Tractable Probabilistic μ-Calculus that Expresses Probabilistic Temporal Logics*

Pablo Castro[1,2], Cecilia Kilmurray[1,2], and Nir Piterman[3]

1 **Departamento de Computación, FCEFQyN, Universidad Nacional de Río Cuarto, Río Cuarto, Argentina, {ckilmurray,pcastro}@dc.exa.unrc.edu.ar**
2 **Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)**
3 **Department of Computer Science, University of Leicester, Leicester, UK, nir.piterman@leicester.ac.uk**

Abstract

We revisit a recently introduced probabilistic μ-calculus and study an expressive fragment of it. By using the probabilistic quantification as an atomic operation of the calculus we establish a connection between the calculus and obligation games. The calculus we consider is strong enough to encode well-known logics such as PCTL and PCTL*. Its game semantics is very similar to the game semantics of the classical μ-calculus (using parity obligation games instead of parity games). This leads to an optimal complexity of NP∩co-NP for its finite model checking procedure. Furthermore, we investigate a (relatively) well-behaved fragment of this calculus: an extension of PCTL with fixed points. An important feature of this extended version of PCTL is that its model checking is only exponential w.r.t. the alternation depth of fixed points, one of the main characteristics of Kozen's μ-calculus.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases μ-calculus, Probabilistic Logics, Model Checking, Game Semantics

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

In recent years, probabilistic model checking has received an increasing attention in the area of system verification; tools like PRISM [8] and LiQuor [4] enable the automatic verification of quantitative properties of systems (and a lot more); these kinds of properties are essential for the verification of network protocols, critical systems and randomized algorithms, to name a few examples.

Some of the most prominent probabilistic temporal logics used for model checking are PCTL, the probabilistic counterpart of CTL, and PCTL*, the probabilistic counterpart of CTL*. In particular, PCTL has a clear semantics, and its model checking procedure can be performed in polynomial time. The definition of a probabilistic μ-calculus that provides a unifying formalism for probabilistic temporal logics has been an active field of research in the area, such a formalism could provide to probabilistic model checking the same benefits as those given by Kozen's μ-calculus to qualitative model checking. The μ-calculus [12] is a powerful temporal logic that combines many useful features. It generalizes modal logic by adding fixpoint operators, it has a compact, extremely powerful, and very pleasing mathematical theory, its model checking problem is polynomial in the length of the formula and only

* This work was partially supported by the MEALS project and EPSRC project.

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

exponential in its alternation depth [7]. Most of the temporal logics used in computer science can be encoded into fragments of it; and, in addition, it has strong connections to two-player games and automata theory, which lead to optimal decision procedures for this logic.

Here, we revisit the probabilistic μ-calculus introduced by Mio and Simpson in [14, 15], however, we suggest to use probabilistic quantification as an atomic operation. The resulting probabilistic μ-calculus (named μ^p-calculus) enjoys many of the qualities of the discrete μ-calculus. We show that the logic is expressive enough to capture PCTL and PCTL*. We establish a tight connection between our logic and the recently introduced obligation parity games [3]. In particular, we provide a game semantics for μ^p-calculus using such games. When considering finite-state model checking, the games provide an optimal decision procedure in NP $\cap$ co-NP (compared with 3EXPTIME for the logic of Mio and Simpson); where optimality is w.r.t. model checking the discrete μ-calculus, which has the same complexity. In contrast to the "normal" μ-calculus, we lose the connection between the alternation depth of the formula and the complexity of model checking. We also propose a well-behaved fragment of μ^p-calculus, this logic is mainly an extension of PCTL with fixpoints, we prove that the complexity of model checking for this fragment is only exponential in the alternation depth of quantifiers; as mentioned above, this is an important characteristic of standard μ-calculus.

The paper is organized as follows. In Section 2 we introduce the basic definitions needed to tackle the rest of the paper. The probabilistic μ-calculus is introduced in Section 3 and then its expressivity is investigated. We then present the game semantics in Section 4. In Section 5 we show that a well-known "hard" problem in NP $\cap$ co-NP can be reduced to model checking formulas of μ^p-calculus with only one fixpoint operator. A well-behaved fragment of this logic is described in Section 6. Finally, we discuss some related work and add some final remarks. For the sake of clarity, long proofs are gathered in the Appendix.

2 Preliminaries

In this section we briefly introduce some basic concepts. We denote the set of real numbers between 0 and 1 as $[0,1]$. Given a set S we denote by $\vec{0}(S)$ the function $\vec{0}(S)(s) = 0$ for every $s \in S$ and by $\vec{1}(S)$ the function $\vec{1}(S)(s) = 1$ for every $s \in S$. When S is clear from the context we write $\vec{0}$ and $\vec{1}$. Given a universe U and a subset $S \subseteq U$ we write χ_S for the function $\chi_S(s) = 1$ if $s \in S$ and $\chi_S(s) = 0$ for $s \notin S$.

A *Kripke structure* over a set AP of atomic propositions is a tuple $K = \langle S, R, L, s_0 \rangle$, where S is a (countable) set of locations, $R \subseteq S \times S$ is a relation such that for every $s \in S$ we have that $R(s)$ is finite, $L : AP \to 2^S$ is a labeling function and $s_0 \in S$ is an initial location. A *Markov chain* over a set AP of atomic letters is a tuple $M = \langle K, P \rangle$, where K is a Kripke structure and $P : R \to (0,1]$ is such that for every $s \in S$ we have $\sum_{(s,s') \in R} P(s,s') = 1$. Sometimes it will be convenient to consider $P : S \times S \to [0,1]$ by associating $P(s,s') = 0$ for every $(s,s') \notin R$. For a location $s \in S$ we denote by M_s the Markov chain obtained from M by setting s to the initial location. A path $\pi = s_0, s_1, \ldots$ is a finite or infinite sequence of locations such that for every $0 \leq i < n$ we have $P(s_i, s_{i+1}) > 0$. If $\pi = s_0, \ldots, s_n$ is finite, we denote by $\mathsf{measure}_M(\pi) = \prod_{i=0}^{n-1} P(s_i, s_{i+1})$ the measure of (the set of infinite paths that extend) π. Given a (Borel) set of paths Π starting from the same state s, we denote by $\mathsf{measure}_M(\Pi)$ the measure of Π. Note that every Markov chain can be interpreted as a Kripke structure by looking on the embedded Kripke structure.

