2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science

Monadic Second-Order Logic and Bisimulation Invariance for Coalgebras

Sebastian Enqvist*†, Fatemeh Seifan*, Yde Venema*

*ILLC, Universiteit van Amsterdam
†Department of Philosophy, Lund University
Sebastian.Enqvist@fil.lu.se, F.Seifan@uva.nl, Y.Venema@uva.nl

***Abstract*—Generalizing standard monadic second-order logic for Kripke models, we introduce monadic second-order logic MSO(T) interpreted over coalgebras for an arbitrary set functor T. Similar to well-known results for monadic second-order logic over trees, we provide a translation of this logic into a class of automata, relative to the class of T-coalgebras that admit a tree-like supporting Kripke frame. We then consider invariance under behavioral equivalence of MSO(T)-formulas; more in particular, we investigate whether the coalgebraic mu-calculus is the bisimulation-invariant fragment of MSO(T). Building on recent results by the third author we show that in order to provide such a coalgebraic generalization of the Janin-Walukiewicz Theorem, it suffices to find what we call an adequate uniform construction for the functor T. As applications of this result we obtain a partly new proof of the Janin-Walukiewicz Theorem, and bisimulation invariance results for the bag functor (graded modal logic) and all exponential polynomial functors.**

Finally, we consider in some detail the monotone neighborhood functor M, which provides coalgebraic semantics for monotone modal logic. It turns out that there is no adequate uniform construction for M, whence the automata-theoretic approach towards bisimulation invariance does not apply directly. This problem can be overcome if we consider global bisimulations between neighborhood models: one of our main results provides a characterization of the monotone modal mu-calculus extended with the global modalities, as the fragment of monadic second-order logic for the monotone neighborhood functor that is invariant for global bisimulations.

***Keywords*-coalgebra; monadic second-order logic; automata; bisimulation invariance; modal mu-calculus**

I. Introduction

A. Logic, automata and coalgebra

The aim of this paper is to strengthen the link between the areas of logic, automata and coalgebra. More in particular, we provide a coalgebraic generalization of the automata-theoretic approach towards monadic second-order logic (MSO), and we address the question whether the Janin-Walukiewicz Theorem can be generalized from Kripke structures to the setting of arbitrary coalgebras.

The connection between *monadic second-order logic* and *automata* is classic, going back to the seminal work of Büchi, Rabin, and others. For instance, Rabin's decidability result for the monadic second-order theory of binary trees, or $S2S$, makes use of a translation of monadic second-order logic into a class of automata, thus reducing the satisfiability problem for $S2S$ to the non-emptiness problem for the corresponding automata [1]. The link between MSO and automata over trees with arbitrary branching was further explored by Walukiewicz [2]. Janin and Walukiewicz considered monadic second-order logic interpreted over Kripke structures, and used automata-theoretic techniques to obtain a van Benthem-like characterization theorem for monadic second-order logic, identifying the modal μ-calculus as the bisimulation invariant fragment of MSO [3]. Given the fact that in many applications bisimilar models are considered to represent the *same* process, one has little interest in properties of models that are *not* bisimulation invariant. Thus the Janin-Walukiewicz Theorem can be seen as an expressive completeness result, stating that all *relevant* properties in monadic second-order logic can be expressed in the modal μ-calculus.

Coalgebra enters naturally into this picture. Recall that Universal Coalgebra [4] provides the notion of a *coalgebra* as the natural mathematical generalization of state-based evolving systems such as streams, (infinite) trees, Kripke models, (probabilistic) transition systems, and many others. This approach combines simplicity with generality and wide applicability: many features, including input, output, nondeterminism, probability, and interaction, can easily be encoded in the coalgebra type T (formally an endofunctor on the category $\mathbf{Set}$ of sets as objects with functions as arrows). Starting with Moss' seminal paper [5], coalgebraic logics have been developed for the purpose of specifying and reasoning about *behavior*, one of the most fundamental concepts that allows for a natural coalgebraic formalization. And with Kripke structures constituting key examples of coalgebras, it should come as no surprise that most coalgebraic logics are some kind of modification or generalization of *modal logic* [6].

The coalgebraic modal logics that we consider here originate with Pattinson [7]; they are characterized by a completely standard syntax, in which the semantics of each modality is determined by a so-called *predicate lifting* (see Definition 2 below). Many well-known variations of modal logic in fact arise as the coalgebraic logic $\mathtt{ML}_\Lambda$ associated with a set Λ of such predicate liftings; examples include both standard and (monotone) neighborhood modal logic, graded and probabilistic modal logic, coalition logic, and conditional logic. Extensions of coalgebraic modal logics with fixpoint operators, needed for describing *ongoing* behavior, were developed in [8],

1043-6871/15 $31.00
DOI 10.1109/LICS.2015.41

[9].

The link between coalgebra and automata theory is by now well-established. For instance, finite state automata operating on finite words have been recognized as key examples of coalgebra from the outset [4]. More relevant for the purpose of this paper is the link with precisely the kind of automata mentioned earlier, since the (potentially infinite) objects on which these devices operate, such as streams, trees and Kripke frames, usually are coalgebras. Thus, the automata-theoretic perspective on modal fixpoint logic could be lifted to the abstraction level of coalgebra [8], [10]. In fact, many key results in the theory of automata operating on infinite objects, such as Muller & Schupp's Simulation Theorem [11] can in fact be seen as instances of more general theorems in Universal Coalgebra [12].

B. Coalgebraic monadic second-order logic

Missing from this picture is, to start with, a coalgebraic version of *(monadic) second-order logic*. Filling this gap is the first aim of the current paper, which introduces a notion of *monadic second-order logic* $\mathtt{MSO}_{\mathsf{T}}$ for coalgebras of type T. Our formalism combines two ideas from the literature. First of all, we looked for inspiration to the coalgebraic versions of *first-order logic* of Litak & alii [13]. These authors introduced Coalgebraic Predicate Logic as a common generalisation of first-order logic and coalgebraic modal logic, combining first-order quantification with coalgebraic syntax based on predicate liftings. Our formalism $\mathtt{MSO}_{\mathsf{T}}$ will combine a similar syntactic feature with second-order quantification. Second, following the tradition in automata-theoretic approaches towards monadic second-order logic, our formalism will be *one-sorted*. That is, we *only* allow second-order quantification in our language, relying on the fact that individual quantification, when called for, can be encoded as second-order quantification relativized to singleton sets. Since predicate liftings are defined as families of maps on powerset algebras, these two ideas fit together very well, to the effect that our second-order logic is in some sense simpler than the first-order formalism of [13].

In section III we will define, for any set Λ of monotone[1] predicate liftings, a formalism $\mathtt{MSO}_{\Lambda}$, and we let $\mathtt{MSO}_{\mathsf{T}}$ denote the logic obtained by taking for Λ the set of *all* monotone predicate liftings. Clearly we will make sure that this definition generalizes the standard case, in the sense that the standard version of $\mathtt{MSO}$ for Kripke structures instantiates the logic $\mathtt{MSO}_{\{\Diamond\}}$ and is equivalent to the coalgebraic logic $\mathtt{MSO}_{\mathcal{P}}$ (where $\mathcal{P}$ denotes the power set functor).

The introduction of a monadic second-order logic $\mathtt{MSO}_{\mathsf{T}}$ for T-coalgebras naturally raises the question, for which T does the coalgebraic modal μ-calculus for T correspond to the bisimulation-invariant fragment of $\mathtt{MSO}_{\mathsf{T}}$.

Question 1. *Which functors* T *satisfy* $\mu\mathtt{ML}_{\mathsf{T}} \equiv \mathtt{MSO}_{\mathsf{T}}/{\simeq}$*?*

[1] In the most general case, restricting to monotone predicate liftings is not needed, one could define $\mathtt{MSO}_{\mathsf{T}}$ as the logic obtained by taking for Λ the set of *all* predicate liftings. However, in the context of this paper, where we take an automata-theoretic perspective on $\mathtt{MSO}$, this restriction makes sense.

C. Automata for coalgebraic monadic second-order logic

In order to address Question 1, we take an automata-theoretic perspective on the logics $\mathtt{MSO}_{\mathsf{T}}$ and $\mu\mathtt{ML}_{\mathsf{T}}$, and as the second contribution of this paper we introduce a class of parity automata for $\mathtt{MSO}_{\mathsf{T}}$.

As usual, the operational semantics of our automata is given in terms of a two-player acceptance game, which proceeds in *rounds* moving from one basic position to another, where a basic position is a pair consisting of a state of the automaton and a point in the coalgebra structure under consideration. In each round, the two players, $\exists$ and $\forall$, focus on a certain local 'window' on the coalgebra structure. This 'window' takes the shape of a *one-step* T*-model*, that is, a triple (X, α, V) consisting of a set X, a *chosen object* $\alpha \in \mathsf{T}X$, and a valuation V interpreting the states of the automaton as subsets of X. More specifically, during each round of the game it is the task of $\exists$ to come up with a valuation V that creates a one-step model in which a certain *one-step formula* δ (determined by the current basic position in the game) is true.

Generally, our automata will have the shape $\mathbb{A} = (A, \Delta, \Omega, a_I)$ where A is a finite carrier set with initial state $a_I \in A$, and Ω and Δ are the parity and transition map of $\mathbb{A}$, respectively. The flavour of such an automaton is largely determined by the co-domain of its transition map Δ, the so-called *one-step language* which consists of the one-step formulas that feature in the acceptance game as described.

Each one-step language $\mathtt{L}$ induces its own class of automata $Aut(\mathtt{L})$. For instance, the class of automata corresponding to the coalgebraic fixpoint logic $\mu\mathtt{ML}_{\Lambda}$ can be given as $Aut(\mathtt{ML}_{\Lambda})$, where $\mathtt{ML}_{\Lambda}$ is the set of positive modal formulas of depth one that use modalities from Λ [10]. Basically then, the problem of finding the right class of automata for the coalgebraic monadic second-order logic $\mathtt{MSO}_{\Lambda}$ consists in the identification of an appropriate one-step language. Our proposal comprises a one-step *second-order logic* which uses predicate liftings to describe the chosen object of the one-step model.

Finally, note that similar to the case of standard $\mathtt{MSO}$, the equivalence between formulas in $\mathtt{MSO}_{\mathsf{T}}$ and automata in $Aut(\mathtt{SO})$ is only guaranteed to hold for coalgebras that are 'tree-like' in some sense (to be defined further on).

Theorem 1 (Automata for coalgebraic $\mathtt{MSO}$)**.** *For any set* Λ *of monotone predicate liftings for* T *there is an effective construction mapping any formula* $\varphi \in \mathtt{MSO}_{\Lambda}$ *into an automaton* $\mathbb{A}_{\varphi} \in Aut(\mathtt{SO}_{\Lambda})$*, which is equivalent to* φ *over* T*-tree models.*

The proof of Theorem 1 proceeds by induction on the complexity of $\mathtt{MSO}_{\mathsf{T}}$-formulas, and thus involves various *closure properties* of automata, such as closure under complementation, union and projection. In order to establish these results, it will be convenient to take an *abstract* perspective, revealing how closure properties of a class of automata are completely determined at the level of the one-step language.

