $\mathbb{S}$.
(2) In case $\mathbb{S}$ is a conversely well-founded tree, the set of all paths emanating from s_I is a bundle, and therefore *every* subset of T is noetherian.
(3) More generally, if $\mathbb{S}$ is an arbitrary tree, then its noetherian subsets coincide with those that are included in a well-founded subtree of $\mathbb{S}$. In case $\mathbb{S}$ is finitely branching, every well-founded subtree is finite; as a consequence, every noetherian subset is finite.
(4) Let s be some arbitrary node in $\mathbb{S}$, and suppose that the points $s_1, \ldots, s_n$ are all reachable from s (i.e., belong to the set $R^*[s]$). Then, for each i, we may fix a (finite) path π_i from s to s_i. Clearly these paths, taken together, provide a bundle, and so the set $\{s_1, \ldots, s_n\}$ is noetherian.
(5) This means in particular that every singleton is noetherian. Furthermore, if $\mathbb{S}$ is finite, and every point in $\mathbb{S}$ is reachable from s_I, then every subset of T is noetherian.
(6) Similarly, every finite subset of a tree is noetherian. Hence, on finitely branching trees, the noetherian sets coincide with the finite ones.

One-sorted monadic second-order logics.

Definition 3.3. The formulas of the *(one-sorted) monadic second-order language* are defined by the following grammar:

$$\varphi ::= \Downarrow p \mid p \sqsubseteq q \mid R(p, q) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists p.\varphi,$$

where p and q are letters from Prop. We adopt the standard convention that no proposition letter is both free and bound in φ.

As mentioned, the three logics SMSO, WMSO, and NMSO are distinguished by their semantics. Let $\mathbb{S} = \langle T, R, \kappa, s_I \rangle$ be an LTS. The interpretation of the atomic formulas is fixed:

$$\begin{array}{rll} \mathbb{S} \models \Downarrow p & \text{iff} & \kappa^\natural(p) = \{s_I\} \\ \mathbb{S} \models p \sqsubseteq q & \text{iff} & \kappa^\natural(p) \subseteq \kappa^\natural(q) \\ \mathbb{S} \models R(p, q) & \text{iff} & \text{for every } s \in \kappa^\natural(p) \text{ there exists } t \in \kappa^\natural(q) \text{ such that } sRt. \end{array}$$

Furthermore, the interpretation of the Boolean connectives is standard. The interpretation of the existential quantifier is where the logics diverge:

$$\mathbb{S} \models \exists p.\varphi \quad \text{iff} \quad \mathbb{S}[p \mapsto X] \models \varphi \begin{cases} \text{for some} & \text{(SMSO)} \\ \text{for some } \textit{finite} & \text{(WMSO)} \\ \text{for some } \textit{noetherian} & \text{(NMSO)} \end{cases} \Bigg\} \; X \subseteq T.$$

Observe that for a given monadic second-order formula φ, the classes $\mathrm{Mod}_{\mathrm{SMSO}}(\varphi)$, $\mathrm{Mod}_{\mathrm{WMSO}}(\varphi)$, and $\mathrm{Mod}_{\mathrm{NMSO}}(\varphi)$ will generally be different.

Two-sorted monadic second-order logics. The reader may have expected to see the following more standard language for second-order logic.

Definition 3.4. Given a set iVar of individual (first-order) variables, we define the formulas of the *two-sorted monadic second-order language* by the following grammar:

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

where $p \in \mathsf{Prop}$, $x, y \in \mathsf{iVar}$ and $\approx$ is the symbol for equality.

Formulas are interpreted over an LTS $\mathbb{S} = \langle T, R, \kappa, s_I \rangle$ with a variable assignment $g : \mathsf{iVar} \to T$, and the semantics of the language is completely standard. Depending on whether second-order quantification ranges over all subsets, over finite subsets, or over noetherian subsets, we obtain the three two-sorted variations denoted respectively as 2SMSO, 2WMSO, and 2NMSO.

Equivalence of one-sorted and two-sorted MSO. In each variation, the one-sorted and the two-sorted versions can be proved to be equivalent, but there is a subtlety due to the fact that our models have a distinguished state. In the one-sorted language, we use the downarrow $\Downarrow$ to access this distinguished state; in the two-sorted approach, we will use a *fixed* variable v to refer to the distinguished state, and given a formula $\varphi(v)$ of which v is the only free individual variable, we write $\mathbb{S} \models \varphi[s_I]$ rather than $\mathbb{S}[v \mapsto s_I] \models \varphi$. As a consequence, the proper counterpart of the one-sorted language SMSO is the set $2\mathrm{SMSO}(v)$ of those 2SMSO-formulas that have precisely v as their unique free variable.

More in particular, with $L \in \{\mathrm{SMSO}, \mathrm{WMSO}, \mathrm{NMSO}\}$, we say that $\varphi \in L$ is *equivalent to* $\psi(v) \in L(v)$ if

$$\mathbb{S} \models \varphi \text{ iff } \mathbb{S} \models \psi[s_I]$$

for every model $\mathbb{S} = \langle T, R, \kappa, s_I \rangle$. We can now state the equivalence between the two approaches to monadic second-order logic as follows.

Proposition 3.5. *Let $L \in \{\mathrm{SMSO}, \mathrm{WMSO}, \mathrm{NMSO}\}$ be a monadic second-order logic.*

(1) There is an effective construction transforming a formula $\varphi \in L$ into an equivalent formula $\varphi^t \in 2L(v)$.
(2) There is an effective construction transforming a formula $\psi \in 2L(v)$ into an equivalent formula $\psi^o \in L$.

Proof. Since it is completely straightforward to define a translation $(\cdot)^t$ as required for part (1) of Proposition 3.5, we only discuss the proof of part (2). The key observation here is that a single-sorted language can interpret the corresponding two-sorted language by encoding every individual variable $x \in \mathsf{iVar}$ as a set variable p_x denoting a singleton and that it is easy to write down a formula stating that a variable indeed is interpreted by a singleton. As a consequence, where $2L(\mathsf{P}, \mathsf{X})$ denotes the set of $2L$-formulas with free second-order variables in P and free first-order variables in X, it is not hard to formulate a translation $(\cdot)^m : 2L(\mathsf{P}, \mathsf{X}) \to L(\mathsf{P} \uplus \{p_x \mid x \in \mathsf{X}\})$ such that, for every model $\mathbb{S}$, every variable assignment g and every formula $\psi \in 2L(\mathsf{Prop}, \mathsf{X})$:

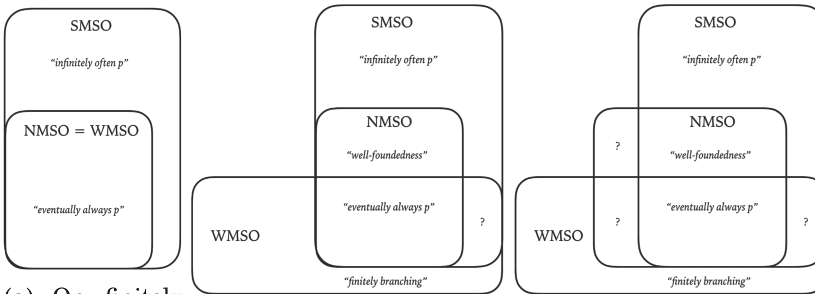
$$\mathbb{S}, g \models \psi \quad \text{iff} \quad \mathbb{S}[p_x \mapsto \{g(x)\} \mid x \in \mathsf{X}] \models \psi^m.$$

From this it is immediate that any $\psi \in 2L(v)$ satisfies

$$\mathbb{S} \models \psi[s_I] \quad \text{iff} \quad \mathbb{S} \models \exists p_v (\Downarrow p_v \wedge \psi^m),$$

so that we may take $\psi^o := \exists p_v (\Downarrow p_v \wedge \psi^m)$. □

Comparing the relative expressive power of the logics SMSO, WMSO, and NMSO on finitely trees, on arbitrary trees, and on arbitrary models, we can make the following observations.



(a) On finitely branching trees (b) On arbitrary trees (c) On arbitrary models

Fig. 1. Expressiveness of three monadic second-order logics.

Finitely branching trees. From Example 3.2(6), it follows that on this subclass of LTS, NMSO and WMSO are equivalent. They are, however, both strictly included in SMSO. First, since being a well-founded subtree is SMSO-definable, NMSO (and thus WMSO) is included in SMSO. Finally, from Rabin [1970], we know that already on binary trees the SMSO-definable property "there is a path on which p is true infinitely often" is not WMSO-definable.

Arbitrary trees. For the same reason as in the case of finitely branching trees, NMSO is strictly included in SMSO. However, WMSO is now incomparable with both NMSO and SMSO. First, it is well known that WMSO can only define properties whose topological complexity is Borel (see, e.g., ten Cate and Facchini [2011]), whereas NMSO can also define non Borel properties, such as being well founded. Second, consider the property of having a node with infinitely many successors. This property is clearly definable in WMSO but not in SMSO. This is due to the fact that on arbitrary trees every SMSO formula is equivalent to a MSO-automata and that every non empty MSO-automata recognises a finitely branching tree (see Walukiewicz [1996]). Since all WMSO-definable languages are closed under complementation, it therefore turns out that the language of finitely branching trees is WMSO-definable, but it is not SMSO-definable.

Arbitrary models. Clearly, the incomparability results on tree models carry over to the more general case; that is, on arbitrary models, WMSO is incomparable with both NMSO and SMSO, and SMSO is not included in NMSO. However, at the moment of writing, we do not know whether on arbitrary models NMSO is still included in SMSO.

These findings are summarized in Figure 1 below. Note, too, that there are many nontrivial properties that can be expressed in all three languages; as an example we mention "eventually always p," see Remark 3.13.

3.2 Some Model-theoretic Observations

Before we turn to the precise syntactic definition of the two fragments of μML that correspond to the monadic second-order logics WMSO and NMSO, we briefly discuss the fundamental *semantic* properties underlying these definitions. Note that such a connection underlies the framework of the modal μ-calculus itself: The syntactic proviso on the formation of fixpoint formulas $\mu q.\varphi$ (viz., the requirement that all occurrences of q in φ are positive) guarantees a semantic property (namely, monotonicity of the associated semantic map $\varphi_q^{\mathbb{S}}$), which is needed to use the Knaster-Tarski theorem to interpret the formula $\mu q.\varphi$.

The idea underlying the definition of the fragments μ_CML and μ_NML is to impose further conditions on the formation of fixpoint formulas $\eta q.\varphi$ to ensure that the semantic map $\varphi_q^{\mathbb{S}}$ satisfies some additional properties to be introduced now.

Definition 3.6. Let $F : \wp S \to \wp S$ be a map. We say that F is *monotone* if $F(X) \subseteq F(Y)$ whenever $X \subseteq Y$ and *continuous* if it is monotone and satisfies

$$F(X) = \bigcup\{F(Y) \mid Y \subseteq X, Y \text{ finite}\}. \tag{18}$$

In case S is the domain of a tree model, we call F *noetherian-based* if it is monotone and satisfies the following condition:

$$F(X) = \bigcup\{F(Y) \mid Y \subseteq X, Y \text{ noetherian}\}. \tag{19}$$

In words, F is continuous if it is completely determined by its action on finite sets, and a similar perspective applies to noetherian-based maps. The name "continuity" is explained by the fact that a map $F : \wp S \to \wp S$ satisfies Equation (18) iff F is continuous with respect to the *Scott topology* on the power set $\wp(S)$ of S. Scott continuity stems from *domain theory* [Abramsky and Jung 1994] and plays a fundamental role in many branches of logic and theoretical computer science that feature ordered structures.

What is of interest here is that we may apply the concepts of Definition 3.6 to *formulas*. To see how this works out for fixpoint formulas, recall the definition of the semantic map $\varphi_q^{\mathbb{S}}$ associated with a formula $\varphi \in \mu$ML and a proposition letter q.

Definition 3.7. Let $\varphi \in \mu$ML and q be a propositional variable. We say that φ *is monotone (respectively, continuous/noetherian) in* q if for every transition system $\mathbb{S}$, the map $\varphi_q^{\mathbb{S}} : \wp S \to \wp S$ is monotone (respectively, continuous/noetherian).

In fact, as part of the *model theory* of the modal μ-calculus, these semantic properties (and many more) can be given rather exact syntactic characterisations.

Definition 3.8. Given a set Q of propositional variables, we define the fragment $\mathsf{Noe}_{\mathrm{Q}}(\mu\mathrm{ML})$ of μML-formulas that are (syntactically) *noetherian* in Q by the following grammar:

$$\varphi ::= q \mid \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu p.\varphi',$$

where $q \in \mathrm{Q}$, ψ is a Q-free μML-formula and $\varphi' \in \mathsf{Noe}_{\mathrm{Q}\cup\{p\}}(\mu\mathrm{ML})$. The *co-noetherian* fragment $\mathsf{CoNoe}_{Q}(\mu\mathrm{ML})$ is defined dually by taking ν instead of μ and stating $\varphi' \in \mathsf{CoNoe}_{\mathrm{Q}\cup\{p\}}(\mu\mathrm{ML})$.

Similarly, we define the fragment of μML *continuous* in Q, denoted by $\mathsf{Con}_{\mathrm{Q}}(\mu\mathrm{ML})$, by induction in the following way:

$$\varphi ::= q \mid \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \mu p.\varphi',$$

where $q, p \in \mathrm{Q}$, ψ is a Q-free μML-formula and $\varphi' \in \mathsf{Con}_{\mathrm{Q}\cup\{p\}}(\mu\mathrm{ML})$. The *co-continuous* fragment $\mathsf{CoCon}_{Q}(\mu\mathrm{ML})$ is defined dually by taking ν instead of μ and $\Box$ instead of $\Diamond$ and stating $\varphi' \in \mathsf{CoCon}_{\mathrm{Q}\cup\{p\}}(\mu\mathrm{ML})$.

Fact 3.9 ([D'Agostino and Hollenberg 2000; Fontaine 2008; Fontaine and Venema 2018]). *The following hold, for any μML-formula formula φ, and any proposition letter q:*

(1) φ is monotone in q iff it is equivalent to a formula φ' that is positive in q;
(2) φ is continuous in q iff it is equivalent to a formula φ' in the fragment $\mathsf{Con}_q(\mu\mathrm{ML})$;
(3) φ is noetherian in q iff it is equivalent to a formula φ' in the fragment $\mathsf{Noe}_q(\mu\mathrm{ML})$.

In passing, we note that in each instance of Fact 3.9, a slightly stronger result can be proved, to the effect that it is *decidable* whether a given μ-calculus formula is monotone (respectively, continuous/noetherian) in a given proposition letter.

3.3 Fragments of the Modal μ-calculus

We are now ready for the definition of the fragments μ_NML and μ_CML. Starting with the first, note that formulas of the modal μ-calculus may be classified according to their *alternation depth*, which roughly is given as the maximal length of a chain of nested alternating least and greatest fixpoint operators [Niwiński 1986]. The *alternation-free fragment* of the modal μ-calculus (μ_NML) is usually defined as the collection of μML-formulas without nesting of least and greatest fixpoint operators. It can also be given a more standard grammatical definition as the fragment of the full language where we restrict the application of the least fixpoint operator μp to formulas that are (syntactically) noetherian in p (and apply a dual condition to the greatest fixpoint operator).

Definition 3.10. The formulas of the *alternation-free* μ-calculus μ_NML are defined by the following grammar:

$$\varphi ::= q \mid \neg q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu p.\varphi' \mid \nu p.\varphi'',$$

where $p, q \in \mathsf{Prop}$, $\varphi' \in \mu_N\mathrm{ML} \cap \mathtt{Noe}_p(\mu\mathrm{ML})$, and dually $\varphi'' \in \mu_N\mathrm{ML} \cap \mathtt{CoNoe}_p(\mu\mathrm{ML})$.

It is then immediate to verify that the above definition indeed captures exactly all formulas without alternation of least and greatest fixpoints. One may prove that a formula $\varphi \in \mu$ML belongs to the fragment μ_NML iff for all subformulas $\mu p.\psi_1$ and $\nu q.\psi_2$ it holds that p is not free in ψ_2 and q is not free in ψ_1.

Similarly, we define μ_CML to be the fragment of μML where the use of the least fixed point operator is restricted to the continuous fragment.

Definition 3.11. Formulas of the fragment μ_CML are given by

$$\varphi ::= q \mid \neg q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mu p.\varphi' \mid \nu p.\varphi'',$$

where $p, q \in \mathsf{Prop}$, $\varphi' \in \mathtt{Con}_p(\mu\mathrm{ML}) \cap \mu_C\mathrm{ML}$, and dually $\varphi'' \in \mathtt{CoCon}_p(\mu\mathrm{ML}) \cap \mu_C\mathrm{ML}$.

Characteristic about μ_CML is that in a formula $\mu p.\varphi \in \mu_C$ML, all occurrences of p in φ are *existential* in the sense that they may be in the scope of a diamond but not of a box. Furthermore, as an immediate consequence of Fact 3.9(2) we may make the following observation.

Corollary 3.12. *For every μ_CML-formula $\mu p.\varphi$, the formula φ is continuous in p.*

Finally, we consider the relative expressiveness of the fixpoint languages μ_CML, μ_NML, and μML. It is immediate from the definitions that $\mu_C\mathrm{ML} \leq \mu_N\mathrm{ML} \leq \mu\mathrm{ML}$. Both inclusions are strict:

($\mu_C\mathrm{ML} \not\geq \mu_N\mathrm{ML}$). Consider the formula $\mu x.\Box x$ in μ_NML, stating that every path starting from the distinguished node of the model is finite. As mentioned, on tree models this formula captures the property of being conversely well founded, which is known not be expressible in WMSO [ten Cate and Facchini 2011] and, hence, since $\mu_C\mathrm{ML} \leq$ WMSO, not in the continuous μ-calculus either.

($\mu_N\mathrm{ML} \not\geq \mu\mathrm{ML}$). The property "on some path, p holds infinitely often" is definable by the μML-formula $\varphi := \nu x.\mu y.((p \vee \Diamond y) \wedge \Diamond x)$. However, this property is not definable in the alternation free fragment μ_NML. This is because, for instance, on trees, every property definable in μ_NML is also recognised by both a Büchi and a co-Büchi automaton (see, e.g., Kupferman and Vardi [2003]), whereas "infinitely often p" is not [Rabin 1970].

Remark 3.13. The discussion above, concerning the property "on some path, p holds infinitely often" may be contrasted with the rather similar-looking formula $\psi := \mu x.\nu y.(\Diamond x \vee (\Diamond y \wedge p))$. This formula states that there is a path in which from a certain point on p always holds ("eventually always p"). Syntactically, ψ is neither in μ_NML (it has one alternation of fixpoints) nor *a fortiori* in μ_CML. However, it is not difficult to see that ψ is equivalent to the formula $\mu x.(\Diamond x \vee \nu y\,(\Diamond y \wedge p))$, which does belong to the continuous μ-calculus and is in particular alternation free.

4 ONE-STEP LOGICS, PARITY AUTOMATA, AND μ-CALCULI

This section introduces and studies the type of parity automata that will be used in the characterisation of WMSO and NMSO on tree models. To define these automata in a uniform way, we introduce, at a slightly higher level of abstraction, the notion of a *one-step logic*, a concept from coalgebraic modal logic [Cîrstea and Pattinson 2004], which provides a nice framework for a general approach toward the theory of automata operating on infinite objects. As salient specimens of such one-step logics, we will discuss monadic first-order logic with equality (FOE_1) and its extension with the infinity quantifier (FOE_1^∞). We then define, parametric in the language L_1 of such a one-step logic, the notions of an L_1-automaton and of a mu-calculus μL_1, and we show how various classes of L_1-automata effectively correspond to fragments of μL_1.

4.1 One-step Logics and Normal Forms

Definition 4.1. Given a finite set A of monadic predicates, a *one-step model* is a pair (D, V) consisting of a *domain* set D and a *valuation* or *interpretation* $V : A \to \wp D$. Where $B \subseteq A$, we say that $V' : A \to \wp D$ is a *B-extension of* $V : A \to \wp D$, notation $V \leq_B V'$, if $V(b) \subseteq V'(b)$ for every $b \in B$ and $V(a) = V'(a)$ for every $a \in A \setminus B$.