PCTL formulas over a set AP are defined as state formulas (Φ) and path formulas (Ψ)

as follows. Let $J = \{>, \geq\} \times [0,1]$ be the set of bounds.

$$\Phi ::= p_i \mid \neg p_i \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \mathcal{P}_J(\Psi) \qquad \Psi ::= \bigcirc\Phi \mid \Phi\,\mathcal{U}\,\Phi \mid \Phi\,\mathcal{W}\,\Phi$$

Here $\mathcal{W}$ is the weak until (i.e., it allows the first operand to hold forever). As usual we introduce the abbreviations F and G. State formulas are *formulas*. The semantics of PCTL associates with every formula a set of states. We denote by $[\![\varphi]\!]_M$ the set of states of M that satisfy φ. For every path formula φ and state s of M, $\mathsf{measure}_M(s, \varphi)$ is the measure of paths starting in s that satisfy φ. The semantics and intuitions of PCTL formulas are as usual, see [1].

We define μ-calculus over Kripke structures with the following syntax.

$$\Phi ::= p_i \mid \neg p_i \mid X_i \mid \Diamond\Phi \mid \Box\Phi \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \mu X_i.\Phi \mid \nu X_i.\Phi$$

Where $p_i \in AP$, $\mathcal{V} = \{X_0, X_1, \dots\}$ is an enumerable set of variables, and $X_i \in \mathcal{V}$. The notions of *open* and *closed* formulas are as usual. The semantics of a μ-calculus formula over a Kripke structure $K = \langle S, R, L, s_0\rangle$ is given w.r.t. assignments to variables in $\mathcal{V}$. An assignment $\rho : \mathcal{V} \to (S \to \{0,1\})$ associates a function from the states to $\{0,1\}$ with every variable in $\mathcal{V}$. Given an assignment ρ we set $\rho[f/X]$ to be the assignment that associates the function f with X and $\rho(Y)$ with every $Y \neq X$. We use the notation of a function into $\{0,1\}$ instead of set notation to facilitate the discussion in the rest of the paper. The semantics of a formula φ in structure K with respect to assignment ρ, denoted $[\![\varphi]\!]_K^\rho$, is defined as follows.

$$\begin{array}{rlrl}
[\![p_i]\!]_K^\rho &= \chi_{L(p_i)} & [\![\neg p_i]\!]_K^\rho &= 1 - \chi_{L(p_i)} \\
[\![X]\!]_K^\rho &= \rho(X) & & \\
[\![\varphi_1 \vee \varphi_2]\!]_K^\rho &= \max([\![\varphi_1]\!]_K^\rho, [\![\varphi_2]\!]_K^\rho) & [\![\varphi_1 \wedge \varphi_2]\!]_K^\rho &= \min([\![\varphi_1]\!]_K^\rho, [\![\varphi_2]\!]_K^\rho) \\
[\![\Diamond\varphi]\!]_K^\rho &= \lambda s.\max_{(s,s')\in R}[\![\varphi]\!]_K^\rho(s') & [\![\Box\varphi]\!]_K^\rho &= \lambda s.\min_{(s,s')\in R}[\![\varphi]\!]_K^\rho(s') \\
[\![\mu X.\varphi]\!]_M^\rho &= \mathrm{lfp}([\![\varphi]\!]_M^{\rho[f/X]}) & [\![\nu X.\varphi]\!]_M^\rho &= \mathrm{gfp}([\![\varphi]\!]_M^{\rho[f/X]})
\end{array}$$

Note that the semantics of a formula where all variables are bound by fixpoint operators is independent of the assignment ρ. The interested reader is referred to [17] for an in-depth introduction to μ-calculus.

3 A Probabilistic μ-calculus

In this section we present our version of probabilistic μ-calculus (denoted μ^p-calculus). Unlike the "normal" μ-calculus, μ^p-calculus is two sorted. We distinguish between qualitative formulas (that get values in $\{0,1\}$) and quantitative formulas (that get values in $[0,1]$).[1] Although the logic is a subset of the probabilistic μ-calculus of Mio and Simpson [15] we give a direct definition of its semantics without relying on their results.

Given an enumerable set of variables $\mathcal{V} = \{X_0, X_1, \dots\}$, the syntax of the logic is given by the following grammar, where Ψ are qualitative formulas, and Φ are quantitative formulas.

$$\begin{aligned}
J &::= \{>, \geq\} \times [0,1] \\
\Psi &::= p_i \mid \neg p_i \mid \Psi_1 \vee \Psi_2 \mid \Psi_1 \wedge \Psi_2 \mid [\Phi]_J \mid \nu X_i.\Psi \mid \mu X_i.\Psi \\
\Phi &::= \Psi \mid X_i \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \Diamond\Phi \mid \Box\Phi \mid \bigcirc\Phi \mid \nu X_i.\Phi \mid \mu X_i.\Phi
\end{aligned} \tag{1}$$

[1] This is not to be confused with qualitative PCTL, where the bounds are restricted to ≥ 1 and > 0.

We say that variable X_i is *bound* in $\sigma X_i.\varphi(X_i)$ for $\sigma \in \{\mu,\nu\}$. A variable that is not bound is *free*. A *formula* is a *qualitative formula* with no free variables. That is, at the top level we consider only formulas that can be evaluated to $\{0,1\}$. Note that we add to the existential and universal next operators of μ-calculus the (probabilistic) next operator and the probabilistic quantification operator.

