Baire Category Quantifier in Monadic Second Order Logic

Henryk Michalewski[1] and Matteo Mio[2(✉)]

[1] University of Warsaw, Warsaw, Poland
[2] CNRS/ENS-Lyon, Lyon, France
momatteo@gmail.com

Abstract. We consider Rabin's *Monadic Second Order* logic (MSO) of the full binary tree extended with Harvey Friedman's "for almost all" second-order quantifier ($\forall^*$) with semantics given in terms of Baire Category. In Theorem 1 we prove that the new quantifier can be eliminated (MSO+$\forall^*$ = MSO). We then apply this result to prove in Theorem 2 that the *finite-SAT* problem for the qualitative fragment of the probabilistic temporal logic pCTL* is decidable. This extends a previous result of Brázdil, Forejt, Křetínský and Kučera valid for qualitative pCTL.

Keywords: Monadic second order logic · Baire category · pCTL*

1 Introduction

The main motivation of this paper is purely logical. We investigate the extension of Rabin's *Monadic Second Order Theory of the Full Binary Tree* [14], henceforth simply shortened as MSO, with an additional "for almost all" second-order quantifier $\forall^*$ whose set-theoretic semantics is defined as:

$$\forall^* X.\phi(X,\vec{Y}) \stackrel{\text{def}}{=} \big\{ \vec{Y} \mid \{X \mid \neg\phi(X,\vec{Y}) \text{ holds}\} \text{ is "topologically small" } \big\}$$

where topologically small is interpreted as *of Baire first category* (or *meager*) in the standard topology on subsets of the full binary tree. Thus, for example, the closed formula $\forall^* X.\phi(X)$ is valid if ϕ holds on all but a meager collection of X's. To the best of our knowledge, the quantifier $\forall^*$ has been first introduced and investigated, in the general context of First Order Logic, by H. Friedman in unpublished manuscripts in 1978–79 (see [17] for an overview of this research).

Another extension of MSO with a *large cardinality* quantifier $\forall^{\aleph_0}$, defined by replacing "is topologically small" with "has cardinality $\leq \aleph_0$" in the equation defining $\forall^*$, has been recently investigated in [3] where it is proved that for every

H. Michalewski—Author supported by Polands National Science Centre grant no. 2014-13/B/ST6/03595.

M. Mio—Author supported by grant "Projet Émergent PMSO" of the École Normale Supérieure de Lyon.

M.M. Halldórsson et al. (Eds.): ICALP 2015, Part II, LNCS 9135, pp. 362–374, 2015.
DOI: 10.1007/978-3-662-47666-6_29

MSO+$\forall^{\aleph_0}$ formula ψ there exists an equivalent MSO (without $\forall^{\aleph_0}$) formula $\hat{\psi}$ such that ψ and $\hat{\psi}$ denote the same set, that is the quantifier $\forall^{\aleph_0}$ can be expressed (i.e., *eliminated*) in ordinary MSO. This implies that the theory MSO+$\forall^{\aleph_0}$ is decidable. We prove a similar result for MSO+$\forall^*$.

Theorem 1. *For every* MSO+$\forall^*$ *formula* ψ *there exists an equivalent MSO (without* $\forall^*$*) formula* $\hat{\psi}$ *such that* ψ *and* $\hat{\psi}$ *denote the same set.*

Corollary 1. *The theory of* MSO+$\forall^*$ *is decidable.*

Our proof uses the fact, first proved in [11, Theorem 6.6], that every MSO–definable set of trees satisfies the *Baire property* (another, somewhat more elementary argument, can be deduced from Kolmogorov's theory of $\mathcal{R}$-sets, see [4, Theorem 3.8] and [9]). As a consequence, the Baire category of regular sets of trees can be determined using the well-known Banach–Mazur game (see, e.g., [12, §8]). Our main observation is that the game itself can be "implemented" via an alternating tree automaton (see Figure 4) which, while technically involved, is conceptually simple. For this reason our proof is radically different from that of [3] which is based on Shelah's *composition method* and does not involve automata constructions. An interesting topic for future research is to verify if, with techniques similar to those used in this work, the quantifier elimination theorem for MSO + $\forall^{\aleph_0}$ of [3] can be proved using purely automata based methods. Further topics of future research and related work are discussed at the end of Section 3. The investigation of sets definable by formulas $\forall^* X.\phi(X, \vec{Y})$, with ϕ specifying Borel, analytic or $\mathcal{R}$–sets, instead of regular sets, has been an active area of research in descriptive set theory with contributions from R. Barua, J. P. Burgess, D. Miller, P. S. Novikov and R. Vaught among others. Our Theorem 1 can be considered an effective, automata–theoretic counterpart of set-theoretic results such as [20, Corollary1.10], [12, Theorem 29.22] and [4, Theorem 6.3].

An application to the theory of pCTL.* We apply our result on MSO+$\forall^*$ to the satisfiability problem of *probabilistic temporal logics* of programs (modeled as Markov chains) such as pCTL* [10] (see also [2, §10.4]). The work of Brázdil, Forejt, Křetínský and Kučera [6] is a main source of results in this area of research. A pCTL* formula can generally be satisfiable but only by infinite models, that is, by infinite Markov chains. Thus we distinguish between the *SAT problem* and the *finite–SAT problem* which, more restrictively, asks about the existence of finite models. In [6] the authors proved that both the SAT and the finite–SAT problems for the *qualitative fragment* of pCTL are decidable. We extend the second of these results to pCTL*.

Theorem 2. *The finite–SAT problem for qualitative pCTL* is decidable.*

Our proof method is based on a reduction of the finite–SAT problem to the satisfiability of MSO+$\forall^*$ which is decidable by Corollary 1. This proof technique is of general applicability and an equivalent of Theorem 2 can be proved even for more expressive probabilistic logics[1]. In contrast, the results of [6] provide much tighter algorithmic information, but the proof methods are specifically tailored to the logic pCTL and their applicability to other logics is not clear and requires a separate study.

