On the logical definability of topologically closed recognizable languages of infinite trees

David Janin
LaBRI
Université de Bordeaux I
351 cours de la Libération
33 405 Talence cedex
France
janin@labri.fr

Giacomo Lenzi
Department of Mathematics
University of Pisa
via Buonarroti 2
56127 Pisa
Italy
lenzi@mail.dm.unipi.it

January 14, 2012

Abstract

In this paper, we prove that for any language L of finitely branching finite and infinite trees, the following properties are equivalent: (1) L is definable by an existential MSO sentence which is bisimulation invariant over graphs, (2) L is definable by a FO-closed existential MSO sentence which is bisimulation invariant over graphs, (3) L is definable in the nu-level of the modal mu-calculus, (4) L is the projection of a locally testable tree language and is bisimulation closed, (5) L is closed in the prefix topology and recognizable by a modal finite tree automaton, (6) L is recognizable by a modal finite tree automaton of index zero.

The equivalence between (3), (4), (5) and (6) is known for quite a long time, although maybe not in such a form, and can be considered as a classical result. The logical characterization of this class of languages given by (1) (and (2)) is new. It is an extension of analogous results over finite structures such as words, trees and grids relating full existential MSO and definability by finite automata.

1 Introduction

Classical logical systems such as monadic second-order logic (MSO) often play, in computer science, the role of basic (assembly-like) languages into which programs or rather programs' specifications can be described or translated. In concurrency, where programs are often modeled as state/transition systems [14], monadic second-order logic is generally considered as a sufficiently expressive logic. In particular, it subsumes most specification languages such as LTL, CTL* [8] or the modal mu-calculus [12].

However, it is not full monadic second-order logic which is needed in concurrency. In fact, when specifying properties of programs, one is generally interested in the behavior of programs rather than in the programs themselves [14, 8]. As program behaviors can be modeled by infinite trees, i.e. trees obtained by unraveling program models, it appears that specifying program behaviors amounts to specifying languages of finite and infinite trees.

In practice, however, when programs are modeled by means of finite state systems, it is important to check, on the finite models of programs, that their potential infinite behaviors (their unravelings) are correct w.r.t. a given specification. In other words, a program specification must define a class of graphs which is, at least, invariant under unraveling. The peculiar status of non-determinism even suggests to consider classes of graphs which are invariant under bisimulation equivalence [14].

This leads to the study of the bisimulation (or counting bisimulation) invariant fragment of monadic second-order logic, i.e. the set of MSO sentences whose classes of models are closed under bisimulation. A first easy observation is that all specification languages mentioned above are part of this fragment and, among them, the mu-calculus is the most expressive language. A result of Walukiewicz and the first author [11] shows that the mu-calculus is even maximal in this respect, i.e. *the bisimulation (resp. counting bisimulation) invariant fragment of monadic second-order logic equals the modal (resp. the counting) mu-calculus.*

Now, among languages of trees definable in the mu-calculus, there are interesting subclasses or complexity levels. In fact, the alternation depth of least and greatest fixed point operators in mu-calculus formulas induces an infinite hierarchy [4]. Then one may ask how this complexity measure for mu-calculus formulas relates with the alternation depth of existential and universal set quantifiers in MSO formulas. Several correspondences have been announced in [9]. In this paper, we focus our attention on the first level of the mu-calculus hierarchy: the nu-level composed of the mu-calculus formulas in positive normal form built without any least fixed point construction.

More precisely, we prove the following theorem.

Theorem 1.1 *For any language L of finitely branching finite and infinite trees, the following properties are equivalent:*

1. *L is definable by an existential MSO sentence which is bisimulation (resp. counting bisimulation) invariant over graphs,*
2. *L is definable by an FO-closed existential MSO sentence which is bisimulation (resp. counting bisimulation) invariant over graphs,*
3. *L is definable in the nu-level of the modal (resp. counting) mu-calculus,*
4. *L is the projection of a locally testable tree language and L is bisimulation closed (resp. counting bisimulation closed),*
5. *L is closed in the prefix topology and recognizable by a modal (resp. counting) finite state tree automaton,*
6. *L is recognizable by a modal (resp. counting) finite state tree automaton of index zero.*

The equivalence between (1) and (6) is a non trivial logical characterization of languages of infinite trees recognizable in a naive sense: by means of finite state automata without any infinitary criterion [20]. Observe that for finite structures such as finite words, trees or grids, recognizability by finite state automata is captured by full existential MSO [21].

FO-closed existential MSO mentioned in (2) is obtained from existential MSO by allowing arbitrary FO quantifiers to be inserted among existential set quantifiers. This fragment, considered in [13], is interesting because it is more robust and, over arbitrary graphs, strictly more expressive than existential MSO. For instance, it is closed under FO transformations. Yet, the equivalence between (1) and (2) shows that it behaves like existential MSO as far as bisimulation invariance is concerned. This result contrasts with the non equivalence observed over trees without the bisimulation invariance requirement [2].

The equivalence between (1) and (3) extends van Benthem's result on FO and modal logic [3], and refines the result obtained by Walukiewicz and the first author regarding MSO and full mu-calculus [11].

The equivalence between (3) and (6) is a classical result (see e.g. [10, 22] for the arguments).

The equivalences between (4), (5) and (6) are easy generalization of known results in the case of the binary tree (see e.g [15, 19]). Proofs are given here for technical reasons and for completeness.

Technically, our results are obtained by combining tree automata techniques, model-theoretic tools and topological arguments. Known results relating monadic second-order logic, fixed point calculi and automata theory [16, 18, 22] are also essential.

Finally, let us stress that in this paper, when we say that a language L of finitely branching trees is *definable* by some sentence φ, we mean that L is the set of all finitely branching trees that satisfy φ, i.e. apart from invariance requirement considerations, we do not pay any attention to the other models (not necessarily trees nor finitely branching) of φ.

Organization of the paper

In Section 2, we review the definition of transition systems, simply called graphs in the sequel, bisimulation and counting bisimulation. We also review the prefix topology over finitely branching trees, which plays a fundamental role in our approach.

Section 3 is dedicated to a presentation of monadic second-order logic (MSO) and modal and counting mu-calculus as fragments of MSO. The notions of bisimulation and counting bisimulation invariant formulas are defined.

In Section 4, an automata-theoretic characterization of these fixed point calculi is given. Focusing on topologically closed languages of finitely branching trees, we prove that recognizable languages of finitely branching trees which are closed in the topological sense are precisely those definable in the nu-level of the mu-calculus, or recognizable by tree automata of index zero. This gives the equivalence between (3), (5) and (6) and it shows that (5) implies (4).