The semantics of a formula ψ over a Markov chain M is defined with respect to an interpretation ρ, which associates a function from states to real values in $[0,1]$ with each variable appearing in ψ. Formally, for $\rho : \mathcal{V} \to (S \to [0,1])$ the semantics $[\![\psi]\!]^\rho_M : S \to [0,1]$ is defined as follows:

$$
\begin{array}{rlrl}
[\![p_i]\!]^\rho_M &= \chi_{L(p_i)} & [\![\neg p_i]\!]^\rho_M &= 1 - \chi_{L(p_i)} \\
[\![X]\!]^\rho_M &= \rho(X) & & \\
[\![\varphi_1 \vee \varphi_2]\!]^\rho_M &= \max([\![\varphi_1]\!]^\rho_M, [\![\varphi_2]\!]^\rho_M) & [\![\varphi_1 \wedge \varphi_2]\!]^\rho_M &= \min([\![\varphi_1]\!]^\rho_M, [\![\varphi_2]\!]^\rho_M) \\
[\![\bigcirc\varphi]\!]^\rho_M &= \lambda s.\sum_{s'} P(s,s')[\![\varphi]\!]^\rho_M(s') & [\![[\varphi]_J]\!]^\rho_M &= ([\![\varphi]\!]^\rho_M(s) J \,?\, 1 : 0) \\
[\![\Diamond\varphi]\!]^\rho_M &= \lambda s.\max_{(s,s')\in R}[\![\varphi]\!]^\rho_M(s') & [\![\Box\varphi]\!]^\rho_M &= \lambda s.\min_{(s,s')\in R}[\![\varphi]\!]^\rho_M(s') \\
[\![\mu X.\varphi]\!]^\rho_M &= \mathrm{lfp}([\![\varphi]\!]^{\rho[f/X]}_M) & [\![\nu X.\varphi]\!]^\rho_M &= \mathrm{gfp}([\![\varphi]\!]^{\rho[f/X]}_M)
\end{array}
$$

That is, the value of the probabilistic next is the average value over successors and the probabilistic quantification compares the value with the given bound. Even though the semantics is quite similar to the semantics of μ-calculus the former is restricted to functions of the type $f : S \to \{0,1\}$ and here the functions are $f : S \to [0,1]$. That is, functions associate real values with states.

It is simple to see that all these transformers are monotonic. In particular, if $\rho_1 \leq \rho_2$, that is for every $X \in \mathcal{V}$ and every $s \in S$ we have $\rho_1(X)(s) \leq \rho_2(X)(s)$, then $[\![\varphi]\!]^{\rho_1}_M \leq [\![\varphi]\!]^{\rho_2}_M$. For instance, consider a formula of the form $[\varphi]_J$. We have to show that, if $[\![[\varphi]_J]\!]^{\rho_1}_M(s) = 1$, then $[\![[\varphi]_J]\!]^{\rho_2}_M = 1$. However, if $\rho_1 \leq \rho_2$ it follows that $[\![\varphi]\!]^{\rho_1}_M \leq [\![\varphi]\!]^{\rho_2}_M$. So, if $[\![\varphi]\!]^{\rho_1}_M J$, then also $[\![\varphi]\!]^{\rho_2}_M J$. It follows from the Knaster-Tarski theorem that fixed-points are well defined.

It is possible to show that our calculus is closed under negation. For this, we need to consider the usual dualizations between the standard operators. In addition, the probabilistic next is its own dual and the probabilistic quantification has to be replaced with the dual probabilistic quantification. That is, $[\cdot]_{>1-p}$ is the dual of $[\cdot]_{\geq p}$ and $[\cdot]_{\geq 1-p}$ is the dual of $[\cdot]_{>p}$. We now show that the definition of qualitative formulas is indeed justified.

▶ **Lemma 1.** *For every qualitative formula φ we have $[\![\varphi]\!]^\rho_M \in \{0,1\}$.*

Proof. We can show that the semantics of all operators in the qualitative fragment are functions whose range is $\{0,1\}$. This holds trivially for propositions. Given two functions whose range is $\{0,1\}$ clearly, min and max return such functions as well.

For $[\varphi]_J$ this follows directly from the definition. ◀

3.1 Expressing the μ-calculus

We show that μ^p-calculus is strong enough to express the μ-calculus over the embedded Kripke structure without using the existential and universal next operators. We include this construction mostly as justification for the hardness of model checking the μ^p-calculus over finite-state Markov chains.

Given a μ-calculus formula φ, let $p(\varphi)$ denote the formula obtained from φ by the following recursive transformation.

$$
\begin{array}{rlrlrl}
p(p_i) &= p_i & p(\psi_1 \vee \psi_2) &= p(\psi_1) \vee p(\psi_2) & p(\Diamond\psi) &= [\bigcirc p(\psi)]_{>0} \\
p(\neg p_i) &= \neg p_i & p(\psi_1 \wedge \psi_2) &= p(\psi_1) \wedge p(\psi_2) & p(\Box\psi) &= [\bigcirc p(\psi)]_{\geq 1} \\
p(X) &= X & p(\mu X.\psi) &= \mu X.p(\psi) & p(\nu X.\psi) &= \nu X.p(\psi)
\end{array}
$$

That is, we replace the existential next operator by a probabilistic quantification of more than 0, and the universal next operator by a probabilistic quantification of at least 1.

▶ **Lemma 2.** *For every Markov chain $M = \langle K, P\rangle$ we have $[\![p(\varphi)]\!]^{\rho}_{M} = [\![\varphi]\!]^{\rho}_{K}$.*

We notice that, in general, it is not clear how to express the universal and existential next operators without including them explicitly. This is because the $[\cdot]_J$ operator also resets the value to 0 or 1. An additional comment regarding these operators is included in Section 5. It follows that μ^p-calculus is strong enough to express all standard temporal logics such as CTL, LTL, and CTL*.

3.2 Expressing PCTL

We show that μ^p-calculus can express PCTL. Given a PCTL formula φ, let $t(\varphi)$ denote the formula obtained from φ by the following recursive transformation.

$$\begin{array}{rlrlrl}
t(p_i) &= p_i & t(\psi_1 \vee \psi_2) &= t(\psi_1) \vee t(\psi_2) & t(\mathcal{P}_J(\psi)) &= [t(\psi)]_J \\
t(\neg p_i) &= \neg p_i & t(\psi_1 \wedge \psi_2) &= t(\psi_2) \wedge t(\psi_2) & t(\bigcirc\psi) &= \bigcirc t(\psi)
\end{array}$$

$$t(\psi_1 \,\mathcal{U}\, \psi_2) = \mu X \,.\, t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X) \qquad t(\psi_1 \,\mathcal{W}\, \psi_2) = \nu X \,.\, t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)$$

That is, we use fixpoint operators to unwind until and weak until operators in the standard way this is done with CTL and μ-calculus. We note that this construction is essentially identical to the encoding of CTL in μ-calculus, which is used also in [15] (though the main complexity in their construction is in expressing the probabilistic quantification, which is part of the syntax in our setting). Due to its importance we include it in full here.