A *one-step language* is a map assigning to any set A a collection $L_1(A)$ of objects that we will refer to as *one-step formulas* over A. We assume that one-step languages come with a *truth* relation $\models$ between one-step formulas and models, writing $(D, V) \models \varphi$ to denote that (D, V) satisfies φ.

Note that we do allow the (unique) one-step model that is based on the empty domain; we will simply denote this model as $(\varnothing, \varnothing)$.

Our chief examples of one-step languages will be variations of modal and first-order logic.

Definition 4.2. For a set A of monadic predicates, the corresponding *basic one-step modal logic* $\mathrm{ML}_1(A)$ is the language defined as:

$$\mathrm{ML}_1(A) := \{\Diamond a, \Box a \mid a \in A\}.$$

The semantics of these formulas is given by

$$\begin{array}{lll} (D,V) \models \Diamond a & \text{iff} & V(a) \neq \varnothing \\ (D,V) \models \Box a & \text{iff} & V(a) = D. \end{array}$$

Definition 4.3. The one-step language $\mathrm{FOE}_1(A)$ of *first-order logic with equality* on a set of predicates A and individual variables iVar is given by the sentences (formulas without free variables) generated by the following grammar, where $a \in A$ and $x, y \in \mathrm{iVar}$.:

$$\varphi ::= a(x) \mid \neg a(x) \mid x \approx y \mid x \not\approx y \mid \exists x.\varphi \mid \forall x.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi. \tag{20}$$

We use FO_1 for the equality-free fragment, where we omit the clauses $x \approx y$ and $x \not\approx y$.

The interpretation of this language in a model (D, V) with $D \neq \varnothing$ is completely standard. Formulas of FO_1 and FOE_1 are interpreted inductively by augmenting the pair (D, V) with a variable assignment $g : \mathrm{iVar} \to D$. The semantics then defines the desired truth relation $(D, V), g \models \varphi$ between one-step models, assignments, and one-step formulas. As usual, the variable assignment g can and will be omitted when we are dealing with sentences—and note that we only take sentences as one-step formulas. For the interpretation in one-step models with empty domain we refer to Definition 4.5.

We now introduce an extension of first-order logic with two additional quantifiers, which first appeared in the context of Mostowski's study [Mostowski 1957] of generalised quantifiers. The first, written $\exists^\infty x.\varphi$, expresses that there exist infinitely many elements satisfying a formula φ. Its dual, written $\forall^\infty x.\varphi$, expresses that there are *at most finitely many* elements *falsifying* the formula φ. Formally:

$$\begin{array}{lll} (D,V),g \models \exists^\infty x.\varphi(x) & \text{iff} & |\{s \in D \mid (D,V),g[x \mapsto s] \models \varphi(x)\}| \geq \omega \\ (D,V),g \models \forall^\infty x.\varphi(x) & \text{iff} & |\{s \in D \mid (D,V),g[x \mapsto s] \not\models \varphi(x)\}| < \omega. \end{array} \tag{21}$$

Definition 4.4. The one-step language $\mathrm{FOE}_1^\infty(A)$ is defined by adding to the grammar (20) of $\mathrm{FOE}_1(A)$ the cases $\exists^\infty x.\varphi$ and $\forall^\infty x.\varphi$. In the case of non-empty models, the truth relation $(D,V),g \models \varphi$ is defined by extending the truth relation for $\mathrm{FOE}_1(A)$ with the clauses (21).

In the case of models with empty domain, we cannot give an inductive definition of the truth relation using variable assignments. Nevertheless, a definition of truth can be provided for formulas that are Boolean combinations of sentences of the form $Qx.\varphi$, where $Q \in \{\exists, \exists^\infty, \forall, \forall^\infty\}$ is a quantifier.

Definition 4.5. For the empty one-step model $(\varnothing,\varnothing)$, we define the truth relation as follows: For every sentence $Qx.\varphi$, where $Q \in \{\exists, \exists^\infty, \forall, \forall^\infty\}$, we set

$$\begin{array}{ll} (\varnothing,\varnothing) \not\models Qx.\varphi & \text{if } Q \in \{\exists, \exists^\infty\} \\ (\varnothing,\varnothing) \models Qx.\varphi & \text{if } Q \in \{\forall, \forall^\infty\}, \end{array}$$

and we extend this definition to arbitrary FOE_1^∞-sentences via the standard clauses for the Boolean connectives.

For various reasons, it will be convenient to assume that our one-step languages are closed under taking (Boolean) duals. Here we say that the one-step formulas φ and ψ are *Boolean duals* if for every one-step model we have $(D,V) \models \varphi$ iff $(D,V^c) \not\models \psi$, where V^c is the complement valuation given by $V^c(a) := D \setminus V(a)$ for all a.

As an example, it is easy to see that for the basic one-step modal logic ML_1 the formulas $\Diamond a$ and $\Box a$ are each other's dual. In the case of the monadic predicate logics FO_1, FOE_1, and FOE_1^∞ we can define the Boolean dual of a formula φ by a straightforward induction.

Definition 4.6. For $L_1 \in \{\mathrm{FO}_1, \mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$, we define the following operation on formulas:

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

We leave it for the reader to verify that the operation $(\cdot)^\delta$ indeed provides a Boolean dual for every one-step sentence.

The following semantic properties will be essential when studying the parity automata and μ-calculi associated with one-step languages.

Definition 4.7. Given a one-step language $L_1(A)$, $\varphi \in L_1(A)$, and $B \subseteq A$,

- φ is *monotone* in B if for all pairs of one step models (D,V) and (D,V') with $V \leq_B V'$, $(D,V) \models \varphi$ implies $(D,V'),g \models \varphi$.
- φ is *B-continuous* if φ is monotone in B and, whenever $(D,V) \models \varphi$, then there exists $V' : A \to \wp(D)$ such that $V' \leq_B V$, $(D,V') \models \varphi$ and $V'(b)$ is finite for all $b \in B$.
- φ is *B-cocontinuous* if its dual φ^δ is continuous in B.

Example 4.8. Fix a set A of monadic predicates, a subset $B \subseteq A$ and a $b \in B$.

(1) It is easy to see that a formula φ is monotone in B if all predicates from B occur only positively in φ (i.e., in the scope of an even number of negations). For simple formulas that are not monotone in b, consider for instance $\neg\exists x b(x)$ or $\forall x(a(x) \vee \neg b(x))$. However, the formula $\forall x(b(x) \vee \neg b(x))$, although it features a negative occurrence of b, is monotone in b.
(2) Typical formulas that are continuous in b are $\exists x\, b(x)$ and $\exists x_1 \exists x_2(x_1 \not\approx x_2 \wedge b(x_1) \wedge b(x_2))$. For a typical counterexample, take the formula $\forall x\, b(x)$.
(3) Particularly interesting FOE_1^∞-formulas that are continuous in B may be obtained using the abbreviated quantifier $\mathbf{W}$ given by

$$\mathbf{W}x.(\varphi, \psi) := \forall x.(\varphi(x) \vee \psi(x)) \wedge \forall^\infty x.\psi(x). \tag{22}$$

In words, $\mathbf{W}x.(\varphi, \psi)$ states that every element of the domain validates $\varphi(x)$ or $\psi(x)$, but only finitely many need to validate $\varphi(x)$. As a consequence, if φ is continuous in B, and no $b \in B$ occurs in ψ, then the formula $\mathbf{W}x.(\varphi, \psi)$ will be continuous in B. Thus $\forall^\infty$ makes a certain use of the universal quantifier compatible with the notion of continuity.

We recall from Carreiro et al. [2018] syntactic characterisations of these semantic properties, relative to the monadic predicate logics FO_1, FOE_1, and FOE_1^∞. We first discuss characterisations of monotonicity and (co)continuity given by grammars.

Definition 4.9. For $L_1 \in \{\mathrm{FO}_1, \mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$, we define the *positive* fragment of $L_1(A)$, written $L_1^+(A)$, as the set of sentences generated by the grammar we obtain by leaving out the clause $\neg a(x)$ from the grammar for L_1.

For $B \subseteq A$, the *B-continuous* fragment of $\mathrm{FOE}_1(A)$, written $\mathsf{Con}_B(\mathrm{FOE}_1(A))$, is the set of sentences generated by the following grammar:

$$\varphi ::= b(x) \mid \psi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi, \tag{23}$$

for $b \in B$ and $\psi \in \mathrm{FOE}_1^+(A \setminus B)$. When $\psi \in \mathrm{FO}_1^+(A \setminus B)$, the grammar (23) above generates the B-continuous fragment $\mathsf{Con}_B(\mathrm{FO}_1(A))$ of $\mathrm{FO}_1(A)$. Finally, the *B-continuous* fragment of $\mathrm{FOE}_1^\infty(A)$, written $\mathsf{Con}_B(\mathrm{FOE}_1^\infty(A))$, is generated as follows:

$$\varphi ::= b(x) \mid \psi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \mathbf{W}x.(\varphi, \psi), \tag{24}$$

where $b \in B$, $\psi \in (\mathrm{FOE}_1^\infty)^+(A \setminus B)$, and $\mathbf{W}x.(\varphi, \psi)$ is defined as in Equation (22).

For $L_1 \in \{\mathrm{FO}_1, \mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$ and $B \subseteq A$, the *B-cocontinuous* fragment of $L_1(A)$, written $\mathsf{CoCon}_B(L_1(A))$, is the set $\{\varphi \mid \varphi^\delta \in \mathsf{Con}_B(L_1(A))\}$.

Note that we do allow the clause $x \not\approx y$ in the positive fragments of FOE_1 and FOE_1^∞.

The following result provides syntactic characterizations for the mentioned semantics properties.

THEOREM 4.10 ([CARREIRO ET AL. 2018]). *For $L_1 \in \{\mathrm{FO}_1, \mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$, let $\varphi \in L_1(A)$ be a one-step formula. Then*

(1) $\varphi \in L_1(A)$ is A-monotone iff it is equivalent to some $\psi \in L_1^+(A)$.
(2) $\varphi \in L_1(A)$ is B-continuous iff it is equivalent to some $\psi \in \mathsf{Con}_B(L_1(A))$.
(3) $\varphi \in L_1(A)$ is B-cocontinuous iff it is equivalent to some $\psi \in \mathsf{CoCon}_B(L_1(A))$.

PROOF. The first two statements are proved in Carreiro et al. [2018]. The third one can be verified by a straightforward induction on φ. □

In some of our later proofs we need more precise information on the shape of formulas belonging to certain syntactic fragments. For this purpose we introduce normal forms for positive sentences in FO_1, FOE_1, and FOE_1^∞.

Definition 4.11. A *type* T is just a subset of A. It defines a FOE_1-formula

$$\tau_T^+(x) := \bigwedge_{a\in T} a(x).$$

Given a one-step model (D,V), $s\in D$ *witnesses* a type T if $(D,V), g[x\mapsto s]\models \tau_T^+(x)$ for some (or, equivalently, each) assignment g. The predicate $\mathrm{diff}(\overline{\mathbf{y}})$, stating that the elements $\overline{\mathbf{y}}$ are distinct, is defined as $\mathrm{diff}(y_1,\dots,y_n) := \bigwedge_{1\le m<m'\le n}(y_m \not\approx y_{m'})$.

A formula $\varphi\in\mathrm{FO}_1(A)$ is said to be in *basic form* if it is of the shape $\varphi=\bigvee\nabla^+_{\mathrm{FO}}(\Sigma,\Pi)$, where for sets $\Sigma,\Pi\subseteq A$ of types, the formula $\nabla^+_{\mathrm{FO}}(\Sigma,\Pi)\in\mathrm{FO}_1(A)$ is defined as

$$\nabla^+_{\mathrm{FO}}(\Sigma,\Pi) := \bigwedge_{S\in\Sigma}\exists x\, \tau_S^+(x)\wedge\forall z.\bigvee_{S\in\Pi}\tau_S^+(z).$$

We say that $\varphi\in\mathrm{FOE}_1(A)$ is in *basic form* if it is a disjunction of formulas of the form $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}},\Pi)$, where each disjunct is of the form

$$\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}},\Pi) := \exists\overline{\mathbf{x}}.\left(\mathrm{diff}(\overline{\mathbf{x}})\wedge\bigwedge_i\tau^+_{T_i}(x_i)\wedge\forall z.(\mathrm{diff}(\overline{\mathbf{x}},z)\to\bigvee_{S\in\Pi}\tau_S^+(z))\right),$$

such that $\overline{\mathbf{T}}=(T_1,\dots,T_k)\in\wp(A)^k$ for some k and $\Pi\subseteq\{T_1,\dots,T_k\}$.

Finally, we say that $\varphi\in\mathrm{FOE}_1^\infty(A)$ is in *basic form* if it is a disjunction of formulas of the form $\nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}},\Pi,\Sigma)$, where each disjunct is of the form

$$\begin{aligned}\nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}},\Pi,\Sigma) &:= \nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}},\Pi\cup\Sigma)\wedge\nabla^+_\infty(\Sigma)\\ \nabla^+_\infty(\Sigma) &:= \bigwedge_{S\in\Sigma}\exists^\infty y.\tau_S^+(y)\wedge\forall^\infty y.\bigvee_{S\in\Sigma}\tau_S^+(y)\end{aligned}$$

for some sets of types $\Pi,\Sigma\subseteq\wp A$ and $T_1,\dots,T_k\subseteq A$.

Intuitively, the basic FO_1-formula $\nabla^+_{\mathrm{FO}}(\Sigma,\Pi)$ simply states that every type $S\in\Sigma$ is witnessed in the model and at the same time, every element of the domain witnesses some type in Π.[3] The formula $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}},\Pi)$ says that each one-step model satisfying it admits a partition of its domain in two parts: distinct elements $t_1,\dots,t_n$ witnessing types $T_1,\dots,T_n$, and all the remaining elements witnessing some type S of Π. The formula $\nabla^+_\infty(\Sigma)$ extends the information given by $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}},\Pi\cup\Sigma)$ by saying that (1) for every type $S\in\Sigma$, there are infinitely many elements witnessing each $S\in\Sigma$ and (2) only finitely many elements do not satisfy any type in Σ.

The next theorem states that the basic formulas indeed provide normal forms.

Theorem 4.12 ([Carreiro et al. 2018]). *For each $L_1\in\{\mathrm{FO}_1,\mathrm{FOE}_1,\mathrm{FOE}_1^\infty\}$ there is an effective procedure transforming any sentence $\varphi\in L_1^+(A)$ into an equivalent sentence $\varphi^\bullet$ in basic L_1-form.*

One may use these normal forms to provide a tighter syntactic characterisation for the notion of continuity, in the cases of FO_1 and FOE_1^∞.

[3]In fact, it is not hard to show that every formula of the form $\nabla^+_{\mathrm{FO}}(\Sigma,\Pi)$ is equivalent to a disjunction of formulas of the form $\nabla^+_{\mathrm{FO}}(\Gamma,\Gamma)$, which are closely related to the disjunctive formulas that feature in the work of Janin and Walukiewicz. In principle, we could simplify our basic FO_1 form further to formulas of the form $\nabla^+_{\mathrm{FO}}(\Gamma,\Gamma)$; for the characterisation of continuity in Theorem 4.13, however, this format is not suitable.

Theorem 4.13 ([Carreiro et al. 2018]).

(1) A formula $\varphi \in \mathrm{FO}_1(A)$ is continuous in $B \subseteq A$ iff it is equivalent to a formula, effectively obtainable from φ, in the basic form $\bigvee \nabla^+_{\mathrm{FO}}(\Sigma, \Pi)$, where we require that $B \cap \bigcup \Pi = \varnothing$ for every Π.

(2) A formula $\varphi \in \mathrm{FOE}^\infty_1(A)$ is continuous in $B \subseteq A$ iff it is equivalent to a formula, effectively obtainable from φ, in the basic form $\bigvee \nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$, where we require that $B \cap \bigcup \Sigma = \varnothing$ for every Σ.

Remark 4.14. We focussed on normal form results for monotone and (co)continuous sentences, as these are the ones relevant to our study of parity automata. However, generic sentences both of FO_1, FOE_1, and FOE^∞_1 also enjoy normal form results, with the syntactic formats given by variations of the "basic form" above. The interested reader may find in Carreiro et al. [2018] a detailed overview of these results.

We finish this section with a disucssion of the notion of *separation.*

Definition 4.15. Fix a one-step language L_1, and two sets A and B with $B \subseteq A$. Given a one-step model (D, V), we say that $V : A \to \wp D$ *separates* B if $|V^{-1}(d) \cap B| \leq 1$ for every $d \in D$, where $V^{-1}(d) = \{a \in A \mid d \in V(a)\}$. A formula $\varphi \in L_1(A)$ is B-*separating* if φ is monotone in B and, whenever $(D, V) \models \varphi$, then there exists a B-separating valuation $V' : A \to \wp(D)$ such that $V' \leq_B V$ and $(D, V') \models \varphi$.

Intuitively, a formula φ is B-separating if its truth in a one-step model never requires an element of the domain to satisfy two distinct predicates in B at the same time.

Example 4.16. Let A be a set of monadic predicates and let $B = \{b_1, b_2\} \subseteq A$.

(1) A typical example of a formula that is *not* B-separating is $\exists x\, b_1 \wedge \forall y\, b_2(y)$. The point is that any one-step model satisfying this formula will have at least one element satisfying both b_1 and b_2. As another example of a formula that is not B-separating, take $\varphi := \exists x\, b_1(x) \wedge \exists x b_2(x)$. This formula is in fact easily satisfiable in a B-separating model, but if we consider a model (D, V) for φ in which there is a element d such that $V(b_1) = V(b_2) = \{d\}$, then we cannot shrink V to a B-separating valuation V' such that $(D, V') \models \varphi$.

(2) For an example of a formula that is B-separating, consider the formula $\exists x_1 \exists x_2 (x_1 \not\approx x_2 \wedge b(x_1) \wedge b(x_2))$. This example is generalised in Proposition 4.17.

We do not need a full syntactic characterisation of this notion, but the following sufficient condition is used later.

Proposition 4.17.

(1) Let $\varphi \in \mathrm{FOE}^+_1(A)$ be a formula in basic form, $\varphi = \bigvee \nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$. Then φ is B-separating if, for each disjunct, $|S \cap B| \leq 1$ for each $S \in \{T_1, \ldots, T_k\} \cup \Pi$.

(2) Let $\varphi \in \mathrm{FOE}^{\infty+}_1(A)$ be a formula in basic form, $\varphi = \bigvee \nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$. Then φ is B-separating if, for each disjunct, $|S \cap B| \leq 1$ for each $S \in \{T_1, \ldots, T_k\} \cup \Pi \cup \Sigma$.

Proof. We only discuss the case $L_1 = \mathrm{FOE}^\infty_1$: a simplification of the same argument yields the case $L_1 = \mathrm{FOE}_1$. Aassume that $(D, V) \models \varphi$ for some model (D, V). We want to construct a valuation $V' \leq_B V$ witnessing the B-separation property. First, we fix one disjunct $\psi = \nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$ of φ such that $(D, V) \models \psi$. The syntactic shape of ψ implies that (D, V) can be partitioned in three sets D_1, D_2, and D_3 as follows: D_1 contains elements $s_1, \ldots, s_k$ witnessing types $T_1, \ldots, T_k$, respectively; among the remaining elements, there are infinitely many witnessing some $S \in \Sigma$ (these form D_2) and finitely many not witnessing any $S \in \Sigma$ but each witnessing some $R \in \Pi$ (these form D_3). In other words, we have assigned to each $d \in D$ a type $S_d \in \{T_1, \ldots, T_k\} \cup \Pi \cup \Sigma$ such that d witnesses

S_d. Now consider the valuation U that we obtain by pruning V to the extent that $U(a) := V(a)$ for $a \in A \setminus B$, while $U(b) := \{d \in D \mid b \in S_d\}$. It is then easy to see that we still have $(D, U) \models \psi$, while it is obvious that U separates B and that $U \leq_B A$. Therefore ψ is B-separating and so φ is, too. □

4.2 Parity Automata

Throughout the rest of the section, we fix, next to a set P of proposition letters, a one-step language L_1, as defined in Subsection 4.1. In addition, we assume that we have isolated fragments $L_1^+(A)$, $\mathtt{Con}_B(L_1(A))$, and $\mathtt{CoCon}_B(L_1(A))$ consisting of one-step formulas in $L_1(A)$ that are respectively monotone, B-continuous, and B-co-continuous for $B \subseteq A$.