In Section 5, we study bisimulation and counting bisimulation invariance within existential second-order logic. Using a standard corollary of Łos' theorem, we prove that bisimulation or counting bisimulation invariant existential formulas define languages of finitely branching trees that are closed in the topological sense. As this applies to the case of both existential MSO and closed existential MSO, this proves that (1) or (2) imply (5).

This completes the proof of Theorem 1.1 since the implications (4) $\Rightarrow$ (1) and (1) $\Rightarrow$ (2) hold for syntactic reasons.

2 Graphs, trees and tree topology

We review here the definition of transition systems, bisimulation equivalence and κ-expansion of transition systems which capture in some sense bisimulation equivalence. We also define a notion of counting bisimulation by adding a "local bijection" constraint to the notion of bisimulation. This new definition makes statements more uniform: counting bisimulation is the equivalence induced by unraveling as bisimulation is, in a sense, the equivalence induced by κ-expansions.

Because a transition system is simply a directed graph with a distinguished vertex called its root, we use the vocabulary of (directed) rooted graphs. In order to simplify statements and proofs, we consider only graphs built over a single binary relation symbol. All the results presented here can easily be generalized to (finitely) labeled directed graphs, i.e. graphs built over a finite set of binary relation symbols.

As trees are important when dealing with bisimulation, we also review some standard notation and definitions of trees. In particular, we recall the definition of the standard prefix topology over finitely branching trees. It is one of the main ingredients in the proof of the main result.

2.1 Graphs, bisimulation and trees

Let *Pred* be a finite set of unary predicate symbols and let E be a binary relation symbol. A *rooted graph*, simply called *graph* in the sequel, is a tuple:

$$M = \langle V^M, r^M, E^M, \{p^M\}_{p \in \mathit{Pred}} \rangle$$

with a set V^M of *vertices*, a *root* $r^M \in V^M$, a binary *successor relation* $E^M \subseteq V^M \times V^M$ and for each $p \in \mathit{Pred}$, a subset $p^M \subseteq V^M$. We say that a vertex v is a successor of u when $(u, v) \in E^M$. The set of all successors of u is denoted by $\mathit{succ}(u)$. We write $M \cong N$ to mean that the graphs M and

N are isomorphic. We also use the notation $dom(M)$ for the domain V^M of the graph M.

If u is a vertex of a graph M, the *degree* of u is the number of its successors. The *degree* of a graph M is the supremum (be it finite or infinite) of the degrees of its vertices. A graph is called *finitely branching* if every vertex has finite degree. Observe that a finitely branching graph does not necessarily have finite degree.

Two graphs M and N are called *bisimilar* when there exists a relation $R \subseteq V^M \times V^N$, called a *bisimulation relation*, such that $(r^M, r^N) \in R$ and for every $(u, v) \in R$ and $p \in \mathit{Pred}$, $u \in p^M$ iff $v \in p^N$, and whenever $(u, u') \in E^M$ for some u', then there exists v' such that $(v, v') \in E^N$ and $(u', v') \in R$, and whenever $(v, v') \in E^N$ for some v', then there exists u' such that $(u, u') \in E^M$ and $(u', v') \in R$.

If, in addition, for each $(u, v) \in R$, R establishes a bijection between $succ(u)$ and $succ(v)$, then we say that R is a *counting bisimulation*. In this case, we say that graph M and N are *counting bisimilar*.

Given any non zero cardinal κ, a *κ-indexed path* in M is a non empty finite or infinite word $w \in V^M.(\kappa.V^M)^* \cup V^M.(\kappa.V^M)^\omega$ such that whenever $w = w_1.u.k.u'.w_2$ with $w_1 \in (V^M.\kappa)^*$, $u \in V^M$, $k \in \kappa$, $u' \in V^M$ and $w_2 \in (\kappa.V^M)^* \cup (\kappa.V^M)^\omega$ one has $(u, u') \in E^M$. The length $|w|$ of a finite κ-indexed path w is defined as the number of occurrences of elements of V^M in w, i.e. when $w = u_0.k_1.u_1.\cdots.k_n.u_n$ then $|w| = n + 1$. In this case, we say u_0 is the source of w, u_n is the target of w and w is a (κ-indexed) path from u_0 to u_n.

Observe that when $\kappa = 1$, κ-indexed paths are nothing but the usual (directed) paths in a graph.

Let u and v be vertices of a graph M. We say that v is *reachable* from u if there is a path from u to v. Then, the *distance* $d(u, v)$ from u to v is defined as $|w| - 1$ where w is a minimal length path from u to v. In the case v is not reachable from u, we declare the distance to be infinite.

The *reachable part* of the graph M, denoted $\mathit{Reach}(M)$, is the graph M restricted to the vertices reachable from the root. The *height* of a reachable vertex is its distance from the root.

Given a integer h, the *h-th prefix* of M, denoted by $P_h(M)$, is the set of all vertices with height at most h. Observe that for every graph M we have $\mathit{Reach}(M) = \bigcup_{h \in \mathbb{N}} P_h(M)$. In the sequel, we also use the notation $P_h(M)$ for the subgraph of M induced by this set of vertices.

The *κ-expansion* $T^\kappa(M)$ of a system M is defined as follows : let $V^{T^\kappa(M)}$ be the set of all finite κ-indexed paths of M with source r^M, the root $r^{T^\kappa(M)}$ equal r^M, the relation $E^{T^\kappa(M)}$ be the set of all pairs of the form $(w.u, w.u.k'.u') \in V^{T^\kappa(M)} \times V^{T^\kappa(M)}$ with $w \in (V^M.\kappa)^*$, u and $u' \in V^M$ and $k' \in \kappa$ such that $(u, u') \in E^M$. Moreover, for any $p \in Prop$, let $p^{T^\kappa(M)}$ be the set of all κ-indexed paths of the form $w.u \in V^{T^\kappa(M)}$ with $w \in (V^M.\kappa)^*$ and $u \in p^M$.

When $\kappa = 1$, the κ-expansion of M, from now on denoted by $T(M)$, is nothing but what is usually called the *unraveling* of the graph M from its root r^M. In particular, vertices of $T(M)$ are all the finite paths from the root r^M in M. This notion allows us to define trees in a very simple way: a *tree* is a graph isomorphic to its unraveling.

Observe that for every cardinal κ and for every graph M, the κ-expansion $T^\kappa(M)$ of M is a tree. It follows that, in particular, $Reach(T^\kappa(M)) \cong T^\kappa(M)$.

The notion of unraveling (or 1-expansion) is related to counting bisimulation as follows.

Fact 2.1 *For any graphs M and N, M and N are counting bisimilar iff their unravelings $T(M)$ and $T(N)$ are isomorphic.*

Proof. This fact follows immediately from the following observations: the functional relation from $V^{T(M)}$ to V^M that maps each finite path to its target is a counting bisimulation between $T(M)$ and M; and, over trees, a counting bisimulation relation is just an isomorphism. □

In a quite similar way, the notion of κ-expansion also gives in some sense canonical representatives of equivalence classes under bisimulation.