▶ **Lemma 3.** *For every Markov chain M and PCTL formula φ we have $[\![\varphi]\!]_M = [\![t(\varphi)]\!]^{\rho}_{M}$.*

The conversion of PCTL* to μ^p-calculus is also possible. As for PCTL, it is essentially identical to the translation of CTL* to μ-calculus, with the caveat that we have to replace nondeterministic automata by deterministic automata. The usage of deterministic automata is, similarly, required for the handling of LTL for probabilistic model checking [2]. We include the full construction in Appendix A. We note that this implies that PCTL* is expressible (through the same construction with the additional encoding of the probabilistic thresholds) also in the probabilistic μ-calculus of Mio and Simpson.

4 Game Semantics

First, we describe the intuition behind the game semantics, and only then formally define the games. Given a formula φ and a Markov chain $M = \langle K, P\rangle$, where $K = \langle S, R, L, s^{in}\rangle$, we define a game whose configurations correspond to locations of M and subformulas of φ. The semantics is defined in terms of a two-player stochastic obligation parity game [3]. Such games include configurations of players 0 and 1 as well as probabilistic configurations. The winning condition is a combination of a parity condition and obligations (for how much player 0 has to win) on some configurations. Player 0 is the *verifier*, who tries to prove that the formula holds, and player 1 is the *refuter*, who tries to prove that the formula does not hold. Each configuration has a valuation for each player. In general, the value of a configuration, denoted by $val_i(s, \varphi)$ for $i \in \{0, 1\}$, is a value in $[0, 1]$; $val_i(s, \varphi) = 1$ means that player i wins (completely) from a configuration. For every qualitative (sub)formula the value of (s, φ) is either 0 or 1. Intuitively, if $val_0(s, \varphi) = 1$, then the formula is true in M. For propositions, (s, p), player 0 wins when $s \in L(p)$ and she loses otherwise (configurations with $\neg p$ are dual). Configurations $(s, \varphi_1 \vee \varphi_2)$ are verifier configurations, and she chooses a successor (s, φ_i). Configurations $(s, \varphi_1 \wedge \varphi_2)$ are refuter configurations, and she selects a successor (s, φ_i).

For a fixpoint $\sigma \in \{\mu, \nu\}$, from configuration $(s, \sigma X.\varphi)$ the game progresses to (s, φ); while from configurations (s, X) the game progresses to $(s, \sigma X.\varphi)$ where $\sigma X.\varphi$ is the subformula binding X. Interesting cases are the probabilistic operators: from configuration $(s, [\varphi]_J)$ the game progresses with no choice to (s, φ). However, the former configurations have the obligation J associated with them. That is, from these obligation states player 0 wins completely (value 1) if she wins with a value satisfying J from the successor configuration. These three types of configurations (fixpoint related and probabilistic quantification) are deterministic configurations. We associate them with the probabilistic player and assign the probability 1 to the single successor. The next operators are treated as follows. A configuration of the form $(s, \Diamond\varphi)$ is a verifier configuration from where she chooses a successor s' of s and moves to configuration (s', φ). A configuration of the form $(s, \Box\varphi)$ is a refuter configuration from where she chooses a successor s' of s and moves to configuration (s', φ). A configuration of the form $(s, \bigcirc\varphi)$ is a probabilistic configuration with successors (s', φ) for every successors s' of s. Furthermore, the probability of $((s, \bigcirc\varphi), (s', \varphi))$ is $\kappa(s, s')$. It follows that the only (meaningful) probabilistic configurations are those corresponding to the probabilistic next of the calculus. The parity condition in the game arises from the alternation depth of formulas.

4.1 Parity Obligation Games

We give a short introduction to obligation parity games. The notion of winning (and value) in an obligation game is quite involved and we refer the reader to [3] for an in-depth introduction. Parts of the definition are included in the Appendix as part of the proof of correctness of the game semantics.

A parity obligation game is $G = (V, (V_0, V_1, V_p), E, \kappa, \mathcal{G})$, where V is a set of configurations, V_0, V_1, and V_p form a partition of V to player 0, player 1, and stochastic configurations, respectively, $E \subseteq V \times V$ is the set of edges, κ associates a probabilistic distribution with the edges leaving every configuration in V_p, i.e., for every $v \in V_p$ we have $\Sigma_{(v,v')\in E}\kappa(v, v') = 1$ and for every $(v, v') \notin E$ we have $\kappa(v, v') = 0$, and $\mathcal{G} = (c, O)$ is the winning condition, where $c : V \to [0..m]$ is a parity priority function, with m as its *index*, and $O : V \to \{\bot\} \cup (\{>, \geq\} \times [0, 1])$ is the obligation function. A configuration v such that $O(v) \neq \bot$ is called an *obligation configuration.*

▶ **Theorem 4.** [3] *For every configuration $v \in V$ there is a value $val_i(v) \in [0, 1]$ such that $val_0(v) + val_1(v) = 1$. Furthermore, for every obligation configuration v we have $val_i(v) \in \{0, 1\}$. For a configuration v of a finite parity obligation game, one can decide whether $val_i(v) \geq r$ in NP ∩ co-NP and $val_i(v)$ can be computed in exponential time.*

4.2 Model Checking Game

We are now ready to formally define the model checking games. Let $sub(\varphi)$ denote the subformulas of φ according to the grammar in (1). We use the notion of *alternation depth* as defined, e.g., in [7]. Roughly speaking, the alternation depth of a formula is a measure of its complexity. Essentially, it is the largest number of μ and ν alternations that appear in the formula. A formal definition is included in Appendix. Furthermore, let d be $ad(\varphi)$, with every subformula φ' of φ we can associate a *color* $c(\varphi')$ as follows. If $\varphi' = \nu X.\psi$ then

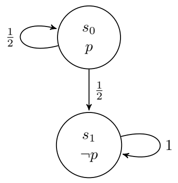

Figure 1 Markov chain M.