We first recall the definition of a general parity automaton, adapted to this setting.

Definition 4.18 (Parity Automata). A *parity automaton* based on the one-step language L_1 and the set P of proposition letters, or briefly: An L_1*-automaton* is a tuple $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ such that A is a finite set of states, also called the *carrier* of $\mathbb{A}$; $a_I \in A$ is the initial state; $\Delta : A \times \wp(\mathsf{P}) \to L_1^+(A)$ is the transition map; and $\Omega : A \to \mathbb{N}$ is the priority map. The class of such automata will be denoted by $Aut(L_1)$.

Acceptance of a P-transition system $\mathbb{S} = \langle T, R, \kappa, s_I \rangle$ by $\mathbb{A}$ is determined by the *acceptance game* $\mathcal{A}(\mathbb{A}, \mathbb{S})$ of $\mathbb{A}$ on $\mathbb{S}$. This is the parity game defined according to the rules of the following table.

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

$\mathbb{A}$ *accepts* $\mathbb{S}$ if $\exists$ has a winning strategy in $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, s_I)$ and *rejects* $\mathbb{S}$ if (a_I, s_I) is a winning position for $\forall$. We write $\mathsf{Mod}(\mathbb{A})$ for the class of transition systems that are accepted by $\mathbb{A}$ and $\mathsf{TMod}(\mathbb{A})$ for the class of tree models in $\mathsf{Mod}(\mathbb{A})$.

The acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$ proceeds in rounds, with each round moving from one basic position $(a, s) \in A \times T$ to the next. At such a basic position, it is $\exists$'s task to turn the set $R(s)$ of successors of s into the domain of a one-step model for the formula $\Delta(a, \kappa(s)) \in L_1(A)$. That is, she needs to come up with a valuation $V : A \to \wp(R[s])$ such that $(R[s], V) \models \Delta(a, \kappa(s))$ (and if she cannot find such a valuation, she loses immediately). One may think of the set $\{(b, t) \mid t \in V(b)\}$ as a collection of witnesses to her claim that, indeed, $(R[s], V) \models \Delta(a, \kappa(s))$. The round ends with $\forall$ picking one of these witnesses, which then becomes the basic position at the start of the next round. (Unless, of course, $\exists$ managed to satisfy the formula $\Delta(a, \kappa(s))$ with an empty set of witnesses, in which case $\forall$ gets stuck and looses immediately.)

Many properties of parity automata can already be determined at the one-step level. An important example concerns the notion of complementation, which will be used later in this section. Recall the notion of (Boolean) *dual* of a one-step formula (Definition 4.1). Following ideas from Muller and Schupp [1987] and Kissig and Venema [2009], we can use duals, together with a *role switch* between $\forall$ and $\exists$, to define a negation or complementation operation on automata.

Definition 4.19. Assume that, for some one-step language L_1, the map $(\cdot)^\delta$ provides, for each set A, a Boolean dual $\varphi^\delta \in L_1(A)$ for each $\varphi \in L_1(A)$. We define the *complement* of a given L_1-automaton $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ as the automaton $\mathbb{A}^\delta := \langle A, \Delta^\delta, \Omega^\delta, a_I \rangle$, where $\Delta^\delta(a, c) := (\Delta(a, c))^\delta$, and $\Omega^\delta(a) := 1 + \Omega(a)$ for all $a \in A$ and $c \in \wp(\mathsf{P})$.

Proposition 4.20. *Let L_1 and $(\cdot)^\delta$ be as in the previous definition. For each $\mathbb{A} \in Aut(L_1)$ and $\mathbb{S}$ we have that $\mathbb{A}^\delta$ accepts $\mathbb{S}$ if and only if $\mathbb{A}$ rejects $\mathbb{S}$.*

The proof of Proposition 4.20 is based on the fact that the *power* of $\exists$ in $\mathcal{A}(\mathbb{A}^\delta, \mathbb{S})$ is the same as that of $\forall$ in $\mathcal{A}(\mathbb{A}, \mathbb{S})$, as defined in Kissig and Venema [2009]. As an immediate consequence, one may show that if the one-step language L_1 is closed under duals, then the class $Aut(L_1)$ is closed under taking complementation. Later we will use Proposition 4.20 to show that the same may apply to some subclasses of $Aut(L_1)$.

The automata-theoretic characterisation of WMSO and NMSO will use classes of parity automata constrained by two additional properties. To formulate these, we first introduce the notion of a *cluster*.

Definition 4.21. Let L_1 be a one-step language, and let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be in $Aut(L_1)$. Write $\prec$ for the reachability relation in $\mathbb{A}$, i.e., the transitive closure of the "occurrence relation" $\{(a, b) \mid b$ occurs in $\Delta(a, c)$ for some $c \in \wp(\mathsf{P})\}$; in case $a \prec b$ we say that b is *active* in a. A *cluster* of $\mathbb{A}$ is a cell of the equivalence relation generated by the relation $\prec \cap \succ$ (i.e., the intersection of $\prec$ with its converse). A cluster is called *degenerate* if it consists of a single element that is not active in itself.

Remark 4.22. Observe that any cluster of an automaton is either degenerate or else each of its states is active in itself and in any other state of the cluster. Observe, too, that there is a natural order on clusters: We may say that one cluster is *higher* than another if each member of the second cluster if active in each member of the first. We may assume without loss of generality that the initial state belongs to the highest cluster of the automaton.

We can now formulate the mentioned requirements on L_1-automata as follows.

Definition 4.23. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be some L_1-automaton. We say that $\mathbb{A}$ is *weak* if $\Omega(a) = \Omega(b)$ whenever a and b belong to the same cluster. For the property of *continuity* we require that, for any cluster M, any state $a \in M$ and any $c \in \wp\mathsf{P}$, we have that $\Omega(a) = 1$ implies $\Delta(a, c) \in \mathsf{Con}_M(L_1(A))$ and $\Omega(a) = 0$ implies $\Delta(a, c) \in \mathsf{CoCon}_M(L_1(A))$.

We call a parity automaton $\mathbb{A} \in Aut(L_1)$ *weak-continuous* if it satisfies both properties, weakness and continuity. The classes of weak and weak-continuous automata are denoted as $Aut_w(L_1)$ and $Aut_{wc}(L_1)$, respectively.

Intuitively, weakness forbids an automaton to register non-trivial properties concerning the vertical "dimension" of input trees such as "there is a path in which infinitely many nodes satisfy p," whereas continuity expresses a constraint on how much of the horizontal "dimension" of an input tree the automaton is allowed to process. In terms of second-order logic, they correspond respectively to quantification over "vertically" finite (i.e., included in well-founded subtrees) and "horizontally" finite (i.e., included in finitely branching subtrees) sets. The conjunction of weakness and continuity thus corresponds to quantification over finite sets.

Remark 4.24. Any weak parity automaton $\mathbb{A}$ is equivalent to a special weak automaton $\mathbb{A}'$ with $\Omega : A' \to \{0, 1\}$. This is because *(weakness)* prevents states of different parity to occur infinitely often in acceptance games; so we may just replace any even priority with 0 and any odd priority with 1. We shall assume such a restricted priority map for weak parity automata.

4.3 μ-Calculi

We now see how to associate, with each one-step language L_1, the following variant μL_1 of the modal μ-calculus. These logics are of a fairly artificial nature; their main use is to smoothen the translations from automata to second-order formulas further on.

Definition 4.25. Given a one-step language L_1, we define the language μL_1 of the *μ-calculus over* L_1 by use of the following grammar:

$$\varphi ::= q \mid \neg q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc_\alpha(\varphi_1, \ldots, \varphi_n) \mid \mu p.\varphi' \mid \nu p.\varphi',$$

where $p, q \in \mathsf{Prop}$, $\alpha(a_1, \ldots, a_n) \in L_1^+$ and φ' is monotone in p.

As in the case of the modal μ-calculus μML, we will freely use standard syntactic concepts and notations related to this language.

Observe that the language μL_1 generally has a wealth of modalities: one for each one-step formula in L_1.

The semantics of this language is given as follows.

Definition 4.26. Let $\mathbb{S}$ be a transition system. The satisfaction relation $\Vdash$ is defined in the standard way, with the following clause for the modality $\bigcirc_\alpha$:

$$\mathbb{S} \Vdash \bigcirc_\alpha(\varphi_1, \dots, \varphi_n) \quad \text{iff} \quad (R[s_I], V_{\overline{\varphi}}) \models \alpha(a_1, \dots, a_n), \tag{25}$$

where $V_{\overline{\varphi}}$ is the one-step valuation given by

$$V_{\overline{\varphi}}(a_i) := \{t \in R[s_I] \mid \mathbb{S}.t \Vdash \varphi_i\}. \tag{26}$$

Example 4.27.

(1) If we identify the modalities $\bigcirc_{\Diamond a}$ and $\bigcirc_{\Box a}$ of the basic modal one-step language ML_1 (cf. Definition 4.2) with the standard $\Diamond$ and $\Box$ operators, then we may observe that $\mu(\mathrm{ML}_1)$ corresponds to the standard modal μ-calculus: $\mu(\mathrm{ML}_1) = \mu\mathrm{ML}$.

(2) Consider the one-step formulas $\alpha = \exists x(a_1(x) \land \forall y\, a_2(y))$, $\beta = \exists xy(x \not\approx y \land a_1(x) \land a_1(y))$, and $\gamma = \mathbf{W}x(a_1(x), a_2(x))$. Then $\bigcirc_\alpha(\varphi_1, \varphi_2)$ is equivalent to the modal formula $\Diamond\varphi_1 \land \Box\varphi_2$ and $\bigcirc_\beta(\varphi)$ expresses that the current state has at least two successors where φ holds. The formula $\bigcirc_\gamma(\varphi_1, \varphi_2)$ holds at a state s if all its successors satisfy φ_1 or φ_2, while at most finitely many successors refute φ_2. Neither $\bigcirc_\beta$ nor $\bigcirc_\gamma$ can be expressed in standard modal logic.

(3) If the one-step language L_1 is closed under taking disjunctions (conjunctions, respectively), then it is easy to see that $\bigcirc_{\alpha\lor\beta}(\overline{\varphi}) \equiv \bigcirc_\alpha(\overline{\varphi}) \lor \bigcirc_\beta(\overline{\varphi})$ ($\bigcirc_{\alpha\land\beta}(\overline{\varphi}) \equiv \bigcirc_\alpha(\overline{\varphi}) \land \bigcirc_\beta(\overline{\varphi})$, respectively).

Alternatively but equivalently, one may interpret the language game-theoretically.

Definition 4.28. Given a μL_1-formula φ and a model $\mathbb{S}$, we define the *evaluation game* $\mathcal{E}(\varphi, \mathbb{S})$ as the two-player infinite game whose rules are given in the next table.

Position	Player	Admissible moves
(q, s), with $q \in FV(\varphi) \cap \kappa(s)$	$\forall$	$\emptyset$
(q, s), with $q \in FV(\varphi) \setminus \kappa(s)$	$\exists$	$\emptyset$
$(\neg q, s)$, with $q \in FV(\varphi) \cap \kappa(s)$	$\exists$	$\emptyset$
$(\neg q, s)$, with $q \in FV(\varphi) \setminus \kappa(s)$	$\forall$	$\emptyset$
$(\psi_1 \lor \psi_2, s)$	$\exists$	$\{(\psi_1, s), (\psi_2, s)\}$
$(\psi_1 \land \psi_2, s)$	$\forall$	$\{(\psi_1, s), (\psi_2, s)\}$
$(\bigcirc_\alpha(\varphi_1, \dots, \varphi_n), s)$	$\exists$	$\{Z \subseteq \{\varphi_1, \dots, \varphi_n\} \times R[s] \mid (R[s], V^*_Z) \models \alpha(\overline{a})\}$
$Z \subseteq Sfor(\varphi) \times S$	$\forall$	$\{(\psi, s) \mid (\psi, s) \in Z\}$
$(\mu p.\varphi, s)$	$-$	$\{(\varphi, s)\}$
$(\nu p.\varphi, s)$	$-$	$\{(\varphi, s)\}$
(p, s), with $p \in BV(\varphi)$	$-$	$\{(\delta_p, s)\}$

For the admissible moves at a position of the form $(\bigcirc_\alpha(\varphi_1, \dots, \varphi_n), s)$, we consider the valuation $V^*_Z : \{a_1, \dots, a_n\} \to \wp(R[s])$, given by $V^*_Z(a_i) := \{t \in R[s] \mid (\varphi_i, t) \in Z\}$. The winning conditions of $\mathcal{E}(\varphi, \mathbb{S})$ are standard: $\exists$ wins those infinite matches of which the highest variable that is unfolded infinitely often during the match is a μ-variable.

The following proposition, stating the adequacy of the evaluation game for the semantics of μL_1, is formulated explicitly for future reference. We omit the proof, which is completely routine.

FACT 4.29 (ADEQUACY). *For any formula $\varphi \in \mu L_1$ and any model $\mathbb{S}$ the following equivalence holds:*

$$\mathbb{S} \Vdash \varphi \quad \textit{iff} \quad (\varphi, s_I) \textit{ is a winning position for } \exists \textit{ in } \mathcal{E}(\varphi, \mathbb{S}).$$

We will be specifically interested in two fragments of μL_1, associated with the properties of being noetherian and continuous, respectively, and with the associated variants of the μ-calculus μL_1 where the use of the fixpoint operator μ is restricted to formulas belonging to these two respective fragments.

Definition 4.30. Let Q be a set of proposition letters. We first define the fragment $\mathsf{Noe}_{\mathsf{Q}}(\mu L_1)$ of μL_1 of formulas that are syntactically *noetherian* in Q by the following grammar:

$$\varphi ::= q \mid \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc_\alpha(\varphi_1, \ldots, \varphi_n) \mid \mu p.\varphi',$$

where $q \in \mathsf{Q}$, ψ is a Q-free μML-formula, $\alpha(a_1, \ldots, a_n) \in L_1^+$ and $\varphi' \in \mathsf{Noe}_{\mathsf{Q}\cup\{p\}}(\mu L_1)$. The *co-noetherian* fragment $\mathsf{CoNoe}_Q(\mu L_1)$ is defined dually.

Similarly, we define the fragment $\mathsf{Con}_{\mathsf{Q}}(\mu L_1)$ of μL_1-formulas that are syntactically *continuous* in Q as follows:

$$\varphi ::= q \mid \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc_\alpha(\varphi_1, \ldots, \varphi_k, \psi_1, \ldots, \psi_m) \mid \mu p.\varphi',$$

where $p \in \mathsf{Prop}$, $q \in \mathsf{Q}$, ψ, ψ_i are Q-free μL_1-formula, $\alpha(a_1, \ldots, a_k, b_1, \ldots, b_m) \in \mathsf{Con}_{\overline{a}}(L_1)(\overline{a}, \overline{b})$, and $\varphi' \in \mathsf{Con}_{\mathsf{Q}\cup\{p\}}(\mu L_1)$. The *co-continuous* fragment $\mathsf{CoCon}_Q(\mu L_1)$ is defined dually.

Based on this, we can now define the mentioned variants of the μ-calculus μL_1, where the use of the least (greatest) fixpoint operator can only be applied to formulas that belong to, respectively, the noetherian (co-noetherian) and continuous (co-continuous) fragment of the language that we are defining.

Definition 4.31. The formulas of the *alternation-free* μ-calculus $\mu_N L_1$ are defined by the following grammar:

$$\varphi ::= q \mid \neg q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc_\alpha(\varphi_1, \ldots, \varphi_n) \mid \mu p.\varphi' \mid \nu p.\varphi'',$$

where $\alpha(a_1, \ldots, a_n) \in L_1^+$, $\varphi' \in \mu_N L_1 \cap \mathsf{Noe}_p(\mu L_1)$, and dually $\varphi'' \in \mu_N L_1 \cap \mathsf{CoNoe}_p(\mu L_1)$.

Similarly, the formulas of the *continuous* μ-calculus $\mu_C L_1$ are given by the grammar

$$\varphi ::= q \mid \neg q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc_\alpha(\varphi_1, \ldots, \varphi_n) \mid \mu p.\varphi' \mid \nu p.\varphi'',$$

where $\alpha(a_1, \ldots, a_n) \in L_1^+$, $\varphi' \in \mu_C L_1 \cap \mathsf{Con}_p(\mu L_1)$, and dually $\varphi'' \in \mu_C L_1 \cap \mathsf{CoCon}_p(\mu L_1)$.

Example 4.32. Following up on Example 4.27, it is easy to verify that $\mu_N\mathrm{ML}_1 = \mu_N\mathrm{ML}$ and $\mu_C\mathrm{ML}_1 = \mu_C\mathrm{ML}$.

4.4 From Automata to Fixpoint Formulas and Back

In the context of modal fixpoint logics and automata operating on (possibly) infinite objects, it is well known that there are effective translations from parity automata to fixpoint formulas and vice versa [Grädel et al. 2002]. The results that we need in this article are the following.

THEOREM 4.33. *There is an effective procedure that, given an automaton $\mathbb{A}$ in $Aut(L_1)$, returns a formula $\xi_{\mathbb{A}} \in \mu L_1$, which satisfies the following properties:*

(1) $\xi_{\mathbb{A}}$ is equivalent to $\mathbb{A}$;
(2) $\xi_{\mathbb{A}} \in \mu_N L_1$ if $\mathbb{A} \in Aut_w(L_1)$;
(3) $\xi_{\mathbb{A}} \in \mu_C L_1$ if $\mathbb{A} \in Aut_{wc}(L_1)$.

Theorem 4.34. *Let L_1 be a on-step language that is closed under taking conjunctions and disjunctions. Then there is an effective procedure that, given a formula $\xi \in \mu L_1$, returns an automaton $\mathbb{A}_\xi$ in $Aut(L_1)$, which satisfies the following properties:*

(1) $\mathbb{A}_\xi$ is equivalent to ξ;
(2) $\mathbb{A}_\xi \in Aut_w(L_1)$ if $\xi \in \mu_N L_1$;
(3) $\mathbb{A}_\xi \in Aut_{wc}(L_1)$ if $\xi \in \mu_C L_1$.

For both theorems, the first item can be proved by standard methods. To prove the second and third items, some care is needed to ensure that the translation obtained in the proof of the first item lands in the right fragment of the language and, respectively, produces an automaton of the right kind. In both cases, these proofs are not entirely trivial, but they are fairly straightforward and space-consuming. For this reason, we refer the interested reader to the technical report for the details.

5 AUTOMATA FOR WMSO

In this section, we start looking at the automata-theoretic characterisation of WMSO. That is, we introduce the following automata, corresponding to this version of monadic second-order logic; these WMSO-*automata* are the continuous-weak automata for the one-step language FOE_1^∞, cf. Definition 4.23.

Definition 5.1. A WMSO-*automaton* is an automaton in the class $Aut_{wc}(\mathrm{FOE}_1^\infty)$.

Recall that our definition of continuous-weak automata is syntactic in nature, i.e., if $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ is a WMSO-automaton, then for any pair of states a, b with $a \prec b$ and $b \prec a$, and any $c \in C$, we have $\Delta(a, c) \in \mathtt{Con}_b(\mathrm{FOE}_1^\infty(A)^+)$ if $\Omega(a)$ is odd and $\Delta(a, c) \in \mathtt{CoCon}_b(\mathrm{FOE}_1^\infty(A)^+)$ if $\Omega(a)$ is even.

The main result of this section states one direction of the automata-theoretic characterisation of WMSO.

Theorem 5.2. *There is an effective construction transforming a* WMSO-*formula φ into a* WMSO-*automaton $\mathbb{A}_\varphi$ that is equivalent to φ on the class of trees.*

The proof of Theorem 5.2, provided at the end of this section, proceeds by induction on the complexity of φ. The case $\varphi = \exists p.\psi$ requires most of the work. First, we need to define a closure operation on classes of tree models corresponding to the semantics of WMSO quantification.