Fact 2.2 (see [11]) *For any infinite cardinal κ and for any graphs M and N of cardinality at most κ, M and N are bisimilar iff their κ-expansions $T^\kappa(M)$ and $T^\kappa(N)$ are isomorphic.*

Proof. Let R be a bisimulation relation between M and N and let cardinal κ be as above. Let R' be the relation between vertices of both $T^\kappa(M)$ and $T^\kappa(N)$ that relates any two κ-indexed paths of the same length whose targets belongs to R. Relation R' is a bisimulation relation. Moreover, provided κ is infinite (so that, with the help of the axiom of choice, $\kappa.\kappa = \kappa$) and big enough (actually not smaller than the degree of M and N), one can check that relation R' can be refined to become the relation of an isomorphism.

The converse is immediate since any graph M is bisimilar with its κ-expansion $T^\kappa(M)$. $\square$

The assumption, in Fact 2.2, that κ is infinite is essential. In fact, let M (resp. N) be the graph defined by a single edge (resp. two edges only) from the root, with all vertices labeled identically. They are bisimilar (and not counting bisimilar). However, for any finite cardinal κ distinct from zero, $T^\kappa(M)$ and $T^\kappa(N)$ are not isomorphic.

2.2 The prefix topology over finitely branching trees

We consider a topology on the set of all *finitely branching* trees. This topology is a straightforward generalization of the classical prefix topology on words, binary trees, or, more generally, k-ary trees, where k is a fixed integer.

Let $FBT(\mathit{Pred})$ be the class of all finitely branching trees over a finite set Pred of predicate symbols. We define the prefix topology on $FBT(\mathit{Pred})$ by taking as the basic open sets, the sets of the form:

$$\{M \in FBT(\mathit{Pred}) \mid P_h(M) \cong F\},$$

where h is a positive integer and F is a finite tree. This topology is Hausdorff as shown by the following lemma.

Lemma 2.3 (Gluing Lemma) *Let M and N be two finitely branching trees. The trees M and N are isomorphic if and only if for infinitely many h, $P_h(M)$ and $P_h(N)$ are isomorphic.*

Proof. The only non-trivial argument which we need to make is to show that if M and N are infinite and $P_h(M) \cong P_h(N)$ for infinitely many h (hence for all h) then $M \cong N$. In order to do so, let $I_{M,N}$ be the set of isomorphisms from $P_h(M)$ to $P_h(N)$, for h ranging over positive integers. The set $I_{M,N}$ ordered by inclusion forms an infinite finitely branching trees. Hence, by Koenig's Lemma, it has an infinite branch which defines an isomorphism between M and N. $\square$

As a curiosity, note that the lemma does *not* extend to arbitrary trees M and N. As an example consider for any n, a unary tree M_n with n vertices, and a unary, infinite tree M_∞. Let M be the graph obtained from the disjoint union of the M_n's by adding a new root on top of them (hence the root of M has countably many successors: one per graph M_n). And let N be the graph obtained in a similar way from the disjoint union of the M_n's and M_∞. For any $h \in \omega$, both $P_h(M)$ and $P_h(N)$ are trees formed by a copy of M_k for each $k < h$, plus countably many copies of M_h. But M and N are not isomorphic, because N has an infinite branch and M has none.

Observe that the prefix topology can be defined by a metric. In fact, given two trees M and N, let $d(M, N) = 0$ when M and N are isomorphic and $d(M, N) = 2^{-k}$ otherwise where k is the biggest integer such that $P_k(M)$ and $P_k(N)$ are isomorphic (which exists by the gluing Lemma). The function d is obviously a metric that defines the prefix topology.

The prefix topology satisfies a weak form of compactness. More precisely, we define the *skeleton* of a tree M to be the tree (over zero predicates) $Sk(M) = \langle V^M, r^M, E^M \rangle$, i.e. $Sk(M)$ is the structure obtained from M by forgetting all unary predicates. Then,

Lemma 2.4 *Let $(M_n)_{n \in I\!N}$ be a sequence in $FBT(Pred)$. If the sequence $\{Sk(M_n)\}_{n \in I\!N}$ has a converging subsequence so does have $\{M_n\}_{n \in I\!N}$.*

Proof. Let M_n be such a sequence. Assuming the sequence $\{Sk(M_n)\}_{n \in I\!N}$ has a converging subsequence, let $I \subseteq I\!N$ be an infinite set such that $\{Sk(M_n)\}_{n \in I}$ converges.

By induction, we build a strictly decreasing sequence $\{I_h\}_{h \in I\!N}$ of infinite sets of positive integers such that $I_0 = I$ and, for any $h > 0$, m and $n \in I_h$, $P_h(M_n) = P_h(M_m)$. This enables us to conclude as it implies that the sequence $\{M_{\min(I_h)}\}_{h \in I\!N}$ converges.

More precisely, assume that, for some $h \in I\!N$, the finite sequence of infinite sets $I = I_0 \supset I_1 \supset \cdots \supset I_h$ has already been built.

Since the sequence $\{Sk(M_n)\}_{n \in I}$ converges, there exists $m_h \in I$ such that, for any $n \in I$ with $n \geq m_h$, $P_{h+1}(Sk(M_{m_h})) = P_{h+1}(Sk(M_n))$. Now, as there are finitely many trees in $FBT(Pred)$ with skeleton $P_{h+1}(Sk(M_{m_h}))$ and the set I_h is infinite, there exists an infinite subset I_{h+1} of I_h (which can be chosen distinct from I_h) such that, for any n and $m \in I_{h+1}$, $P_{h+1}(M_n) = P_{h+1}(M_m)$.
□

The prefix topology is not compact as shown, for instance, by any sequence of trees M_n where the root has degree n which has no converging subsequence.

3 Logic and modal or counting mu-calculus

In this section we review the definition of first-order logic (FO) and monadic second-order logic (MSO), and the counting and modal propositional mu-calculus [12]. All logics are interpreted over graphs. Any graph M, as defined above, is a FO-structure on the vocabulary $\{r, E\} \cup \mathit{Pred}$ with r a constant symbol standing for the root, E a binary relation symbol and *Pred* a set of unary relation symbols.

3.1 First-order and monadic second-order logic

Let $\mathit{var} = \{x, y, \cdots\}$ and $\mathit{Var} = \{X, Y, \cdots\}$ be disjoint sets of, respectively, first-order and monadic second-order variable symbols.