D. Bisimulation Invariance

With automata-theoretic characterizations in place for both coalgebraic $\mathtt{MSO}$ and the coalgebraic μ-calculus $\mu\mathtt{ML}$, we can address Question 1 by considering the following question:

Question 2. *Which functors* T *satisfy* $Aut(\mathtt{ML}) \equiv Aut(\mathtt{SO})/{\simeq}$*?*

Continuing the program of the third author [14], we will approach this question *at the level of the one-step languages*, SO and ML. To start with, observe that any translation (from one-step formulas in) SO to (one-step formulas in) ML naturally induces a translation from SO-automata to ML-automata. A new observation we make here is that any so-called *uniform construction* on the class of one-step models for the functor T that satisfies certain *adequacy* conditions, provides (1) a translation $(\cdot)^* : \mathtt{SO} \to \mathtt{ML}$, together with (2) a construction $(\cdot)_*$ transforming a pointed T-model $(\mathbb{S}, s)$ into a tree model $(\mathbb{S}_*, s_*)$ which is a coalgebraic pre-image of $(\mathbb{S}, s)$ satisfying

$$\mathbb{A} \text{ accepts } (\mathbb{S}_*, s_*) \text{ iff } \mathbb{A}^* \text{ accepts } (\mathbb{S}, s).$$

From this it easily follows that an SO-automaton $\mathbb{A}$ is bisimulation invariant iff it is equivalent to the ML-automaton $\mathbb{A}^*$.

On the basis of these observations we can prove the following generalisation of the Janin-Walukiewicz Theorem.

Theorem 2 (Coalgebraic Bisimulation Invariance)**.** *If the set functor* T *admits an adequate uniform construction, then*

$$\mu\mathtt{ML}_{\mathsf{T}} \equiv \mathtt{MSO}_{\mathsf{T}}/{\simeq}.$$

In our eyes, the significance of Theorem 2 is twofold. First of all, the proof separates the 'clean', abstract part of bisimulation-invariance results from the more functor-specific parts. As a consequence, Theorem 2 can be used to obtain immediate results in particular cases. Examples include the power set functor (standard Kripke structures), where the adequate uniform construction roughly consists of taking ω-fold products (see Example 1), the bag functor (Example 2), and all exponential polynomial functors (Corollary 3). Second, in case the functor does *not* admit an adequate uniform construction, Theorem 2 may still be of use in proving alternative characterization results for the functor.

Instantiating the latter phenomenon is the *monotone neighborhood functor* $\mathcal{M}$ (see the next section for its definition). The importance of this functor lies, among other things, in it providing a coalgebraic semantics for monotone modal logic [15]. The coalgebraic monadic second-order language $\mathtt{MSO}_{\mathcal{M}}$ is equivalent to a natural second-order language for reasoning about monotone neighborhood structures that we shall denote by MMSO, and $\mu\mathtt{ML}_{\mathcal{M}}$ is equivalent to the fixpoint-extension of monotone modal logic, denoted μMML. As we shall see in Proposition 12 below, $\mathcal{M}$ does *not* admit an adequate uniform construction.[2] This, however, is not the end of the story. It turns out that we *can* find an adequate uniform construction for a *variant* $\mathcal{M}^\star$ of the functor $\mathcal{M}$ (see Proposition 14). As a corollary, we obtain a characterization of the fragment of MMSO that is invariant under *global* bisimulations (bisimulations that are full on both domain and codomain). This fragment turns out to be exactly the extension of the monotone μ-calculus with the global modalities (for precise definitions we refer to section VI), which we shall denote $\mu\mathtt{MML}_g$.

In this notation, our final contribution is the following characterization result:

Theorem 3. *A formula in* MMSO *is invariant for global neighborhood bisimulations if, and only if, it is equivalent to a formula of the logic* $\mu\mathtt{MML}_g$.

[2]This does not mean that the monotone μ-calculus $\mu\mathtt{ML}_{\mathcal{M}}$ does *not* correspond to the bisimulation-invariant fragment of $\mathtt{MSO}_{\mathcal{M}}$, but it does mean that a proof of such a result will necessarily involve techniques that differ from the ones employed here.

II. Some technical background

In this paper we assume familiarity with the basic theory of modal (fixpoint) logic, monadic second-order logic, coalgebra, coalgebraic modal (fixpoint) logic, and parity games. Here we fix some notation and terminology.

A. Kripke models and their logics

We restrict to the theory of modal logic with one modality (and hence, one accessibility relation). Let *Var* be a fixed infinite supply of variables. A *Kripke model* is a structure $\mathbb{S} = (S, R, V)$ where S is a set, $R \subseteq S \times S$ and $V : \mathit{Var} \to \mathcal{P}(S)$ is a *Var*-valuation. Associated with such a valuation V, we define the *conjugate coloring* $V^\dagger : S \to \mathcal{P}(\mathit{Var})$ by $V^\dagger(s) := \{p \in \mathit{Var} \mid s \in V(p)\}$. Given a subset $T \subseteq S$, the valuation $V[p \mapsto T]$ is as V except that it maps the variable p to T. A *pointed* Kripke model is a structure $(\mathbb{S}, u)$ where $\mathbb{S}$ is a Kripke model and u is a point in $\mathbb{S}$. Turning to syntax, we define the formulas of monadic second-order logic MSO through the following grammar:

$$\varphi ::= \mathtt{sr}(p) \mid p \subseteq q \mid R(p,q) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists p.\varphi,$$

with $p, q \in \mathit{Var}$. Formulas are evaluated over pointed Kripke models by the following induction:

- $(S, R, V, u) \vDash \mathtt{sr}(p)$ iff $V(p) = \{u\}$
- $(S, R, V, u) \vDash p \subseteq q$ iff $V(p) \subseteq V(q)$
- $(S, R, V, u) \vDash R(p,q)$ iff for all $v \in V(p)$ there is $w \in V(q)$ with vRw
- standard clauses for the boolean connectives
- $(S, R, V, u) \vDash \exists p.\varphi$ iff $(S, R, V[p \mapsto T], u) \vDash \varphi$ for some $T \subseteq S$.

We present the language of the modal μ-calculus μML in negation normal form, by the following grammar:

$$\varphi ::= p \mid \neg p \mid \bot \mid \top \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \eta p.\varphi$$

where $p \in \mathit{Var}$, $\eta \in \{\mu, \nu\}$, and in the formula $\eta p.\varphi$ no free occurrence of the variable p may be in the scope of a negation.

The satisfaction relation between pointed Kripke models and formulas in μML is defined by the usual induction, with, e.g.

- $(S, R, V, u) \vDash \mu p.\varphi$ iff $u \in \bigcap\{Z \subseteq S \mid \varphi_p(Z) \subseteq Z\}$ where $\varphi_p(Z)$ denotes the truth set of the formula φ in the model $(S, R, V[p \mapsto Z])$.

We assume familiarity with the notion of bisimilarity between two (pointed) Kripke models, and say that a formula of

$\mathtt{MSO}$ is *bisimulation invariant* if it has the same truth value in any pair of bisimilar pointed Kripke models.

Fact 1. *[3] A formula φ of* $\mathtt{MSO}$ *is equivalent to a formula of* $\mu\mathtt{ML}$ *iff φ is invariant for bisimulations.*

B. Coalgebras and models

Our basic semantic structures consist of coalgebras together with valuations. We only consider coalgebras over the base category $\mathbf{Set}$ with sets as objects and functions as arrows. The co- and contravariant power set functors will be denoted by $\mathcal{P}$ and $\mathcal{Q} : \mathbf{Set} \to \mathbf{Set}^{op}$, respectively. Covariant endofunctor on $\mathbf{Set}$ will be called *set functors*.

Definition 1. Let T be a set functor. A T-*coalgebra* is a pair (S, σ) consisting of a set S, together with a map $\sigma : S \to \mathsf{T}S$. A T-*model* is a structure $\mathbb{S} = (S, \sigma, V)$ where (S, σ) is a T-coalgebra and $V : \mathit{Var} \to \mathcal{P}S$. A *pointed* T-model is a structure $(\mathbb{S}, s)$ where $\mathbb{S}$ is a T-model and $s \in S$.

The usual notion of a p-morphism between Kripke models can be generalized as follows: Let $\mathbb{S}_1 = (S_1, \sigma_1, V_1)$ and $\mathbb{S}_2 = (S_2, \sigma_2, V_2)$ be two T-models and let $f : S_1 \to S_2$ be any map. Then f is said to be a T-*model homomorphism* if:

1) for each variable p and each $u \in S_1$, we have $u \in V_1(p)$ iff $f(u) \in V_2(p)$;
2) the map f is a *coalgebra morphism*, i.e. we have

$$\sigma_2 \circ f = \mathsf{T}f \circ \sigma_1.$$

Two pointed coalgebras $(\mathbb{S}, s)$ and $(\mathbb{S}', s')$ are *behaviorally equivalent*, notation: $\mathbb{S}, s \simeq \mathbb{S}', s'$, if s and s' can be identified by coalgebra morphisms $f : \mathbb{S} \to \mathbb{T}$ and $f' : \mathbb{S}' \to \mathbb{T}$ such that $f(s) = f'(s')$.

A *coalgebraic logic* consists of a set $\mathtt{L}$ of *formulas* together with, for each coalgebra (S, σ), a truth or satisfaction relation $\Vdash \subseteq S \times \mathtt{L}$. A formula φ is called *bisimulation invariant*[3] if $\mathbb{S}, s \Vdash \varphi \iff \mathbb{S}', s' \Vdash \varphi$ whenever $\mathbb{S}, s \simeq \mathbb{S}', s'$.

Kripke frames are coalgebras for the (covariant) power set functor $\mathcal{P}$. A functor of particular interest in this paper is the *monotone neighborhood* functor $\mathcal{M}$, usually defined as the subfunctor of $\mathcal{Q} \circ \mathcal{Q}$ given by setting $\mathcal{M}X \subseteq \mathcal{Q}\mathcal{Q}X$ to be:

$$\{N \in \mathcal{Q}\mathcal{Q}X \mid \forall Z, Z' : Z \in N \ \& \ Z \subseteq Z' \Rightarrow Z' \in N\}$$

This functor comes equipped with the following notion of bisimilarity. A *neighborhood bisimulation* between $\mathcal{M}$-models $\mathbb{S}_1$ and $\mathbb{S}_2$ is a relation $R \subseteq S_1 \times S_2$ such that, if $s_1 R s_2$ then:

- $V_1^\dagger(s_1) = V_2^\dagger(s_2)$;
- for all Z_1 in $\sigma_1(s_1)$ there is Z_2 in $\sigma_2(s_2)$ such that for all $t_2 \in Z_2$ there is $t_1 \in Z_1$ with $t_1 R t_2$;
- for all Z_2 in $\sigma_2(s_2)$ there is Z_1 in $\sigma_1(s_1)$ such that for all $t_1 \in Z_1$ there is $t_2 \in Z_2$ with $t_1 R t_2$.

[3]Strictly speaking, behavioral equivalence and bisimilarity are distinct concepts. However, in many concrete cases, behavioural equivalence and bisimilarity coincide, so we shall be content to use the more common parlance of "bisimulation invariance" rather than "invariance for behavioural equivalence".

C. Coalgebraic μ-calculus & coalgebra automata

The modal μ-calculus is just one in a family of logical systems that may collectively be referred to as the *coalgebraic μ-calculus* [9]. These logics essentially make use of *predicate liftings*.