Definition 5.3. Fix a set P of proposition letters, a proposition letter $p \notin P$, and a language C of $\mathrm{P} \cup \{p\}$-labeled trees. The *finitary projection* of C over p is the language of P-labeled trees defined as

$$\exists_F p.\mathrm{C} := \{\mathbb{T} \mid \text{ there is a finite } p\text{-variant } \mathbb{T}' \text{ of } \mathbb{T} \text{ with } \mathbb{T}' \in \mathrm{C}\}.$$

A collection of classes of tree models is *closed under finitary projection over* p if it contains the class $\exists_F p.\mathrm{C}$ whenever it contains the class C itself.

The case $\varphi = \exists p.\psi$ of the proof of Theorem 5.2 will require a *projection construction* that, given a WMSO-automaton $\mathbb{A}$, provides one recognising $\exists_F p.\mathsf{TMod}(\mathbb{A})$. In other words, this will prove that the collection of classes that are recognisable by WMSO-automata is closed under finitary projection. The next subsection is devoted to a preliminary result, allowing us to put WMSO-automata in a suitable shape for the projection construction.

5.1 Simulation Theorem for WMSO-automata

For SMSO-automata, the analogous projection construction (mimicking SMSO quantification) crucially uses the following *simulation theorem*: Every SMSO-automaton $\mathbb{A}$ is equivalent to a *non-deterministic* automaton $\mathbb{A}'$ [Walukiewicz 1996]. Semantically, non-determinism yields the appealing property that every node of the input model $\mathbb{T}$ is associated with at most one state of $\mathbb{A}'$ during the acceptance game, which means that we may assume $\exists$'s strategy f in $\mathcal{A}(\mathbb{A}', \mathbb{T})$ to be *functional* (cf. Definition 5.9 below). This is particularly helpful in case we want to define a p-variant of $\mathbb{T}$ that is accepted by the projection construct on $\mathbb{A}'$: Our decision whether to label a node s with p will crucially depend on the value $f(a, s)$, where a is the unique state of $\mathbb{A}'$ that is associated with s. Now, in the case of WMSO-automata, we are interested in guessing *finitary* p-variants, which requires f to be functional only on a *finite* set of nodes. Thus, the idea of our simulation theorem is to turn a WMSO-automaton $\mathbb{A}$ into an equivalent device $\mathbb{A}^F$ that consists of an initial, non-deterministic part, together with a final part that is a copy of the (generally alternating) automaton $\mathbb{A}$ itself; by tweaking the transition and priority function, we can then make sure that $\mathbb{A}^F$ behaves non-deterministically on a *finite* part of any accepted tree.

For SMSO-automata, the simulation theorem is based on a powerset construction: If the starting automaton has set of states A, then the resulting non-deterministic automaton is based on "macro-states" from the set $\wp A$. Analogously, for WMSO-automata, we will associate the non-deterministic behaviour with macro-states. However, as explained above, the automaton $\mathbb{A}^F$ that we construct has to be non-deterministic just on finitely many nodes of the input and may behave as $\mathbb{A}$ (i.e., in "alternating mode") on the others. To this aim, $\mathbb{A}^F$ will be "two-sorted," roughly consisting of a copy of $\mathbb{A}$ (with set of states A) together with a variant of its powerset construction, based both on A and $\wp A$. For any accepted $\mathbb{T}$, the idea is to make any match π of $\mathcal{A}(\mathbb{A}^F, \mathbb{T})$ consist of two parts:

(*Non-deterministic mode*). For finitely many rounds π is played on macro-states, i.e., positions belong to the set $\wp A \times T$. In her strategy, player $\exists$ assigns macro-states (from $\wp A$) only to *finitely many* nodes and states (from A) to the rest. Also, her strategy is functional in $\wp A$, i.e., it assigns *at most one macro-state* to each node.

(*Alternating mode*). At a certain round, π abandons macro-states and turns into a match of the game $\mathcal{A}(\mathbb{A}, \mathbb{T})$, i.e., all subsequent positions are from $A \times T$ (and are played according to a not necessarily functional strategy).

Therefore, successful runs of $\mathbb{A}^N$ will have the property of processing only a *finite* amount of the input with $\mathbb{A}^N$ being in a macro-state and all the rest with $\mathbb{A}^N$ behaving exactly as $\mathbb{A}$. We now proceed in steps toward the construction of $\mathbb{A}^N$. First, recall from Definition 4.11 that an *A-type* is just a subset of A. We now define a notion of liftings for sets of types, which is instrumental in translating the transition function from states on macro-states.

Definition 5.4. The *lifting* of a type $S \in \wp A$ is defined as the following $\wp A$-type:

$$S^{\Uparrow} := \begin{cases} \{S\} & \text{if } S \neq \varnothing \\ \varnothing & \text{if } S = \varnothing. \end{cases}$$

This definition is extended to sets of A-types by putting $\Sigma^{\Uparrow} := \{S^{\Uparrow} \mid S \in \Sigma\}$.

The distinction between empty and non-empty elements of Σ is to ensure that the empty type on A is lifted to the empty type on $\wp A$. Notice that the resulting set $\Sigma^{\Uparrow}$ is either empty or contains exactly one $\wp A$-type. This property is important for functionality, see below.

Next we define a translation on the sentences associated with the transition function of the original WMSO-automaton. Following the intuition given above, we want to work with sentences that can be made true by assigning macro-states (from $\wp A$) to finitely many nodes in the model and ordinary states (from A) to all the other nodes. Moreover, each node should be associated with *at most one* macro-state because of functionality. These desiderata are expressed for one-step formulas as *$\wp A$-continuity* and *$\wp A$-separability*, see the Definitions 4.7 and 4.15. For the language FOE_1^∞, Theorem 4.12 and Proposition 4.17 guarantee these properties when formulas are in a certain syntactic shape. The next definition will provide formulas that conform to this particular shape.

Definition 5.5. Let $\varphi \in \mathrm{FOE}_1^{\infty+}(A)$ be a formula of shape $\nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$ for some $\Pi, \Sigma \subseteq \wp A$ and $\overline{\mathbf{T}} = \{T_1, \dots, T_k\} \subseteq \wp A$. We define $\varphi^F \in \mathrm{FOE}_1^{\infty+}(A \cup \wp A)$ as the formula $\nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}^{\Uparrow}, \Pi^{\Uparrow} \cup \Sigma^{\Uparrow}, \Sigma)$, which means

$$\varphi^F := \exists \overline{\mathbf{x}}.\left(\mathrm{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_{0 \le i \le n} \tau^+_{T_i^{\Uparrow}}(x_i) \wedge \forall z.(\mathrm{diff}(\overline{\mathbf{x}}, z) \to \bigvee_{S \in \Pi^{\Uparrow} \cup \Sigma^{\Uparrow} \cup \Sigma} \tau^+_S(z))\right) \wedge \bigwedge_{P \in \Sigma} \exists^\infty y.\tau^+_P(y) \wedge \forall^\infty y. \bigvee_{P \in \Sigma} \tau^+_P(y). \tag{27}$$

We combine the previous definitions to form the transition function for macro-states.

Definition 5.6. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be a WMSO-automaton. Fix $c \in C$ and $Q \in \wp A$. By Theorem 4.12 for some $\Pi, \Sigma \subseteq \wp A$ and $T_i \subseteq A$, there is a sentence $\Psi_{Q,c} \in \mathrm{FOE}_1^{\infty+}(A)$ in the basic form $\bigvee \nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$ such that $\bigwedge_{a \in Q} \Delta(a, c) \equiv \Psi_{Q,c}$. By definition $\Psi_{Q,c}$ is of the form $\bigvee_i \varphi_i$, with each φ_i of shape $\nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$. We put $\Delta^\sharp(Q, c) := \bigvee_i \varphi_i^F$, where the translation $(-)^F$ is given as in Definition 5.5. Observe that $\Delta^\sharp(Q, c)$ is of type $\mathrm{FOE}_1^{\infty+}(A \cup \wp A)$.

We have now all the ingredients to define our two-sorted automaton.

Definition 5.7. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be a WMSO-automaton. We define the *finitary construct over* $\mathbb{A}$ as the automaton $\mathbb{A}^F = \langle A^F, \Delta^F, \Omega^F, a_I^F \rangle$ given by

$$\begin{array}{lll} A^F := A \cup \wp A & \Omega^F(a) := \Omega(a) & \Delta^F(a, c) := \Delta(a, c) \\ a_I^F := \{a_I\} & \Omega^F(R) := 1 & \Delta^F(Q, c) := \Delta^\sharp(Q, c) \vee \bigwedge_{a \in Q} \Delta(a, c). \end{array}$$

Remark 5.8. In the standard powerset construction of non-deterministic parity automata (Walukiewicz [2002], see also Venema [2012] and Arnold and Niwiński [2001]), macro-states are required to be *relations* rather than sets to determine whether a run through macro-states is accepting. This is not needed in our construction: Macro-states will never be visited infinitely often in accepting runs, and thus they may simply be assigned the priority 1.

The idea behind this definition is that $\mathbb{A}^F$ is enforced to process only a finite portion of any accepted tree while in the non-deterministic mode. This is encoded in game-theoretic terms through the notion of functional and finitary strategy.

Definition 5.9. Given a WMSO-automaton $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ and transition system $\mathbb{T}$, a strategy f for $\exists$ in $\mathcal{A}(\mathbb{A}, \mathbb{T})$ is *functional in* $B \subseteq A$ (or simply functional, if $B = A$) if for each node s in $\mathbb{T}$ there is at most one $b \in B$ such that (b, s) is a reachable position in an f-guided match. Also f is *finitary* in B if there are only finitely many nodes s in $\mathbb{T}$ for which a position (b, s) with $b \in B$ is reachable in an f-guided match.

The next proposition establishes the desired properties of the finitary construct.

Theorem 5.10 (Simulation Theorem for WMSO-automata). *Let $\mathbb{A}$ be a WMSO-automaton and $\mathbb{A}^F$ its finitary construct.*

(1) $\mathbb{A}^F$ *is a* WMSO-*automaton.*
(2) *For any tree model* $\mathbb{T}$, *if* (a_I^F, s_I) *is a winning position for* $\exists$ *in* $\mathcal{A}(\mathbb{A}^F, \mathbb{T})$ *, then she has a winning strategy that is both functional and finitary in* $\wp A$.
(3) $\mathbb{A} \equiv \mathbb{A}^F$.

Proof.

(1) Observe that any cluster of $\mathbb{A}^F$ involves states of exactly one sort, either A or $\wp A$. For clusters on sort A, weakness and continuity of $\mathbb{A}^F$ follow by the same properties of $\mathbb{A}$. For clusters on sort $\wp A$, weakness follows by observing that all macro-states in $\mathbb{A}^F$ have the same priority. Concerning continuity, by definition of Δ^F any macro-state can only appear inside a formula of the form $\varphi^F = \nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}^{\Uparrow}, \Pi^{\Uparrow} \cup \Sigma^{\Uparrow}, \Sigma)$ as in Equation (27). Because $\wp A \cap \bigcup \Sigma = \varnothing$, by Theorem 4.13 φ^F is continuous in each $Q \in \wp A$.

(2) Let f be a (positional) winning strategy for $\exists$ in $\mathcal{A}(\mathbb{A}^F, \mathbb{T})@(a_I^F, s_I)$. We define a strategy f' for $\exists$ in the same game as follows:

(a) On basic positions of the form $(a, s) \in A \times T$, let $V : A \to \wp R[s]$ be the valuation suggested by f. We let the valuation suggested by f' be the restriction V' of V to A. Observe that, as no predicate from $A^F \setminus A = \wp A$ occurs in $\Delta^F(a, \kappa(s)) = \Delta(a, \kappa(s))$, then V' also makes that sentence true in $R[s]$.

(b) For winning positions of the form $(R, s) \in \wp A \times T$, let $V_{R,s} : (\wp A \cup A) \to \wp R[s]$ be the valuation suggested by f. As f is winning, $\Delta^F(R, \kappa(s))$ is true in the model $V_{R,s}$. If this is because the disjunct $\bigwedge_{a \in R} \Delta(a, \kappa(s))$ is made true, then we can let f' suggest the restriction to A of $V_{R,s}$, for the same reason as in (a).

Otherwise, the disjunct $\Delta^\sharp(R, \kappa(s)) = \bigvee_i \varphi_i^F$ is made true. This means that for some i, $(R[s], V_{R,s}) \models \varphi_i^F$. Now, by construction of φ_i^F as in Equation (27), we have $\wp A \cap \bigcup \Sigma = \varnothing$. By Theorem 4.13, this implies that φ_i^F is continuous in $\wp A$. Thus we have a restriction $V'_{R,s}$ of $V_{R,s}$ that verifies φ_i^F and assigns only finitely many nodes to predicates from $\wp A$. Moreover, by construction of φ_i^F, for each $S \in \{T_1^{\Uparrow}, \ldots, T_k^{\Uparrow}\} \in \Pi^{\Uparrow} \cup \Sigma^{\Uparrow}$, S contains at most one element from $\wp A$. Thus, by Proposition 4.17, φ_i^F is $\wp A$-separable. But then we may find a separating valuation $V''_{R,s} \leq_{\wp A} V''_{R,s}$ such that $V''_{R,s}$ verifies φ_i^F. Separation means that $V''_{R,s}$ associates with each node at most one predicate from $\wp A$, and the fact that $V''_{R,s} \leq_{\wp A} V''_{R,s}$, combined with the $\wp A$-continuity of $V'_{R,s}$ ensures $\wp A$-continuity of $V''_{R,s}$. In this case, we let f' suggest $V''_{R,s}$ at position (R, s).

The strategy f' defined as above is immediately seen to be surviving for $\exists$. It is also winning, since at every basic winning position for $\exists$, the set of possible next basic positions offered by f' is a subset of those offered by f. By this observation, it also follows that any f'-guided match visits basic positions of the form $(R, s) \in \wp A \times C$ only finitely many times, as those have odd parity. By definition, the valuation suggested by f' only assigns finitely many nodes to predicates in $\wp A$ from positions of that shape, and no nodes from other positions. It follows that f' is finitary in $\wp A$. Functionality in $\wp A$ also follows immediately by definition of f'.

(3) For the direction from left to right, it is immediate by definition of $\mathbb{A}^F$ that a winning strategy for $\exists$ in $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{T})@(a_I, s_I)$ is also winning for $\exists$ in $\mathcal{G}^F = \mathcal{A}(\mathbb{A}^F, \mathbb{T})@(a_I^F, s_I)$.

For the direction from right to left, let f be a winning strategy for $\exists$ in $\mathcal{G}^F$. The idea is to define a strategy f' for $\exists$ in stages, while playing a match π' in $\mathcal{G}$. In parallel to π', a shadow match π in $\mathcal{G}^F$ is maintained, where $\exists$ plays according to the strategy f. For each round z_i, we want to keep the following relation between the two matches:

Either (1) positions of the form $(Q,s) \in \wp A \times T$ and $(a,s) \in A \times T$ occur respectively in π and π', with $a \in Q$, or (2) the same position of the form $(a,s) \in A \times T$ occurs in both matches.

(‡)

The key observation is that, because f is winning, a basic position of the form $(Q,s) \in \wp A \times T$ can occur only for finitely many initial rounds $z_0, \ldots, z_n$ that are played in π, whereas for all successive rounds $z_n, z_{n+1}, \ldots$ only basic positions of the form $(a,s) \in A \times T$ are encountered. Indeed, if this was not the case, then either $\exists$ would get stuck or the highest priority occurring infinitely often would be odd, since states from $\wp A$ all have priority 1.

It follows that enforcing a relation between the two matches as in (‡) suffices to prove that the defined strategy f' is winning for $\exists$ in π'. For this purpose, first observe that (‡).1 holds at the initial round, where the positions visited in π' and π are, respectively, $(a_I, s_I) \in A \times T$ and $(\{a_I\}, s_I) \in A^F \times T$. Inductively, consider any round z_i that is played in π' and π, respectively, with basic positions $(a,s) \in A \times T$ and $(q,s) \in A^F \times T$. To define the suggestion of f' in π', we distinguish two cases.

— First suppose that (q,s) is of the form $(Q,s) \in \wp A \times T$. By (‡) we can assume that a is in Q. Let $V_{Q,s} : A^F \to \wp(R[s])$ be the valuation suggested by f, verifying the sentence $\Delta^F(Q, \kappa(s))$. We distinguish two further cases, depending on which disjunct of $\Delta^F(Q, \kappa(s))$ is made true by $V_{Q,s}$.
  (i) If $(R[s], V_{Q,s}) \models \bigwedge_{b \in Q} \Delta(b, \kappa(s))$, then we let $\exists$ pick the restriction to A of the valuation $V_{Q,s}$.
  (ii) If $(R[s], V_{Q,s}) \models \Delta^\sharp(Q, \kappa(s))$, then we let $\exists$ pick a valuation $V_{a,s} : A \to \wp(R[s])$ defined by putting, for each $b \in A$:

$$V_{a,s}(b) := \bigcup_{b \in Q'} \{t \in R[s] \mid t \in V_{Q,s}(Q')\} \cup \{t \in R[s] \mid t \in V_{Q,s}(b)\}.$$

  It can be readily checked that the suggested move is legitimate for $\exists$ in π, i.e., it makes $\Delta(a, \kappa(s))$ true in $R[s]$.

  For case (ii), observe that the nodes assigned to b by $V_{Q,s}$ have to be assigned to b also by $V_{a,s}$, as they may be necessary to fulfill the condition, expressed with $\exists^\infty$ and $\forall^\infty$ in $\Delta^\sharp$, that infinitely many nodes witness (or that finitely many nodes do not witness) some type.

  We now show that (‡) holds at round z_{i+1}. If (i) is the case, then any next position $(b,t) \in A \times T$ picked by player $\forall$ in π' is also available for $\forall$ in π, and we end up in case (‡.2). Suppose instead that (ii) is the case. Given a move $(b,t) \in A \times T$ by $\forall$, by definition of $V_{a,s}$ there are two possibilities. First, (b,t) is also an available choice for $\forall$ in π, and we end up in case (‡.2) as before. Otherwise, there is some $Q' \in \wp A$ such that b is in Q' and $\forall$ can choose (Q', t) in the shadow match π. By letting π advance at round z_{i+1} with such a move, we are able to maintain (‡.1) also in z_{i+1}.

— In the remaining case, inductively we are given the same basic position $(a,s) \in A \times T$ both in π and in π'. The valuation V suggested by f in π verifies $\Delta^F(a, \kappa(s)) = \Delta(a, \kappa(s))$, and thus we can let the restriction of V to A be the valuation chosen by $\exists$ in the match π'. It is immediate that any next move of $\forall$ in π' can be mirrored by the same move in

π, meaning that we are able to maintain the same position—whence the relation (‡.1)—also in the next round.

In both cases, the suggestion of strategy f' was a legitimate move for $\exists$ maintaining the relation (‡) between the two matches for any next round z_{i+1}. It follows that f' is a winning strategy for $\exists$ in $\mathcal{G}$. □

5.2 From Formulas to Automata

In this subsection, we conclude the proof of Theorem 5.2. We first focus on the case of projection with respect to finite sets, which exploits our simulation result, Theorem 5.10. The next definition yields a projection construction for WMSO-automata.

Definition 5.11. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be a WMSO-automaton on alphabet $\wp(\mathsf{P} \cup \{p\})$, and $\mathbb{A}^F = \langle A^F, \Delta^F, \Omega^F, a_I^F \rangle$ the finitary construct over $\mathbb{A}$, as given in Definition 5.7. Recall that $A^F = A \cup \wp A$. We define the automaton $\exists p.\mathbb{A} = \langle A^F, \Delta^\exists, \Omega, a_I \rangle$ on alphabet $\wp\mathsf{P}$ by putting, for each $a \in A$ and $Q \in \wp A$,

$$\Delta^\exists(a, c) \;:=\; \Delta^F(a, c) \qquad\qquad \Delta^\exists(Q, c) \;:=\; \Delta^F(Q, c) \vee \Delta^F(Q, c \cup \{p\}).$$

The automaton $\exists p.\mathbb{A}$ is called the *finitary projection construct of* $\mathbb{A}$ *over* p.