The semantics of qualitative pCTL* is based on the (measure-theoretic) probabilistic concepts of *null set*, *set of positive measure* and *set of measure* 1. To reduce it to MSO+$\forall^*$, which can express the concept of Baire category, we apply a remarkable result of Ludwig Staiger ([16, Theorem 4]) which says that a regular set $L\subseteq\Sigma^\omega$ of infinite words is comeager if and only if $\mu(L)=1$, where μ is the standard Lebesgue measure. This result is also used in [21] to develop a theory of fairness for concurrent systems. Using Staiger's theorem we prove that pCTL* with its standard probabilistic semantics and pCTL* with an alternative "Baire-categorical" semantics, where the state-formula $\mathbb{P}_{=1}\phi$ is interpreted as: "$s\models\mathbb{P}_{=1}\phi \Leftrightarrow$ the set of paths starting from s and satisfying the path–formula ϕ is comeager", agree on all finite models. This kind of observation is not new and has been already made (with respect to another logic) in the recent literature [1]. A proof of Theorem 2 is then obtained by combining these facts and by showing that the Baire–categorical semantics of pCTL* can be interpreted in MSO+$\forall^*$.

2 Background in Topology, Logic and Automata

Topology. Our exposition of topological and set–theoretical notions follows [12]. Given a topological space X, a set $A\subseteq X$ is *nowhere dense* if the interior of its closure is the empty set, that is $(\mathtt{int}(\mathtt{cl}(A))=\emptyset$. A set $A\subseteq X$ is of *(Baire) first category* (or *meager*) if A can be expressed as countable union of nowhere dense sets. A set $A\subseteq X$ which is not meager is *of the second (Baire) category*. The complement of a meager set is called *comeager*. A set $B\subseteq X$ has the *Baire property* if $B=U\triangle M$, for some open set $U\subseteq X$ and meager set $M\subseteq X$, where $\triangle$ is the operation of symmetric difference $U\triangle M=(U\cup M)\setminus(U\cap M)$.

The set of natural numbers is denoted by ω. A topological space X is Polish if it is separable and completely metrizable. A main example of Polish space is the *Cantor space* $\{0,1\}^\omega$ of infinite sequences of bits endowed with the product topology. The Cantor space is *zero-dimensional*, i.e., it has a basis of *clopen* (both open and closed) sets. We now describe the well-known Banach–Mazur game (see [12, 8.H] for a detailed overview) which characterizes Baire category.

Definition 1 (Banach–Mazur Game). *Let X be a zero–dimensional Polish space. For a given payoff set $A\subseteq X$ the infinite duration game* $\mathbf{BM}(X,A)$ *is played by Player I and Player II by sequentially choosing non-empty clopen sets*

$$\begin{array}{lcccc} \textit{Player I} & U_0 & & U_2 & \dots \\ \textit{Player II} & & U_1 & & U_3 \dots \end{array}$$

[1] E.g., Theorem 2 can be proved for the, easily conceivable but to our knowledge never appeared in published work, probabilistic version of the non–probabilistic logic ECTL* (see, e.g., [18] for an introduction to non–probabilistic ECTL*).

with $U_{n+1} \subsetneq U_n$. *Player I wins if* $\bigcap_{n\in\omega} U_n \cap A \neq \emptyset$ *and Player II wins otherwise.*

Theorem 3. *Let* X *be a zero–dimensional Polish space. If* $A \subseteq X$ *has the Baire property, then* $\mathbf{BM}(X, A)$ *is determined and Player II wins iff* A *is meager.*

Monadic Second Order Logic. We assume familiarity of the reader with exposition of monadic second order logic (MSO). A standard reference is [19].

The set $\{L, R\}^*$ of finite words over the alphabet $\{L, R\}$ is called the *full binary tree* and each $w \in \{L, R\}^*$ is referred to as a *vertex*. The functions $\texttt{Succ}_L$ ($w \mapsto w.L$) and $\texttt{Succ}_R$ ($w \mapsto w.R$) are called *successor operations*. Given a finite alphabet Σ the function space $(\{L, R\}^* \to \Sigma)$ is denoted by $\mathcal{T}_\Sigma$ and an element $t \in \mathcal{T}_\Sigma$ is called a Σ-labeled tree, or just a Σ-tree. We identify $\{0, 1\}$-labeled trees, seen as characteristic functions, with sets of vertices. A tree $t \in \mathcal{T}_\Sigma$ is called *regular* if it has only finitely many subtrees up-to isomorphism. The space $\mathcal{T}_\Sigma$ has a natural topology homeomorphic to the Cantor space. A basis for the topology consists of clopen sets of Σ-trees extending a given finite prefix.

The language of MSO consists of *first order variables* x, y (ranging over vertices $w \in \{L, R\}^*$), second order variables X, Y (ranging over sets of vertices $t \in \mathcal{T}_{0,1}$), the set-theoretic membership relation $x \in X$ (interpreted as usual), the operations $\texttt{Succ}_L$ and $\texttt{Succ}_R$ (with interpretation given as above), the usual Boolean connectives ($\vee$, $\wedge$, $\neg$), first order quantifiers ($\forall x.\phi$, $\exists x.\phi$) and the second order quantifiers ($\forall X.\phi$, $\exists X.\phi$). In the rest of this paper we will consider MSO formulas whose free variables are all second order. This is not a significant restriction since it is well known (see, e.g., [19]) how to present MSO as a purely second order (i.e., without first order variables and quantifiers) theory. For a vector $\vec{Y} = (Y_1, \ldots, Y_n)$ and a variable X we write $\phi(X, \vec{Y})$ to denote that ϕ has precisely $n+1$ free variables $X, Y_1, \ldots, Y_n$. The set theoretic semantics of $\phi(X, \vec{Y})$ is the collection of $n + 1$-tuples of $\{0, 1\}$-trees $\langle t_0, t_1, \ldots, t_n\rangle$ satisfying the formula ϕ. Equivalently, $\phi(X, \vec{Y})$ defines a collection of Σ-trees with $\Sigma = \{0, 1\}^{n+1}$. A subset $A \subseteq \mathcal{T}_\Sigma$ is *regular* if it is definable by a MSO formula. Given a formula $\phi(X, \vec{Y})$ and a tuple $\vec{t} = \langle t_1, \ldots, t_n\rangle$ of $\{0, 1\}$-trees, the formula $\phi(X, \vec{t})$ with *parameters* $\vec{t}$ denotes the *section* $\{t_0 \mid \langle t_0, t_1, \ldots, t_n\rangle \in \phi(X, \vec{Y})\} \subseteq \mathcal{T}_{0,1}$. Note that $\phi(X, \vec{t})$ needs not be regular, but is regular when $\vec{t}$ is a regular tree.