Definition 2. Given a set functor T, an *n-place predicate lifting* for T is a natural transformation

$$\lambda : \mathcal{Q}(-)^n \to \mathcal{Q} \circ \mathsf{T},$$

where $\mathcal{Q}(-)^n$ denotes the n-fold product of $\mathcal{Q}$ with itself. A predicate lifting λ is said to be *monotone* if

$$\lambda_X(Y_1, ..., Y_n) \subseteq \lambda_X(Z_1, ..., Z_n),$$

whenever $Y_i \subseteq Z_i$ for each i. The *Boolean dual* λ^d of λ is defined by

$$(Z_1, ..., Z_n) \mapsto \mathsf{T}X \setminus (\lambda_X(X \setminus Z_1, ..., X \setminus Z_n)).$$

Given a set functor T, the language $\mu\mathtt{ML}_\mathsf{T}$ of the coalgebraic μ-calculus for T is defined thus:

$$\varphi ::= p \mid \neg p \mid \bot \mid \top \mid \lambda(\varphi_1, ..., \varphi_n) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \eta p.\varphi$$

where $p \in \mathit{Var}$, λ is any monotone n-place predicate lifting for T, $\eta \in \{\mu, \nu\}$, and, in $\eta p.\varphi$, no free occurrence of the variable p is in the scope of a negation. If we restrict the formulas $\lambda(\varphi_1, ..., \varphi_n)$ so that λ must come frome some distinguished set of liftings Λ, then we denote the corresponding sublanguage of $\mu\mathtt{ML}_\mathsf{T}$ by $\mu\mathtt{ML}_\Lambda$.

The semantics of formulas in a pointed T-model is defined as follows:

- $(\mathbb{S}, s) \vDash p$ iff $s \in V(p)$ and $(\mathbb{S}, s) \vDash \neg p$ iff $s \notin V(p)$
- $(\mathbb{S}, s) \vDash \lambda(\varphi_1, ..., \varphi_n)$ iff $\sigma(s) \in \lambda_S(\|\varphi_1\|, ..., \|\varphi_n\|)$, where $\|\varphi_i\| = \{t \in S \mid (\mathbb{S}, t) \vDash \varphi_i\}$ denotes the "truth set" of φ_i in $\mathbb{S}$
- standard clauses for the boolean connectives
- $(\mathbb{S}, s) \vDash \mu p.\varphi$ iff $s \in \bigcap\{X \subseteq S \mid \varphi_p(X) \subseteq X\}$, where $\varphi_p(Z)$ denotes the truth set of the formula φ in the T-model $(S, \sigma, V[p \mapsto Z])$.

It is routine to prove that all formulas in $\mu\mathtt{ML}_\mathsf{T}$ are bisimulation invariant.

Turning to the parity automata corresponding to the language $\mu\mathtt{ML}_\Lambda$, we first define the *modal one-step language* $\mathtt{ML}^1_\Lambda$. Its set $\mathtt{ML}^1_\Lambda(A)$ of *modal one-step formulas* over a set A of variables is given by the following grammar:

$$\varphi ::= \bot \mid \top \mid \lambda(\psi_1, ..., \psi_n) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$

where $\psi_1, ..., \psi_n$ are formulas built up from variables in A using disjunctions and conjunctions.

Definition 3. Given a functor T and a set of variables A, a *one-step model* over A is a triple (X, α, V) where X is any set, $\alpha \in \mathsf{T}X$ and $V : A \to \mathcal{P}(X)$ is a valuation.

The semantics of formulas in the modal one-step language in a one-step model is given as follows:

- standard clauses for the boolean connectives,

- $(X, \alpha, V) \vDash_1 \lambda(\psi_1, ..., \psi_n)$ iff $\alpha \in \lambda_X(\|\psi_1\|, ..., \|\psi_n\|)$, where $\|\psi_i\| \subseteq X$ is the (classical) truth set of the formula ψ_i under the valuation V.

We can now define the class of automata used to characterize the coalgebraic μ-calculus.

Definition 4. Let P be a finite set of variables and Λ a set of predicate liftings. Then a *(P-chromatic) modal Λ-automaton* is a tuple (A, Δ, Ω, a_I) where A is a finite set of states with $a_I \in A$,

$$\Delta : A \times \mathcal{P}(P) \to \mathtt{ML}^1_\Lambda(A)$$

is the transition map of the automaton, and $\Omega : A \to \omega$ is the parity map. The class of these automata is denoted as $Aut(\mathtt{ML}_\Lambda)$.

The acceptance game for an automaton $\mathbb{A} = (A, \Delta, \Omega, a_I)$ and a T-model (S, σ, V) is given by the following table:

Position	Pl'r	Admissible moves
$(a, s) \in A \times S$	$\exists$	$\{U \in (\mathcal{P}S)^A \mid (S, \sigma(s), U) \vDash_1 \Delta(a, V^\dagger(s))\}$
$U : A \to \mathcal{P}S$	$\forall$	$\{(b, t) \mid t \in U(b)\}$

The loser of a finite match is the player who got stuck, and the winner of an infinite match is $\exists$ if the greatest parity that appears infinitely often in the match is even, and the winner is $\forall$ if this parity is odd. The automaton $\mathbb{A}$ *accepts* the pointed model $(\mathbb{S}, s)$ if $\exists$ has a winning strategy in the acceptance game from the starting position (a_I, s). We say that an automaton $\mathbb{A}$ is *equivalent* to a formula $\varphi \in \mu\mathtt{ML}_\Lambda$ if, for every pointed T-model $(\mathbb{S}, s)$, we have that $\mathbb{A}$ accepts $(\mathbb{S}, s)$ iff $(\mathbb{S}, s) \vDash \varphi$.

Fact 2. *[10] Let T be a set functor, and Λ a set of monotone predicate liftings for T, closed under Boolean duals. Then*

$$\mu\mathtt{ML}_\Lambda \equiv Aut(\mathtt{ML}_\Lambda).$$

That is, there are effective transformations of formulas in $\mu\mathtt{ML}_\Lambda$ into equivalent automata in $Aut(\mathtt{ML}_\Lambda)$, and vice versa.

III. Coalgebraic MSO

We now introduce coalgebraic monadic second-order logic for a set functor T and a set of liftings Λ and show how MSO can be recovered as a special case. We define the syntax of the monadic second-order logic $\mathtt{MSO}_\mathsf{T}$ by the following grammar:

$$\varphi ::= \bot \mid \mathtt{sr}(p) \mid p \subseteq q \mid \lambda(p, q_1, .., q_n) \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists p.\varphi$$

where λ is any n-place monotone predicate lifting and $p, q, q_1, ..., q_n \in Var$. More generally, restricting to a set Λ of monotone liftings for T, we define the sublanguage $\mathtt{MSO}_\Lambda \subseteq \mathtt{MSO}_\mathsf{T}$ by the same grammar except that we require the liftings to be in Λ.

For the semantics, let $(\mathbb{S}, s)$ be a pointed T-model. We define the satisfaction relation $\vDash \subseteq S \times \mathtt{MSO}_\mathsf{T}$ as follows:

- $(\mathbb{S}, u) \vDash \mathtt{sr}(p)$ iff $V(p) = \{u\}$,
- $(\mathbb{S}, u) \vDash p \subseteq q$ iff $V(p) \subseteq V(q)$,
- $(\mathbb{S}, u) \vDash \lambda(p, q_1, ..., q_n)$ iff $\sigma(v) \in \lambda_S(V(q_1), .., V(q_n))$ for all $v \in V(p)$,
- standard clauses for the Boolean connectives
- $(\mathbb{S}, u) \vDash \exists p.\varphi$ iff $(S, \sigma, V[p \mapsto Z], u) \vDash \varphi$, some $Z \subseteq S$.

We introduce the following abbreviations:

- $p = q$ for $p \subseteq q \wedge q \subseteq p$,
- $\mathtt{Em}(p)$ for $\forall q.(q \subseteq p \to q = p)$,
- $\mathtt{Sing}(p)$ for $\neg\mathtt{Em}(p) \wedge \forall q(q \subseteq p \to (\mathtt{Em}(q) \vee q = p))$

expressing, respectively, that p and q are equal, that p denotes the empty set, and that p denotes a singleton.

Clearly, standard MSO is the logic $\mathtt{MSO}_{\{\Diamond\}}$, where $\Diamond$ is the predicate lifting corresponding to the usual diamond modality over Kripke models. Obviously then, $\mathtt{MSO}_\mathcal{P}$ contains MSO. In order to see that the languages are in fact equivalent in expressive power, we need the notion of *expressive completeness*, which plays an important role in this paper.

Definition 5. A set of monotone liftings Λ for a set functor T is said to be *expressively complete* if, for every finite set of variables A and every monotone predicate lifting $\lambda : \mathcal{Q}(-)^A \to \mathcal{Q} \circ \mathsf{T}$, there exists a formula $\varphi \in \mathtt{ML}^1_\Lambda(A)$ such that, for every one-step model (X, α, V) with $V : A \to \mathcal{Q}(X)$, we have

$$(X, \alpha, V) \vDash_1 \varphi \text{ iff } \alpha \in \lambda_X(V).$$

If Λ is expressively complete, then clearly $\mu\mathtt{ML}_\Lambda$ is equivalent in expressive power to the full language $\mu\mathtt{ML}_\mathsf{T}$. It is not much harder to show that, under the same conditions, $\mathtt{MSO}_\Lambda$ is equivalent in expressive power to the full language $\mathtt{MSO}_\mathsf{T}$. Furthermore, expressive completeness can often be obtained fairly easily if we make use of an application of the Yoneda lemma to represent n-place predicate liftings as subsets of $\mathsf{T}(2^n)$, a method developed in [16]. In particular, since the liftings $\{\Box, \Diamond\}$ for $\mathcal{P}$ are expressively complete and $\Box$ is clearly definable in $\mathtt{MSO}_{\{\Diamond\}}$, one can show that $\mathtt{MSO} = \mathtt{MSO}_{\{\Diamond\}}$ is equivalent in expressive power to the full coalgebraic logic $\mathtt{MSO}_\mathcal{P}$. Furthermore, $\mu\mathtt{ML}_\mathcal{P}$ is equivalent to $\mu\mathtt{ML}_{\{\Box,\Diamond\}}$. As a second example, involving the monotone neighborhood functor $\mathcal{M}$, let $\Box$ here be the predicate lifting defined by $\alpha \in \Box_X(Z)$ iff $Z \in \alpha$, and let $\Diamond$ be its dual. Then the language $\mathtt{MSO}_\mathcal{M}$ is equivalent to $\mathtt{MSO}_{\{\Box,\Diamond\}}$, and also $\mu\mathtt{ML}_\mathcal{M}$ is equivalent to $\mu\mathtt{ML}_{\{\Box,\Diamond\}}$.

Finally, as mentioned in the introduction, the key question in this paper will be to compare the expressive power of coalgebraic monadic second-order logic to that of the coalgebraic μ-calculus. The following observation, of which the (routine) proof is omitted, provides the easy part of the link.

Proposition 1. *Let Λ be a set of monotone predicate lfitings for the set functor T. There is an inductively defined translation $(\cdot)^\diamond$ mapping any formula $\varphi \in \mu\mathtt{ML}_\Lambda$ to an equivalent formula $\varphi^\diamond \in \mathtt{MSO}_\Lambda$.*

IV. Automata for coalgebraic MSO

In this section we introduce automata for coalgebraic monadic second-order logic.