Lemma 5.12. *Let* $\mathbb{A}$ *be a* WMSO-*automaton on alphabet* $\wp(\mathsf{P} \cup \{p\})$. *Then* $\mathbb{A}$ *is a* WMSO-*automaton on alphabet* $\wp\mathsf{P}$ *satisfying*

$$\mathsf{TMod}(\exists p.\mathbb{A}) \equiv \exists_F p.\mathsf{TMod}(\mathbb{A}).$$

Proof. By definition, we need to show that for any tree $\mathbb{T} = \langle T, R, \kappa : \mathsf{P} \to \wp T, s_I \rangle$:

$$\exists p.\mathbb{A} \text{ accepts } \mathbb{T} \text{ iff there is a finite } p\text{-variant } \mathbb{T}' \text{of } \mathbb{T} \text{ such that } \mathbb{A} \text{ accepts } \mathbb{T}'.$$

For the direction from left to right, by the equivalence between $\mathbb{A}$ and $\mathbb{A}^F$ it suffices to show that if $\exists p.\mathbb{A}$ accepts $\mathbb{T}$ then there is a finite p-variant $\mathbb{T}'$ of $\mathbb{T}$ such that $\mathbb{A}^F$ accepts $\mathbb{T}'$. First, we first observe that the properties stated by Theorem 5.10, which hold for $\mathbb{A}^F$ by assumption, by construction hold for $\exists p.\mathbb{A}$ as well. Thus we can assume that the given winning strategy f for $\exists$ in $\mathcal{G}_\exists = \mathcal{A}(\exists_F p.\mathbb{A}^F, \mathbb{T})@(a_I^F, s_I)$ is functional and finitary in $\wp A$. Functionality allows us to associate with each node s either none or a unique state $Q_s \in \wp A$ such that (Q_s, s) is winning for $\exists$. We now want to isolate the nodes that f treats "as if they were labeled with p." For this purpose, let V_s be the valuation suggested by f from a position $(Q_s, s) \in \wp A \times T$. As f is winning, V_s makes $\Delta^\exists(Q, \kappa(s))$ true in $R[s]$. We define a p-variant $\mathbb{T}' = \langle T, R, \kappa' : \mathsf{P} \cup \{p\} \to \wp T, s_I \rangle$ of $\mathbb{T}$ by defining $\kappa' := \kappa[p \mapsto X_p]$, that is, by colouring with p all nodes in the following set:

$$X_p := \{s \in T \mid (R[s], V_s) \models \Delta^F(Q_s, \kappa(s) \cup \{p\})\}. \tag{28}$$

The fact that f is finitary in $\wp A$ guarantees that X_p is finite, whence $\mathbb{T}'$ is a finite p-variant. It remains to show that $\mathbb{A}^F$ accepts $\mathbb{T}'$: We claim that f itself is winning for $\exists$ in $\mathcal{G} = (\mathbb{A}^F, \mathbb{T}')@(a_I, s_I)$. To see that, let us construct in stages an f-guided match π of $\mathcal{G}$ and an f-guided shadow match $\tilde{\pi}$ of $\mathcal{G}_\exists$. The inductive hypothesis we want to bring from one round to the next is that the same basic position occurs in both matches, as this suffices to prove that f is winning for $\exists$ in $\mathcal{G}$.

First, we consider the case of a basic position $(Q, s) \in A^F \times T$, where $Q \in \wp A$. By assumption f provides a valuation V_s that makes $\Delta^\exists(Q, \kappa(s))$ true in $R[s]$. Thus, V_s verifies either $\Delta^F(Q, \kappa(s))$ or $\Delta^F(Q, \kappa(s) \cup \{p\})$. Now, the match π^F is played on the p-variant $\mathbb{T}'$, where the labeling $\kappa'(s)$ is decided by the membership of s to X_p. According to Equation (28), if V_s verifies $\Delta^F(Q, \kappa(s) \cup \{p\})$, then s is in X_p, meaning that it is labeled with p in $\mathbb{T}'$, i.e., $\kappa'(s) = \kappa(s) \cup \{p\}$. Therefore V_s also verifies $\Delta^F(Q, \kappa'(s))$ and it is a legitimate move for $\exists$ in match π^F. In the remaining case, V_s verifies $\Delta^F(Q, \kappa(s))$ but falsifies $\Delta^F(Q, \kappa(s) \cup \{p\})$, implying by definition that s is not in X_p. This means that s is not labeled with p in $\mathbb{T}'$, i.e., $\kappa'(s) = \kappa(s)$. Thus again V_s verifies $\Delta^F(Q, \kappa'(s))$ and it is a legitimate move for $\exists$ in match π^F.

It remains to consider the case of a basic position $(a, s) \in A^F \times T$ with $a \in A$ a state. By definition $\Delta^{\exists}(a, \kappa(s))$ is just $\Delta^F(a, \kappa(s))$. As (a, s) is winning, we can assume that no position (Q, s) with Q a macro-state is winning according to the same f, as making $\Delta^{\exists}$-sentences true never forces $\exists$ to mark a node both with a state and a macro-state. Therefore, s is not in X_p either, meaning that it it is not labeled with p in the p-variant $\mathbb{T}'$ and thus $\kappa'(s) = \kappa(s)$. This implies that f makes $\Delta^F(a, \kappa'(s)) = \Delta^F(a, \kappa(s))$ true in $R[s]$ and its suggestion is a legitimate move for $\exists$ in match π^F. To conclude the proof, observe that for all positions that we consider the same valuation is suggested to $\exists$ in both games: This means that any next position that is picked by player $\forall$ in π^F is also available for $\forall$ in the shadow match $\tilde{\pi}$.

We now show the direction right to left of the statement. Let $\mathbb{T}'$ be a finite p-variant of $\mathbb{T}$, with labeling function κ', and g a winning strategy for $\exists$ in $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{T}')@(a_I, s_I)$. Our goal is to define a strategy g' for $\exists$ in $\mathcal{G}_{\exists}$. g' will be constructed in stages, while playing a match π' in $\mathcal{G}_{\exists}$. In parallel to π', a *bundle* $\mathcal{B}$ of g-guided shadow matches in $\mathcal{G}$ is maintained, with the following condition enforced for each round z_i:

(1) If the current basic position in π' is of the form $(Q, s) \in \wp A \times T$, then for each $a \in Q$ there is an g-guided (partial) shadow match π_a at basic position $(a, s) \in A \times T$ in the current bundle $\mathcal{B}_i$. Also, either $\mathbb{T}'_s$ is not p-free (i.e., it does contain a node s' with $p \in \kappa'(s')$) or s has some sibling t such that $\mathbb{T}'_t$ is not p-free.
(2) Otherwise, the current basic position in π' is of the form $(a, s) \in A \times T$ and $\mathbb{T}'_s$ is p-free. Also, the bundle $\mathcal{B}_i$ only consists of a single g-guided match π_a whose current basic position is also (a, s). (‡)

We recall the idea behind (‡). Point (‡.1) describes the part of match π' where it is still possible to encounter nodes that are labeled with p in $\mathbb{T}'$. As $\Delta^{\exists}$ only takes the letter p into account when defined on macro-states in $\wp A$, we want π' to visit only positions of the form $(Q, s) \in \wp A \times T$ in that situation. Anytime we visit such a position (Q, s) in π', the role of the bundle is to provide one g-guided shadow match at position (a, s) for each $a \in Q$. Then g' is defined in terms of what g suggests from those positions.

Point (‡.2) describes how we want the match π' to be played on a p-free subtree: As any node that one might encounter has the same label in $\mathbb{T}$ and $\mathbb{T}'$, it is safe to let $\exists_F p.\mathbb{A}^F$ behave as $\mathbb{A}$ in such situation. Provided that the two matches visit the same basic positions, of the form $(a, s) \in A \times T$, we can let g' just copy g.

The key observation is that, as $\mathbb{T}'$ is a *finite* p-variant of $\mathbb{T}$, nodes labeled with p are reachable only for finitely many rounds of π'. This means that, provided that (‡) hold at each round, (‡.1) will describe an initial segment of π', whereas (‡.2) will describe the remaining part. Thus our proof that g' is a winning strategy for $\exists$ in $\mathcal{G}_{\exists}$ is concluded by showing that (‡) holds for each stage of construction of π' and $\mathcal{B}$.

For this purpose, we initialize π' from position $(a_I^{\sharp}, s) \in \wp A \times T$ and the bundle $\mathcal{B}$ as $\mathcal{B}_0 = \{\pi_{a_I}\}$, with π_{a_I} the partial g-guided match consisting only of the position $(a_I, s) \in A \times T$. The situation described by (‡.1) holds at the initial stage of the construction. Inductively, suppose that at round z_i we are given a position $(q, s) \in A^F \times T$ in π^F and a bundle $\mathcal{B}_i$ as in (‡). To show that (‡) can be maintained at round z_{i+1}, we distinguish two cases, corresponding, respectively, to situation (‡.1) and (‡.2) holding at round z_i.

(A) If (q, s) is of the form $(Q, s) \in \wp A \times T$, then by inductive hypothesis we are given g-guided shadow matches $\{\pi_a\}_{a \in Q}$ in $\mathcal{B}_i$. For each match π_a in the bundle, we are provided with a valuation $V_{a,s} : A \to \wp(R[s])$ making $\Delta(a, \kappa'(s))$ true. Then we further distinguish the following two cases.

(i) Suppose first that $\mathbb{T}'_s$ is not p-free. We let the suggestion $V' : A^F \to \wp(R[s])$ of g' from position (Q, s) be defined as follows:

$$V'(q') := \begin{cases} \bigcap\limits_{\substack{b \in q', \\ a \in Q}} \{t \in R[s] \mid t \in V_{a,s}(b)\} & q' \in \wp A \\ \bigcup\limits_{a \in Q} \{t \in R[s] \mid t \in V_{a,s}(q') \text{ and } \mathbb{T}'.t \text{ is } p\text{-free}\} & q' \in A \end{cases}.$$

The definition of V' on $q' \in \wp A$ is standard (cf. Zanasi [2012, Prop. 2.21]) and guarantees a correspondence between the states assigned by the valuations $\{V_{a,s}\}_{a \in Q}$ and the macro-states assigned by V'. The definition of V' on $q' \in A$ aims at fulfilling the conditions, expressed via $\exists^\infty$ and $\forall^\infty$, on the number of nodes in $R[s]$ witnessing (or not) some A-types. Those conditions are the ones that $\Delta^\sharp(Q, \kappa'(s))$—and thus also $\Delta^F(Q, \kappa'(s))$—"inherits" by $\bigwedge_{a \in R} \Delta(a, \kappa'(s))$, by definition of $\Delta^\sharp$. Notice that we restrict $V'(q')$ to the nodes $t \in V_{a,s}(q')$ such that $\mathbb{T}'.t$ is p-free. As $\mathbb{T}'$ is a *finite* p-variant, only *finitely many* nodes in $V_{a,s}(q')$ will not have this property. Therefore their exclusion, which is crucial for maintaining condition (‡) (cf. case (a) below), does not influence the fulfilling of the cardinality conditions expressed via $\exists^\infty$ and $\forall^\infty$ in $\Delta^\sharp(Q, \kappa'(s))$.

On the base of these observations, one can check that V' makes $\Delta^\sharp(Q, \kappa'(s))$—and thus also $\Delta^F(Q, \kappa'(s))$—true in $R[s]$. In fact, to be a legitimate move for $\exists$ in π', V' should make $\Delta^\exists(Q, \kappa(s))$ true: This is the case for $\Delta^F(Q, \kappa'(s))$ is either equal to $\Delta^F(Q, \kappa(s))$, if $p \notin \kappa'(s)$, or to $\Delta^F(Q, \kappa(s) \cup \{p\})$ otherwise. To check that, we can maintain (‡) and let $(q', t) \in A^F \times T$ be any next position picked by $\forall$ in π' at round z_{i+1}. As before, we distinguish two cases:

(a) If q' is in A, then, by definition of V', $\forall$ can choose (q', t) in some shadow match π_a in the bundle $\mathcal{B}_i$. We dismiss the bundle—i.e., make it a singleton—and bring only π_a to the next round in the same position (q', t). Observe that, by definition of V', $\mathbb{T}'.t$ is p-free and thus (‡.2) holds at round z_{i+1}.

(b) Otherwise, q' is in $\wp A$. The new bundle $\mathcal{B}_{i+1}$ is given in terms of the bundle $\mathcal{B}_i$: For each $\pi_a \in \mathcal{B}_i$ with $a \in Q$, we determine for some $b \in q'$ the position (b, t) is a legitimate move for $\forall$ at round z_{i+1}; if so, then we bring π_a to round z_{i+1} at position (b, t) and put the resulting (partial) shadow match π_b in $\mathcal{B}_{i+1}$. Observe that if $\forall$ is able to pick such position (q', t) in π', then by definition of V' the new bundle $\mathcal{B}_{i+1}$ is non-empty and consists of a g-guided (partial) shadow match π_b for each $b \in q'$. In this way, we are able to keep condition (‡.1) at round z_{i+1}.

(ii) Let us now consider the case in which $\mathbb{T}'_s$ is p-free. We let g' suggest the valuation V' that assigns to each node $t \in R[s]$ all states in $\bigcup_{a \in Q}\{b \in A \mid t \in V_{a,s}(b)\}$. It can be checked that V' makes $\bigwedge_{a \in Q} \Delta(a, \kappa'(s))$—and then also $\Delta^F(Q, \kappa'(s))$—true in $R[s]$. As $p \notin \kappa(s) = \kappa'(s)$, it follows that V' also makes $\Delta^\exists(Q, \kappa(s))$ true, whence it is a legitimate choice for $\exists$ in π'. Any next basic position picked by $\forall$ in π' is of the form $(b, t) \in A \times T$, and thus condition (‡.2) holds at round z_{i+1} as shown in (i.a).

(B) In the remaining case, (q, s) is of the form $(a, s) \in A \times T$ and by inductive hypothesis we are given with a bundle $\mathcal{B}_i$ consisting of a single f-guided (partial) shadow match

π_a at the same position (a, s). Let $V_{a,s}$ be the suggestion of $\exists$ from position (a, s) in π_a. Since by assumption s is p-free, we have that $\kappa'(s) = \kappa(s)$, meaning that $\Delta^{\exists}(a, \kappa(s))$ is just $\Delta(a, \kappa(s)) = \Delta(a, \kappa'(s))$. Thus the restriction V' of V to A makes $\Delta(a, \kappa'(t))$ true, and we let it be the choice for $\exists$ in $\tilde{\pi}$. It follows that any next move made by $\forall$ in $\tilde{\pi}$ can be mirrored by $\forall$ in the shadow match π_a. □

5.2.1 Closure under Boolean Operations. Here we show that the collection of *Aut*(WMSO)-recognizable classes of tree models is closed under the Boolean operations. For union, we use the following result, leaving the straightforward proof as an exercise to the reader.

Lemma 5.13. *Let $\mathbb{A}_0$ and $\mathbb{A}_1$ be* WMSO*-automata. Then there is a* WMSO*-automaton $\mathbb{A}$ such that* TMod($\mathbb{A}$) *is the union of* TMod($\mathbb{A}_0$) *and* TMod($\mathbb{A}_1$).

For closure under complementation, we reuse the general results established in Section 4 for parity automata.

Lemma 5.14. *Let $\mathbb{A}$ be an* WMSO*-automaton. Then the automaton $\overline{\mathbb{A}}$ defined in Definition 4.19 is a* WMSO*-automaton recognizing the complement of* TMod($\mathbb{A}$).

Proof. It suffices to check that Proposition 4.20 restricts to the class $Aut_{wc}(\text{FOE}_1^{\infty})$ of WMSO-automata. First, the fact that FOE_1^{∞} is closed under Boolean duals (Definition 4.6) implies that it holds for the class $Aut(\text{FOE}_1^{\infty})$. It then remains to check that the dual automata construction $\overline{(\cdot)}$ preserves weakness and continuity. But this is straightforward, given the self-dual nature of these properties. □

We are now finally able to conclude the direction from formulas to automata of the characterisation theorem.

Proof of Theorem 5.2. The proof is by induction on φ.

- For the base case, we consider the atomic formulas $\Downarrow p$, $p \sqsubseteq q$ and $R(p, q)$.
  - The WMSO-automaton $\mathbb{A}_{\Downarrow p} = \langle A, \Delta, \Omega, a_I \rangle$ is given by putting

$$A := \{a_0, a_1\} \qquad a_I := a_0 \qquad \Omega(a_0) := 0 \qquad \Omega(a_1) := 0$$
$$\Delta(a_0, c) := \begin{cases} \forall x.a_1(x) \text{ if } p \in c \\ \bot \qquad\quad\;\; \text{otherwise} \end{cases} \qquad \Delta(a_1, c) := \begin{cases} \forall x.a_1(x) \text{ if } p \notin c \\ \bot \qquad\quad\;\; \text{otherwise} \end{cases}.$$

  - The WMSO-automaton $\mathbb{A}_{p \sqsubseteq q} = \langle A, \Delta, \Omega, a_I \rangle$ is given by $A := \{a\}$, $a_I := a$, $\Omega(a) := 0$ and $\Delta(a, c) := \forall x\, a(x)$ if $p \notin c$ or $q \in c$, and $\Delta(a, c) := \bot$ otherwise.
  - The WMSO-automaton $\mathbb{A}_{R(p,q)} = \langle A, \Delta, \Omega, a_I \rangle$ is given below:

$$A := \{a_0, a_1\} \qquad a_I := a_0 \qquad \Omega(a_0) := 0 \qquad \Omega(a_1) := 1$$
$$\Delta(a_0, c) := \begin{cases} \exists x.a_1(x) \land \forall y.a_0(y) \text{ if } p \in c \\ \forall x\ (a_0(x)) \qquad\qquad\quad \text{otherwise} \end{cases} \qquad \Delta(a_1, c) := \begin{cases} \top \text{ if } q \in c \\ \bot \text{ otherwise} \end{cases}.$$

- For the Boolean cases, where $\varphi = \psi_1 \lor \psi_2$ or $\varphi = \neg\psi$ we refer to the Boolean closure properties that we just established in the Lemmas 5.13 and 5.14, respectively.
- The case $\varphi = \exists p.\psi$ follows by the following chain of equivalences, where $\mathbb{A}_\psi$ is given by the inductive hypothesis and $\exists_F p.\mathbb{A}_\psi$ is constructed according to Definition 5.11:

$$\begin{aligned} \exists_F p.\mathbb{A}_\psi \text{ accepts } \mathbb{T} &\text{ iff } \mathbb{A}_\psi \text{ accepts } \mathbb{T}[p \mapsto X], \text{ for some } X \subseteq_\omega T && \text{(Lemma 5.12)} \\ &\text{ iff } \mathbb{T}[p \mapsto X] \models \psi, \text{ for some } X \subseteq_\omega T && \text{(induction hyp.)} \\ &\text{ iff } \mathbb{T} \models \exists p.\psi && \text{(semantics WMSO)} \end{aligned}$$

□

6 AUTOMATA FOR NMSO

In this section, we introduce the automata that capture NMSO. These will be the weak automata associated with the one-step language FOE_1^∞, cf. Definition 4.23.

Definition 6.1. An NMSO-*automaton* is an automaton in the class $Aut_w(\mathrm{FOE}_1^\infty)$.

Analogous to the previous section, our main goal here is to construct an equivalent NMSO-automaton for each NMSO-formula.

THEOREM 6.2. *There is an effective construction transforming an* NMSO-*formula* φ *into an* NMSO-*automaton* $\mathbb{A}_\varphi$ *that is equivalent to* φ *on the class of trees.*

The proof for Theorem 6.2 will closely follow the steps for proving the analogous result for WMSO (Theorem 5.2). Again, the crux of the matter is to show that the collection of classes of tree models that are recognisable by some NMSO-automaton is closed under the relevant notion of projection. Where this was finitary projection for WMSO (Definition 5.3), the notion mimicking NMSO-quantification is *noetherian* projection.