Alternating Tree Automata. The importance of MSO stems from the fact that the theory is decidable. An approach to the proof of decidability taken in [13] is based on *alternating tree automata*. We include a brief exposition of alternating automata which follows the presentation in [13, Appendix C].

Definition 2 (Alternating automaton). *Given a finite set* X*, we denote with* $\mathcal{DL}(X)$ *the set of expressions* e *generated by the grammar* $e ::= x \in X \mid e \wedge e \mid e \vee e$. *An* alternating tree automaton *over a finite alphabet* Σ *is a tuple* $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \mathcal{F}\rangle$ *where* Q *is a finite* set of states*,* $q_0 \in Q$ *is the* initial state*,* $\delta : Q \times \Sigma \to \mathcal{DL}(\{L, R\} \times Q)$ *is the* alternating transition function*,* $\mathcal{F} \subseteq \mathcal{P}(Q)$ *is a set of subsets of* Q *called the* Muller condition*. The Muller condition* $\mathcal{F}$ *is* *called a* parity condition *if there exists a* parity assignment $\pi : Q \to \omega$ *such that:* $\mathcal{F} = \{F \subseteq Q \mid (\max_{q \in F} \pi(q)) \text{ is even}\}$.

An alternating automaton $\mathcal{A}$ over the alphabet Σ defines, or "accepts", a set of Σ-trees. The *acceptance* of a tree $t \in \mathcal{T}_\Sigma$ is defined via a two-player ($\exists$ and $\forall$) game of infinite duration denoted as $\mathcal{A}(t)$. Game states of $\mathcal{A}(t)$ are of the form $\langle \vec{x}, q\rangle$ or $\langle \vec{x}, e\rangle$ with $\vec{x} \in \{L, R\}^*$, $q \in Q$ and $e \in \mathcal{DL}(\{L, R\} \times Q)$.

The game $\mathcal{A}(t)$ starts at state $\langle \epsilon, q_0\rangle$. Game states of the form $\langle \vec{x}, q\rangle$, including the initial state, have only one successor state, to which the game progresses automatically. The successor state is $\langle \vec{x}, e\rangle$ with $e = \delta(q, a)$, where $a = t(\vec{x})$ is the labeling of the vertex $\vec{x}$ given by t. The dynamics of the game at states $\langle \vec{x}, e\rangle$ depends on the possibly nested shape of e. If $e = e_1 \vee e_2$, then Player $\exists$ moves either to $\langle \vec{x}, e_1\rangle$ or $\langle \vec{x}, e_2\rangle$. If $e = e_1 \wedge e_2$, then Player $\forall$ moves either to $\langle \vec{x}, e_1\rangle$ or $\langle \vec{x}, e_2\rangle$. If $e = (L, q)$ then the game progresses automatically to the state $\langle \vec{x}.L, q\rangle$. Lastly, if $e = (R, q)$ the game progresses automatically to the state $\langle \vec{x}.R, q\rangle$. Thus a play in the game $\mathcal{A}(t)$ is a sequence Π of game–states, that looks like: $\Pi = (\langle \epsilon, q_0\rangle, \dots, \langle L, q_1\rangle, \dots, \langle LR, q_2\rangle, \dots, \langle LRL, q_3\rangle, \dots, \langle LRLL, q_4\rangle, \dots)$, where the dots represent part of the play in game–states of the form $\langle \vec{x}, e\rangle$. Let $\infty(\Pi)$ be the set of automata states $q \in Q$ occurring infinitely often in configurations $\langle \vec{x}, q\rangle$ of Π. We then say that the play Π of $\mathcal{A}(t)$ is winning for $\exists$, if $\infty(\Pi) \in \mathcal{F}$. The play Π is winning for $\forall$ otherwise. The set (or "language") of Σ-trees defined by $\mathcal{A}$ is the collection $\{t \in \mathcal{T}_\Sigma \mid \exists \text{ has a winning strategy in the game } \mathcal{A}(t)\}$.

Definition 3. *An alternating automaton $\mathcal{A}$ is called* non-deterministic *if for all $q \in Q$ and $a \in \Sigma$, the expression $\delta(q, a)$ is n-ary disjunction $e_1 \vee \dots \vee e_n$ where each disjunct e_i is a binary conjunction of the form $\langle L, q_1\rangle \wedge \langle R, q_2\rangle$ with q_1, q_2 not necessarily distinct.*

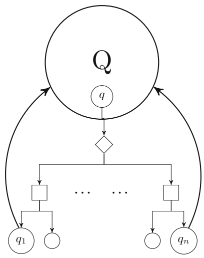

Fig. 1. A non–deterministic automaton $\mathcal{A}$ with states Q

We will visualize alternating automata by diagrammatic pictures with the convention that ◇-shaped and □-shaped positions mark decisions of Player $\exists$ and Player $\forall$, respectively. For example, Figure 1 illustrates the shape of a non-deterministic automaton.

The following theorem is of fundamental importance and states that alternating and nondeterministic automata have the same expressive power.

Theorem 4. *[13] For every alternating automaton $\mathcal{A}$ there exists a non–deterministic parity automaton $\mathcal{B}$ defining the same set as $\mathcal{A}$.*

3 The Quantifier $\forall^*$ in MSO

In this section we introduce the extension of MSO with Friedman's "for almost all" quantifier interpreted using the concept of Baire category.