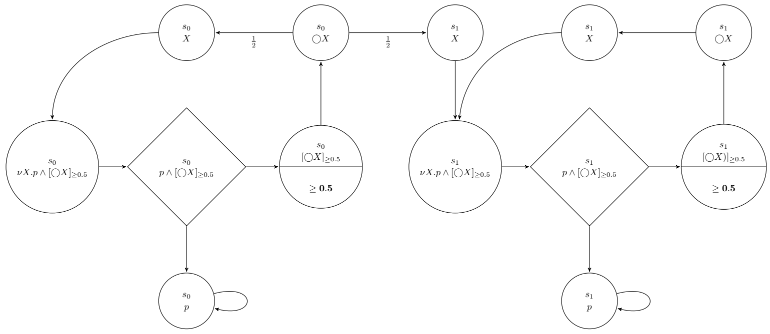

Figure 2 The game $G_{M,\varphi}$.

$c(\varphi') = 2(d - ad(\varphi'))$. If $\varphi' = \mu X.\psi$ then $c(\varphi') = 2(d - ad(\varphi')) + 1$. For every other formula φ' we set $c(\varphi') = 2d - 1$. It follows that $c(\varphi')$ is in the range $[0..2d - 1]$.

▶ **Definition 5.** Consider a Markov chain $M = \langle K, P\rangle$, where $K = \langle S, R, L, s^{in}\rangle$ and a formula φ. We define the game $G_{M,\varphi} = (V, E, (V_0, V_1, V_p), \kappa, \mathcal{G})$ as follows:

- $V = \{(s, \psi) \mid s \in S \wedge \psi \in sub(\varphi)\}$,
- $V_0 = \{(s, \psi_1 \vee \psi_2), (s, \Diamond\psi)\}$, $V_1 = \{(s, \psi_1 \wedge \psi_2), (s, \Box\psi)\}$, and $V_p = V \setminus (V_0 \cup V_1)$,
- $E = \{((s, p), (s, p)), ((s, \neg p), (s, \neg p)) \mid p \text{ is a proposition }\} \cup \{((s, [\psi]_J), (s, \psi))\}$
  $\cup \{((s, \psi_1 \vee \psi_2), (s, \psi_i)) \mid i \in \{1, 2\}\} \quad \cup \quad \{((s, \psi_1 \wedge \psi_2), (s, \psi_i)) \mid i \in \{1, 2\}\}$
  $\cup \{((s, \Diamond\psi), (s', \psi)) \mid P(s, s') > 0\} \quad \cup \quad \{((s, \Box\psi), (s', \psi)) \mid P(s, s') > 0\}$
  $\cup \{((s, \bigcirc\psi), (s', \psi)) \mid P(s, s') > 0\} \quad \cup \quad \{((s, \sigma X.\psi), (s, \psi)) \mid \sigma \in \{\nu, \mu\}\}$
  $\cup \{((s, X), (s, \sigma X.\psi)) \mid \sigma X.\psi$ is the subformula binding X and $\sigma \in \{\mu, \nu\}\}$
- $\kappa((s, \bigcirc\psi)(s', \psi)) = P(s, s')$, and $\kappa((s, \psi)(s, \psi')) = 1$ for every other $(s, \psi) \in V_p$ and $((s, \psi), (s, \psi')) \in E$.
- $\mathcal{G} = (c, O)$, where $O(s, [\psi]_J) = J$ and $O(s, \psi) = \perp$ for every other formula;

$$c(s, \psi) \quad = \quad \begin{cases} c(\psi) & \text{If } \psi \text{ is not a proposition.} \\ 0 & \text{If } (\psi = p \text{ and } s \in L(p)) \text{ or } (\psi = \neg p \text{ and } s \notin L(p)) \\ 1 & \text{If } (\psi = p \text{ and } s \notin L(p)) \text{ or } (\psi = \neg p \text{ and } s \in L(p)) \end{cases}$$

Let us present a simple example to obtain a first taste of μ^p-calculus and its game semantics. Consider Markov chain M in Fig. 1 and the formula $\varphi : \nu X.p \wedge [\bigcirc X]_{\geq 0.5}$. The alternation depth of φ is 1. It follows that $c(s_0, p) = 0$, $c(s_1, p) = 1$, and for every other configuration $c(v) = 0$. The game obtained from φ and M is shown in Fig. 2. In this graphic, we use circles to denote probabilistic configurations and diamonds to denote player 1 configurations. Note that there are no player 0 configurations in this game. The only configurations with obligations are $(s_0, [\bigcirc X]_{\geq 0.5})$ and $(s_1, [\bigcirc X]_{\geq 0.5})$. Let us calculate the value of $(s_0, \nu X.p \wedge [\bigcirc X]_{\geq 0.5})$, the unique successor of this configuration is a configuration where the refuter plays. The configuration (s_0, p) is colored 0 as $p \in L(s_0)$. Thus, the refuter should avoid this sink state as it is winning for verifier and select the other successor. This is a probabilistic configuration with obligation $\geq \frac{1}{2}$. Then note that player 0 can ensure that with probability at least $\frac{1}{2}$ she either wins by reading (s_0, p) or gets to the same obligation configuration, with color 0 the minimal in the loop. Player 0 can repeat this pattern forever. It follows that player 0 meets her obligation and that the value of $(s_0, [\bigcirc X]_{\geq 0.5})$ is 1. We conclude that $val_0(s_0, \nu X.p \wedge [\bigcirc X]_{\geq 0.5}) = 1$. Thus, the formula holds over this structure. Intuitively, there is a location where p holds and for at least $\frac{1}{2}$ of its successors the same property holds again.

The following theorem shows that these games capture the semantics of μ^p-calculus.

▶ **Theorem 6.** *For every Markov chain M, every location s, and every formula φ we have $[\![\varphi]\!]^{\rho}_{M}(s) = val_0(s,\varphi)$, where $val_0(s,\varphi)$ is the value of configuration (s,φ) in game $G_{M,\varphi}$.*

▶ **Corollary 7.** *Given a finite Markov chain M and a formula φ we can decide whether $[\![\varphi]\!]^{\rho}_{M} = 1$ in NP $\cap$ co-NP.*

Proof. From Theorem 4 we can determine whether the value of configuration (s,φ) in $G_{M,\varphi}$ is at least one in NP $\cap$ co-NP. The size of $G_{M,\varphi}$ is polynomial in the size of M and in the size of φ. The result follows. ◀

We note that the game captures also the semantics of quantitative subformulas. It follows that for a quantitative subformula ψ we can decide whether $[\![\psi]\!]^{\rho}_{M}(s) > p$ in NP $\cap$ co-NP and compute it in exponential time.