Definition 6.3. Given a set P of proposition letters, $p \notin P$ and a class C of $\mathrm{P} \cup \{p\}$-labeled trees, we define the *noetherian projection* of C over p as the language of P-labeled trees given as

$$\exists_N p.\mathrm{C} := \{\mathbb{T} \mid \text{there is a noetherian } p\text{-variant } \mathbb{T}' \text{ of } \mathbb{T} \text{ with } \mathbb{T}' \in \mathrm{C}\}.$$

A collection of classes of tree models is *closed under noetherian projection over* p if it contains the class $\exists_N p.\mathrm{C}$ whenever it contains the class C itself.

6.1 Simulation Theorem for NMSO-automata

Just as for WMSO-automata, also for NMSO-automata the projection construction will rely on a simulation theorem, constructing a two-sorted automaton $\mathbb{A}^N$ consisting of a copy of the original automaton, based on states A, and a variation of its powerset construction, based on macro-states $\wp A$. For any accepted $\mathbb{T}$, we want any match π of $\mathcal{A}(\mathbb{A}^N, \mathbb{T})$ to split in two parts:

(*Non-deterministic mode*). For finitely many rounds π is played on macro-states, i.e., positions are of the form $\wp A \times T$. The strategy of player $\exists$ is functional in $\wp A$, i.e., it assigns *at most one macro-state* to each node.

(*Alternating mode*). At a certain round, π abandons macro-states and turns into a match of the game $\mathcal{A}(\mathbb{A}, \mathbb{T})$, i.e., all next positions are from $A \times T$ (and are played according to a non-necessarily functional strategy).

The only difference with the two-sorted construction for WMSO-automata is that, in the non-deterministic mode, the cardinality of nodes to which $\exists$'s strategy assigns macro-states is irrelevant. Indeed, NMSO's finiteness is only on the vertical dimension: Assigning an odd priority to macro-states will suffice to guarantee that the non-deterministic mode processes just a well-founded portion of any accepted tree.

We now proceed in steps toward the construction of $\mathbb{A}^N$. First, the following lifting from states to macro-states parallels Definition 5.5, but for the one-step language FOE_1 proper of NMSO-automata. It is based on the basic form for FOE_1-formulas, see Definition 4.11.

Definition 6.4. Let $\varphi \in {\mathrm{FOE}_1}^+(A)$ be of shape $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$ for some $\Pi \subseteq \wp A$ and $\overline{\mathbf{T}} = \{T_1, \ldots, T_k\} \subseteq \wp A$. We define φ^N as $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}^{\Uparrow}, \Pi^{\Uparrow}) \in {\mathrm{FOE}_1}^+(\wp A)$, that means

$$\varphi^N \ := \ \exists \overline{\mathbf{x}}.\left(\mathrm{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_{0 \le i \le n} \tau^+_{T_i^{\Uparrow}}(x_i) \wedge \forall z.(\mathrm{diff}(\overline{\mathbf{x}}, z) \to \bigvee_{S \in \Pi^{\Uparrow}} \tau^+_S(z))\right). \tag{29}$$

It is instructive to compare (29) with its WMSO-counterpart (27): The difference is that, because the quantifiers $\exists^\infty$ and $\forall^\infty$ are missing, the sentence does not impose any cardinality requirement but only enforces $\wp A$-separability—cf. Section 4.1.

Lemma 6.5. *Let $\varphi \in {\mathrm{FOE}_1}^+(A)$ and $\varphi^N \in {\mathrm{FOE}_1}^+(\wp A)$ be as in Definition 6.4. Then φ^N is separating in $\wp A$.*

Proof. Each element of $\overline{\mathbf{T}}^{\Uparrow}$ and $\Pi^{\Uparrow}$ is by definition either the empty set or a singleton $\{Q\}$ for some $Q \in \wp A$. Then the statement follows from Proposition 4.17. □

We are now ready to define the transition function for macro-states. The following adapts Definition 5.6 to the one-step language FOE_1 of NMSO-automata and its normal form result, Theorem 4.12.

Definition 6.6. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be an NMSO-automaton. Fix any $c \in C$ and $Q \in \wp A$. By Theorem 4.12 there is a sentence $\Psi_{Q,c} \in {\mathrm{FOE}_1}^+(A)$ in the basic form $\bigvee \nabla_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$, for some $\Pi \subseteq \wp A$ and $T_i \subseteq A$, such that $\bigwedge_{a \in Q} \Delta(a, c) \equiv \Psi_{Q,c}$. By definition, $\Psi_{Q,c} = \bigvee_n \varphi_n$, with each φ_n of shape $\nabla_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Pi)$. We put $\Delta^\flat(Q, c) := \bigvee_n \varphi_n^N \in {\mathrm{FOE}_1}^+(\wp A)$, where the translation $(\cdot)^N$ is as in Definition 6.4.

We now have all the ingredients for the two-sorted construction over NMSO-automata.

Definition 6.7. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be an NMSO-automaton. We define the *noetherian construct over* $\mathbb{A}$ as the automaton $\mathbb{A}^N = \langle A^N, \Delta^N, \Omega^N, a_I^N \rangle$ given by

$$\begin{array}{llll} A^N := A \cup \wp A & \Omega^N(a) := \Omega(a) & \Delta^N(a,c) & := \Delta(a,c) \\ a_I^N := \{a_I\} & \Omega^N(R) := 1 & \Delta^N(Q,c) & := \Delta^\flat(Q,c) \vee \bigwedge_{a \in Q} \Delta(a,c). \end{array}$$

The construction is the same as the one for WMSO-automata (Definition 5.7) but for the definition of the transition function for macro-states, which is now free of any cardinality requirement.

Definition 6.8. We say that a strategy f in an acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{T})$ is *noetherian* in $B \subseteq A$ when in any f-guided match there can be only finitely many rounds played at a position of shape (q, s) with $q \in B$.

Theorem 6.9 (Simulation Theorem for NMSO-automata). *Let $\mathbb{A}$ be an NMSO-automaton and $\mathbb{A}^N$ its noetherian construct.*

(1) $\mathbb{A}^N$ is an NMSO-automaton.
(2) For any $\mathbb{T}$, if $\exists$ has a winning strategy in $\mathcal{A}(\mathbb{A}^N, \mathbb{T})$ from position (a_I^N, s_I) then she has one that is functional in $\wp A$ and noetherian in $\wp A$.
(3) $\mathbb{A} \equiv \mathbb{A}^N$.

Proof. The proof follows the same steps as the one of Proposition 5.10, minus all the concerns about continuity of the constructed automaton and any associated winning strategy f being finitary. One still has to show that f is noetherian in $\wp A$ ("vertically finitary"), but this is enforced by macro-states having an odd parity: Visiting one of them infinitely often would mean $\exists$'s loss. □

Remark 6.10. As mentioned, the class $Aut(\mathrm{FOE}_1)$ of automata characterising SMSO [Janin and Walukiewicz 1996] also enjoys a simulation theorem [Walukiewicz 1996], turning any automaton into an equivalent non-deterministic one. Given that the class $Aut_w(\mathrm{FOE}_1)$ only differs for the weakness constraint, one may wonder whether the simulation result for $Aut(\mathrm{FOE}_1)$ could not actually be restricted to $Aut_w(\mathrm{FOE}_1)$, making our two-sorted construction redundant. This is actually not the case: Not only does Walukiewicz's simulation theorem [Walukiewicz 1996] fail to preserve the weakness constraint, but even without this failure our purposes would not be served: A fully non-deterministic automaton is instrumental in guessing a p-variant of any accepted tree, but it does not guarantee that the p-variant is also noetherian, as the two-sorted construct does.

6.2 From Formulas to Automata

We can now conclude one direction of the automata characterisation of NMSO. Analogously to the case of WMSO-automata, we can define the *noeatherian projection construct* of an NMSO-automaton $\mathbb{A}$, on alphabet $\wp(\mathsf{P} \cup \{p\})$, as an NMSO-automaton $\exists p.\mathbb{A}$, on alphabet $\wp\mathsf{P}$: The only difference with Definition 5.11 is that $\exists p.\mathbb{A}$ is based on the *noetherian* construct $\mathbb{A}^N$ over $\mathbb{A}$.

Lemma 6.11. *For each* NMSO-*automaton* $\mathbb{A}$ *on alphabet* $\wp(\mathsf{P} \cup \{p\})$,

$$\mathsf{TMod}(\exists p.\mathbb{A}) \equiv \exists_N p.\mathsf{TMod}(\mathbb{A}).$$

Proof. The argument is the same as for WMSO-automata (Lemma 5.12). As in that proof, the inclusion from left to right relies on the simulation result (Theorem 6.9): $\exists p.\mathbb{A}$ is two-sorted and its non-deterministic mode can be used to guess a noetherian p-variant of any accepted tree. □

Proof of Theorem 6.2. As for its WMSO-counterpart Theorem 5.2, the proof is by induction on $\varphi \in$ NMSO. The Boolean inductive cases are handled by the NMSO-versions of Lemma 5.13 and 5.14. The projection case follows from Lemma 6.11. □

7 FIXPOINT OPERATORS AND SECOND-ORDER QUANTIFIERS

In this section, we will show how to translate some of the mu-calculi that we encountered until now into the appropriate second-order logics. Given the equivalence between automata and fixpoint logics that we established in Section 4, and the embeddings of WMSO and NMSO into, respectively, the automata classes $Aut_{wc}(\mathrm{FOE}_1^\infty)$ and $Aut_w(\mathrm{FOE}_1)$ that we provided in the Sections 5 and 6 for the class of tree models, the results here provide the missing link in the automata-theoretic characterizations of the monadic second-order logics WMSO and NMSO:

$$\begin{aligned} \mu_C(\mathrm{FOE}_1^\infty) &\equiv \mathrm{WMSO} && \text{(over the class of all tree models)} \\ \mu_N(\mathrm{FOE}_1) &\equiv \mathrm{NMSO} && \text{(over the class of all tree models).} \end{aligned}$$

7.1 Translating μ-calculi into Second-order Logics

More specifically, our aim in this section is to prove the following result.

Theorem 7.1.

(1) *There is an effective translation* $(\cdot)^* : \mu_N\mathrm{FOE}_1 \to$ NMSO *such that* $\varphi \equiv \varphi^*$ *for every* $\varphi \in \mu_N\mathrm{FOE}_1$*; that is:*

$$\mu_N\mathrm{FOE}_1 \leq \mathrm{NMSO}.$$

(2) *There is an effective translation* $(\cdot)^* : \mu_C\mathrm{FOE}_1^\infty \to$ WMSO *such that* $\varphi \equiv \varphi^*$ *for every* $\varphi \in \mu_C\mathrm{FOE}_1^\infty$*; that is:*

$$\mu_C\mathrm{FOE}_1^\infty \leq \mathrm{WMSO}.$$

Two immediate observations on this Theorem are in order. First, note that we use the same notation $(\cdot)^*$ for both translations; this should not cause any confusion, since the maps agree on formulas belonging to their common domain. Consequently, in the remainder, we will speak of a single translation $(\cdot)^*$. Second, as the target language of the translation $(\cdot)^*$ we will take the *two-sorted* version of second-order logic, as discussed in Section 3.1, and thus we will need Proposition 3.5 to obtain the result as formulated in Theorem 7.1, that is, for the one-sorted versions of MSO. We reserve a fixed individual variable v for this target language, i.e., every formula of the form φ^* will have this v as its unique free variable; the equivalence $\varphi \equiv \varphi^*$ is to be understood accordingly.

The translation $(\cdot)^*$ will be defined by a straightforward induction on the complexity of fixpoint formulas. The two clauses of this definition that deserve some special attention are the ones related to the fixpoint operators and the modalities.

Fixpoint operators. It is important to realise that our clause for the fixpoint operators *differs* from the one used in the standard inductive translation $(\cdot)^s$ of μML into standard MSO, where we would inductively translate $(\mu p.\varphi)^*$ as

$$\forall p\,(\forall w\,(\varphi^*[w/v] \to p(w)) \to p(v)), \tag{30}$$

which states that v belongs to any prefixpoint of φ with respect to p. To understand the problem with this translation in the current context, suppose, for instance, that we want to translate some continuous μ-calculus into WMSO. Then the formula in Equation (30) expresses that v belongs to the intersection of all *finite* prefixpoints of φ, whereas the least fixpoint is identical to the intersection of *all* prefixpoints. As a result, Equation (30) does not give the right translation for the formula $\mu p.\varphi$ into WMSO.

To overcome this problem, we will prove that least fixpoints in restricted calculi like $\mu_N\mathrm{FOE}_1$, $\mu_C\mathrm{FOE}_1^\infty$ and many others in fact satisfy a rather special property, which enables an alternative translation. We need the following definition to formulate this property.

Definition 7.2. Let $F : \wp(S) \to \wp(S)$ be a functional; for a given $X \subseteq S$, we define the *restricted* map $F_{\restriction_X} : \wp(S) \to \wp(S)$ by putting $F_{\restriction_X}(Y) := FY \cap X$. In case F is monotone, we will use $\mathit{LFP}.F$ to denote its least fixpoint.

The observations formulated in the proposition below provide the crucial insight underlying our embedding of various alternation-free and continuous μ-calculi into, respectively, NMSO and WMSO.

Proposition 7.3. *Let $\mathbb{S}$ be an LTS, and let r be a point in $\mathbb{S}$.*

(1) For any formula φ with $\mu p.\varphi \in \mu_N\mathrm{FOE}_1$ we have

$$r \in [\![\mu p.\varphi]\!]^{\mathbb{S}} \text{ iff there is a noetherian set } X \text{ such that } r \in \mathit{LFP}.(\varphi_p^{\mathbb{S}})_{\restriction_X}. \tag{31}$$

(2) For any formula φ with $\mu p.\varphi \in \mu_C\mathrm{FOE}_1^\infty$ we have

$$r \in [\![\mu p.\varphi]\!]^{\mathbb{S}} \text{ iff there is a finite set } X \text{ such that } r \in \mathit{LFP}.(\varphi_p^{\mathbb{S}})_{\restriction_X}. \tag{32}$$

Remark 7.4. In fact, the statements in Proposition 7.3 can be generalised to the setting of a fixpoint logic μL_1 associated with an arbitrary one-step language L_1.

The right-to-left direction of both Equations (31) and (32) follow from the following, more general, statement, which can be proved by a routine argument.

Proposition 7.5. *Let $F, G : \wp(S) \to \wp(S)$ be monotone, and assume that $F(Y) \subseteq G(Y)$ for every $Y \in \wp(S)$. Then $\mathit{LFP}.F \subseteq \mathit{LFP}.G$.*

The left-to-right direction of Equations (31) and (32) will be proved in the next two subsections. Note that in the continuous case we will in fact prove a slightly stronger result, which applies to *arbitrary* continuous functionals.

The point of Proposition 7.3 is that it naturally suggests the following translation for the least fixpoint operator, as a subtle but important variation of Equation (30):

$$(\mu p.\varphi)^* := \exists q\left(\forall p \subseteq q.\,\left(p \in \mathit{PRE}((\varphi_p^{\mathbb{S}})_{\restriction_q}) \to p(v)\right)\right), \tag{33}$$

where $p \in PRE((\varphi^{\mathbb{S}}_p)_{\upharpoonright_q})$ expresses that $p \subseteq q$ is a prefixpoint of the map $(\varphi^{\mathbb{S}}_p)_{\upharpoonright_q}$, that is,

$$p \in PRE((\varphi^{\mathbb{S}}_p)_{\upharpoonright_q}) := \forall w\,((q(w) \wedge \varphi^*[w/v]) \to p(w)).$$

Modalities. Finally, before we can give the definition of the translation $(\cdot)^*$, we briefly discuss the clause involving the modalities. Here we need to understand the role of the *one-step formulas* in the translation. First, an auxiliary definition.

Definition 7.6. Let $\mathbb{S} = \langle T, R, \kappa, s_I\rangle$ be a P-LTS, let A be a set of new variables, and let $V : A \to \wp(X)$ be a valuation on a subset $X \subseteq T$. The $(\mathsf{P} \cup A)$-LTS $\mathbb{S}^V := \langle T, R, \kappa^V, s_I\rangle$, given by defining the marking $\kappa^V : T \to \wp(\mathsf{P} \cup A)$, where

$$\kappa^V(s) := \begin{cases} \kappa(s) & \text{if } s \notin X \\ \kappa(s) \cup \{a \in A \mid s \in V(a)\} & \text{else} \end{cases},$$

is called the V*-expansion* of $\mathbb{S}$.

The following proposition states that at the one-step level, the formulas that provide the semantics of the modalities of $\mu\mathrm{FOE}_1$ and $\mu\mathrm{FOE}^\infty_1$ can indeed be translated into, respectively, NMSO and WMSO.

Proposition 7.7. *There is a translation* $(\cdot)^\dagger : \mathrm{FOE}^\infty_1(A) \to \mathrm{WMSO}$ *such that for every model* $\mathbb{S}$ *and every valuation* $V : A \to \wp(R[s_I])$*:*

$$(R[s_I], V) \models \alpha \textit{ iff } \mathbb{S}^V \models \alpha^\dagger[s_I].$$

Moreover, $(\cdot)^\dagger$ *restricts to first-order logic, i.e.,* $\alpha^\dagger$ *is a first-order formula if* $\alpha \in \mathrm{FOE}_1$.

Proof. Basically, the translation $(\cdot)^\dagger$ *restricts* all quantifiers to the collection of successors of v. In other words, $(\cdot)^\dagger$ is the identity on basic formulas, as it commutes with the propositional connectives, and for the quantifiers $\exists$ and $\exists^\infty$ we define:

$$\begin{array}{ll} (\exists x\,\alpha)^\dagger & := \exists x\,(Rvx \wedge \alpha^\dagger) \\ (\exists^\infty x\,\alpha)^\dagger & := \forall p \exists x\,(Rvx \wedge \neg p(x) \wedge \alpha^\dagger). \end{array}$$

We leave it for the reader to verify the correctness of this definition—observe that the clause for the infinity quantifier $\exists^\infty$ is based on the equivalence between WMSO and FOE^∞, established by Väänänen [1977]. □

We are now ready to define the translation used in the main result of this section.

Definition 7.8. By an induction on the complexity of formulas we define the following translation $(\cdot)^*$ from $\mu\mathrm{FOE}^\infty$-formulas to formulas of monadic second-order logic:

$$\begin{array}{ll} p^* & := p(v) \\ (\neg\varphi)^* & := \neg\varphi^* \\ (\varphi \vee \psi)^* & := \varphi^* \vee \psi^* \\ (\bigcirc_\alpha(\overline{\varphi}))^* & := \alpha^\dagger[\varphi^*_i/a_i \mid i \in I], \end{array}$$

where $\alpha^\dagger$ is as in Proposition 7.7 and $[\varphi^*_i/a_i \mid i \in I]$ is the substitution that replaces every occurrence of an atomic formula of the form $a_i(x)$ with the formula $\varphi^*_i(x)$ (i.e., the formula φ^*_i but with the free variable v substituted by x).

Finally, the inductive clause for a formula of the form $\mu p.\varphi$ is given as in Equation (33).