First-order logic over the vocabulary $\{r, E\} \cup \mathit{Pred}$ can be defined as follows. The set of FO formulas is the smallest set containing the formulas $p(t)$, $t = t'$, $E(t, t')$, $X(t)$ for $p \in \mathit{Pred}$, $X \in \mathit{Var}$ and $t, t' \in \mathit{var} \cup \{r\}$, which is closed under negation $\neg$, disjunction $\vee$, conjunction $\wedge$ and existential $\exists$ and universal $\forall$ quantifications over FO variables.

The set of monadic second-order (MSO) formulas over the same vocabulary is the smallest set containing all FO formulas and closed under negation $\neg$, disjunction $\vee$, conjunction $\wedge$ and existential $\exists$ and universal $\forall$ quantifications over FO and MSO variables.

Existential monadic second-order logic (EMSO) is defined as the set of all formulas of the form $\exists X_1 \cdots \exists X_n \varphi$ with φ some FO formula.

Closed existential monadic second-order logic (CEMSO) is defined as the set of all formulas of the form $\theta_1 \overline{x}_1 \exists X_1 \cdots \theta_n \overline{x}_n \exists X_n \varphi$ where φ is an FO formula, the $(\theta_i x_i)$s are finite sequences of FO quantifications. Ajtai et al. proved that, over finite models, CEMSO is stricly more expressive than EMSO [13]. Arnold et al. [2] show that the same holds over infinite trees.

We denote by $\varphi(x_1, \cdots, x_m, X_1, \cdots, X_n)$ an MSO formula φ with free first-order variables among $\{x_1, \cdots, x_m\}$ and free set variables among $\{X_1, \cdots, X_n\}$. If M, $v_1, \ldots, v_m \in V^M$, and $V_1, \ldots, V_n \subseteq V^M$, we write

$$M \models \varphi(v_1, \cdots, v_m, V_1, \cdots, V_n)$$

to say that formula φ is true in M, or M satisfies φ, under the interpretation mapping each FO variable x_i to the vertex v_i and each set variable X_j to the set V_j. This satisfaction relation is defined in the standard way [17, 7].

A class $\mathcal{C}$ of graphs is called *MSO definable* when there exists a sentence $\varphi \in MSO$, i.e. an MSO formula with no free variable, such that $M \in \mathcal{C}$ iff $M \models \varphi$. A class $\mathcal{C}$ of transition systems is *bisimulation closed* (resp. *counting bisimulation closed*) if whenever $M \in \mathcal{C}$ and M' is bisimilar (resp. counting bisimilar) to M then $M' \in \mathcal{C}$. A sentence φ is *bisimulation invariant* (resp. *counting bisimulation invariant*) if the class of transition systems it defines is bisimulation closed (resp. counting bisimulation closed). Observe that bisimulation invariance implies counting bisimulation invariance since a counting bisimulation relation is a bisimulation relation.

Remark 3.1 *In this paper, we do consider bisimulation invariance over all models and not only over trees. Observe that in EMSO it makes a difference. In fact, bisimulation invariance over trees only is a less restrictive notion than bisimulation invariance over arbitrary graphs. For instance, the (even FO !) formula $\exists x p(x)$ is bisimulation invariant over trees while it is equivalent to no bisimulation invariant formula over graphs in EMSO. In fact, the bisimulation closed class of graph which contains, as trees, the class of trees that satisfy $\exists x p(x)$, can be defined by the formula $\exists x \in Reach(r) \ \wedge p(x)$. It is however not definable in EMSO.*

With more expressive languages such as full monadic second-order logic, this distinction is no longer needed since bisimulation invariance over trees only or bisimulation invariance over arbitrary graphs coincide in terms of resulting expressive power (see [11] and [9] for more details).

Though in this paper, we are mainly interested in *monadic* second-order logic, it will be useful to consider also full second order, possibly non-monadic, formulas. The difference with monadic second-order logic is that we are now given second-order variables $R_k, \ldots$, ranging over relations between vertices, of any finite nonzero arity $k, \ldots$ (when $k = 1$ we have just sets of vertices, like in the monadic case); here the additional atomic formulas will have the form $R_k(t_1, \ldots, t_k)$, where $t_1, \ldots, t_k$ are first-order variables or constants.

3.2 Modal and counting mu-calculus

The set of the modal μ-calculus formulas is the smallest set containing $Pred \cup Var$ which is closed under negation, disjunction and the following formation rules:

- if α is a formula then $\Diamond\alpha$ and $\Box\alpha$ are formulas,

- if α is a formula and X occurs only positively (i.e. under an even number of negations) in α then $\mu X.\alpha$ and $\nu X.\alpha$ are formulas.

The set of counting μ-calculus formulas is defined as above replacing standard modalities $\Diamond$ and $\Box$ by counting modalities $\Diamond_k$ and $\Box_k$ for any integer k.

A formula φ is said in *positive normal form* when negation only applies to atomic sub-formulas, i.e. constant or variable predicates of $\mathit{Pred} \cup \mathit{Var}$.

We use the same convention as for MSO with free variables, i.e. we denote by $\alpha(X_1, \cdots, X_n)$ a formula with free variables among $\{X_1, \cdots, X_n\}$. For convenience, we may also omit these free set variables in formula α considering implicitly that graphs have been built over the set of unary predicate symbols $\mathit{Pred}' = \mathit{Pred} \cup \{X_1, \cdots, X_n\}$. In the sequel, we call *fixed point formula* any formula of the modal or counting μ-calculus.

With each fixed point formula α, we associate an unary MSO predicate $\varphi_\alpha(x)$ with the same free variables (implicitly added to the vocabulary) defined as follows. Let $p \in Pred$, α and β be fixed point formulas, X be a set variable, x and z be FO variables, and $\overline{z} = (z_1, \cdots, z_k)$ be a k-tuple of FO variables.

- Atomic formulas :
  $\varphi_p(x) = p(x)$ and $\varphi_X = X(x)$,
- Boolean connectives :
  $\varphi_{\alpha\wedge\beta}(x) = \varphi_\alpha(x) \wedge \varphi_\beta(x)$, $\varphi_{\alpha\vee\beta}(x) = \varphi_\alpha(x) \vee \varphi_\beta(x)$
  and $\varphi_{\neg\alpha}(x) = \neg\varphi_\alpha(x)$
- Modalities :
  $\varphi_{\Diamond\alpha}(x) = \exists z\ E(x, z) \wedge \varphi_\alpha(z)$,
  $\varphi_{\Box\alpha}(x) = \forall z\ (E(x, z) \Rightarrow \varphi_\alpha(z))$
- Counting modalities :
  $\varphi_{\Box_k\alpha}(x) = \forall\overline{z}\ ((\mathit{diff}(\overline{z}) \wedge \bigwedge_{i\in[1,k]} E(x, z_i)) \Rightarrow \bigvee_{i\in[1,k]} \varphi_\alpha(z_i))$,
  $\varphi_{\Diamond_k\alpha}(x) = \exists\overline{z}\ \mathit{diff}(\overline{z}) \wedge \bigwedge_{i\in[1,k]} E(x, z_i) \wedge \varphi_\alpha(z_i)$,
- Fixed points :
  $\varphi_{\mu X.\alpha(X)}(x) = \forall X((\varphi_{\alpha(X)} \subseteq X) \Rightarrow X(x))$,
  $\varphi_{\nu X.\alpha(X)}(x) = \exists X(X \subseteq \varphi_{\alpha(X)}) \wedge X(x)$.