Definition 4 ($\mathrm{MSO}+\forall^*$)**.** *The syntax of* $\mathrm{MSO}+\forall^*$ *extends that of* MSO *with the new second-order quantifier* $\forall^*$ *whose set-theoretic semantics is defined as:*

$$\forall^* X.\phi(X,\vec{Y}) \stackrel{\text{def}}{=} \left\{\vec{t} \mid \neg\phi(X,\vec{t}) \subseteq \mathcal{T}_{0,1} \text{ is of first category}\right\}$$

The dual quantifier $\exists^* X.\phi$*, derivable as* $\exists^* X.\phi = \neg\forall^* X.(\neg\phi)$ *denotes the set* $\exists^* X.\phi(X,\vec{Y}) = \left\{\vec{t} \mid \phi(X,\vec{t}) \subseteq \mathcal{T}_{0,1} \text{ is of second category}\right\}$.

The set denoted by $\forall^* X.\phi(X,\vec{Y})$ can be illustrated as in Figure 2, as the collection of trees $\vec{t}$ having a *large* (comeager) section $\phi(X,\vec{t})$. Informally, $\langle t_1,\ldots,t_n\rangle \in \forall^* X.\phi(X,\vec{Y})$ if, "for almost all" $t_0 \in \mathcal{T}_{0,1}$, the tuple $\langle t_0,t_1,\ldots,t_n\rangle$ satisfies ϕ.

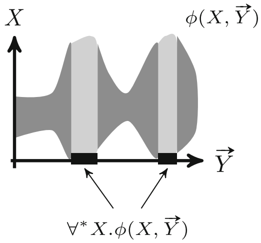

Fig. 2. Large section interpretation of Friedman's quantifier $\forall^*$. The large sections selected by quantifier $\forall^*$ are marked with lines.

Clearly, other kinds of "large section" quantifiers can be considered. Among others, the two quantifiers $\forall^{\aleph_0}$ and $\forall^{=1}$ obtained by replacing "is of first category" with "has cardinality $\leq \aleph_0$" and "has Lebesgue measure 0", are particularly natural since the σ-ideals of countable sets and of Lebesgue null sets are important set-theoretic notions of smallness. As already mentioned in the Introduction, the theory $\mathrm{MSO}+\forall^{\aleph_0}$ has been first studied in [3]. Instead, to the best of our knowledge, the theories $\mathrm{MSO}+\forall^*$ and $\mathrm{MSO}+\forall^{=1}$ have never been investigated before.

Related and Future Work. The study of the system $\mathrm{MSO}+\forall^{=1}$ appears to be an interesting topic of future research, especially in connection with investigations on probabilistic logics of programs. In this direction a relevant work is the recent paper of Carayol, Haddad and Serre [7] where the authors have developed a theory of nondeterministic tree automata with the usual acceptance condition on runs "every path must be accepting" replaced by "the set of accepting paths is of measure 1"[2]. Languages definable by such automata are called in [7] *qualitative tree languages*. Furthermore, Olivier Serre has explored in his habilitation thesis [15] and in a recent paper with Carayol [8], similar types of automata and languages obtained by replacing "measure 1" with "comeager" and "uncountable" in the acceptance condition.

The automata-based work of [7] and the approach based on extensions of MSO with large section quantifiers, followed in this paper and in [3], are to some extend similar, however the resulting theories diverge. Qualitative tree languages are not closed under complementation [7, Proposition 15], hence a comparison with a logic with negation such as $\mathrm{MSO}+\forall^{=1}$ may be problematic.

[2] Other acceptance conditions based on measure are also investigated in [7].

4 Elimination of $\forall^*$ from MSO+$\forall^*$

This section is devoted to the proof of Theorem 1. The proof is by induction on the structure of the MSO + $\forall^*$ formula $\psi(\vec{Y})$ and consists in the construction of an alternating tree automaton $\mathcal{A}_\psi$ accepting the language defined by ψ. From the automaton $\mathcal{A}_\psi$, by Theorem 4 and by Rabin's theorem [14], one can effectively construct a purely MSO formula $\hat{\psi}(\vec{Y})$. The crucial step in the proof is associated with the case of $\psi(\vec{Y})$ having the form $\psi = \exists^* X.\phi(X, \vec{Y})$. By definition, $\exists^* X.\phi(X, \vec{Y})$ defines the set of n-tuples $\vec{t}$ of trees $\{\vec{t} \mid \phi(X, \vec{t})$ is not meager$\}$. The dual case of $\psi(\vec{Y}) = \forall^* X.\phi(X, \vec{Y})$ follows by complementation of automata.

By induction hypothesis, we can assume that the sub-formula ϕ is an ordinary MSO formula. So let $\mathcal{A}$ be a nondeterministic and parity automaton, schematically representable as in Figure 1, accepting the set of $(n{+}1)$-tuples of $\{0,1\}$-trees defined by ϕ. Equivalently, the automaton $\mathcal{A}$ accepts Σ-trees with $\Sigma = \{0,1\}^{n+1}$. We identify elements of Σ with sequences of bits $\langle b_0, b_1 \dots b_n\rangle$ of length $n+1$. We now describe the well-known construction of automata $\mathcal{A}^\exists$ and $\mathcal{A}^\forall$ recognizing respectively the languages $\exists X.\phi(X, \vec{Y})$ and $\forall X.\phi(X, \vec{Y})$ over the restricted alphabet $\Sigma' = \{0,1\}^n$. See [19] for a standard exposition.