Proof of Theorem 7.1. First, it is clear that in both cases the translation $(\cdot)^*$ lands in the correct language. For both parts of the theorem, we thence prove that $(\cdot)^*$ is truth preserving by a straightforward formula induction. For example, for part (2) we need to show that, for an arbitrary formula $\varphi \in \mu_C\mathrm{FOE}_1^\infty$ and an arbitrary model $\mathbb{S}$:

$$\mathbb{S} \Vdash \varphi \text{ iff } \mathbb{S} \models \varphi^*[s_I]. \tag{34}$$

As discussed in the main text, the two critical cases concern the inductive steps for the modalities and the least fixpoint operators. Let $L_1^+ \in \{\mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$. We start verifying the case of modalities. Hence, consider the formula $\bigcirc_\alpha(\varphi_1, \ldots, \varphi_n)$ with $\alpha(a_1, \ldots, a_n) \in L_1^+$. By induction hypothesis, $\varphi_\ell \equiv \varphi_\ell^*$ for $\ell = 1, \ldots, n$. Now, let $\mathbb{S}$ be a transition system. We have that

$$\begin{aligned}
\mathbb{S} \Vdash \bigcirc_\alpha(\varphi_1, \ldots, \varphi_n) &\text{ iff } (R[s_I], V_{\overline{\varphi}}) \models \alpha(a_1, \ldots, a_n) && \text{(by (25))}\\
&\text{iff } \mathbb{S}^{V_{\overline{\varphi}}} \models \alpha^\dagger[s_I] && \text{(by Prop. 7.7)}\\
&\text{iff } \mathbb{S} \models \alpha^\dagger[\varphi_i^*/a_i \mid i \in I][s_I] && \text{(by Equation (26), Definition 7.6 and IH)}
\end{aligned}$$

The inductive step for the least fixpoint operator will be justified by Proposition 7.3. In more detail, given a formula of the form $\mu x.\psi \in \mu_Y L_1^+$, with $Y = D$ for $L_1^+ = \mathrm{FOE}_1$, and $Y = C$ for $L_1^+ = \mathrm{FOE}_1^\infty$, consider the following chain of equivalences:

$$\begin{aligned}
& s_I \in [\![\mu p.\psi]\!]^{\mathbb{S}} \\
\text{iff } & s_I \in \mathit{LFP}.(\psi_p^{\mathbb{S}})_{\upharpoonright_Q} \text{ for some } \begin{cases}\text{finite}\\ \text{noetherian}\end{cases} \text{set } Q && \text{(by (31)/(32))}\\
\text{iff } & s_I \in \bigcap \left\{P \subseteq Q \mid P \in \mathit{PRE}((\psi_p^{\mathbb{S}})_{\upharpoonright_Q})\right\} \text{ for some } \begin{cases}\text{finite}\\ \text{noetherian}\end{cases} \text{set } Q \\
\text{iff } & \mathbb{S} \models \exists q. \left(\forall p \subseteq q. \left(p \in \mathit{PRE}((\psi_p^{\mathbb{S}})_{\upharpoonright_q}) \to p(s_I)\right)\right) \\
\text{iff } & \mathbb{S} \models (\mu p.\psi)^*[s_I]. && \text{(IH)}
\end{aligned}$$

This concludes the proof of Equation (34). □

7.2 Fixpoints of Continuous Maps

It is well known that continuous functionals are *constructive*. That is, if we construct the least fixpoint of a continuous functional $F : \wp(S) \to \wp(S)$ using the ordinal approximation $\varnothing, F\varnothing, F^2\varnothing, \ldots, F^\alpha\varnothing, \ldots$, then we reach convergence after at most ω many steps, implying that $\mathit{LFP}.F = F^\omega\varnothing$. We will see now that this fact can be strengthened to the following observation, which is the crucial result needed in the proof of Proposition 7.3.

Theorem 7.9. *Let $F : \wp(S) \to \wp(S)$ be a continuous functional. Then for any $s \in S$:*

$$s \in \mathit{LFP}.F \text{ iff } s \in \mathit{LFP}.F_{\upharpoonright_X}, \text{ for some finite } X \subseteq S. \tag{35}$$

Proof. The direction from right to left of Equation (35) is a special case of Proposition 7.5. For the opposite direction of Equation (35), a bit more work is needed. Assume that $s \in \mathit{LFP}.F$; we claim that there are sets $U_1, \ldots, U_n$, for some $n \in \omega$, such that $s \in U_n$, $U_1 \subseteq_\omega F(\varnothing)$, and $U_{i+1} \subseteq_\omega F(U_i)$ for all i with $1 \leq i < n$.

To see this, first observe that since F is continuous, we have $\mathit{LFP}.F = F^\omega(\varnothing) = \bigcup_{n\in\omega} F^n(\varnothing)$, and so we may take n to be the least natural number such that $s \in F^n(\varnothing)$. By a downward induction, we now define sets $U_n, \ldots, U_1$, with $U_i \subseteq F^i(\varnothing)$ for each i. We set up the induction by putting $U_n := \{s\}$, and then $U_n \subseteq F^n(\varnothing)$ by our assumption on n. For $i < n$, we define U_i as follows. Using the inductive fact that $U_{i+1} \subseteq_\omega F^{i+1}(\varnothing) = F(F^i(\varnothing))$, it follows by continuity of F that for each $u \in U_{i+1}$ there is a set $V_u \subseteq_\omega F^i(\varnothing)$ such that $u \in F(V_u)$. We then define $U_i := \bigcup\{V_u \mid u \in U_{i+1}\}$, so that clearly $U_{i+1} \subseteq_\omega F(U_i)$ and $U_i \subseteq_\omega F^i(\varnothing)$. Continuing like this, ultimately we arrive at stage $i = 1$, where we find $U_1 \subseteq F(\varnothing)$ as required.

Finally, given the sequence $U_n, \ldots, U_1$, we define

$$X := \bigcup_{0<i\leq n} U_i.$$

It is then straightforward to prove that $U_i \subseteq LFP.F_{\restriction_X}$, for each i with $0 < i \leq n$, and so in particular we find that $s \in U_n \subseteq LFP.F_{\restriction_X}$. This finishes the proof of the implication from left to right in Equation (35). □

As an almost immediate corollary of this result, we obtain the second part of Proposition 7.3.

Proof of Proposition 7.3(2). Take an arbitrary formula $\mu p.\varphi \in \mu_C\mathrm{FOE}_1^\infty$, and then by definition we have $\varphi \in \mu_C\mathrm{FOE}_1^\infty \cap \mathsf{Con}_p(\mu\mathrm{FOE}_1^\infty)$. But it follows from a routine inductive proof that every formula $\psi \in \mu_C\mathrm{FOE}_1^\infty \cap \mathsf{Con}_\mathsf{Q}(\mu\mathrm{FOE}_1^\infty)$ is continuous in each variable in Q. Thus φ is continuous in p, and so the result is immediate by Theorem 7.9. □

7.3 Fixpoints of Noetherian Maps

We will now see how to prove Proposition 7.3(1), which is the key result that we need to embed alternation-free μ-calculi such as $\mu_N\mathrm{FOE}_1$ and $\mu_N\mathrm{ML}$ into noetherian second-order logic. Perhaps suprisingly, this case is slightly more subtle than the characterisation of fixpoints of continuous maps.

We start with stating some auxiliary definitions and results on monotone functionals, starting with a game-theoretic characterisation of their least fixpoints [Venema 2012].

Definition 7.10. Given a monotone functional $F : \wp(S) \to \wp(S)$, we define the *unfolding game* $\mathcal{U}_F$ as follows:

- at any position $s \in S$, $\exists$ needs to pick a set X such that $s \in FX$;
- at any position $X \in \wp(S)$, $\forall$ needs to pick an element of X
- all infinite matches are won by $\forall$.

A positional strategy $f : S \to \wp(S)$ for $\exists$ in $\mathcal{U}_F$ is *descending* if, for all ordinals α,

$$s \in F^{\alpha+1}(\varnothing) \text{ implies } f(s) \subseteq F^\alpha(\varnothing). \tag{36}$$

It is not the case that *all* positional winning strategies for $\exists$ in $\mathcal{U}_F$ are descending, but the next result shows that there always is one.

Proposition 7.11. *Let $F : \wp(S) \to \wp(S)$ be a monotone functional.*

(1) For all $s \in S$, $s \in Win_\exists(\mathcal{U}_F)$ iff $s \in LFP.F$;
(2) If $s \in LFP.F$, then $\exists$ has a descending winning strategy in $\mathcal{U}_F@s$.

Proof. Point (1) corresponds to Venema [2012, Theorem 3.14(2)]. For part (2) one can simply take the following strategy. Given $s \in LFP.F$, let α be the least ordinal such that $s \in F^\alpha(\varnothing)$; it is easy to see that α must be a successor ordinal, say, $\alpha = \beta + 1$. Now, simply put, $f(s) := F^\beta(\varnothing)$. □

Definition 7.12. Let $F : \wp(S) \to \wp(S)$ be a monotone functional, let f be a positional winning strategy for $\exists$ in $\mathcal{U}_F$, and let $r \in S$. Define $T_{f,r} \subseteq S$ to be the set of states in S that are f-reachable in $\mathcal{U}_F@r$. This set has a tree structure induced by the map f itself, where the children of $s \in T_{f,r}$ are given by the set $f(s)$; we will refer to $T_{f,r}$ as the *strategy tree* of f.

Note that a strategy tree $T_{f,r}$ will have no infinite paths, since we define the notion only for a *winning* strategy f.

Proposition 7.13. *Let $F : \wp(S) \to \wp(S)$ be a monotone functional, let $r \in S$, and let f be a descending winning strategy for $\exists$ in $\mathcal{U}_F$. Then*

$$r \in LFP.F \text{ implies } r \in LFP.F\restriction_{T_{f,r}}. \tag{37}$$

Proof. Let F, r, and f be as in the formulation of the proposition. Assume that $r \in LFP.F$, and then clearly $r \in F^\alpha(\varnothing)$ for some ordinal α; furthermore, $T_{f,r}$ is defined, and clearly we have $r \in T_{f,r}$. Abbreviate $T := T_{f,r}$. It then suffices to show that for all ordinals α we have

$$F^\alpha(\varnothing) \cap T \subseteq (F\restriction_T)^\alpha(\varnothing). \tag{38}$$

We will prove Equation (38) by transfinite induction. The base case, where $\alpha = 0$, and the inductive case where α is a limit ordinal are straightforward, so we focus on the case where α is a successor ordinal, say, $\alpha = \beta + 1$. Take an arbitrary state $u \in F^{\beta+1}(\varnothing) \cap T$; then, we find $f(u) \subseteq F^\beta(\varnothing)$ by our assumption (36) and $f(u) \subseteq T$ by definition of T. Then the induction hypothesis yields that $f(u) \subseteq (F\restriction_T)^\beta(\varnothing)$, and so we have $f(u) \subseteq (F\restriction_T)^\beta(\varnothing) \cap T$. But since f is a winning strategy, and u is a winning position for $\exists$ in $\mathcal{U}_F$ by Claim 7.11(i), $f(u)$ is a legitimate move for $\exists$, and so we have $u \in F(f(u))$. Thus by monotonicity of F we obtain $u \in F((F\restriction_T)^\beta(\varnothing) \cap T)$, and since $u \in T$ by assumption, this means that $u \in (F\restriction_T)^{\beta+1}(\varnothing)$ as required. □

We now turn to the specific case where we consider the least fixed point of a functional F, which is induced by some formula $\varphi(p) \in \mu_N L_1$ on some LTS $\mathbb{S}$. By Proposition 7.11 and Fact 4.29, $\exists$ has a winning strategy in $\mathcal{E}(\mu p.\varphi(p), \mathbb{S})@(\mu p.\varphi(p), s)$ if and only if she has a winning strategy in $\mathcal{U}_F@s$, too, where $F := \varphi_p^{\mathbb{S}}$ is the monotone functional defined by $\varphi(p)$. The next Proposition makes this correspondence explicit when $L_1 = \text{FOE}$.

First, we need to introduce some auxiliary concepts and notations. Given a winning strategy f for $\exists$ in $\mathcal{E}(\mu p.\varphi, \mathbb{S})@(\mu p.\varphi, s)$, we denote by $B(f)$ the set of all finite f-guided, possibly partial, matches in $\mathcal{E}(\psi, \mathbb{S})@(\psi, s)$ in which no position of the form $(\nu q.\psi, r)$ is visited. Let f be a positional winning strategies for $\exists$ in $\mathcal{U}_F@s$ and f' a winning strategy for her in $\mathcal{E}(\mu p.\varphi, \mathbb{S})@(\mu p.\varphi, s)$. We call f and f' *compatible* if each point in $T_{f,s}$ occurs on some path belonging to $B(f')$.

Proposition 7.14. *Let $\varphi(p) \in \mu_N\text{FOE}p$ and $s \in [\![\mu p.\varphi]\!]^{\mathbb{S}}$. Then there is a descending winning strategy for $\exists$ in $\mathcal{U}_F@s$ compatible with a winning strategy for $\exists$ in $\mathcal{E}(\mu p.\varphi, \mathbb{S})@(\mu p.\varphi, s)$.*

Proof. Let $F := \varphi_p^{\mathbb{S}}$ be the monotone functional defined by $\varphi(p)$. From $s \in [\![\mu p.\varphi]\!]^{\mathbb{S}}$, we get that $s \in LFP.F$. Applying Proposition 7.11 to the fact that $s \in LFP.F$ yields that $\exists$ has a descending winning strategy $f : S \to \wp(S)$ in $\mathcal{U}_F@s$. We define $\exists$'s strategies f' in $\mathcal{E}(\mu p.\varphi, \mathbb{S})@(\mu p.\varphi, s)$, and f^* in $\mathcal{U}_F@s$ as follows:

(1) In the evaluation games $\mathcal{E}$, after the initial automatic move, the position of the match is (φ, s); there $\exists$ first plays her positional winning strategy f_s from $\mathcal{E}(\varphi(p), \mathbb{S}[p \mapsto f(s)])@(\varphi(p), s)$, and we define her move $f^*(s)$ in the unfolding game $\mathcal{U}$ as the set of all nodes $t \in f(s)$ such that there is a f_s-guided match in $B(f_s)$, whose last position is (p, t).
(2) Each time a position (p, t) is reached in the evaluation games $\mathcal{E}$, distinguish cases:
  (a) If $t \in Win_\exists(\mathcal{U}_F)$, then $\exists$ continues with the positional winning strategy f_t from $\mathcal{E}(\varphi(p), \mathbb{S}[p \mapsto f(t)])@(\varphi(p), t)$, and we define her move $f^*(t)$ in $\mathcal{U}$ as the set of all nodes $w \in f(t)$ such that there is a f_t-guided match in $B(f_t)$ whose last position (p, w);
  (b) If $t \notin Win_\exists(\mathcal{U}_F)$, then $\exists$ continues with a random positional strategy, and we define $f^*(t) := \varnothing$.
(3) For any position (p, t) that was not reached in the previous steps, $\exists$ sets $f^*(t) := \varnothing$.

By construction, f' and f^* are compatible. Moreover, $f^*(t) \subseteq f(t)$, for $t \in S$, meaning that f^* is descending. We verify that both f' and f^* are actually winning strategies for $\exists$ in the respective games.

First, observe that in every position of the form (p, t) reached during a f'-guided match, we have $t \in Win_\exists(\mathcal{U}_F)$. This can be proved by induction on the number of position of the form (p, t) visited during an f'-guided match. For the inductive step, assume $w \in Win_\exists(\mathcal{U}_F)$. Hence f_w is winning for $\exists$ in $\mathcal{E}(\varphi, \mathbb{S}[p \mapsto f(w)])@(\varphi, w)$. This means that if a position of the form (p, t) is reached, then the variable p must be true at t in the model $\mathbb{S}[p \mapsto f(w)]$, meaning that it belongs to the set $f(w)$. By assumption f is a winning strategy for $\exists$ in $\mathcal{U}_F$, and therefore any element of $f(w)$ is again a member of the set $Win_\exists(\mathcal{U}_F)$.

Finally, let π be an arbitrary f'-guided match of $\mathcal{E}(\varphi, \mathbb{S}[p \mapsto f(w)])@(\varphi, w)$. We verify that π is winning for $\exists$. First observe that since f is winning for her in $\mathcal{U}_F@s$, the fixpoint variable p is unfolded only finitely many times during π. Let (p, t) be the last basic position in π where p occurs. Then, from now on, f' and f_t coincide, yielding that the match is winning for $\exists$.

We finally verify that f^* is winning for $\exists$ in the unfolding game $\mathcal{U}_F@s$. First, since f' is winning, $B(f')$ does not contain an infinite ascending chain of f'-guided matches, and thence any f^*-guided match in $\mathcal{U}_F@s$ is finite. It therefore remains to verify that for every f^*-guided match π in $\mathcal{U}_F@s$ such that $\mathsf{last}(\pi)$ is an $\exists$ position, she can always move. We do it by induction on the length of a f^*-guided match. At each step, we use compatibility and thus keep track of the corresponding position in the evaluation game $\mathcal{E}(\mu p.\varphi, \mathbb{S})@(\mu p.\varphi, s)$. The initial position for her is $s \in S$. Notice that $f^*(s) = f(s) \cap B(\xi')$ and therefore f' corresponds to f_s on $\mathcal{E}(\varphi(p), \mathbb{S}[p \mapsto f^*(s)])@(\varphi(p), s)$ and it is therefore winning for $\exists$. In particular, this means that $s \in F(f^*(s))$. Hence, as initial move, $\exists$ is allowed to play $f^*(s)$. Moreover, any subsequent choice for $\forall$ is such that there is a winning match $\pi \in B(\xi_s)$ for $\exists$ such that $\mathsf{last}(\pi) = (p, w)$. For the induction step, assume $\forall$ has chosen $t \in f^*(w)$, where $f^*(w) = f(w) \cap B(\xi')$, f' corresponds to the winning strategy f_w on $\mathcal{E}(\varphi(p), \mathbb{S}[p \mapsto f^*(w)])@(\varphi(p), w)$, and there is a winning match $\pi \in B(\xi_w)$ for $\exists$ such that $\mathsf{last}(\pi) = (p, w)$. By construction, f' corresponds to the winning strategy f_t for $\exists$ on $\mathcal{E}(\varphi(p), \mathbb{S}[p \mapsto f(t)])@(\varphi(p), t)$. Because $f^*(t) = f(s) \cap B(\xi')$, f_t is also winning for her in $\mathcal{E}(\varphi(p), \mathbb{S}[p \mapsto f^*(t)])@(\varphi(p), t)$, meaning that $s \in F(f^*(s))$. The move $f^*(t)$ is therefore admissible, and any subsequent choice for $\forall$ is such that there is a winning match $\pi \in B(\xi_t)$ for $\exists$ with $\mathsf{last}(\pi) = (p, w)$. □

Proof of Proposition 7.3(1). Let $\mathbb{S}$ be an LTS and $\varphi(p) \in \mu_N\mathrm{FOE}_1 p$.

The right-to-left direction of Equation (31) being proved by Proposition 7.5, we check the left-to-right direction. We first verify that winning strategies in evaluation games for noetherian fixpoint formulas naturally induce bundles. More precisely: □

Claim. *Let $B^{\mathbb{S}}(f)$ be the projection of $B(f)$ on S, that is, the set of all paths in $\mathbb{S}$ that are a projection on S of a f-guided (partial) match in $B(f)$. Then $B^{\mathbb{S}}(f)$ is a bundle.*

Proof Of Claim. Assume toward a contradiction that $B^{\mathbb{S}}(f)$ contains an infinite ascending chain $\pi_0 \sqsubset \pi_1 \sqsubset \cdots$. Let π be the limit of this chain and consider the set of elements in $B(f)$ that, projected on S, are prefixes of π. By König's Lemma, this set contains an infinite ascending chain whose limit is an infinite f-guided match in $\mathcal{E}(\mu p.\varphi, \mathbb{S})$, which starts at $(\mu p.\varphi, s)$ and of which π is the projection on S. By definition of $B(f)$, the highest bound variable of $\mu p.\varphi$ that gets unravelled infinitely often in ρ is a μ-variable, meaning that the match is winning for $\forall$, a contradiction. ◀

Assume that $s \in [\![\mu p.\varphi]\!]^{\mathbb{S}}$, and let $F := \varphi_p^{\mathbb{S}}$ be the monotone functional defined by $\varphi(p)$. By Proposition 7.14, $\exists$ has a winning strategy f' in $\mathcal{E}(\mu p.\varphi, \mathbb{S})@(\mu p.\varphi, s)$ compatible with a descending winning strategy f in $\mathcal{U}_F@s$. By Proposition 7.13, we obtain that $s \in LFP.F{\upharpoonright_{T_{f,s}}}$.

Because of compatibility, every node in $T_{f,s}$ occurs on some path of $B(f')$. From the Claim, we know that $B^{\mathbb{S}}(f')$ is a bundle, meaning that $T_{f,s}$ is noetherian as required.

8 EXPRESSIVENESS MODULO BISIMILARITY

In this section, we use the tools developed in the previous parts to prove the main results of the article on expressiveness modulo bisimilarity, viz., Theorem 1.1 stating

$$\mu_N\mathrm{ML} \equiv \mathrm{NMSO}/\underline{\leftrightarrow}, \tag{39}$$

$$\mu_C\mathrm{ML} \equiv \mathrm{WMSO}/\underline{\leftrightarrow}. \tag{40}$$

Proof of Theorem 1.1. The structure of the proof is the same for the statements (39) and (40). In both cases, we will need three steps to establish a link between the modal language on the left-hand side of the equation to the bisimulation-invariant fragment of the second-order logic on the right-hand side.