There, $\mathit{diff}(\overline{z})$ is the quantifier-free FO formula stating that $z_i \neq z_j$ for all $i \neq j$, $\varphi_{\alpha(X)} \subseteq X$ is the MSO formula $\forall z(\varphi_{\alpha(X)}(z) \Rightarrow X(z))$, and, similarly, $X \subseteq \varphi_{\alpha(X)}$ is the MSO formula $\forall z(X(x) \Rightarrow \varphi_{\alpha(X)}(z))$.

For any fixed point formula α, we write $M \models \alpha$ when $M \models \varphi_\alpha(r)$. We say that an MSO sentence φ is equivalent to a fixed point formula α when $\models \varphi_\alpha(r) \Leftrightarrow \varphi$. Likewise, two fixed point formulas α and β are said to be equivalent when $\models \varphi_\alpha(r) \Leftrightarrow \varphi_\beta(r)$.

Fact 3.2 *Any fixed point formula is equivalent to a fixed point formula in positive normal form.*

In the sequel, we will assume (often implicitly) that the formulas we are dealing with, are in positive normal form.

Observe that modalities $\Diamond$ and $\Diamond_1$ on the one hand, and modalities $\Box$ and $\Box_1$ on the other hand, have equal meaning. So the counting mu-calculus is an extension of the modal mu-calculus. This extension is proper. In fact, it is an easy exercise to show that, for instance, predicate $\Diamond_2 p$ is not definable in the modal mu-calculus.

As is well known, the interpretation over a graph M of the predicate defined by $\mu X.\alpha(X)$ (resp. $\nu X.\alpha(X)$) is the least (resp. the greatest) solution of (the interpretation over $\mathit{dom}(M)$ of) the set equation defined by $X = \alpha(X)$.

Further analysis of the basic properties of the modal (resp. the counting) fixed point calculus shows that it does not distinguish bisimilar (resp. counting bisimilar) models.

Fact 3.3 (Folklore) *For any modal (resp. counting) fixed point formula α, formula $\varphi_\alpha(r)$ is bisimulation invariant (resp. counting bisimulation invariant).*

In particular, since any graph M is bisimilar (or even counting bisimilar) to its reachable part $\mathit{Reach}(M)$, the truth of a fixed point formula in the root r of a given graph M only depends on the subgraph induced by the set $\mathit{Reach}(M)$ of all vertices reachable from the root r^M.

The following theorem shows that the bisimulation invariance (resp. counting bisimulation invariance) not only holds but even characterizes modal (resp. counting) fixed point calculi as fragments of MSO. In fact:

Theorem 3.4 (from Walukiewicz [22]) *An MSO sentence is invariant under counting bisimulation iff it is equivalent to some counting mu-calculus formula.*

Remark 3.5 *To be precise, the counting mu-calculus is not mentioned in Walukiewicz's paper. What is proved there is rather the equivalence, over arbitrary trees, between MSO sentences and counting automata as defined in the next section (see Theorem 4.2). But it is an easy exercise to check that transition specifications in counting automata can be defined by counting modalities (see for instance Courcelle's remarks on MSO over amorphous sets [6]). Then usual techniques, as described for instance in [10] apply, to translate counting automata to equivalent counting mu-calculus formulas.*

Theorem 3.6 (Janin-Walukiewicz [11]) *An MSO sentence is invariant under bisimulation iff it is equivalent to some modal mu-calculus formula.*

Finally, let us mention that alternation of least and greatest fixed point constructions induces a hierarchy in modal or counting mu-calculus. In the modal case it has been proven infinite by Bradfield [4]. A similar result by Arnold [1] shows it is infinite as well in the counting case.

Actually, Arnold's result is stated for the modal mu-calculus on the binary tree. But this implies the more general statement above, as both counting and modal mu-calculus are equivalent on the binary tree, and the binary tree itself is definable in the counting mu-calculus by a formula with greatest fixed point constructions only.

In this paper, we are mainly interested in the first level of the counting and the modal mu-calculus hierarchy: the set of fixed point formulas, in positive normal form, defined with greatest fixed point operator only. For this reason, in the sequel, this level is called the nu-level.

4 Tree automata

We review here a notion of tree automata that characterizes the expressive power of the counting and modal mu-calculus [10, 22]. Our aim in this section is to obtain an automata-theoretic (and mu-calculus) characterization of topologically closed languages of finitely branching trees.

4.1 Infinite tree automata

Before defining the notion of tree automata, we need to specify sets of formulas which are used to specify tree automata transitions. Given a finite set $Q = \{q_0, q_1, \cdots, q_n\}$ of unary predicate symbols, let $CNT(Q)$ be the set of finite disjunctions of FO-formulas of the form

$$\exists x_1, \ldots, x_k . \mathit{diff}(x_1, \ldots, x_k) \wedge \bigwedge_{i \in \{1, \cdots, k\}} q_{j_i}(x_i) \wedge \left(\forall z . \mathit{diff}(z, x_1, \ldots, x_k) \Rightarrow \bigvee_{j \in J} q_j(z) \right)$$

where $j_1, \ldots, j_k \in \{0, \cdots, n\}$, $J \subseteq \{0, \cdots, n\}$. Let also $MDL(Q)$ be the set of finite disjunctions of FO-formulas of the form

$$\exists x_1, \ldots, x_k . \bigwedge_{i \in \{1, \cdots, k\}} q_{j_i}(x_i) \wedge \forall z . \bigvee_{j \in J} q_j(z)$$

with the same notation as above.

A counting (resp. modal) tree automaton with parity conditions is defined as a tuple :

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

with Q a finite set of states, $q_0 \in Q$ an initial state, $\Omega : Q \to I\!N$ a priority function, and $\delta : Q \times \Sigma \to CNT(Q)$ (resp. $\delta : Q \times \Sigma \to MDL(Q)$) a counting (resp. modal) transition specification function.

The *index* of the automaton $\mathcal{A}$ is defined as $\max(\Omega(Q))$. As Q is finite, this is well defined.

In the sequel, the *alphabet* Σ is defined as the powerset $\mathcal{P}(\mathit{Pred})$. The intuition behind this is that a vertex v in a tree M is labeled by the letter $\lambda(v) = \{p \in \mathit{Pred} : v \in p^M\}$ which belongs to Σ.