A. A general perspective on parity automata

Standard monadic second-order formulas can be translated to equivalent automata over *trees*, but this equivalence is not guaranteed to extend to arbitrary Kripke models. In the case of general coalgebra, we should expect having to introduce a coalgebraic concept of "tree-like" models.

Definition 6. Given a set S and $\alpha \in \mathsf{T}S$, a subset $X \subseteq S$ is said to be a *support* for α if there is some $\beta \in \mathsf{T}X$ with $\mathsf{T}\iota_{X,S}(\beta) = \alpha$. A *supporting Kripke frame* for a T-coalgebra (S, σ) is a binary relation $R \subseteq S \times S$ such that, for all $u \in S$, $R(u) = \{v \mid uRv\}$ is a support for $\sigma(u)$.

Definition 7. A T*-tree model* is a structure $(\mathbb{S}, R, u)$ where $\mathbb{S} = (S, \sigma, V)$ is a T-model and $u \in S$, such that R is a supporting Kripke frame for the coalgebra (S, σ), and furthermore (S, R) is a tree rooted at u, so that there is a unique R-path from u to w for each $w \in S$.

Our goal is to translate formulas in $\mathtt{MSO}_\mathsf{T}$ to equivalent automata over T-tree models. We start by introducing a very general type of automaton, originating with [14].

Definition 8. Given a finite set A, a *generalized predicate lifting* over A comprises an assignment of a map

$$\varphi_X : (\mathcal{Q}X)^A \to \mathcal{Q}\mathsf{T}X.$$

to every set X. Concepts like *Boolean dual* and *monotonicity* apply to these liftings in the obvious way.

The difference with respect to standard predicate liftings is that the components of a generalized predicate lifting do not need to form a natural transformation.[4]

Definition 9. A *one-step language* $\mathtt{L}$ consists of a collection $\mathtt{L}(A)$ of generalized predicate liftings for every finite set A. The semantics of a generalized predicate lifting φ in a one-step model (X, α, V) is given by

$$(X, \alpha, V) \vDash_1 \varphi \textit{ iff } \alpha \in \varphi_X(V).$$

Our automata will be indexed by a (finite) set of variables involved, corresponding to the set of free variables of the $\mathtt{MSO}_\mathsf{T}$-formula.

Definition 10. Let $P \subseteq \mathit{Var}$ be a finite set of variables and let $\mathtt{L}$ be a one-step language for functor T. A *(P-chromatic)* $\mathtt{L}$*-automaton* is a structure (A, Δ, Ω, a_I) where

- A is a finite set, with $a_I \in A$,
- $\Omega : A \to \omega$ is a parity map, and
- $\Delta : A \times \mathcal{P}(P) \to \mathtt{L}(A)$ is the transition map of $\mathbb{A}$.

The *acceptance game* of $\mathbb{A}$ with respect to a T-tree model (T, R, σ, V, u) is given by Table I. We say that the automaton $\mathbb{A}$ accepts the model (T, R, σ, V, u) if $\exists$ has a winning strategy in this game (initialized at position (a_I, u)).

[4]In the style of abstract logic, it would make sense to require a general predicate lifting to be natural with respect to certain maps, in particular, bijections. For the purpose of this paper such a restriction is not needed, however.

B. Closure properties

This abstract level is useful for establishing some simple closure properties of automata, based on properties of the one-step language. The first, easy, results establish sufficient conditions for closure under union and complementation.

Proposition 2. *If the one-step language* $\mathtt{L}$ *is closed under disjunction, then the class of* $\mathtt{L}$*-automata is closed under union.*

Proposition 3. *If the monotone fragment of the one-step language* $\mathtt{L}$ *is closed under Boolean duals, then the class of* $\mathtt{L}$*-automata is closed under complementation.*

The most interesting property concerns closure under existential projection. The following terminology is taken from [3], but instead of relying on a particular syntactic shape of one-step formulas, we define the concepts in purely semantic terms.

Definition 11. A predicate lifting φ over A is said to be *special basic* if, for every one-step model (X, α, V) such that

$$(X, \alpha, V) \vDash_1 \varphi$$

there is a valuation $V^* : A \to \mathcal{Q}(X)$ such that

- $V^*(a) \subseteq V(a)$ for each $a \in A$,
- $V^*(a) \cap V^*(b) = \emptyset$ whenever $a \neq b$, and
- $(X, \alpha, V^*) \vDash_1 \varphi$.

Call an $\mathtt{L}$-automaton *non-deterministic* if every lifting $\Delta(a, c)$ is special basic.

It is easy to see that if the language $\mathtt{L}$ is closed under disjunctions, then so is its fragment of special basic liftings. From this we obtain the following.

Proposition 4. *If the one-step language* $\mathtt{L}$ *is closed under disjunction, then the class of non-deterministic* $\mathtt{L}$*-automata is closed under existential projection over* T*-tree models.*

Proof: Suppose $\mathbb{A} = (A, \Delta, a_I, \Omega)$ is a non-deterministic $\mathtt{L}$-automaton for the variable set P. Define the $P \setminus q$-chromatic automaton $\exists q.\mathbb{A} = (A, \Delta^*, a_I, \Omega)$ by setting

$$\Delta^*(a, c) = \Delta(a, c) \vee \Delta(a, c \cup \{q\}).$$

It is easy to see that every T-tree model accepted by $\mathbb{A}$ is also accepted by $\exists p.\mathbb{A}$. Conversely, suppose $\exists p.\mathbb{A}$ accepts some T-tree model (S, R, σ, V, s_I). For each winning position (a, s) in the acceptance game, let $V_{(a,s)}$ be the valuation chosen by $\exists$ according to some given winning strategy χ. Note that we can assume that χ is a *positional* winning strategy, since $\exists p.\mathbb{A}$ is a parity automaton. It is not difficult to see that the automaton $\exists p.\mathbb{A}$ is a non-deterministic automaton, and so for each winning position (a, s) there is a valuation $V^*_{(a,s)} : A \to \mathcal{P}(R(s))$, which is an admissible move for $\exists$, such that $V^*_{(a,s)}(b) \subseteq V_{(a,s)}(b)$ and such that for all $b_1 \neq b_2 \in A$ we have $V^*_{(a,s)}(b_1) \cap V^*_{(a,s)}(b_2) = \emptyset$. Define the strategy χ^* by letting $\exists$ choose the valuation $V^*_{(a,s)}$ at each winning position (a, s) - this is still a winning strategy, since the valuations chosen by $\exists$ are smaller and so no new

Position	Player	Admissible moves	Parity
$(a,s) \in A \times T$	$\exists$	$\{U : A \to \mathcal{P}(R(s)) \mid (R(s), \sigma(s), U) \vDash_1 \Delta(a, V^\dagger(s))\}$	$\Omega(a)$
$U : A \to \mathcal{P}(T)$	$\forall$	$\{(b,t) \mid t \in U(b)\}$	0

TABLE I
ACCEPTANCE GAME FOR PARITY AUTOMATA.

choices for $\forall$ are introduced. Furthermore, χ^* is clearly still a positional winning strategy.

From these facts it follows by a simple induction on the depth of the nodes in the supporting tree that the strategy χ^* is *scattered*, i.e. that for every $s \in S$ there is at most one automaton state a such that (a,s) appears in a χ^*-guided match of the acceptance game. So we can define a valuation V' like V except we evaluate q to be true at all and only the states s such that

$$(R(s), \sigma(s), V^*_{(a_s,s)}) \vDash_1 \Delta(a_s, c \cup \{q\}),$$

where a_s is a necessarily *unique* automaton state such that (a,s) appears in some χ^*-guided match, and c is the color consisting of the variables true under V at s. It is not hard to show that $\mathbb{A}$ accepts (S, R, σ, V', s_I). ■

C. Second-order automata

We now introduce a more concrete one-step language for a given set functor T and a given set of (natural) liftings Λ, and show that $\mathtt{MSO}_\Lambda$ can be translated into the corresponding class of automata.

Let Λ be a set of monotone predicate liftings for T. The set of *second-order one-step formulas* over any set of variables A and relative to the set of liftings Λ is defined by the grammar:

$$\varphi ::= a \subseteq b \mid \lambda(a_1, ..., a_n) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists a.\varphi,$$

where $a, b, a_1, ..., a_n \in A$ and λ is any predicate lifting in Λ. Fixing an infinite set of "one-step variables" Var_1, and given a finite set A, the set of *second-order one-step sentences* over A, denoted $\mathtt{SO}^1_\Lambda(A)$, is the set of one-step formulas over $A \cup Var_1$, with all free variables belonging to A. We write $\mathtt{SO}^1_\mathsf{T}(A)$ when Λ comprises all monotone liftings for T.

The semantics of a one-step second-order A-formula in a one-step model (X, α, V) (with $V : A \to \mathcal{P}(X)$) is defined by the following clauses:

- $(X, \alpha, V) \vDash_1 p \subseteq q$ iff $V(p) \subseteq V(q)$,
- $(X, \alpha, V) \vDash_1 \lambda(p_1, ..., p_n)$ iff $\alpha \in \lambda_X(V(p_1), ..., V(p_n))$,
- standard clauses for the Boolean connectives,
- $(X, \alpha, V) \vDash_1 \exists p.\varphi$ iff $(X, \alpha, V[p \mapsto S]) \vDash_1 \varphi$ for some $S \subseteq X$.

Any one-step second-order A-sentence φ can be regarded as a generalized predicate lifting over A, with

$$\varphi_X(V) = \{\alpha \in \mathsf{T}X \mid (X, \alpha, V) \vDash_1 \varphi\}.$$

Note that the syntax of $\mathtt{SO}^1_\mathsf{T}$ allows negations, implying that not all these predicate liftings are monotone.

Definition 12. Let Λ be a set of monotone predicate liftings for T. A *second-order Λ-automaton* is an $\mathtt{L}$-automaton for $\mathtt{L}$ being the assignment of the one-step second-order A-sentences $\mathtt{SO}^1_\Lambda(A)$ to every set of variables A. We write $Aut(\mathtt{SO}_\Lambda)$ to denote this class, and $Aut(\mathtt{SO}_\mathsf{T})$ in case Λ is the set of *all* monotone predicate liftings for T.

Our aim is to prove that every formula of $\mathtt{MSO}_\Lambda$ can be translated into an equivalent second-order Λ-automaton (over rooted T-tree models), and the main problem here is to obtain closure under existential projection.

The key to this step is a simulation theorem.

First, a useful trick due to Walukiewicz [2] allows us to transform any second-order automaton into one in which all the one-step formulas are monotone, when regarded as generalized predicate liftings. We call such an automaton a *monotone* automaton.