Definition 5. *Let $\mathcal{A} = (\Sigma, Q, q_0, \delta, \mathcal{F}_\pi)$ be the nondeterministic parity automaton (with parity assignment $\pi : Q \to \omega$) accepting the language over the alphabet $\Sigma = \{0,1\}^{n+1}$ defined by ϕ. The automaton $\mathcal{A}^\exists$, over the restricted alphabet $\Sigma' = \{0,1\}^n$, is defined as $\mathcal{A}^\exists = (\Sigma', Q, q_0, \delta^\exists, \mathcal{F}_\pi)$ where $\delta^\exists$ is defined as*

$$\delta^\exists(q, \vec{b}) = e_0 \vee e_1 \;\Leftrightarrow\; \Big(\delta(q, 0.\vec{b}) = e_0 \text{ and } \delta(q, 1.\vec{b}) = e_1\Big)$$

and $\vec{b} = \langle b_1, \dots, b_n\rangle$. Similarly, $\mathcal{A}^\forall$ is defined as: $\mathcal{A}^\forall(\Sigma', Q, q_0, \delta^\forall, \mathcal{F}_\pi)$ where

$$\delta^\forall(q, \vec{b}) = e_0 \wedge e_1 \;\Leftrightarrow\; \Big(\delta(q, 0.\vec{b}) = e_0 \text{ and } \delta(q, 1.\vec{b}) = e_1\Big).$$

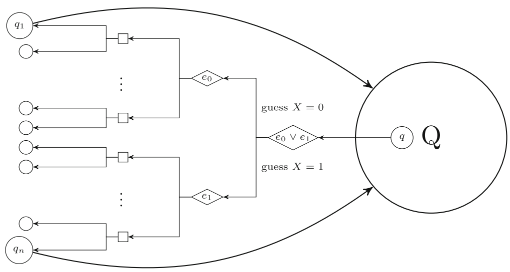

Fig. 3. Automaton $\mathcal{A}^\exists$ constructed from $\mathcal{A}$

The three automata $\mathcal{A}$ (schematically depicted in Figure 1), $\mathcal{A}^\exists$ (Figure 3) and $\mathcal{A}^\forall$ have the same set of states Q and parity condition $\mathcal{F}_\pi$ with $\pi : Q \to \omega$. Note that $\mathcal{A}$ and $\mathcal{A}^\exists$ are non-deterministic while $\mathcal{A}^\forall$ is not. The following is a standard result.

Proposition 1. *[13, 19] The automata $\mathcal{A}^\exists$ and $\mathcal{A}^\forall$ accept the languages defined by the formulas $\exists X.\phi(X, \vec{Y})$ and $\forall X.\phi(X, \vec{Y})$, respectively.*

We now introduce some convenient terminology. In the automaton $\mathcal{A}^\exists$, if the current state is q and the automata reads the letter $\vec{b}$ the transition $e_0 \vee e_1$ is reached. Here Player $\exists$ has two options:

Either choose the expression e_0, thus simulating the transition of $\mathcal{A}$ at q reading the letter $\langle 0, \vec{b}\rangle$. Then we say that "$\exists$ **chooses to label** X **with** 0",

Or choose the expression e_1, thus simulating the transition of $\mathcal{A}$ at q reading the letter $\langle 1, \vec{b} \rangle$. Then we say that "$\exists$ **chooses to label** X **with** 1".

Similarly, in the automaton $\mathcal{A}^\forall$, at the expression $e_0 \wedge e_1$, we say that "$\forall$ chooses to label X with 0" if Player $\forall$ moves to e_0 and that "$\forall$ chooses to label X with 1" if Player $\forall$ moves to e_1.

The alternating automaton $\mathcal{A}_\psi$ we are going to construct, recognizing the language of $\psi = \exists^* X.\phi(X, \vec{Y})$, is obtained by combining together the two automata $\mathcal{A}^\exists$ and $\mathcal{A}^\forall$. In what follows, to avoid confusion, we rename every state $q \in Q$ of the automaton $\mathcal{A}^\forall$ to q' so that $\mathcal{A}^\forall = (Q', q'_0, \delta^\forall, \mathcal{F}'_\pi)$, where π (and therefore $\mathcal{F}'_\pi$) and $\delta^\forall$ are defined over Q' as they were formerly defined over Q.

Before proceeding with the formal definition of $\mathcal{A}_\psi$ we describe informally the main ideas. The automaton $\mathcal{A}^\exists$ can be understood as a modified copy of $\mathcal{A}$ where player $\exists$ can choose (or "guess") labels of X, i.e., the $\{0,1\}$-labeled tree associated with the variable X in $\phi(X, \vec{Y})$. Similarly, in $\mathcal{A}^\forall$ choices related to X are made by Player $\forall$. In the automaton $\mathcal{A}_\psi$ we will implement the dynamics occurring in the Banach–Mazur game (see Definition 1) where Player I (as Player $\exists$) and Player II (as Player $\forall$) take turns in choosing how to label the tree associated with X. We will do so by defining $\mathcal{A}_\psi$ as an automaton consisting of two disjoint components $\mathcal{A}^\exists$ and $\mathcal{A}^\forall$ with special transitions allowing moving back and forth between these components. An appropriate Muller condition will be defined on $\mathcal{A}_\psi$ to enforce infinitely many alternations between components.

Definition 6. *$\mathcal{A}_\psi = (\Sigma', Q \cup Q', q_0, \delta^\psi, \mathcal{F}_\psi)$ where $q_0 \in Q$ is the initial state of $\mathcal{A}^\exists$ and, for $\vec{q} = \langle q_1, \ldots, q_n \rangle \in \Sigma'$, the transition function δ^ψ is defined as:*

$$\delta^\psi(q, \vec{b}) = e^\exists \vee e^\forall \iff \left(\delta^\exists(q, \vec{b}) = e^\exists \text{ and } \delta^\forall(q', \vec{b}) = e^\forall\right),$$

$$\delta^\psi(q', \vec{b}) = e^\exists \wedge e^\forall \iff \left(\delta^\exists(q, \vec{b}) = e^\exists \text{ and } \delta^\forall(q', \vec{b}) = e^\forall\right),$$

and $\mathcal{F}_\psi = \{S \subseteq \mathcal{P}(Q \cup Q') \mid$ the condition $A \vee (B \wedge C)$ holds$\}$, *where* $A =$ "S does not contain any $q \in Q$", $B =$ "S contains some $q' \in Q'$" *and, lastly,* $C =$ "*$proj(S) = \{q \in Q \mid q \in S \vee q' \in S\} \in \mathcal{F}_\pi$*".