5 Hardness of Model Checking

As we have shown in Section 3, there is a simple translation from the μ-calculus to our logic. The exact complexity of model checking the μ-calculus is a long standing open problem. It is well-known that its complexity lies in UP $\cap$ co-UP [11] and is equivalent to the complexity of solving parity games [7]. However, the complexity arises from the alternation of fixpoint operators. Here, we show that in our logic already the fraction that uses only the least fixpoint (and only one fixpoint) is as hard as some of the "hard" problems known to be in NP $\cap$ co-NP but not known to be in P.

5.1 Two-player Stochastic Reachability Games

A two-player stochastic reachability game is $G = (V, (V_0, V_1, V_p), E, \kappa, T)$, where V, V_0, V_1, V_p, E, and κ are just like in parity obligation games and $T \subseteq V$ is a set of target configurations. A strategy for player 0 is $\sigma : V_0 \to V$ such that for every $v \in V_0$ we have $(v, \sigma(v)) \in E$. A strategy for player 1 is defined similarly. We intentionally consider only deterministic memoryless strategies[2]. Given strategies σ and π for players 0 and 1, respectively, the Markov chain $G_{\sigma,\pi}$ is the result of fixing the choices of the players according to their strategies. For a configuration $v \notin T$, let $\Pi_v = \{v\} \cdot V^* \cdot T \cdot V^{\omega}$ be the set of paths that start in v and visit T. Then, the value of a configuration $v \in V \setminus T$ for player 0 is $val_0(v) = \sup_{\sigma} \inf_{\pi} \mathsf{measure}_{G_{\sigma,\pi}}(\Pi_v)$.

▶ **Theorem 8.** [6, 11, 19] *For every configuration $v \in V \setminus T$ deciding if $val_0(v) > p$ for some $p \in [0,1]$ is in* NP $\cap$ co-NP*. The decision problem of whether a configuration in a 2-player parity/mean-payoff/discounted is winning for player 0 can be reduced to deciding $val_0(v) > p$.*

5.2 Encoding Games as Model Checking

Consider a two-player stochastic reachability game $G = (V, (V_0, V_1, V_p), E, \kappa, T)$, a configuration $v \in V \setminus T$ and a value $p \in [0,1]$. We show how to construct a Markov chain M_G and a formula φ_R such that $[\![\varphi_R]\!]^{\rho}_{M_G}(s_0) = 1$ iff $val_0(v) > p$, where s_0 is the initial state of M_G. Let $M_G = \langle K, P\rangle$ be a Markov chain, where $K = \langle V, E, L, v\rangle$, and $P(v, v')$ is $\kappa(v, v')$ if

[2] It is well known that in two-player stochastic reachability games there are optimal deterministic memoryless strategies for both players [6].

$v \in V_p$ and $P(v, v') = \frac{1}{|E(v)|}$ otherwise. The labeling L uses four propositions: p_0, p_1, and p_p marking configurations of player 0, player 1, and stochastic, and p_g marking configurations in T as the goal.

Let $\psi_{_R} = p_g \vee ((p_p \rightarrow \bigcirc X) \wedge (p_0 \rightarrow \Diamond X) \wedge (p_1 \rightarrow \Box X))$. Then $\varphi_{_R} = [\mu X.\psi_{_R}]_{>p}$.

▶ **Lemma 9.** $[\![\varphi_R]\!]^{\rho}_{M_G}(v) = 1$ *iff* $val_0(v) > p$.

▶ **Corollary 10.** *Model checking alternation free μ^p-calculus formulas is as hard as solving parity/mean-payoff/discounted games.*

We note that this result relies on the usage of the existential and universal next operators. Indeed, the proof relies on our ability to "keep" the value of existential and universal configurations in the original game in the formula. We do not know whether it is possible to prove a similar result for a calculus without the existential and universal next operators. We suspect that these next operators increase the expressive power of the logic. We also do not know if by removing these two operators the "normal" complexity hierarchy of the μ-calculus that relies on alternation depth is introduced. We note that parity obligation games can clearly encode the reachability of stochastic games. Thus, showing that the μ^p-calculus without existential and universal next operators enjoys the same hierarchy would require other techniques for model checking this calculus. A hardness result that does not use the existential and universal next operators is by encoding the μ-calculus in μ^p-calculus, as we do in Subsection 3.1. This hardness result does rely on fixpoint alternation.

We note that a similar encoding can represent the value of an obligation game (with finitely many different obligation values) as the result of model checking a μ^p-calculus formula over a Markov chain. As before, the structure of the game is encoded into the Markov chain. The encoding is more involved as we have to include propositions that will identify the exact obligations of configurations. Using these additional propositions the correct probabilistic quantification can be included in the formula. The structure of the formula is very similar to the classical encoding of the solution of parity games as μ-calculus model checking. That is, a prefix with fixpoints binding the variables according to the parity condition followed by a body that includes the association of configurations with player 0, player 1, or probabilistic (as above) with the inclusion of probabilistic quantification as well. We leave further details of this construction as future work.

6 μ-PCTL

We now introduce a fragment of μ^p-calculus that is expressive enough for encoding PCTL and whose model checking is exponential only w.r.t. alternations of quantifiers. Thus, for formulas with a bounded number of fixpoint alternations the model checking of this fragment is polynomial. We believe that this logic may serve as a basis for defining other useful extensions of PCTL.

Let AP be a set $\{p_0, p_1, \dots\}$ of atomic propositions and let $\mathcal{V} = \{X_0, X_1, X_2, \dots\}$ be an enumerable set of variables; the sets Φ and Ψ of location and path formulas, respectively, are mutually recursively defined as follows:

$$J ::= \{>, \geq\} \times [0, 1]$$
$$\Phi ::= p_i \mid \neg p_i \mid X_i \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid [\Psi]_J \mid \nu X_i.\Phi \mid \mu X_i.\Phi$$
$$\Psi ::= \mathsf{X}\Phi \mid \Phi\,\mathcal{U}\,\Phi \mid \Phi\,\mathcal{W}\,\Phi$$

We assume that in every formula there is no repetition of bound variables; it is straightforward to see that every formula can be rewritten to satisfy this requirement. In general, we are interested in formulas in which all variables are bound. The definition of alternation depth is as before.