The first step is to connect the fragments μ_NML and μ_CML of the modal μ-calculus to, respectively, the weak and the continuous-weak automata for first-order logic without equality. That is, in Theorem 8.1 below we prove the following:

$$\mu_N\mathrm{ML} \equiv Aut_w(\mathrm{FO}_1), \tag{41}$$

$$\mu_C\mathrm{ML} \equiv Aut_{wc}(\mathrm{FO}_1). \tag{42}$$

Second, the main observation that we shall make in this section is that

$$Aut_w(\mathrm{FO}_1) \equiv Aut_w(\mathrm{FOE}_1)/\underline{\leftrightarrow}, \tag{43}$$

$$Aut_{wc}(\mathrm{FO}_1) \equiv Aut_{wc}(\mathrm{FOE}_1^\infty)/\underline{\leftrightarrow}. \tag{44}$$

That is, for Equation (43) we shall see in Theorem 8.4 below that a weak FOE_1-automaton $\mathbb{A}$ is bisimulation invariant iff it is equivalent to a weak FO_1-automaton $\mathbb{A}^\diamond$ (effectively obtained from $\mathbb{A}$) and similarly for Equation (44).

Finally, we use the automata-theoretic characterisations of NMSO and WMSO that we obtained in earlier sections:

$$Aut_w(\mathrm{FOE}_1) \equiv \mathrm{NMSO}, \tag{45}$$

$$Aut_{wc}(\mathrm{FOE}_1^\infty) \equiv \mathrm{WMSO}. \tag{46}$$

Then it is obvious that Equation (39) follows from Equations (41), (43), and (45), while similarly Equation (40) follows from Equations (42), (44), and (46). □

It is left to prove Equations (41) and (42) and Equations (43) and (44); we will take care of this in the two subsections below.

8.1 Automata for μ_NML and μ_CML

In this subsection, we consider the automata corresponding to the continuous and the alternation-free μ-calculus. That is, we verify Equations (41) and (42).

Theorem 8.1.

(1) *There is an effective construction transforming a formula $\varphi \in \mu$ML into an equivalent automaton in $Aut(\mathrm{FO}_1)$ and vice versa.*
(2) *There is an effective construction transforming a formula $\varphi \in \mu_N$ML into an equivalent automaton in $Aut_w(\mathrm{FO}_1)$ and vice versa.*

(3) *There is an effective construction transforming a formula* $\varphi \in \mu_C\mathrm{ML}$ *into an equivalent automaton in* $Aut_{wc}(\mathrm{FO}_1)$ *and vice versa.*

Proof. In each of these cases, the direction from left to right is easy to verify, so we omit details. For the opposite direction, we focus on the hardest case, that is, we will only prove that $Aut_{wc}(\mathrm{FO}_1) \leq \mu_C\mathrm{ML}$. By Theorem 4.33, it suffices to show that $\mu_C\mathrm{FO}_1 \leq \mu_C\mathrm{ML}$, and we will in fact provide a direct, inductively defined, truth-preserving translation $(\cdot)^t$ from $\mu_C\mathrm{FO}_1(\mathrm{P})$ to $\mu_C\mathrm{ML}(\mathrm{P})$. Inductively, we will ensure that, for every set $\mathrm{Q} \subseteq \mathrm{P}$,

$$\varphi \in \mathsf{Con}_{\mathrm{Q}}(\mu\mathrm{FO}_1) \text{ implies } \varphi^t \in \mathsf{Con}_{\mathrm{Q}}(\mu\mathrm{ML}) \tag{47}$$

and that the dual property holds for cocontinuity.

Most of the clauses of the definition of the translation $(\cdot)^t$ are completely standard: For the atomic clause we take $p^t := p$ and $(\neg p)^t := \neg p$, for the Boolean connectives we define $(\varphi_0 \vee \varphi_1)^t := \varphi_0^t \vee \varphi_1^t$ and $(\varphi_0 \wedge \varphi_1)^t := \varphi_0^t \wedge \varphi_1^t$, and for the fixpoint operators we take $(\mu p.\varphi)^t := \mu p.\varphi^t$ and $(\nu p.\varphi)^t := \nu p.\varphi^t$—to see that the latter clauses indeed provide formulas in $\mu_C\mathrm{ML}$ we use Equation (47) and its dual. In all of these cases, it is easy to show that Equation (47) holds (or remains true, in the inductive cases).

The only interesting case is where φ is of the form $\bigcirc_\alpha(\varphi_1, \dots, \varphi_n)$. By definition of the language $\mu_C\mathrm{FO}_1$, we may assume that $\alpha(a_1, \dots, a_n) \in \mathsf{Con}_B(\mathrm{FO}_1(A))$, where $A = \{a_1, \dots, a_n\}$ and $B = \{a_1, \dots, a_k\}$, that for each $1 \leq i \leq k$ the formula φ_i belongs to the set $\mathsf{Con}_{\mathrm{Q}}(\mu_C\mathrm{FO}_1)$, and that for each $k+1 \leq j \leq n$ the formula φ_j is Q-free. It follows by the induction hypothesis that $\varphi_l \equiv \varphi_l^t \in \mu_C\mathrm{ML}$ for each l, that $\varphi_i^t \in \mathsf{Con}_{\mathrm{Q}}(\mu\mathrm{ML})$ for each $1 \leq i \leq k$, and that the formula φ_j^t is Q-free for each $k+1 \leq j \leq n$. The key observation is now that by Theorem 4.13 we may without loss of generality assume that α is in *normal form*; that is, a disjunction of formulas of the form $\alpha_{\Sigma,\Pi} = \nabla^+_{\mathrm{FO}}(\Sigma, \Pi)$, where every Σ and Π is a subset of $\wp(A)$, $B \cap \bigcup \Pi = \varnothing$ for every Π, and

$$\nabla^+_{\mathrm{FO}}(\Sigma, \Pi) := \bigwedge_{S \in \Sigma} \exists x \bigwedge_{a \in S} a(x) \;\wedge\; \forall x \bigvee_{S \in \Pi} \bigwedge_{a \in S} a(x).$$

We now define

$$\begin{aligned} \left(\bigcirc_{\alpha_{\Sigma,\Pi}}(\overline{\varphi})\right)^t &:= \textstyle\bigwedge_{S \in \Sigma} \Diamond \bigwedge_{a_l \in S} \varphi_l^t \;\wedge\; \Box \bigvee_{S \in \Pi} \bigwedge_{a_j \in S} \varphi_j^t \\ \varphi^t &:= \textstyle\bigvee \left(\bigcirc_{\alpha_{\Sigma,\Pi}}(\overline{\varphi})\right)^t. \end{aligned}$$

It is then obvious that φ and φ^t are equivalent, so it remains to verify Equation (47). But this is immediate by the observation that all formulas φ_j^t in the scope of the $\Box$ are associated with an a_j belonging to a set $S \subseteq A$ that has an empty intersection with the set B; that is, each a_j belongs to the set $\{a_{k+1}, \dots, a_n\}$ and so φ_j^t is Q-free. $\square$

8.2 Bisimulation Invariance, One Step at a Time

In this subsection, we will show how the bisimulation invariance results in this article can be proved by automata-theoretic means. Following Janin and Walukiewicz [1996], we will define a construction that, for $L_1 \in \{\mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$, transforms an arbitrary L_1-automaton $\mathbb{A}$ into an FO_1-automaton $\mathbb{A}^\Diamond$ such that $\mathbb{A}$ is bisimulation invariant iff it is equivalent to $\mathbb{A}^\Diamond$. In addition, we will make sure that this transformation preserves both the weakness and the continuity condition. The operation $(\cdot)^\Diamond$ is completely determined by the following translation at the one-step level.

Definition 8.2. Recall from Theorem 4.12 that any formula in ${\mathrm{FOE}_1}^+(A)$ is equivalent to a disjunction of formulas of the form $\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}}, \Sigma)$, whereas any formula in $\mathrm{FOE}_1^{\infty+}(A)$ is equivalent to a disjunction of formulas of the form $\nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$. Based on these normal forms, for both one-step languages $L_1 = \mathrm{FOE}_1$ and $L_1 = \mathrm{FOE}_1^\infty$, we define the translation $(\cdot)^\Diamond : {L_1}^+(A) \to \mathrm{FO}_1^+(A)$ by setting

$$\left.\begin{array}{l}\left(\nabla^+_{\mathrm{FOE}}(\overline{\mathbf{T}},\Sigma)\right)^{\Diamond} \\ \left(\nabla^+_{\mathrm{FOE}^\infty}(\overline{\mathbf{T}},\Pi,\Sigma)\right)^{\Diamond}\end{array}\right\} := \bigwedge_i \exists x_i.\tau^+_{T_i}(x_i) \wedge \forall x. \bigvee_{S\in\Sigma} \tau^+_S(x),$$

and for $\alpha = \bigvee_i \alpha_i$ we define $\alpha^{\Diamond} := \bigvee \alpha_i^{\Diamond}$.

This definition propagates to the level of automata in the obvious way.

Definition 8.3. Let $L_1 \in \{\mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$ be a one-step language. Given an automaton $\mathbb{A} = \langle A, \Delta, \Omega, a_I\rangle$ in $Aut(L_1)$, define the automaton $\mathbb{A}^{\Diamond} := \langle A, \Delta^{\Diamond}, \Omega, a_I\rangle$ in $Aut(\mathrm{FO}_1)$ by putting, for each $(a,c) \in A \times C$:

$$\Delta^{\Diamond}(a,c) := (\Delta(a,c))^{\Diamond}.$$

The main result of this section is the theorem below. For its formulation, recall that $\mathbb{S}^\omega$ is the ω-unravelling of the model $\mathbb{S}$ (as defined in the preliminaries). As an immediate corollary of this result, we see that Equations (43) and (44) hold.

Theorem 8.4. *Let $L_1 \in \{\mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$ be a one-step language and let $\mathbb{A}$ be an L_1-automaton.*

(1) The automata $\mathbb{A}$ and $\mathbb{A}^{\Diamond}$ are related as follows, for every model $\mathbb{S}$:

$$\mathbb{A}^{\Diamond} \textit{ accepts } \mathbb{S} \textit{ iff } \mathbb{A} \textit{ accepts } \mathbb{S}^\omega. \tag{48}$$

(2) The automaton $\mathbb{A}$ is bisimulation invariant iff $\mathbb{A} \equiv \mathbb{A}^{\Diamond}$.

(3) If $\mathbb{A} \in Aut_w(L_1)$, then $\mathbb{A}^{\Diamond} \in Aut_w(\mathrm{FO}_1)$, and if $\mathbb{A} \in Aut_{wc}(\mathrm{FOE}_1^\infty)$, then $\mathbb{A}^{\Diamond} \in Aut_{wc}(\mathrm{FO}_1)$.

The remainder of this section is devoted to the proof of Theorem 8.4. The key proposition is the following observation on the one-step translation, which we take from the companion paper [Carreiro et al. 2018].

Proposition 8.5. *Let $L_1 \in \{\mathrm{FOE}_1, \mathrm{FOE}_1^\infty\}$. For every one-step model (D,V) and every $\alpha \in L_1^+(A)$ we have*

$$(D,V) \models \alpha^{\Diamond} \textit{ iff } (D\times\omega, V_\pi) \models \alpha, \tag{49}$$

where V_π is the induced valuation given by $V_\pi(a) := \{(d,k) \mid d \in V(a), k \in \omega\}$.

Proof of Theorem 8.4. The proof of the first part is based on a fairly routine comparison, based on Proposition 8.5, of the acceptance games $\mathcal{A}(\mathbb{A}^{\Diamond}, \mathbb{S})$ and $\mathcal{A}(\mathbb{A}, \mathbb{S}^\omega)$. (In a slightly more general setting, the details of this proof can be found in Venema [2014].)

For part 2, the direction from right to left is immediate by Theorem 8.1. The opposite direction follows from the following equivalences, where we use the bisimilarity of $\mathbb{S}$ and $\mathbb{S}^\omega$ (Fact 2.4):

$$\begin{aligned} \mathbb{A} \text{ accepts } \mathbb{S} &\text{ iff } \mathbb{A} \text{ accepts } \mathbb{S}^\omega && \mathbb{A} \text{ bisimulation invariant} \\ &\text{ iff } \mathbb{A}^{\Diamond} \text{ accepts } \mathbb{S} && \text{equivalence (48)} \end{aligned}$$

It remains to be checked that the construction $(\cdot)^{\Diamond}$, which has been defined for arbitrary automata in $Aut(L_1)$, transforms both WMSO-automata and NMSO-automata into automata of the right kind. This can be verified by a straightforward inspection at the one-step level. □

Remark 8.6. In fact, we are dealing here with an instantiation of a more general phenomenon that is essentially coalgebraic in nature. In Venema [2014], it is proved that if L_1 and L_1' are two one-step languages that are connected by a translation $(\cdot)^{\Diamond} : L_1' \to L_1$ satisfying a condition similar to Equation (49), then we find that $Aut(L_1)$ corresponds to the bisimulation-invariant fragment of $Aut(L_1')$: $Aut(L_1) \equiv Aut(L_1')/\underline{\leftrightarrow}$. This subsection can be generalized to prove similar results relating $Aut_w(L_1)$ to $Aut_w(L_1')$ and $Aut_{wc}(L_1)$ to $Aut_{wc}(L_1')$.

REFERENCES

Samson Abramsky and Achim Jung. 1994. Domain theory. In *Handbook for Logic in Computer Science*, S. Abramsky, M. Gabbay Dov, and T. S. M. Maibaum (Eds.). Clarendon Press, Oxford.

André Arnold and Damian Niwiński. 1992. Fixed point characterization of weak monadic logic definable sets of trees. In *Tree Automata and Languages*. 159–188.

André Arnold and Damian Niwiński. 2001. *Rudiments of μ-calculus*. Studies in Logic and the Foundations of Mathematics, Vol. 146. North-Holland, 1–39.

Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. 2017. Deciding parity games in quasipolynomial time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computin (STOC'17)*, Hamed Hatami, Pierre McKenzie, and Valerie King (Eds.). ACM, 252–263. DOI: https://doi.org/10.1145/3055399.3055409

Facundo Carreiro. 2015. *Fragments of Fixpoint Logics*. Ph.D. Dissertation. Institute for Logic, Language and Computation, Universiteit van Amsterdam.

Facundo Carreiro, Alessandro Facchini, Yde Venema, and Fabio Zanasi. 2014. Weak MSO: Automata and expressiveness modulo bisimilarity. In *Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (CSL-LICS'14)*. 27:1–27:27. DOI: https://doi.org/10.1145/2603088.2603101

Facundo Carreiro, Alessandro Facchini, Yde Venema, and Fabio Zanasi. 2018. Model theory of monadic predicate logic with the infinity quantifier. *CoRR* abs/1809.03262 (2018). http://arxiv.org/abs/1809.03262.

Corina Cîrstea and Dirk Pattinson. 2004. Modular construction of modal logics. In *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04)*, Ph. Gardner and N. Yoshida (Eds.), Lecture Notes in Computer Science, Vol. 3170. Springer, 258–275.

Giovanna D'Agostino and Marco Hollenberg. 2000. Logical questions concerning the μ-calculus. *J. Symbol. Logic* 65, 1 (2000), 310–332.

Anuj Dawar and Martin Otto. 2009. Modal characterisation theorems over special classes of frames. *Ann. Pure Appl. Logic* 161, 1 (2009), 1–42.

Ernest Allen Emerson and Charanjit S. Jutla. 1991. Tree automata, μ-calculus and determinacy (extended abstract). In *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science (FOCS'91)*. IEEE Computer Society, 368–377.

Alessandro Facchini, Yde Venema, and Fabio Zanasi. 2013. A characterization theorem for the alternation-free fragment of the modal μ-calculus. In *Proceedings of the ACM/IEEE Symposium on Logic in Computer Science (LICS'13)*. IEEE Computer Society, 478–487.

Gaëlle Fontaine. 2008. Continuous fragment of the mu-calculus. In *Proceedings of the Conference on Computer Science Logic (CSL'08)*, Michael Kaminski and Simone Martini (Eds.), Lecture Notes in Computer Science, Vol. 5213. Springer, 139–153.

Gaëlle Fontaine and Yde Venema. 2018. Some model theory for the modal μ-calculus: Syntactic characterisations of semantic properties. *Logic. Methods Comput. Sci.* 14, 1 (2018).

Erich Grädel, Wolfgang Thomas, and Thomas Wilke (Eds.). 2002. *Automata, Logics, and Infinite Games: A Guide to Current Research*. Lecture Notes in Computer Science, Vol. 2500. Springer.

David Janin and Igor Walukiewicz. 1996. On the expressive completeness of the propositional μ-calculus with respect to monadic second order logic. In *Proceedings of the 7th International Conference on Concurrency Theory (CONCUR'96)*. Springer-Verlag, London, UK, 263–277.

Christian Kissig and Yde Venema. 2009. Complementation of coalgebra automata. In *Proceedings of the Conference on Algebra and Coalgebra in Computer Science (CALCO'09)*, Alexander Kurz, Marina Lenisa, and Andrzej Tarlecki (Eds.), Lecture Notes in Computer Science, Vol. 5728. Springer, 81–96.

Orna Kupferman and Moshe Y. Vardi. 2003. $\Pi_2 \cap \Sigma_2 \equiv$ AFMC. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP'03)*. 697–713.

Andrzej Mostowski. 1957. On a generalization of quantifiers. *Fundam. Math.* 44, 1 (1957), 12–36. http://eudml.org/doc/213418.

Andrzej Mostowski. 1991. *Games with Forbidden Positions*. Technical Report 78. University of Gdansk.

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

David E. Muller and Paul E. Schupp. 1987. Alternating automata on infinite trees. *Theor. Comput. Sci.* 54, 2–3 (1987), 267–276. DOI: https://doi.org/10.1016/0304-3975(87)90133-2

Damian Niwiński. 1986. On fixed-point clones (extended abstract). In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP'86)* Laurent Kott (Ed.), Lecture Notes in Computer Science, Vol. 226. Springer, 464–473.

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

Eric Rosen. 1997. Modal logic over finite structures. *J. Logic Lang. Inf.* 6, 4 (1997), 427–439.

Balder ten Cate and Alessandro Facchini. 2011. Characterizing EF over infinite trees and modal logic on transitive graphs. In *Proceedings of the International Symposium on Mathematical Foundations of Computer Science (MFCS'11)*, Filip Murlak and Piotr Sankowski (Eds.), Lecture Notes in Computer Science, Vol. 6907. Springer, 290–302.

Jouko Väänänen. 1977. Remarks on generalized quantifiers and second-order logics. In *Set Theory and Hierarchy Theory*. Prace Naukowe Instytutu Matematyki Politechniki Wroclawskiej, Wroclaw, 117–123.

Johan van Benthem. 1977. *Modal Correspondence Theory*. Ph.D. Dissertation. Universiteit van Amsterdam.

Johan van Benthem. 2006. Modal frame correspondences and fixed-points. *Stud. Logic.* 83, 1–3 (2006), 133–155.

Yde Venema. 2014. Expressiveness modulo bisimilarity: A coalgebraic perspective. To appear in: A. Baltag and S. Smets (eds.), *Johan van Benthem on Logic and Information Dynamics*. 33–65. DOI: 10.1007/978-3-319-06025-5_2

Yde Venema. October 2012. Lectures on the modal μ-calculus. Lecture Notes, Institute for Logic, Language and Computation, University of Amsterdam.

Igor Walukiewicz. 1996. Monadic second order logic on tree-like structures. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS'96)*, Claude Puech and Rüdiger Reischuk (Eds.), Lecture Notes in Computer Science, Vol. 1046. Springer, 401–413.

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

Fabio Zanasi. 2012. *Expressiveness of Monadic Second Order Logics on Infinite Trees of Arbitrary Branching Degree*. Master's thesis. ILLC, Universiteit van Amsterdam, the Netherlands.

Received September 2018; revised July 2019; accepted September 2019