We can put automata to work, in such a way that they recognize languages of trees. If one is interested in defining these notions over arbitrary graphs instead of trees, the most convenient way of defining the notion of runs and accepting runs is probably by means of games [22]. We use here an equivalent, yet more direct, definition for trees.

Given an automaton $\mathcal{A} = \langle Q, \Sigma, q_0, \Omega, \delta \rangle$ and a tree M, a *run of $\mathcal{A}$ over M* is defined as a mapping $\rho : V^M \to Q$ such that:

1. $\rho(r^M) = q_0$,

2. for each $v \in V^M$, given $succ(v)$ the set of immediate successors of node v, given $m_{\rho,v} = \langle succ(v), \rho^{-1} \rangle$ the FO-structure over the vocabulary Q induced by this set and the run function ρ^{-1} (by interpreting any state q by the set $q^{m_{\rho,v}} = \rho^{-1}(q) \cap succ(v)$), we have $m_{\rho,v} \models \delta(q, \lambda(v))$.

A run ρ is said to be accepting when, moreover,

3. for any infinite path $v_0.v_1.v_2.\cdots$ of M, the least priority occurring infinitely often in the sequence $\Omega(\rho(v_0)).\Omega(\rho(v_1)).\Omega(\rho(v_2)).\cdots$ is even.

The language of trees $L(\mathcal{A})$ accepted, or recognized, by the automaton $\mathcal{A}$ is defined as follows.

1. If $\mathcal{A}$ is a *counting automaton* then $L(\mathcal{A})$ is the set of all trees M such that there exists an accepting run ρ of $\mathcal{A}$ over M.

2. If $\mathcal{A}$ is a *modal automaton* with κ states then $L(\mathcal{A})$ is the set of all trees M such that there exists an accepting run ρ of $\mathcal{A}$ over $T^\kappa(M)$, i.e. over the κ-expansion of the tree M.

Remark 4.1 *When $\mathcal{A}$ is a modal automaton, the "shift" to the κ-expansion in the definition of $L(\mathcal{A})$ ensures that $L(\mathcal{A})$ is, indeed, closed under bisimulation equivalence. Without this "shift" to κ-expansions, the language defined by a modal automaton may be not bisimulation closed. Intuitively, it prevents automaton $\mathcal{A}$ from counting successors as illustrated by the following example.*

Let $\mathcal{A} = \langle \{q_0, q_1\}, \{a\}, q_0, \Omega, \delta \rangle$ where, for $i = 0$ or 1, $\Omega(q_i) = 0$ and $\delta(q_i, a) = \exists x_0\ x_1\ (q_0(x_0) \wedge q_1(x_1))$. One can check that modal automaton $\mathcal{A}$ has an accepting run on the a-labeled binary tree. But it has no run on the a-labeled unary tree allthough this unary tree is bisimilar to the binary tree.

With these notions of modal and counting automata, we have:

Theorem 4.2 ([10] and [22]) *Let Σ be a finite alphabet, for any languages L of Σ-labeled trees the following properties are equivalent :*

1. *L is definable in MSO (resp. in bisimulation invariant MSO),*

2. *L is definable in counting mu-calculus (resp. in modal mu-calculus),*

3. *L is recognized by a counting automaton (resp. a modal automaton).*

Moreover, L is definable in the nu-level of counting (resp. modal) mu-calculus if and only if it is recognized by a counting (resp. modal) automaton of index zero.

In particular, this gives the equivalence between (3) and (6) in Theorem 1.1.

4.2 More on recognizable closed languages

Here, we consider the languages of trees which are closed in the topological sense. We prove that the languages of finitely branching trees accepted by modal or counting automata that are closed in the topological sense, are exactly those definable by means of (counting or modal) fixed point formulas of the nu-level.

The following lemma asserts that the restriction to finitely branching trees is harmless, in the sense that recognizable languages are characterized by the finitely branching trees they contain.

Lemma 4.3 (Finitely branching lemma [18]) *Two recognizable languages of trees are equal if and only if they contain the same set of finitely branching trees.*

The first step in the proof of Theorem 1.1 is then given by the following statement.

Proposition 4.4 *For any MSO-definable language L of finitely branching Σ-labeled trees, the following properties are equivalent:*

1. *L is closed in the prefix topology (resp. closed in the prefix topology and closed under bisimulation),*
2. *L is recognized by a counting (resp. modal) automaton of index zero.*

Proof. Let L be an MSO-definable language. Applying Theorem 4.2, there is a modal or counting (depending on whether L is bisimulation closed or not) automaton $\mathcal{A} = \langle Q, \Sigma, q_0, \Omega, \delta \rangle$ such that $L = L(\mathcal{A})$.

Without loss of generality, we may assume that any state $q \in Q$ is productive, i.e. for any state q there is at least one tree which is recognized from this state. We may also assume that any transition is productive, i.e. for any state q, any $a \in \Sigma$ such that $\delta(q, a)$ is satisfiable, there is at least one (finitely branching) tree $T_{q,a}$ such that the root of $T_{q,a}$ is labeled by a and there exists an accepting run of $\mathcal{A}$ with initial state q instead of q_0 over the tree $T_{q,a}$. Since non-productive states or transitions cannot occur in an accepting run, all such states or transitions can be deleted from $\mathcal{A}$ without altering the accepted language $L(\mathcal{A})$.

Let then $\overline{\mathcal{A}} = \langle Q, \Sigma, q_0, 0, \delta \rangle$ be the automaton obtained from $\mathcal{A}$ just replacing the priority function Ω with the constant function 0.

To prove the equivalence, it is sufficient to prove that (over finitely branching trees) $L(\overline{\mathcal{A}})$ is the topological closure $\overline{L(\mathcal{A})}$ of $L(\mathcal{A})$.

We prove first that $L(\overline{\mathcal{A}})$ is closed. As the prefix topology can be defined by a metric, it is sufficient to show that if $\{M_n\}_{n\in\mathbb{N}}$ is a sequence of trees in $L(\overline{\mathcal{A}})$ which converges to a tree M then $M \in L(\overline{\mathcal{A}})$.

For each $n \in \mathbb{N}$, let ρ_n be an accepting run of $\overline{\mathcal{A}}$ over M_n. Considering the sequence $\{\langle M_n, \rho_n\rangle\}_{n\in\mathbb{N}}$ of $\mathcal{P}(\mathit{Pred} \cup Q)$-labeled trees, we know that its induced skeleton sequence converges (since $\{M_n\}_{n\in\mathbb{N}}$ converges). Applying Lemma 2.4 shows that it has a converging subsequence. The limit of that subsequence must be of the form $\langle M, \rho\rangle$ where M is the limit of $\{M_n\}_{n\in\mathbb{N}}$ and ρ is a run of $\overline{\mathcal{A}}$ over M. Now, as $\overline{\mathcal{A}}$ is of index zero, the run ρ is accepting hence $M \in L(\overline{\mathcal{A}})$.