The semantics of this logic can be straightforwardly obtained from the semantics for μ^p-calculus given in Section 3, taking into account the fixpoint semantics of path operators; and similarly for its game semantics. That is, we replace $\mathsf{X}\psi$ by $\bigcirc\psi$, $\psi_1 \,\mathcal{U}\, \psi_2$ by $\mu X.\psi_2 \vee (\psi_1 \wedge \bigcirc X)$, and $\psi_1 \,\mathcal{W}\, \psi_2$ by $\nu X.\psi_2 \vee (\psi_1 \wedge \bigcirc X)$.

Before presenting the model-checking algorithm we introduce some further notations. We use a collection of (global) set variables $S_i \in 2^S$, where each variable S_i represents the valuation of a variable X_i appearing in the formula. Let $c_0, c_1, \ldots$ be a set of fresh propositions, and we denote by $M[c_i \leftarrow S_i]$ the structure over $AP \cup \{c_1, \ldots, c_n\}$ obtained from M by setting $L(c_i) = S_i$. For the formula φ, let $\varphi[X_j \leftarrow c_j]$ be the formula obtained from φ, by replacing every reference to X_j by c_j.

We are now ready to present the model-checking algorithm for μ-PCTL. Our procedure, called *eval*, is presented as Algorithm 1. The procedure takes a Markov chain $M = \langle S, R, L, s_0 \rangle$ and a μ-PCTL formula φ and returns the set of states satisfying φ. We assume that variables S_i, where X_i is bound by a least fixpoint, are initialized to the empty set; and variables S_i, where X_i is bound by a greatest fixpoint, are initialized to the set of all states S. This algorithm uses the well-known way of calculating fixed points by using the Knaster-Tarski theorem and it assumes a polynomial model checking for PCTL (denoted $evalPCTL$).

The algorithm is similar to that proposed in [7] to model check standard μ-calculus, fixed points are calculated in the standard way, new constants are used for reducing subformulas to PCTL formulas, and we only reset the values of variables when the nesting of two different fixed points are found, otherwise previous calculation of fixed points are employed; to do so, we use some auxiliary functions: $Parent(\varphi_i)$ returns the fixpoint σX_j surrounding φ_i such that X_j appears free in that formula, and $OpenSub(\varphi_i)$ returns the set subformulas of φ_i that are bound by the same fixpoint operators and in which X_i is free.

▶ **Theorem 11.** *For a formula φ, $s \in \mathrm{eval}(M, \varphi)$ iff $[\![\varphi]\!]^{\rho}_{M}(s) = 1$.*

We note that this procedure is exponential only w.r.t. alternation depth. Thus, if the alternation depth is fixed the procedure is polynomial.[3]

▶ **Theorem 12.** *Procedure* eval *runs in time $O((|M|^k \cdot |\phi|^{\frac{3}{2}})^{ad(\phi)+1})$, where the constant k depends on the model checker used for* PCTL *formulas.*

Furthermore, we prove that this fragment of μ^p-calculus is strictly more expressive than PCTL.

▶ **Theorem 13.** *μ-PCTL is strictly more expressive than PCTL.*

Proof. Consider the formula $\nu X.p \wedge [\bigcirc X]_{>0}$, one can see that it is equivalent to the CTL formula $\mathsf{EG}p$. Theorem 14.45 in [1] shows that there is no qualitative PCTL formula that is equivalent to it. It is possible to extend their proof to cover also quantitative probabilistic quantification of PCTL. Thus, formula $\nu X.p \wedge [\bigcirc X]_{>0}$ cannot be expressed in PCTL. ◀

[3] We also note that if a similar approach would be applied to finite obligation parity games the result would be an exponential number of calls to an NP ∩ co-NP algorithm. Indeed, the search for the sets of obligations that can be used to satisfy other obligations can follow the same search pattern by using maximal and minimal fixpoints. However, checking that each obligation is met, which corresponds to the PCTL model checking in *eval*, would be a solution of a finite turn-based stochastic parity-reachability game, which is in NP ∩ co-NP.

Input: A Markov Chain M and a formula ϕ
Output: Set of states satisfying ϕ

switch *the form of* φ **do**
  case ϕ *is a* PCTL *formula* **return** $evalPCTL(M,\phi)$;
  case $\phi = p_i$ **return** $L(p_i)$;
  case $\phi = c_i$ **return** S_i ;
  case $\phi = \phi_1 \wedge \phi_2$ **return** $eval(M,\phi_1) \cap eval(M,\phi_2)$;
  case $\phi = \phi_1 \vee \phi_2$ **return** $eval(M,\phi_1) \cup eval(M,\phi_2)$;
  case $\phi = \nu X_i.\phi'$
    if $Parent(\phi) = \mu X_j$ **then**
      forall the $\nu X_k \in OpenSub(\phi)$ **do** $S_k = S$;
    end
    repeat
      $S'_i = S_i$;
      $S_i = eval(M[c_i \leftarrow S_i], \phi'[X_i \leftarrow c_i])$;
    until $S_i = S'_i$;
    return S_i;
  end
  case $\phi = \mu X_i.\phi'$
    if $Parent(\phi) = \nu X_j$ **then**
      forall the $\mu X_k \in OpenSub(\phi)$ **do** $S_k = \emptyset$;
    end
    repeat
      $S'_i = S_i$;
      $S_i = eval(M[c_i \leftarrow S_i], \phi'[X_i \leftarrow c_i])$;
    until $S_i = S'_i$;
    return S_i;
  end
endsw

Algorithm 1: Recursive Procedure *eval*

To summarize, μ-PCTL formulas with bounded alternation depth admit a polynomial model-checking procedure, μ-PCTL is more expressive than PCTL. Finally, note that μ-PCTL may be particularly useful to capture properties about repeating patterns of executions with measure 0. For instance, the formula $\nu X.p \wedge [\bigcirc X]_{\geq 0.5}$ allows one to separate the model of Figure 1 from the model obtained from it by removing the loop in state s_0. We leave as further work a careful investigation of this logic.