Proposition 5. *Let Λ be any set of monotone predicate liftings. Every automaton $\mathbb{A} \in Aut(\mathtt{SO}_\Lambda)$ is equivalent to a monotone second-order $\mathbb{A} \in Aut(\mathtt{SO}_\Lambda)$.*

Proof: Enumerate A as $\{a_1, ..., a_k\}$, and just replace each formula $\Delta(a,c)$ by

$$\exists Z_1 ... \exists Z_k . Z_1 \subseteq a_1 \wedge ... \wedge Z_k \subseteq a_k \wedge \Delta(a,c)[Z_i/a_i]$$

where $\Delta(a,c)[Z_i/a_i]$ is the result of substituting the variable Z_i for each open variable a_i in $\Delta(a,c)$. This new formula is monotone in the variables A and the resulting automaton is equivalent to $\mathbb{A}$. ■

The intuition behind the simulation theorem is the same as that behind the standard "powerset construction" for word automata: the states of the new non-deterministic automaton $\mathbb{A}_n$ are "macro-states" representing several possible states of $\mathbb{A}$ at once. Formally, the states of $\mathbb{A}_n$ will be binary relations over A, and given a macro-state R, its range gives an exact description of the states in $\mathbb{A}$ that are currently being visited simultaneously. In fact, it is safe to think of the macro-states as subsets of A: the only reason that we have binary relations over A as states rather than just subsets is to have a memory device so that we can keep track of traces in infinite matches. For each macro-state R and each colour c we want to be able to say that the one-step formulas corresponding to each state in the range of R hold, so we want to translate the one-step formulas over A into one-step formulas over the set of macro-states. In order to translate a formula $\Delta(a,c)$ to a new one-step formula with macro-states as variables, we have to replace the variable b in $\Delta(a,c)$ with a new variable that acts as a stand-in for b. For this purpose we introduce a new, existentally quantified variable Z_b, together with a formula stating explicitly that Z_b is to represent the union of the values of all those macro states that contain b. Furthermore we want all the one-step formulas to be special basic, and for this purpose we simply add a conjunct "disj" to each one-step formula, stating that the values of any pair of distinct variables appearing in the formula are to be disjoint. Finally, in order to turn $\mathbb{A}_n$ into a parity automaton, we use a stream automaton to detect bad traces (see for instance [17] for the details in a more specific case). We omit the details of the proof.

Theorem 4 (Simulation)**.** *Let Λ be a set of monotone predicate liftings for T. For any monotone automaton $\mathbb{A} \in Aut(\mathtt{SO}_\Lambda)$ there exists an equivalent non-deterministic $\mathbb{A}' \in Aut(\mathtt{SO}_\Lambda)$.*

Combining Proposition 4 with Theorem 4, we easily obtain the following closure property.

Proposition 6. *Let Λ be a set of monotone predicate liftings for a set functor T. Over T-tree models, the class of second-order Λ-automata is closed under existential projection.*

We can now use the closure properties we have established for second-order automata to give the desired translation of $\mathtt{MSO}_\mathsf{T}$ into second-order automata.

Proposition 7. *For every formula $\varphi \in \mathtt{MSO}_\Lambda$ with free variables in P, there exists a P-chromatic automaton $\mathbb{A}_\varphi \in Aut(\mathtt{SO}_\Lambda)$ which is equivalent to φ over T-tree models.*

Proof: Proceeding by a straightforward induction on the complexity of φ, we leave it to the reader to construct appropriate automata for the atomic formulas. The inductive cases for disjunction and negation follow by the Propositions 2 and 3, together with the easy observation that the one-step language $\mathtt{SO}_\Lambda$ is closed under disjunction and Boolean duals. The case of existential quantification is taken care of by Proposition 6. ■

Theorem 1 is immediate from this, as is the following.

Corollary 1. *Suppose Λ is any set of monotone predicate liftings for T such that $\mathtt{MSO}_\mathsf{T} \equiv \mathtt{MSO}_\Lambda$. Then for every formula of $\mathtt{MSO}_\mathsf{T}$, there exists an equivalent second-order Λ-automaton over T-tree models. In particular, this holds whenever Λ is expressively complete.*

V. Bisimulation invariance

This section continues the program of [14], making use of the automata-theoretic translation of $\mathtt{MSO}_\mathsf{T}$ we have just established. The gist of our approach is that, in order to characterize a coalgebraic fixpoint logic $\mu\mathtt{ML}_\mathsf{T}$ as the bisimulation-invariant fragment of $\mathtt{MSO}_\mathsf{T}$, it suffices to establish a certain type of translation between the corresponding one-step languages. First we need some definitions.

Definition 13. Given sets X, Y, a mapping $h : X \to Y$ and a valuation $V : A \to \mathcal{Q}(Y)$, we define the valuation $V_{[h]} : A \to \mathcal{Q}(X)$ by setting $V_{[h]}(b) = h^{-1}[V(b)]$ for each $b \in A$.

The most important concept that we take from [14] is that of a *uniform translation* (called *uniform correspondence* in [14]). For this we need a few auxiliary definitions:

Definition 14. A *one-step frame* is a pair (X, α) with $\alpha \in \mathsf{T}X$. A *homomorphism* of one-step frames $h : (X', \alpha') \to (X, \alpha)$ is a map $h : X' \to X$ with $\mathsf{T}h(\alpha') = \alpha$. A one-step frame (X', α') together with a homomorphism $h : (X', \alpha') \to (X, \alpha)$ is called a *cover* of (X, α).

We can now define the notions of uniform translations and uniform constructions:

Definition 15. Given a functor T, a *uniform construction* F for T is an assignment of a cover $h_\alpha : (X_*, \alpha_*) \to (X, \alpha)$ to every one-step frame (X, α).

Definition 16. We say that the second-order one-step language $\mathtt{SO}^1_\Lambda(A)$ *admits uniform translations* if, given any natural number k, there exists a uniform construction F and an assignment of a monotone (natural) predicate lifting

$$\varphi^* : \mathcal{Q}(-)^A \to \mathcal{Q} \circ \mathsf{T}$$

to each monotone one-step formula $\varphi \in \mathtt{SO}^1_\Lambda$ with free variables A and quantifier depth at most k, such that for any one-step model (X, α, V), we have

$$(X, \alpha, V) \vDash_1 \varphi^* \text{ iff } (X_*, \alpha_*, V_{[h_\alpha]}) \vDash_1 \varphi.$$

Remark 1. It is easy to see that every monotone predicate lifting $\lambda : \mathcal{Q}(-)^A \to \mathcal{Q} \circ \mathsf{T}$ is equivalent to an atomic formula of $\mathtt{ML}^1_\mathsf{T}(A)$. In the following we shall not take care to distinguish between such a monotone predicate lifting and the corresponding atomic formula.

Definition 17. Any translation $(\cdot)^* : \mathtt{SO}^1_\Lambda \to \mathtt{ML}^1_\mathsf{T}$ induces a construction on automata, transforming a second-order Λ-automaton $\mathbb{A} = (A, \Delta, a_I, \Omega)$ into the modal automaton $\mathbb{A}^* = (A, \Delta^*, a_I, \Omega)$, with Δ^* given by $\Delta^*(a, c) := (\Delta(a, c))^*$.

Since the proof of the following result closely follows that of the main result in [14], we omit the details. The main difference with [14] is that here we need an "unravelling"-like component.

Proposition 8. *Assume that $\mathtt{SO}^1_\Lambda$ admits a uniform translation $(\cdot)^*$, and let $\mathbb{A}$ be a second-order Λ-automaton. Then for each pointed T-model $(\mathbb{S}, s)$ there is a T-tree model $(\mathbb{T}, R, t)$, with a T-model homomorphism f from $\mathbb{T}$ to $\mathbb{S}$, mapping t to s, and such that*

$$\mathbb{A} \textit{ accepts } (\mathbb{T}, R, t) \textit{ iff } \mathbb{A}^* \textit{ accepts } (\mathbb{S}, s).$$

Furthermore, given that $\mathbb{S} = (S, \sigma, V)$, if the map $h_{\sigma(s)} : S_* \to S$ is surjective, so is f.

From this, a routine argument yields the following result.

Theorem 5 (Characterization Theorem 1)**.** *Let Λ be an expressively complete set of monotone predicate liftings for a set functor T, and assume that $\mathtt{SO}_\Lambda(A)$ (for any set of variables A) admits uniform translations. Then $\mu\mathtt{ML}_\Lambda$ is the bisimulation-invariant fragment of $\mathtt{MSO}_\Lambda$.*

The existence of uniform translations for the one-step language [14] involves two components: a translation on the syntactic side and a uniform construction on the semantic side. However, as we shall now see, we can focus entirely on finding a suitable uniform construction for the one-step models; the syntactic translation will come for free.

Definition 18. Let φ be any formula of $\mathtt{SO}^1_\Lambda(A)$ of quantifier depth $\leq k$, and let F be a uniform construction for k. Then, we define the generalized predicate lifting $\varphi^* : \mathcal{Q}(-)^A \to \mathcal{Q} \circ \mathsf{T}$ by setting, for a given set X and valuation $V : A \to \mathcal{Q}(X)$:

$$\varphi^*_X(V) := \{\alpha \in \mathsf{T}X \mid (X_*, \alpha_*, V_{[h_\alpha]}) \vDash_1 \varphi\}.$$

The following is obvious:

Proposition 9. *If φ is a monotone formula then φ^* is a monotone generalized predicate lifting.*

Note that, in order for $\mathtt{SO}^1_\Lambda(A)$ to admit a uniform translation, it suffices that there exists for any k a uniform construction F such that, for every formula φ of quantfier depth $\leq k$, the generalized lifting φ^* is natural. An equivalent formulation of this condition is the following.

Proposition 10. *Let φ be any one-step formula in $\mathtt{SO}^1_\Lambda(A)$ and let F be a uniform construction. Then the lifting φ^* is natural if, for any pair of sets X, Y, any map $f : X \to Y$ and any valuation $V : A \to \mathcal{Q}(Y)$, we have*

$$(\star) \quad (X_*, \alpha_*, V_{[f \circ h_\alpha]}) \vDash_1 \varphi \text{ iff } (Y_*, \beta_*, V_{[h_\beta]}) \vDash_1 \varphi$$

provided that $\mathsf{T}f(\alpha) = \beta$.

The following diagram clarifies condition $(\star)$:

$$\begin{tikzcd}
(X_*, \alpha_*, V_{[f \circ h_\alpha]}) \arrow[r, Leftrightarrow, "\varphi"] \arrow[d, "h_\alpha"'] & (Y_*, \beta_*, V_{[h_\beta]}) \arrow[d, "h_\beta"] \\
(X, \alpha, V_{[f]}) \arrow[r, "f"'] & (Y, \beta, V)
\end{tikzcd}$$

Definition 19. A uniform construction F is said to be *adequate* for k, and with respect to the liftings Λ, if the equivalence $(\star)$ holds for all (monotone) formulas in $\mathtt{SO}^1_\Lambda(A)$ of quantifier depth $\leq k$ (for any finite set of variables A).

Since we could of course take the quantifier depth k and the set of liftings as extra inputs for the uniform construction, we shall simply say that the functor T admits an adequate uniform construction if there is an adequate uniform construction for T with respect to every k and every set of monotone liftings. If Λ is an expressively complete set of liftings, this is equivalent to requiring an adequate uniform construction with respect to Λ, for every k.

The following theorem, from which we obtain Theorem 2 by taking for Λ the set of all monotone liftings for T, summarizes the results of this section.