See Figure 4 for a graphical exposition of Definition 6. Some explanations are in order. The automaton $\mathcal{A}_\psi$ starts at the state q_0 which belongs to $\exists$-component. This is because the Banach–Mazur game starts with a move of Player I. When reading the letter $\vec{b}$ of the (root of the) input tree, the transition is of the form $\delta^\psi(q_0, \vec{b}) = e^\exists \vee e^\forall$ meaning that player $\exists$ has two options:

guess move: choose condition $e^\exists$, i.e., the same transition as in the automaton $\mathcal{A}^\exists$. Once transition $e^\exists$ is chosen, the first move associated with the expression $e^\exists$ will correspond to the choice of $\exists$ with regard to the labeling of X ($\exists$ should choose between 0 and 1). Note that the next state to be visited will be again in the $\mathcal{A}^\exists$ component because $e^\exists \in \mathcal{DL}(\{0,1\} \times Q\}$.

skip move: choose condition $e^\forall$ as in the automaton $\mathcal{A}^\forall$. Once transition $e^\forall$ is chosen, the first move associated with the expression $e^\forall$ will correspond to the choice of $\forall$ with regard to the labeling of X ($\forall$ should choose between 0 and 1). Note that the next visited state is, this time, in the $\mathcal{A}^\forall$ component, because $e^\forall \in \mathcal{DL}(\{0,1\} \times Q'\}$.

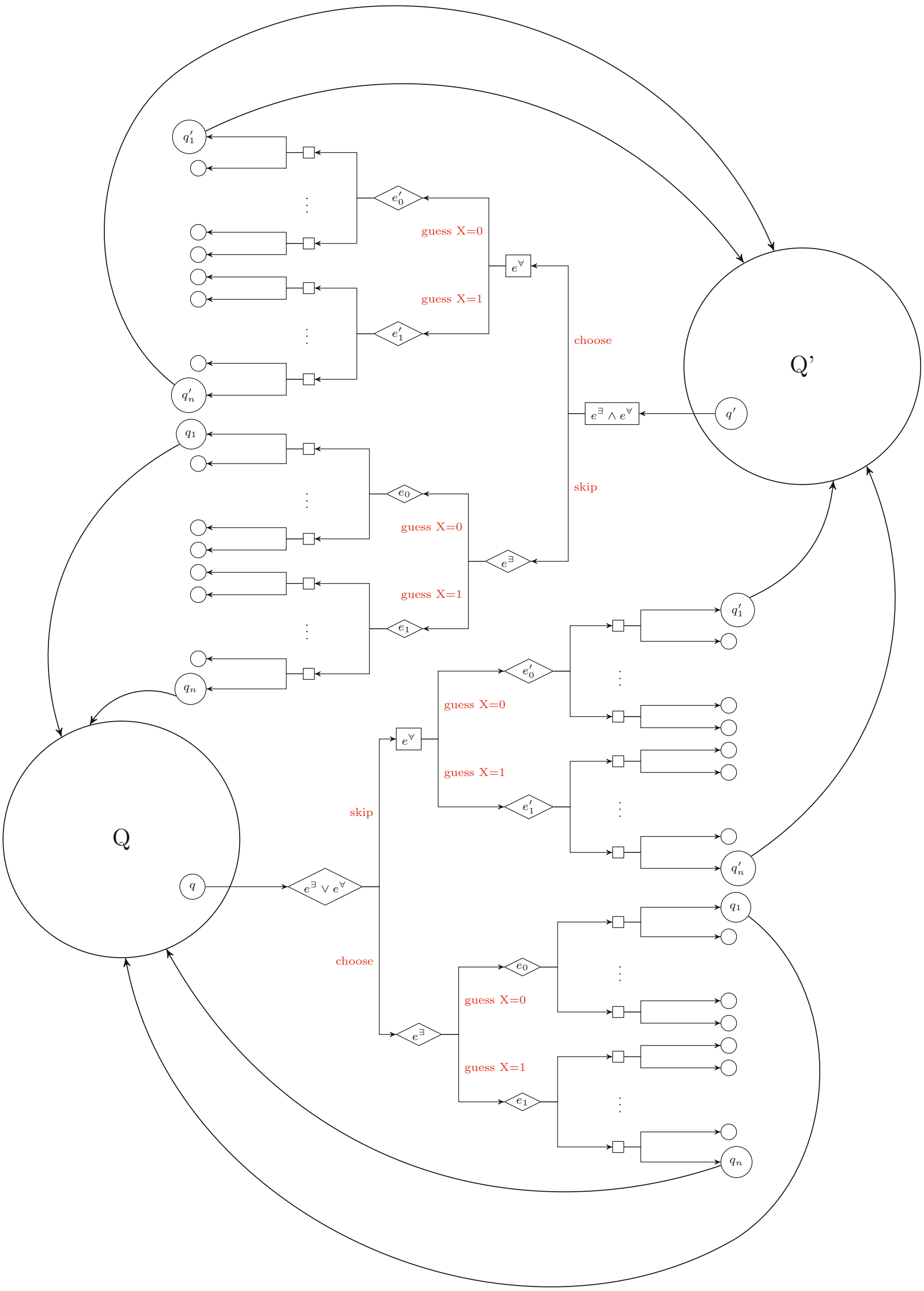

Fig. 4. Automaton $\mathcal{A}_\psi$ with states $Q \cup Q'$ and transitions depicted in this figure; $\Diamond$-shaped positions mark decisionss of player $\exists$ and $\Box$-shaped positions mark decisions of player $\forall$.

Guess and **skip** moves are marked in Figure 4. In a similar way, on states q' when reading the letter $\vec{b}$ the transition is of the form $\delta^{\psi}(q', \vec{b}) = e^{\exists} \wedge e^{\forall}$, hence Player $\forall$ can either make a **guess move** by picking $e^{\forall}$, "choose the labeling of X" and then remain in the $\forall$-component, or a **skip move** by picking $e^{\exists}$, thus allowing $\exists$ to "choose the labeling of X" and move to the $\exists$-component.