To continue the proof, we observe that the inclusion $\overline{L(\mathcal{A})} \subseteq L(\overline{\mathcal{A}})$ is immediate as $L(\mathcal{A}) \subseteq L(\overline{\mathcal{A}})$ and $L(\overline{\mathcal{A}})$ is closed. It remains thus to show that $L(\overline{\mathcal{A}}) \subseteq \overline{L(\mathcal{A})}$.

Let now M be a finitely branching tree in $L(\overline{\mathcal{A}})$ and let ρ be an accepting run of $\overline{\mathcal{A}}$ over M. It is sufficient to show that there is a sequence $\{M_n\}_{n\in\mathbb{N}}$ of (finitely branching) trees in $L(\mathcal{A})$ which converges to M. For each $n \in \mathbb{N}$, let us define M_n as the tree obtained from the finite tree $P_n(M)$ by attaching, under each leaf v of $P_n(M)$, the tree $T_{\rho(v),\lambda(v)}$ (with a root labeled $\lambda(v)$ and accepted by the automaton $\mathcal{A}$ from the initial state $\rho(v)$).

By construction, each tree M_n belongs to $L(\mathcal{A})$ and the sequence $(M_n)_{n\in\mathbb{N}}$ converges to M which concludes the proof.

Strictly speaking, in the case where $\mathcal{A}$ is a modal automaton, we must consider in this proof runs over the κ-expansions $T^\kappa(M_n)$ of the M_ns for $\kappa = |Q|$ instead of runs on the M_ns themselves. However, this makes no difference in the argument as the κ-expansion permutes with limits. □

This proposition gives the equivalence between (5) and (6) in Theorem 1.1. In the binary case, a very similar result is obtained by Mostowski [15].

Observe that, as a consequence of this proposition, we also have:

Corollary 4.5 *Any MSO-definable language of trees which is closed in the prefix topology is definable by means of an MS formula of the form*

$$\exists \overline{X} X_0(r) \wedge \forall x \varphi_\alpha(x, \overline{X})$$

where X_0 is one of the variables occurring in $\overline{X}$, and $\alpha(\overline{X})$ is a depth one counting formula.

Proof. Let L be an MSO-definable language of trees closed in the prefix topology. Applying Proposition 4.4, let $\mathcal{A}$ be a counting (or modal) automaton of rank 0 recognizing this language. The formula of the desired form is then obtained as follows. It expresses the existence of an accepting run of automaton $\mathcal{A}$ with each variable X_q (one per state q) in $\overline{X}$ encoding the set of vertices labeled by state q (with X_0 encoding the intial state) and $\varphi_\alpha(x, \overline{X})$ describing the (local) transition specification. □

Following the standard terminology [20], this corollary can be restated as follows : closed MSO definable languages of infinite trees are projection of locally testable languages of trees. Here, by localy testable, we mean languages that are defined by universaly quantified local FO-formulas.

The corollary 4.5 proves that both (5) or (6) imply (4) in Theorem 1.1.

5 Bisimulation invariance in existential MSO

In this section, we conclude the proof of Theorem 1.1 by proving that language of finitely branching trees defined by bisimulation (resp. counting bisimulation) invariant formulas of EMSO (1) or CEMSO (2) are recognizable and closed in the prefix topology (5).

In order to do so, we prove in Section 5.1, by applying Łos Theorem to existential second-order logic (ESO), that classes of graphs definable in ESO are closed under ultraproduct. In Section 5.2 we prove that the ultraproduct of any converging sequence of finitely branching trees is counting bisimilar with its limit. And in Section 5.3 we apply this result to EMSO and CEMSO as both are fragments of ESO.

5.1 Ultraproducts and existential second-order logic

Let I be a set. An *ultrafilter* over I is a set $U \subseteq \mathcal{P}(I)$ of subsets of I such that $I \in U$, $\emptyset \notin U$, and U is closed under the following rules: for any A and $B \subseteq I$, if $A \in U$ and $A \subseteq B$ then $B \in U$; if A and $B \in U$ then $A \cap B$ in U; and either $A \in U$ or $I \setminus A \in U$. An ultrafilter U is *principal* if it contains a finite set, and non-principal otherwise. Observe that a non-principal ultrafilter over I contains all co-finite subsets of I.

With the help of the axiom of choice (or the Zorn Lemma) one can prove [5] that if I is an infinite set then there is a non-principal ultrafilter over I.

Assuming I is an infinite set, let U be an ultrafilter over I, and let $\{M_i\}_{i\in I}$ be an I-indexed collection of FO-structures over some relational vocabulary τ. The ultraproduct $\Pi_i^U M_i$ of $\{M_i\}_{i\in I}$ modulo U is defined as the quotient of the product structure $\Pi_i M_i$ under the congruence $\simeq_U$ defined, for any u and $v \in dom(\Pi_i M_i)$ by $u \simeq_U v$ when the set $\{i \in I : u_i = v_i\}$ belongs to U. This construction is motivated by the following theorem.

Theorem 5.1 (Łos) *For any FO sentence φ over the vocabulary τ, $\Pi_i^U M_i$ is a model of φ if and only if $\{i \in I : M_i \models \varphi\}$ belongs to U.*

As this holds for an arbitrary vocabulary τ, it leads to the following corollary.

Corollary 5.2 *For any formula φ of existential second-order logic on the vocabulary τ, if $\{i \in I : M_i \models \varphi\} \in U$ then $\Pi_i^U M_i \models \varphi$.*

Proof. By standard syntactic arguments, we can always assume φ is of the form $\exists R \psi(R)$ with $\psi(R)$ a FO formula over the vocabulary $\tau \cup \{R\}$. For each $i \in I$, let R_i be any interpretation of R over $dom(M_i)$ such that $M_i \models \psi(R_i)$ if and only if $M_i \models \exists R\psi(R)$. Assuming that $\{i \in I : M_i \models \exists R\psi\}$ belongs to U and considering ψ (resp. $\{\langle M_i, R_i\rangle\}_{i\in I}$) as a FO sentence (resp. an indexed collection of FO-structures) over the vocabulary $\tau \cup \{R\}$, Łos theorem can be applied to show that the ultraproduct $\Pi_i^U \langle M_i, R_i\rangle$ satisfies $\psi(R)$. It follows that, given R_U the congruence closure of $\Pi_i R_i$, by definition of an ultraproduct, $\Pi_i^U M_i \models \psi(R_U)$ and hence $\Pi_i^U M_i \models \exists R\psi(R)$. $\square$

5.2 Ultraproducts of converging sequences and limits

Lemma 5.3 *Let $\{M_n\}_{n\in\mathbb{N}}$ be a sequence of finitely branching Σ-labeled trees. Assume that $\{M_n\}_{n\in\mathbb{N}}$ converges to a limit $M \in FBT(Pred)$. Let U be a* *non-principal ultrafilter over $I\!N$. The two structures M and $Reach(\Pi_n^U M_n)$ are isomorphic.*

Proof. Let h be a strictly positive integer. We show that $P_h(M)$ and $P_h(\Pi_n^U M_n)$ are isomorphic.