Theorem 6. *Let Λ be any expressively complete set of monotone predicate liftings for the set functor T. If T admits an adequate uniform construction, then*

$$\mu\mathtt{ML}_\Lambda \equiv \mathtt{MSO}_\Lambda/\!\simeq.$$

Example 1. As a first application, the standard Janin-Walukiewicz characterization of the modal μ-calculus can be seen as an instance of the result by taking $\Lambda = \{\Box, \Diamond\}$ and $\mathsf{T} = \mathcal{P}$, recalling that $\mathtt{MSO} = \mathtt{MSO}_{\{\Diamond\}} \equiv \mathtt{MSO}_{\{\Box,\Diamond\}}$. The adequate uniform construction for $\mathcal{P}$ is given as follows: consider a pair (X, α) with $\alpha \in \mathcal{P}(X)$. We take this to $X_* = \alpha_* = \alpha \times \omega$, and we let $h_\alpha : \alpha \times \omega \to \alpha$ be the projection map.

It turns out that several other applications of this result can be obtained in a particularly simple way. Say that a uniform construction F is *strongly adequate* if, for any mapping $f : X \to Y$ and any $\alpha \in \mathsf{T}X$, $\beta \in \mathsf{T}Y$ with $\mathsf{T}f(\alpha) = \beta$, there is a bijection $g : X_* \to Y_*$ such that $\mathsf{T}g(\alpha_*) = \beta_*$ and $f \circ h_\alpha = h_\beta \circ g$. Since it is easy to check that any strongly adequate uniform construction is adequate, we get:

Corollary 2. *If there is a strongly adequate uniform construction for T, then $\mu\mathtt{ML}_\mathsf{T} \equiv \mathtt{MSO}_\mathsf{T}/\!\simeq$.*

Example 2. As a first example, consider the finitary multiset ("bags") functor $\mathcal{B}$, which sends a set X to the set of mappings $f : X \to \omega$ such that the set $\{u \in X \mid f(u) = 0\}$ is cofinite. The action on morphisms is given by letting, for $f \in \mathcal{B}X$ and $h : X \to Y$, the multiset $\mathcal{B}h(f) : Y \to \omega$ be defined by $w \mapsto \sum_{h(v)=w} f(v)$. Given a pair X, α where $\alpha : X \to \omega$ has finite support, we define

$$X_* = \bigcup\{\{u\} \times \alpha(u) \mid u \in X\}.$$

Here, we identify each each $n \in \omega$ with the set $\{0, ..., n-1\}$. The mapping $\alpha_* : X_* \to \omega$ is defined by setting $\alpha_*(w) = 1$ for all $w \in X_*$. The map $h_\alpha : X_* \to X$ is defined by $(u, i) \mapsto u$. It is easy to check that the construction F is strongly adequate, hence $\mu\mathtt{ML}_\mathcal{B} \equiv \mathtt{MSO}_\mathcal{B}/\!\simeq$.

As a final application, consider the set of all *exponential polynomial functors* [18] defined by the "grammar"

$$\mathsf{T} ::= \mathsf{C} \mid \mathsf{Id} \mid \mathsf{T} \times \mathsf{T} \mid \coprod_{i \in I} \mathsf{T}_i \mid \mathsf{T}(-)^\mathsf{C}$$

where C is any constant functor for some set C, and Id is the identity functor on $\mathbf{Set}$. These functors cover many important applications: streams, binary trees, deterministic finite automata and deterministic labelled transition systems are all examples of coalgebras for exponential polynomial functors, as is the so called *game functor* whose coalgebras provide the semantics for "Coalition Logic" [6]. For this last instance, the "game functor" $\mathcal{G}$ for n agents can be written in the form of an exponential polynomial functor as follows:

$$\coprod_{\langle S_0, ..., S_{n-1}\rangle \in (\mathcal{P}(\omega) \setminus \{\emptyset\})^n} \{\langle S_0, ..., S_{n-1}\rangle\} \times \mathsf{Id}^{(S_0 \times ... \times S_{n-1})}$$

Then, for a given set X, an element of $\mathcal{G}X$ will be a pair consisting of a vector $\langle S_0, ..., S_{n-1}\rangle$ of available strategies for each player, together with an "outcome map" f assigning an element of X to each strategy profile in $S_0 \times ... \times S_{n-1}$.

Proposition 11. *Every exponential polynomial functor admits a strongly adequate uniform translation.*

Corollary 3. *For every exponential polynomial functor* T, *we have* $\mu\mathtt{ML}_{\mathsf{T}} \equiv \mathtt{MSO}_{\mathsf{T}}/{\simeq}$.

The following diagram illustrates the connection between the results that have been presented in this section:

$$\begin{tikzcd}
\mathtt{MSO}_\Lambda \;/\simeq \arrow[d, bend right, dash, "trees"'] & \equiv & \mu\mathtt{ML}_\Lambda \arrow[d, bend left, dash, "all\ models"] \\
Aut(\mathtt{SO}^1_\Lambda)/\simeq & \equiv & Aut(\mathtt{ML}^1_\Lambda)
\end{tikzcd}$$

The cases where we can find a strongly adequate uniform construction are the most straightforward applications of Theorem 6 that we know of. The Janin-Walukiewicz theorem is a less direct application: there is no strongly adequate uniform construction for the powerset functor, but there is an adequate uniform construction. In the next section, we shall study an example of a functor where there is no adequate uniform construction at all.

VI. The monotone neighborhood functor

The final section of our paper concerns the monotone neighborhood functor $\mathcal{M}$. Our main result concerns a characterization of the fragment of $\mathtt{MSO}_{\mathcal{M}}$ that is invariant under *global* neighborhood bisimulations, to be introduced below. Our proof applies the method of section V, but not directly: we will first see that the functor $\mathcal{M}$ itself does *not* admit an adequate uniform construction.

A. No adequate uniform construction for $\mathcal{M}$

We first consider the negative result.

Proposition 12. *There is no adequate uniform construction for the monotone neighborhood functor* $\mathcal{M}$.

Proof: To arrive at a contradiction assume that F is adequate. Fix some $a \in A$ and consider the formula $\varphi = \forall Z.(a \subseteq Z)$ expressing that a has empty extension.

Let Y be the set $\{u, v\}$ and let $\beta \in \mathcal{M}Y$ be the neighborhood structure $\{\{u\}, \{u, v\}\}$. Let V be any valuation with $V(a) = \{v\}$. First, we prove that $(Y, \beta, V) \vDash_1 \varphi^*$: to see this, consider the one-step model (Y', β', V') where $\beta' = \{\{u\}\}$ and we recall that $Y' = \{u\}$, and where V' is simply the restriction of V to Y'. It is easy to show that $(Y'_*, \beta'_*, V'_{[h_{\beta'}]}) \vDash_1 \varphi$, and hence $(Y', \beta', V') \vDash_1 \varphi^*$. Since the generalized predicate lifting φ^* is natural by assumption and $\mathcal{M}\iota_{Y',Y}(\beta') = \beta$, we get $(Y, \beta, V) \vDash_1 \varphi^*$ as required.

With this in mind, let X be the set $\{u^*, v^*, w^*\}$ and let $\alpha \in \mathcal{M}X$ be the neighborhood structure

$$\{\{u^*, v^*\}, \{u^*, w^*\}, \{u^*, v^*, w^*\}\}$$

Define the map $f : X \to Y$ by setting $u^* \mapsto u$, $v^* \mapsto v$ and $w^* \mapsto u$. It can easily be checked that $\mathcal{M}f(\alpha) = \beta$. By naturality of the formula φ^*, it follows that $(X, \alpha, V_{[f]}) \vDash_1 \varphi^*$. Hence we must have

$$(X_*, \alpha_*, V_{[f \circ h_\alpha]}) \vDash_1 \varphi$$

hence $V_{[f \circ h_\alpha]}(a) = \emptyset$. Since $v^* \in V_{[f]}(a)$, this means that we have $v^* \notin h_\alpha[X_*]$. But since $\mathcal{M}h_\alpha(\alpha_*) = \alpha$, this means $h_\alpha[X_*]$ must be a support for α. But it is easy to show that α cannot have a support S with $v^* \notin S$, so we have now reached a contradiction showing that F cannot be an adequate construction. ∎

B. The functor $\mathcal{M}^\star$

In this section, as a step towards our main characterization result, we shall consider the language $\mu\mathtt{ML}_{\mathcal{M}^\star}$, where the functor $\mathcal{M}^\star$ is a slight variation of the monotone neighborhood functor $\mathcal{M}$. The functor $\mathcal{M}^\star$ is obtained as the subfunctor of $\mathcal{M} \times \mathcal{P}$ given by

$$X \mapsto \{(\alpha, Y) \in \mathcal{M}X \times \mathcal{P}X \mid Y \text{ supports } \alpha\}$$

This is indeed a subfunctor of $\mathcal{M} \times \mathcal{P}$, because given a map $h : X \to Y$, if Z is a support for $\alpha \in \mathcal{M}X$, then $h[Z]$ is a support for $\mathcal{M}h(\alpha)$. Given $\alpha \in \mathcal{M}^\star X$, we will write $\alpha = (N_\alpha, S_\alpha)$.

Definition 20. For the functor $\mathcal{M}^\star$ we define the unary predicate liftings $\Box$ and E by

$$\Box_X(Z) := \{\alpha \in \mathcal{M}^\star X \mid Z \in N_\alpha\}$$
$$E_X(Z) := \{\alpha \in \mathcal{M}^\star X \mid Z \cap S_\alpha \neq \emptyset\},$$

and we let $\Diamond$ be the dual of $\Box$ and let E^d be the dual of E. The set of liftings $\{\Box, \Diamond, E, E^d\}$ is denoted as Θ.

The set Θ is an expressively complete set of liftings for $\mathcal{M}^\star$. We shall omit the proof of this fact here, and merely state it as the following proposition:

Proposition 13. *Every monotone natural predicate lifting* $\lambda : \mathcal{Q}(-)^A \to \mathcal{Q} \circ \mathcal{M}^\star$ *is equivalent to a formula in* $\mathtt{ML}_\Theta(A)$.

The main technical result of this section states the existence, for all k, of a uniform construction F that is adequate for k and with respect to the set of liftings $\{\Box, E\}$.

Definition 21. Fix a natural number k. Given a set X, and object $\alpha \in \mathcal{M}^\star X$, put

$$X_* := \{(u, i, Z, j) \in (X \times 2^k \times \mathcal{P}(S_\alpha) \times \omega) \mid u \in Z\},$$

and let π_X be the projection map from X_* to X. Define $\alpha_* = (N_{\alpha_*}, S_{\alpha_*}) \in \mathcal{M}^\star(X_*)$ by setting $S_{\alpha_*} = X_*$, and set $Z \in N_{\alpha_*}$ for $Z \subseteq S_{\alpha_*}$ iff $\lceil Y, j \rceil \subseteq Z$ for some $Y \in \alpha$, $Y \subseteq S_\alpha$ and some $j < \omega$, where

$$\lceil Y, j \rceil := \{(u, i, Y, j) \mid u \in Y, i < 2^k\}.$$

The sets of the form $\lceil Z, j \rceil$ will be called the *basic members* of N_{α^*}.

The main goal of this section is to prove the following:

Proposition 14. *The construction given in Definition 21 is an adequate, uniform construction for* k.

It is easy to check that, for all sets X and $\alpha \in \mathcal{M}^\star X$, we have $\mathcal{M}^\star \pi_X(\alpha_F) = \alpha$.