The Muller condition $\mathcal{F}_{\psi}$ captures the following aspects of the gameplay on the game tree $\mathcal{A}_{\psi}(\vec{t})$, where $\vec{t} = \langle t_1, \ldots, t_n \rangle$ with $t_i \in \mathcal{T}_{0,1}$:

A) a branch in $\mathcal{A}_{\psi}(\vec{t})$ is winning for $\exists$ if the $\exists$-component was visited only finitely many times. This intuitively means that, at some point, Player $\forall$ played "unfairly", never giving back the control to $\exists$.
B) if $\forall$ was "fair" and $\exists$ played unfairly, then $\exists$ loses and $\forall$ wins.
C) else, if both players played "fairly" alternating infinitely often between $\exists$ and $\forall$ components, then a branch in the game $\mathcal{A}_{\psi}(\vec{t})$ is winning for $\exists$ if and only if the sequence of visited states (ignoring the distinction between q and its copy q') is winning under the parity condition $\mathcal{F}_{\pi}$ (note that the parity assignment π is identical in all $\mathcal{A}$, $\mathcal{A}^{\exists}$ and $\mathcal{A}^{\forall}$).

Thus the automaton $\mathcal{A}_{\psi}$ implements the policy of infinite alternation between $\exists$ and $\forall$ in "guessing" the set X. Due to space limitations, a detailed proof of the fact that $\mathcal{A}_{\psi}$ accepts the set defined by $\psi = \exists^{*} X.\phi(X, \vec{Y})$ is not included and will appear in an extended version of this paper.

5 Applications to Probabilistic Logics

In this section we consider the *qualitative fragment* of the *probabilistic logic CTL** (pCTL*), as introduced in [10], which can express useful properties of Markov chains. We refer to the book [2, §10.4] for a detailed introduction.

Definition 7 (Markov Chain). *A Markov chain is a triple* (V, E, p)*, where* (V, E) *is a directed graph and* $p: E \rightarrow [0, 1]$ *is a function assigning probabilities to each edge in such a way that the sum of the probabilities of edges leaving every vertex* v *is* 1*. A Markov chain is* finite *if* V *is finite and* $p(e)$ *is a rational number for all* $e \in E$*. A Markov chain is* simple *if every vertex has exactly two outgoing edges both labeled with probability* $\frac{1}{2}$*.*

Definition 8 (Syntax of Qualitative pCTL*). *Given a finite set of atomic predicates* $a_1, \ldots, a_n$*, qualitative pCTL* formulas are generated by the following two-sorted grammar:* state formulas $\Phi, \Psi ::= a_i \mid \neg\Phi \mid \Psi \vee \Phi \mid \phi \mid \mathbb{P}_{=1}\phi$*, and* path formulas $\phi, \psi ::= \Phi \mid \neg\phi \mid \phi \vee \psi \mid \circ\phi \mid \phi\, \mathcal{U}\, \psi$*, where* $\circ$ *and* $\mathcal{U}$ *are the usual* Next *and* Until *operators of linear time logic.*

Definition 9 (Semantics of Qualitative pCTL*). *Given a Markov chain* $M = (V, E, p)$ *and interpretations of the atomic predicates* $||a_i||_M \subseteq V$*, state formulas* Φ *are interpreted as sets* $||\Phi||_M \subseteq V$ *of vertices and path formulas* ϕ *are interpreted as sets* $||\phi||_M$ *of paths in the graph* (V, E)*. The inductive definition is the same as that of CTL* (see [2, Definition 6.81]) with the addition of:* $v \in ||\mathbb{P}_{=1}\phi||_M \Leftrightarrow$ *"*$||\phi||_M$ *has measure* 1 *in the set of paths starting at* v*", where* *the measure on paths is defined in the standard way (see, e.g., [2, §10.4]) inferred from probabilities on the edges of M.*

Definition 10 (Finite Satisfiability). *We say that a formula Φ is* finite satisfiable *(finite–SAT) if there exists a finite Markov chain $M = (V, E, p)$ with interpretations $||a||_M$ of the atomic predicates such that $||\Phi||_M \neq \emptyset$.*

It has been observed in [5] that the finite–SAT problem can be restricted to finite–and–simple Markov chains. The observation follows from the fact, that one can transform any finite Markov chain into a finite and simple one, at the cost of introducing auxiliary "dummy states", by first simulating states with n outgoing edges by a sequence of binary choices and then simulating binary choices having arbitrary rational probabilities with a finite (cyclic) system of binary $\frac{1}{2}$-weighted choices. Furthermore, for every pCTL* formula Φ one can construct a formula $\hat{\Phi}$ (with an additional predicate for "dummy states") such that Φ is finite–SAT if and only if $\hat{\Phi}$ is satisfied by a finite and simple Markov chain. Thanks to this observation we can replace "finite" with "finite and simple" in Definition 10.

Definition 11 (Categorical Semantics of Qualitative pCTL*). *Given a Markov chain $M = (V, E, p)$, the categorical semantics of a given state formula Φ is a set $[\![\Phi]\!]_M \subseteq V$, defined as in the standard semantics $||\Phi||_M$ (Definition 9) on all connectives and on all path formulas except for $\mathbb{P}_{=1}$ which is instead defined as follows: $v \in [\![\mathbb{P}_{=1}\phi]\!]_M \Leftrightarrow$ "$||\phi||_M$ is comeager in the set of paths starting at v".*

Theorem 5. *Let $M = (V, E, p)$ be a finite and simple Markov chain and assume $||a_i||_M = [\![a_i]\!]_M$, for all atomic predicates a_i. Then $||\Phi||_M = [\![\Phi]\!]_M$ for every qualitative pCTL* formula Φ.*

Proof. The proof goes by induction on the structure of Φ with the only non-trivial case being $\Phi = \mathbb{P}_{=1}\phi$. Assume that ϕ is build from state formulas $\Psi_0, \ldots, \Psi_n$. By inductive hypothesis $||\Psi_i||_M = [\![\Psi_i]\!]_M$, for $0 \leq i \leq n$. It follows that $||\phi||_M = [\![\phi]\!]_M$. By standard arguments, $||\phi||_M$ denotes a regular set of paths in the graph (V, E). We only prove $||\Phi||_M \subseteq [\![\Phi]\!]_M$ as the case $[\![\Phi]\!]_M \subseteq ||\Phi||_M$ is similar. Assume, by contradiction, that $v \in ||\Phi||_M$ and $v \notin [\![\Phi]\!]_M$. This means that the regular set $||\phi||$ has measure 1 in the set of paths starting from v. By Staiger's theorem [16, Theorem 4] (see also [21, Theorem 9.8] for a convenient graph–theoretical formulation) this implies that $||\phi||_M$ is comeager in the set of paths starting from v. It then follows that $v \in [\![\Phi]\!]_M$ and thus we have the desired contradiction. □

Theorem 2 in the Introduction, that is the decidability of the finite–SAT problem for qualitative pCTL*, follows from the theorem below along with the decidability of MSO+$\forall^*$ (Corollary 1 of Theorem 1 in this paper).