Since M is the limit of $\{M_n\}_{n \in I\!N}$, there is a number n_h such that, for every $n \geq n_h$, $P_h(M_n)$ and $P_h(M)$ are isomorphic. As $P_h(M)$ is finite, there is also a FO formula φ_h such that, given any model N, $N \models \varphi_h$ if and only if $P_h(N)$ is isomorphic with $P_h(M)$. Then, as U is non-principal, the co-finite set $\{n \in I\!N : M_n \models \varphi_h\}$ belongs to U. By applying Łos' Theorem, we get $\Pi_n^U M_n \models \varphi_h$, and hence $P_h(\Pi_n^U M_n) \models \varphi_h$, so it is isomorphic to $P_h(M)$.

Since this holds for arbitrary $h > 0$, this implies in particular that $Reach(\Pi_n^U M_n) = \bigcup_h P_h(\Pi_n^U M_n)$ is finitely branching and thus, the result now follows from Lemma 2.3. □

5.3 Applications to bisimulation invariance

As EMSO and CEMSO are both fragments of MSO, we have:

Lemma 5.4 *Let L be a language of finitely branching trees definable by a bisimulation or counting bisimulation invariant EMSO or CEMSO sentence. Then L is both recognizable and topologically closed.*

Proof. As bisimulation invariance implies counting bisimulation invariance, we only need to prove this Lemma for counting bisimulation invariant sentences.

Let φ be a EMSO or CEMSO counting bisimulation invariant sentence. Let L be the class of (finitely branching) trees that satisfy φ. Since both EMSO and CEMSO are fragments of MSO, L is recognizable by Theorem 4.2.

Now, let $\{M_n\}_{n \in I\!N}$ be a sequences of finitely branching trees in L that converges towards a finitely branching tree M. In order to conclude the proof, we have to show that $M \in L$.

In order to do so, let U be a non principal ultrafilter over $I\!N$, and let N be the ultraproduct $\Pi_n^U M_n$. By Corollary 5.2 the class of models of φ is closed under ultraproduct so N satisfies φ. By Lemma 5.3, we also have that the limit M of limit of $\{M_n\}_{n \in I\!N}$ is isomorphic to $Reach(N)$ hence counting bisimilar to N. Now, since φ is counting bisimulation invariant, this shows that $M \models \varphi$, that is, $M \in L$. □

In other words, Lemma 5.4 proves that both (1) or (2) imply (5) in Theorem 1.1. As the implications from (4) to (1) and (1) to (2) are immediate for syntactic reasons, this concludes the proof of our main Theorem.

References

[1] A. Arnold. The mu-calculus alternation-depth hierarchy over the binary tree is strict. *Theoretical Informatics and Applications*, 33:329–339, 1999.

[2] A. Arnold, G. Lenzi, and J. Marcinkowski. The hierarchy inside closed monadic Σ_1 collapses on the infinite binary tree. In *IEEE Symp. on Logic in Computer Science (LICS)*, pages 157–166. IEEE Computer Society, 2001.

[3] J.F.A.K. van Benthem. *Modal Correspondance Theory.* PhD thesis, University of Amsterdam, 1976.

[4] J. Bradfield. The modal mu-calculus alternation hierarchy is strict. *Theoretical Comp. Science*, 195:133–153, 1998.

[5] C. C. Chang and H. J. Keisler. *Model Theory.* Studies in Logic and the Foundations of Mathematics vol. 73. North–Holland, 1990.

[6] B. Courcelle. The monadic second order logic of graph I: Recognizable sets of finite graphs. *Inf. and Comp.*, 85:12–75, 1990.

[7] H.D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic.* Undergraduate Texts in Mathematics. Springer-verlag, 1984.

[8] E. Emerson. Temporal and modal logic. In J. Van Leeuwen, editor, *Handbook of Theor. Comp. Science (vol. B)*, pages 995–1072. Elsevier, 1990.

[9] D. Janin and G. Lenzi. Relating levels of the mu-calculus hierarchy and levels of the monadic hierarchy. In *IEEE Symp. on Logic in Computer Science (LICS)*, pages 347–356. IEEE Computer Society, 2001.

[10] D. Janin and I. Walukiewicz. Automata for the modal mu-calculus and related results. In *Mathematical Found. of Comp. Science (MFCS).* LNCS 969, 1995.

[11] D. Janin and I. Walukiewicz. On the expressive completeness of the modal mu-calculus with respect to monadic second order logic. In *Conf. on Concurrency Theory (CONCUR'96)*, pages 263–277. LNCS 1119, 1996.

[12] D. Kozen. Results on the propositional μ-calculus. *Theoretical Comp. Science*, 27:333–354, 1983.

[13] R. Fagin M. Ajtai and L. Stockmeyer. The closure of monadic *np*. *Journal of Computer and System Sciences*, 60:660–716, 2000.

[14] R. Milner. Operational and algebraic semantics of concurrent processes. In J. Van Leeuwen, editor, *Handbook of Theor. Comp. Science (vol. B)*, pages 1201–1242. Elsevier, 1990.

[15] A.W. Mostowski. Hierarchies of weak automata on weak monadic formulas. *Theoretical Comp. Science*, 83:323–335, 1991.

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

[17] J. R. Shoenfield. *Mathematical Logic*. Addison-Wesley, Reading, Mass., 1967.

[18] R. S. Streett and E. A. Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81:249–264, 1989.

[19] M. Takahashi. The greatest fixed points and rational omega-tree languages. *Theoretical Computer Science*, 44:259–274, 1986.

[20] W. Thomas. Automata on infinite objects. In J. Van Leeuwen, editor, *Handbook of Theor. Comp. Science (vol. B)*, pages 133–191. Elsevier, 1990.

[21] W. Thomas. Automata theory on trees and partial orders. In M. Dauchet M. Bidoit, editor, *TAPSOFT'97*, pages 20–38. LNCS 1214, Springer-Verlag, 1997.

[22] I. Walukiewicz. Monadic second order logic on tree-like structures. In *Symp. on Theor. Aspects of Computer Science (STACS)*, 1996. LNCS 1046.