Our main goal in this section is to prove the following result, from which Proposition 14 now follows:

Lemma 1. *Let* X, Y *be any sets,* $\alpha \in \mathcal{M}^\star X$, $\beta \in \mathcal{M}^\star Y$ *and* $V : A \to Q(Y)$. *Suppose that we have a map* $h : X \to Y$ *such that* $\mathcal{M}^\star h(\alpha) = \beta$. *Then we have*

$$(X_*, \alpha_*, V_{[h \circ \pi_X]}) \equiv^k (Y_*, \beta_*, V_{[\pi_Y]})$$

Here, and throughout this section, we write $(X, \alpha, V) \equiv^k (Y, \beta, U)$ to say that two one-step models satisfy the same formulas of $\mathtt{MSO}^1_{\{\Box, E\}}(A)$ with at most n nested quantifiers. Let us keep the data X, Y, α, β, V and h fixed throughout the proof, and assume that $\mathcal{M}^\star h(\alpha) = \beta$. We will also assume, from now on, that N_α and N_β are both non-empty sets: if one of them is empty then both of them are, and in this case the lemma can be proved essentially using an easier version of the argument we use below.

Definition 22. Given a finite set of variables A, a propositional A-*type* τ is a subset of A. Given a set X and a valuation $V : A \to Q(X)$, the propositional A-type of $v \in X$ is defined to be $V^\dagger(v) = \{a \in A \mid v \in V(a)\}$.

Definition 23. Given a basic member $\lceil Z, j \rceil$ either in N_{α_*} or in N_{β_*}, a valuation $V : A \to \mathcal{Q}(X_*)$ or $V : A \to \mathcal{Q}(Y_*)$, and a natural number m, the m-*signature* of $\lceil Z, j \rceil$ over variables A and relative to the valuation V is the mapping $\sigma : \mathcal{P}(A) \to \{0, ..., m\}$ defined by setting $\sigma(t)$ to be $n < m$ if $\lceil Z, j \rceil$ contains exactly n elements of type t under the valuation V, or $\sigma(t) = m$ if $\lceil Z, j \rceil$ contains *at least* m elements of type t.

Definition 24. Let B be any set of variables containing A, and let $V_1 : B \to \mathcal{Q}(X_*)$ and $V_2 : B \to \mathcal{Q}(Y_*)$. Then for any natural number n we write

$$(X_*, \alpha_*, V_1) \approx^n (Y_*, \beta_*, V_2)$$

and say that these one-step models *match up to depth* n, if: for every n-signature σ over variables B, either the number of basic elements of signature σ in N_{α_*} and N_{β_*} respectively are both finite and the same, or both infinite.

Lemma 2. $(X_*, \alpha_*, V_{[h \circ \pi_X]}) \approx^{2^k} (Y_*, \beta_*, V_{[\pi_Y]})$.

We are going to show, by induction on a natural number $m \leq k$, that if two one-step models of the form (X_*, α_*, V_1) and (Y_*, β_*, V_2) match up to depth 2^m, then they satisfy the same formulas of quantifier depth m. For the basis case of $2^0 = 1$, we need the following result:

Lemma 3. *Let* B *be a set of variables containing* A, *and let* $V_1 : B \to \mathcal{Q}(X_*)$ *and* $V_2 : B \to \mathcal{Q}(F^\beta(Y))$ *be valuations such that*

$$(X_*, \alpha_*, V_1) \approx^1 (Y_*, \beta_*, V_2)$$

Then these two one-step models satisfy the same atomic formulas of the one-step language $\mathtt{MSO}^1_{\{\Box, E\}}$.

To clinch the proof of Proposition 14, we now only need the following lemma:

Lemma 4. *Let* B *be a finite set of variables containing* A, *let* $0 < m \leq k$ *and let* $V_1 : B \to Q(X_*)$ *and* $V_2 : B \to \mathcal{Q}(Y_*)$ *be valuations such that*

$$(X_*, \alpha_*, V_1) \approx^{2^m} (Y_*, \beta_*, V_2)$$

Let q *be any fresh variable. Then for any valuation* V_1' *extending* V_1 *with some value for* q, *there exists a valuation* V_2' *extending* V_2, *such that*

$$(X_*, \alpha_*, V_1') \approx^{2^{(m-1)}} (Y_*, \beta_*, V_2')$$

and vice versa.

Proof: We only prove one direction since the other direction can be proved by a symmetric argument. Let V_1' be given. By the hypothesis, for any 2^m-signature σ over the variables B, either the number of basic elements of signature σ in N_{α_*} and N_{β_*} relative to V_1 and V_2 are both finite and the same, or both infinite. Let $\sigma_1, ..., \sigma_k$ be a list of all the distinct 2^m-signatures over B such that the set of basic elements of N_{α_*} and N_{β_*} of signature σ_i, with $1 \leq i \leq k$, is non-empty but finite, and let $\sigma_{k+1}, ..., \sigma_l$ be a list of all the 2^m-signatures such that, for $k+1 \leq i \leq l$, there are infinitely many basic elements of N_{α_*} and of N_{β_*} of signature σ_i. Then, for each $i \in \{1, ..., l\}$, let $\alpha_*[\sigma_i]$ denote the set of basic elements in N_{α_*} of signature σ_i, and similarly let $\beta_*[\sigma_i]$ denote the set of basic elements of N_{β_*} of signature σ_i.

Given the extended valuation V_1' in X_* defined on variables $B \cup \{q\}$, we similarly let $\tau_1, ..., \tau_{k^*}$ be a list of all the 2^{m-1}-signatures over $B \cup \{q\}$ such that, for $1 \leq i \leq k^*$, the set of basic elements of N_{α_*} of 2^{m-1}-signature τ_i is non-empty but finite. We let $\tau_{k^*+1}, ..., \tau_{l^*}$ be a list of all the 2^{m-1}-signatures over $B \cup \{q\}$ such that, for each i with $k^* + 1 \leq i \leq l^*$, the set of basic elements of N_{α_*} of 2^{m-1}-signature τ_i is infinite. Let $\alpha_*[\tau_i]$ denote the set of basic elements of N_{α_*} of 2^{m-1}-signature τ_i, so that the collection $\alpha_*[\tau_1], ..., \alpha_*[\tau_{l^*}]$ constitutes a second partition of the set of basic elements of $N_{\alpha*}$. It will be useful to introduce the abbreviation D_1 for the finite set $\alpha_*[\sigma_1] \cup ... \cup \alpha_*[\sigma_k]$, and the abbreviation D_2 for the finite set $\alpha_*[\tau_1] \cup ... \cup \alpha_*[\tau_{k^*}]$.

For each i with $1 \leq i \leq k$, there is a bijection between the set $\alpha_*[\sigma_i]$ and $\beta_*[\sigma_i]$, and we can paste all these bijections together into a bijective map

$$f : \alpha_*[\sigma_1] \cup ... \cup \alpha_*[\sigma_k] \to \beta_*[\sigma_1] \cup ... \cup \beta_*[\sigma_k]$$

Since every basic element of N_{α_*} not in D_1 belongs to a 2^m-signature of which there are infinitely many basic elements in β_*, and since $D_1 \cup D_2$ is finite, it is easy to see that we can extend the map f to a map g which is an injection from the set $D_1 \cup D_2$ into the set of basic elements of N_{β_*}, such that for each basic element $\lceil Z, j \rceil$ in $D_1 \cup D_2$, $\lceil Z, j \rceil$ and $g(\lceil Z, j \rceil)$ have the same 2^m-signature over B, and such that $g \restriction D_1 = f$.

Each basic element of N_{β_*} not in the image of g must then be of one of the 2^m-signatures $\sigma_{k+1}, ..., \sigma_l$, and so we can partition the set of basic elements of N_{β_*} outside the image of g into the cells $\beta_*[\sigma_{k+1}] \setminus g[D_2], ..., \beta_*[\sigma_l] \setminus g[D_2]$. For each i with $k+1 \leq i \leq l$, let $\gamma_1^i, ..., \gamma_r^i$ list all infinite sets of the form $\alpha_*[\sigma_i] \cap \alpha_*[\tau_j]$ for $k^* + 1 \leq j \leq l^*$. The list $\gamma_1^i, ..., \gamma_r^i$ must be non-empty, and so since the set $\beta_*[\sigma_i] \setminus g[D_2]$ is also infinite, we may partition it into r many infinite cells and list these as $\delta_1^i, ..., \delta_r^i$. Now, for each basic element $\lceil Z, j \rceil$ of β_F, we define a map $W_{\lceil Z,j \rceil}$ from $B \cup \{q\}$ to $\mathcal{P}(\lceil Z, j \rceil)$ by a case distinction as follows:

Case 1: $\lceil Z, j \rceil = g(\lceil Z', j' \rceil)$ for some $\lceil Z', j' \rceil \in D_1 \cup D_2$. Then $\lceil Z, j \rceil$ and $\lceil Z', j' \rceil$ have the same 2^m-signature over B. Using this fact we define the valuation $W_{\lceil Z,j \rceil}$ so that, for each $p \in B$, we have $W_{\lceil Z,j \rceil}(p) = V_2(p) \cap \lceil Z, j \rceil$, and so that $\lceil Z', j' \rceil$ and $\lceil Z, j \rceil$ have the same 2^{m-1}-signature over $B \cup \{q\}$ with respect to the valuations V_1' and $W_{\lceil Z,j \rceil}$. We leave the details of this construction to the reader.

Case 2: $\lceil Z, j \rceil$ is not in the image of g. Then there must be some $i \in \{k^* + 1, ..., l^*\}$ such that $\lceil Z, j \rceil \in \beta_*[\sigma_{k+1}] \setminus g[D_2]$, and this set is partitioned into $\delta_1^i, ..., \delta_r^i$. Let $\lceil Z, j \rceil \in \delta_j^i$, and pick some arbitary element $\lceil Z', j' \rceil$ of the set γ_j^i. Then $\lceil Z', j' \rceil$ and $\lceil Z, j \rceil$ have the same 2^m-signature over B and we can proceed as in Case 1.

We define the valuation V_2' by setting $V_2'(q)$ to be the union of the sets $W_{\lceil Z,j \rceil}(q)$ for $\lceil Z, j \rceil$ a basic element in N_β^F. It is now fairly straightforward to check that

$$(X_*, \alpha_*, V_1') \approx^{2^{(m-1)}} (Y_*, \beta_*, V_2')$$

as required. We omit the details. ∎

Lemma 1 can now be deduced by combining the last three lemmas, by a straightforward argument using Ehrenfeucht-Fraïssé games for the one-step language.

C. Global neighborhood bisimulations

Since the set of liftings $\{\Box, \Diamond\}$ can be shown to be expressively complete for $\mathcal{M}$, and since $\Diamond$ is just the dual of $\Box$, the monadic second order language $\mathtt{MSO}_{\mathcal{M}}$ is equivalent to the logic $\mathtt{MMSO}$ which has its syntax given by

$$\varphi ::= \mathtt{sr}(p) \mid p \subseteq q \mid \Box(p, q) \mid \exists p.\varphi \mid \neg\varphi \mid \varphi \vee \varphi.$$

The semantics of an atomic formula $\Box(p, q)$ in a neighborhood model $\mathbb{S}$ is given, concretely, by the clause: $(S, \sigma, V, u) \vDash \Box(p, q)$ if, for all $v \in V(p)$, there is $Z \in \sigma(v)$ such that $Z \subseteq V(q)$.