Theorem 6. *To each qualitative pCTL* formula Φ with atomic predicates $a_1 \ldots a_n$ one can effectively associate a* MSO+$\forall^*$ *formula $F_\Phi(x, X_{a_0}, \ldots, X_{a_n})$, such that* "$\Phi$ is finite–SAT" $\Leftrightarrow$ "F_Φ is satisfiable."

Proof. As discussed above, we can restrict attention to simple Markov chains M. The full binary tree $\{L,R\}^*$ can be viewed as a simple (infinite) Markov chain by labeling each edge with $\frac{1}{2}$. Furthermore, since pCTL* is invariant under *bisimulation* [2, Theorem 10.67], each Markov chain M (which is a binary graph since the probabilistic information is implicit) can be replaced by its unraveling $\{L,R\}^*$. Each interpretation $||a_i||_M$ of the atomic predicates can then by identified with a corresponding $t_i \in \mathcal{T}_{0,1}$. Hence simple (finite) Markov chains M with interpretations $[\![a_i]\!]_M$ of the atomic predicates can be identified with (regular) Σ-trees with $\Sigma = \{0,1\}^n$. The proof goes by induction on the structure of state formulas Φ by defining formulas $F_\Phi(x, \vec{X_{a_i}})$ with the following property. An arbitrary Σ-tree $\vec{t} = \langle t_0, \ldots, t_n \rangle$ with vertex $w \in \{L,R\}^*$ satisfies F_Φ iff $w \in [\![F_\Phi]\!]_M$ with interpretation of the atomic predicates as $[\![a_i]\!] = t_i$. The construction of F_Φ follows the standard method (see, e.g., [18]) for interpreting CTL* into MSO. The only non-standard step is for $\Phi = \mathbb{P}_{=1}\phi$. The encoding in MSO+$\forall^*$ is not entirely trivial and will appear in an extended version of this paper. By Rabin's theorem [14], the formula F_Φ is satisfiable iff it satisfiable by a regular Σ-tree which can be interpreted as a finite-and-simple Markov chain M_t satisfying Φ. The desired result then follows by Theorem 5. □

References

1. Baier, C., Bertrand, N., Bouyer, P., Brihaye, T., Größer, M.: Probabilistic and topological semantics for timed automata. In: Arvind, V., Prasad, S. (eds.) FSTTCS 2007. LNCS, vol. 4855, pp. 179–191. Springer, Heidelberg (2007)
2. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)
3. Bárány, V., Kaiser, Ł., Rabinovich, A.: Cardinality quantifiers in MLO over trees. In: Grädel, E., Kahle, R. (eds.) CSL 2009. LNCS, vol. 5771, pp. 117–131. Springer, Heidelberg (2009)
4. Barua, R.: *R*-sets and category. Trans. Amer. Math. Soc. **286**(1), 125–158 (1984)
5. Bertrand, N., Fearnley, J., Schewe, S.: Bounded satisfiability for PCTL. In: Proc. of CSL (2012)
6. Brázdil, T., Forejt, V., Křetínský, J., Kučera, A.: The satisfiability problem for probabilistic CTL. In: Proc. of LICS, pp. 391–402 (2008)
7. Carayol, A., Haddad, A., Serre, O.: Randomization in automata on infinite trees. ACM Transactions on Computational Logic **1**5(3) (2014)
8. Carayol, A., Serre, O.: How good is a strategy in a game with nature? In: To appear in Proc. LICS (2015)
9. Gogacz, T., Michalewski, H., Mio, M., Skrzypczak, M.: Measure properties of game tree languages. In: Csuhaj-Varjú, E., Dietzfelbinger, M., Ésik, Z. (eds.) MFCS 2014, Part I. LNCS, vol. 8634, pp. 303–314. Springer, Heidelberg (2014)
10. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing **6**(5), 512–535 (1994)
11. Hjorth, G., Khoussainov, B., Montalban, A., Nies, A.: From automatic structures to Borel structures. In: Proc. of LICS, pp. 431–441 (2008)
12. Kechris, A.S.: Classical Descriptive Set Theory. Springer (1994)

13. Muller, D.E., Schupp, P.E.: Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin. McNaughton and Safra. Theor. Comput. Sci. **141**(1&2), 69–107 (1995)
14. Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. Transactions of American Mathematical Society **141**, 1–35 (1969)
15. Serre, O.: Playing with Trees and Logic. Habilitation Thesis, Université Paris Diderot (Paris 7) (2015)
16. Staiger, L.: Rich omega-words and monadic second-order arithmetic. In: Proc. of CSL, pp. 478–490 (1997)
17. Steinhorn, C.I.: Chapter XVI: Borel Structures and Measure and Category Logics. Perspectives in Mathematical Logic, vol. 8. Springer (1985)
18. Thomas, W.: On chain logic, path logic, and first-order logic over infinite trees. In: Proc. of LICS, pp. 245–256 (1987)
19. Thomas, W.: Languages, automata, and logic. In: Handbook of Formal Languages, pp. 389–455. Springer (1996)
20. Vaught, R.: Invariant sets in topology and logic. Fund. Math., 82, 269–294 (1974/1975), Collection of articles dedicated to Andrzej Mostowski on his sixtieth birthday, VII
21. Völzer, H., Varacca, D.: Defining fairness in reactive and concurrent systems. Journal of the ACM **59**(3) (2012)