Using the techniques in this paper, we cannot characterize the fragment of the language $\mathtt{MMSO}$ that is invariant for arbitrary neighborhood bisimulations. However, the situation changes if we consider *global* bisimulations between neighborhood models.

Definition 25. A *global neighborhood bisimulation* between $\mathcal{M}$-models $\mathbb{S}_1$ and $\mathbb{S}_2$ is a neighborhood bisimulation R that additionally satisfies the conditions:

Forth For every $u \in S_1$ there is some $v \in S_2$ with uRv
Back For every $v \in S_2$ there is some $u \in S_1$ with uRv

We now ask: what is the fragment of $\mathtt{MMSO}$ that is invariant for global neighborhood bisimulations? Since global bisimulations are the natural equivalence relation for modal logic with the *global modalities*, the most reasonable candidate would be: the monotone modal μ-calculus extended with the global modalities. To be precise, let the *monotone modal μ-calculus with global modalities*, denoted $\mu\mathtt{MML}_g$, be the language defined by the grammar:

$$\varphi ::= p \mid \neg p \mid \bot \mid \top \mid \Box\varphi \mid \Diamond\varphi \mid [\forall]\varphi \mid [\exists]\varphi$$
$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mu p.\varphi \mid \nu p.\varphi$$

where the formula φ in $\mu p.\varphi$ and $\nu p.\varphi$ must be positive in the variable p. The new operators $[\forall]$ and $[\exists]$ are the global universal and existential modalities, with their standard semantics: $(\mathbb{S}, u) \vDash [\forall]\varphi$ if $(\mathbb{S}, v) \vDash \varphi$ for all $v \in S$, and $(\mathbb{S}, u) \vDash [\exists]\varphi$ if $(\mathbb{S}, v) \vDash \varphi$ for some $v \in S$.

Given an $\mathcal{M}^\star$-model $\mathbb{S}$, let $\mathbb{S}_{\mathcal{M}}$ be the underlying $\mathcal{M}$-model. Conversely, given an $\mathcal{M}$-model $\mathbb{S} = (S, \sigma, V)$, define the $\mathcal{M}^\star$-model $\mathbb{S}^G = (S, \sigma^G, V)$ by setting $\sigma^G(s) = (\sigma(s), S)$. The main result of this section is the following.

Theorem 7. *A formula in* $\mathtt{MMSO}$ *is invariant for global neighborhood bisimulations if, and only if, it is equivalent to a formula of the logic* $\mu\mathtt{MML}_g$.

Proof: Clearly $\mu\mathtt{MML}_g$ translates into $\mathtt{MMSO}$ and is invariant for global bisimulations.

Conversely, suppose $\varphi \in \mathtt{MMSO}$ is invariant for global neighborhood bisimulations. First observe that φ can be regarded as a formula in $\mathtt{MSO}_{\mathcal{M}^\star}$ as well. More precisely, there is a formula $\varphi^* \in \mathtt{MSO}_{\mathcal{M}^\star}$ such that

$$(\mathbb{T}, t) \vDash \varphi^* \text{ iff } (\mathbb{T}_{\mathcal{M}}, t) \vDash \varphi \tag{1}$$

for any $\mathcal{M}^\star$-model $(\mathbb{T}, t)$. By Corollary 1 there is a second-order $\{\Box, E\}$-automaton $\mathbb{A}_\varphi$ such that

$$\mathbb{A}_\varphi \equiv \varphi^* \text{ (on all } \mathcal{M}^\star\text{-tree models).} \tag{2}$$

Now we use the existence of an adequate, uniform construction for $\mathcal{M}^\star$ (Proposition 14). Let $\mathbb{A}_\varphi^t$ be the corresponding modal Λ-automaton given by Proposition 8, where Λ is the collection of all monotone, natural predicate liftings for $\mathcal{M}^\star$. By Proposition 13 we may in fact assume that $\mathbb{A}_\varphi^t$ is a Θ-automaton, where $\Theta = \{\Box, \Diamond, E, E^d\}$. Let $\psi = \psi_{\mathbb{A}_\varphi^t}$ be the corresponding formula in $\mu\mathtt{ML}_\Theta$. We claim that, for any pointed neighborhood model $(\mathbb{S}, s)$ we have

$$\mathbb{S}, s \vDash \varphi \text{ iff } \mathbb{S}^G, s \vDash \psi. \tag{3}$$

To prove this, consider the $\mathcal{M}^\star$-tree model $(\mathbb{T}, R, r)$ given by Proposition 8, applied to the pointed $\mathcal{M}^\star$-model $(\mathbb{S}^G, s)$. Then there is a surjective $\mathcal{M}^\star$-coalgebra morphism $f : (\mathbb{T}, r) \to (\mathbb{S}^G, s)$, and so in particular, f is the graph of a global neighborhood bisimulation between $\mathbb{T}_{\mathcal{M}}$ and $\mathbb{S}$ relating r to s. Gathering some facts we obtain the following chain of equivalences:

$$\begin{aligned}
\mathbb{S}, s \vDash \varphi \text{ iff } & \mathbb{T}_{\mathcal{M}}, r \vDash \varphi && \text{(assumption on } \varphi) \\
\text{iff } & \mathbb{T}, r \vDash \varphi^* && (1) \\
\text{iff } & \mathbb{T}, R, r \vDash \mathbb{A}_\varphi && (2) \\
\text{iff } & \mathbb{S}^G, s \vDash \mathbb{A}^t_\varphi && \text{(Proposition 8)} \\
\text{iff } & \mathbb{S}^G, s \vDash \psi && \text{(assumption on } \psi)
\end{aligned}$$

which proves (3) indeed.

Finally, let $\psi^\forall \in \mu\mathtt{MML}_g$ be the formula we obtain from ψ by replacing every occurrence of E with $[\exists]$ and every occurrence of E^d with $[\forall]$. It is a routine check to verify that

$$\mathbb{S}^G, s \vDash \psi \text{ iff } \mathbb{S}, s \vDash \psi^\forall. \tag{4}$$

But then the equivalence of $\varphi \in \mathtt{MMSO}$ and $\psi^\forall \in \mu\mathtt{MML}_g$ is immediate from (3) and (4). ■

VII. Future work

For a concise formulation of the contributions of this publication we refer to the abstract.

Here we mention some questions for future research:

1) Is there a good categorical characterization of those set functors T that admit an adequate uniform construction, for instance, in terms of T preserving certain limits or colimits?
2) Can we generalize our work in section VI, to the effect that every set functor T has a companion $\mathsf{T}^\star$ that admits an adequate uniform construction? Can we then use this companion functor to prove invariance results for T-logics, similar to Theorem 7? Relating this to the previous question, we would like to understand *why* $\mathcal{M}^\star$ admits an adequate uniform construction, and $\mathcal{M}$ does not.
3) We intend to further explore the relation between $\mathtt{MSO}_\mathsf{T}$ and the first-order logic of Litak & alii [13] for T-coalgebras. For instance, an interesting question would be whether (on T-tree models) $\mathtt{MSO}_\mathsf{T}$ is equivalent to some extension of this first-order language with certain fixpoint operators.

Finally we note that after submitting the manuscript of the current publication, we could settle the main open question concerning the monotone neighborhood functor $\mathcal{M}$ in the positive. Based on Theorem 7 we can prove that

$$\mu\mathtt{MML} \equiv \mathtt{MMSO}/{\simeq}$$

indeed. That is, a formula in $\mathtt{MMSO}$ is invariant under neighborhood bisimulations if, and only if, it is equivalent to a formula of the monotone μ-calculus $\mu\mathtt{MML}$. We will report on this result in a future publication.

References

[1] M. Rabin, "Decidability of second-order theories and automata on infinite trees," *Transactions of the American Mathematical Society*, vol. 141, pp. 1–35, 1969.

[2] I. Walukiewicz, "Monadic second order logic on tree-like structures," in *STACS*, 1996, pp. 401–413.

[3] D. Janin and I. Walukiewicz, "On the expressive completeness of the propositional μ-calculus w.r.t. monadic second-order logic," in *Proceedings of the Seventh International Conference on Concurrency Theory, CONCUR '96*, ser. LNCS, vol. 1119, 1996, pp. 263–277.

[4] J. Rutten, "Universal coalgebra: A theory of systems," *Theoretical Computer Science*, vol. 249, pp. 3–80, 2000.

[5] L. Moss, "Coalgebraic logic," *Annals of Pure and Applied Logic*, vol. 96, pp. 277–317, 1999, (Erratum published *Ann.P.Appl.Log.* 99:241–259, 1999).

[6] C. Cîrstea, A. Kurz, D. Pattinson, L. Schröder, and Y. Venema, "Modal logics are coalgebraic," *The Computer Journal*, vol. 54, pp. 524–538, 2011.

[7] D. Pattinson, "Coalgebraic modal logic: Soundness, completeness and decidability of local consequence," *Theoretical Computer Science*, vol. 309, pp. 177–193, 2003.

[8] Y. Venema, "Automata and fixed point logic: a coalgebraic perspective," *Information and Computation*, vol. 204, pp. 637–678, 2006.

[9] C. Cîrstea, C. Kupke, and D. Pattinson, "EXPTIME tableaux for the coalgebraic μ-calculus," in *Computer Science Logic 2009*, E. Grädel and R. Kahle, Eds., vol. LNCS 5771. Springer, 2009, pp. 179–193.

[10] G. Fontaine, R. Leal, and Y. Venema, "Automata for coalgebras: an approach using predicate liftings," in *Proc. 37th International Colloquium on Automata, Languages, and Programming, ICALP 2010*, ser. LNCS, vol. 6199. Springer, 2010, pp. 381–392.

[11] D. Muller and P. Schupp, "Simulating alternating tree automata by nondeterministic automata," *Theoretical Computer Science*, vol. 141, pp. 69–107, 1995.

[12] C. Kupke and Y. Venema, "Coalgebraic automata theory: basic results," *Logical Methods in Computer Science*, vol. 4, pp. 1–43, 2008.

[13] T. Litak, D. Pattinson, K. Sano, and L. Schröder, "Coalgebraic predicate logic," in *Proc. 39th International Colloquium on Automata, Languages, and Programming, ICALP 2012*, ser. LNCS, vol. 7392. Springer, 2012, pp. 299–312.

[14] Y. Venema, "Expressiveness modulo bisimilarity: a coalgebraic perspective," in *Johan van Benthem on Logic and Information Dynamics*, ser. Outstanding Contributions to Logic. Springer, 2014, vol. 5, pp. 33–65.

[15] H. Hansen and C. Kupke, "A coalgebraic perspective on monotone modal logic," *Electronic Notes in Theoretical Computer Science*, vol. 106, pp. 121 – 143, 2004, proceedings of the Workshop on Coalgebraic Methods in Computer Science (CMCS).

[16] L. Schröder, "Expressivity of coalgebraic modal logic: the limits and beyond," *Theoretical Computer Science*, pp. 230–247, 2008.

[17] Y. Venema, "Lectures on the modal μ-calculus," 2012, Lecture Notes, ILLC, University of Amsterdam.

[18] B. Jacobs, "Introduction to coalgebra. Towards mathematics of states and observations," 2012, manuscript, Radboud University Nijmegen.