7 Related Work

Several attempts have been made to extend the features of Kozen's μ-calculus to the realm of logics characterizing Markov chains. Huth and Kwiatkowska and, independently, McIver and Morgan considered qualitative μ-calculi over Markov chains [9, 13]. Their definition replaced union by maximum (*max*) and intersection by minimum (*min*) defining a basic probabilistic calculus. The semantics of a formula was changed from a Boolean value of $\{0, 1\}$ to a real value in $[0, 1]$. Their logic, however, does not capture popular probabilistic temporal logics such as PCTL [14]. In particular, these logics do not include the probabilistic quantification central to the notion of PCTL and also did not allow to capture a single probabilistic quantification surrounding an LTL formula. Cleaveland et al. extend the calculus of Huth and Kwiatkowska by adding probabilisitc quantification and allowing a finite number of nesting of probabilisitc quantifications [5]. In particular, they do not allow interaction between fixpoint operators and probabilistic quantification. This restriction makes reasoning about the logic simple by repeating a finite number of times the evaluation of the simpler logic of Huth and Kwiatkowska. At the same time, it severly limits the expressive power of the logic. The resulting logic allows to express PCTL (and PCTL*). However, it cannot express the μ-calculus over the embedded Kripke structure, or even the CTL formula $\mathsf{EG}p$, which we saw can be expressed in μ-PCTL (and consequently in μ^p-calculus). Both types of μ-calculus are subsets of μ^p-calculus.

Recently, Mio and Simpson [15] suggested an extended quantitative μ-calculus that includes various options for join and meet. They include the *max* and *min* suggested previously, but also include some standard operators in Łukasiewicz logics such as $\oplus$ and $\odot$, that have similar pleasing mathematical properties and are generalizations of Boolean disjunction and conjunction. In order to capture probabilistic quantification they also include explicit multiplication by a rational constant. The resulting logic enjoys some of the mathematical properties of the μ-calculus, allowing one to express PCTL probabilistic quantification, for instance. Using the operators $\oplus$ and $\odot$ as atomic operators results in several shortcomings. The game semantics associated with it includes a construct called "independent product" that relies on additional set-theoretic assumptions (on top of the axiom of choice). In particular, though not relevant for the μ-calculus, it is not known whether games with independent product with general Borel winning conditions are determined. Furthermore, the best algorithms for model checking for this logic are either non-elementary or (by reduction to first-order theory of the reals) triple exponential. Probabilistic quantification is expressed as a combination of a fixpoint of one of the new operators along with multiplication by constants. Another distinct advantage of our logic over that of Mio and Simpson is that we can syntactically recognize formulas that are qualitative.

8 Final Remarks

We have presented a probabilistic μ-calculus that uses probabilistic quantification as an atomic operation. Our main goal is to provide a unifying formalism into which the probabilistic temporal logics used in model checking can be encoded. We have shown that PCTL and PCTL* can be captured in this calculus, and we note that similar results can be obtained for other probabilistic logics such as probabilistic linear temporal logic. We have proved some interesting results for this logic; in particular, its model checking problem is in NP $\cap$ co-NP and it admits a simple game semantics. Furthermore, we presented a simple fragment of this logic which we believe may be important for expressing properties that are not expressible in other probabilistic logics, in particular, those predicating about executions with measure 0, we leave as a further work a deeper investigation of this fragment.

The discrete μ-calculus is intrinsically linked to alternating parity tree automata. We believe that a similar connection exists between μ^p-calculus and p-automata [10]. We leave the consideration of this connection as future work.

References

1 C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

2 A. Bianco and L. de Alfaro. Model checking of probabalistic and nondeterministic systems. In *15th Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 1026 of *Lecture Notes in Computer Science*, pages 499–513. Springer, 1995.

3 K. Chatterjee and N. Piterman. Obligation blackwell games and p-automata. Technical report, arXiv:1206.5174, 2012.

4 F. Ciesinski and C. Baier. LiQuor: A tool for qualitative and quantitative linear time analysis of reactive systems. In *QEST*, pages 131–132. IEEE Computer Society, 2006.

5 R. Cleaveland, S. Purushothaman Iyer, and M. Narasimha. Probabilistic temporal logics via the modal mu-calculus. *Theor. Comput. Sci.*, 342(2-3):316–350, 2005.

6 A. Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992.

7 E.A. Emerson and C. Lei. Efficient model checking in fragments of the μ-calculus. In *LICS*. IEEE Computer Society, 1986.

8 A. Hinton, M.Z. Kwiatkowska, G. Norman, and D. Parker. PRISM: a tool for automatic verification of probabilistic systems. In *TACAS*, volume 3920 of *Lecture Notes in Computer Science*. Springer-Verlag, 2006.

9 M. Huth and M.Z. Kwiatkowska. Quantitative analysis and model checking. In *12th IEEE Symposium on Logic in Computer Science*, pages 111–122. IEEE Computer Society, 1997.

10 M. Huth, N. Piterman, and D. Wagner. p-automata: New foundations for discrete-time probabilistic verification. *Performance Evaluation*, 69(7–8):356–378, 2012.

11 M. Jurdzinski. Deciding the winner in parity games is in UP $\cap$ co-UP. *Inf. Process. Lett.*, 68(3):119–124, 1998.

12 D. Kozen. Results on the propositional μ-calculus. In *Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1982.

13 A. McIver and C. Morgan. Results on the quantitative μ-calculus qMμ. *ACM Trans. Comput. Log.*, 8(1), 2007.

14 M. Mio. *Game Semantics for Probabilistic μ-Calculi*. PhD thesis, University of Edinburgh, 2012.

15 M. Mio and A. Simpson. Łukasiewicz μ-calculus. In *FICS*, 2013.

16 N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Logical Methods in Computer Science*, 3(3):e5, 2007.

17 K. Scheider. *Verification of Reactive Systems: Formal Methods and Algorithms*. Springer, 2004.

18 M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.

19 U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.

A Proofs from Section 3

Proof of Lemma 2:

Proof. The proof for propositions and Boolean connectives is immediate. The only non-qualitative part of a translation is the usage of variables and the translation of $\Box\psi$ and $\Diamond\psi$.

For both $\Box\psi$ and $\Diamond\psi$ the translation is of the form $[\bigcirc p(\psi)]_J$. Thus, it is clearly qualitative. It follows that we can restrict attention to qualitative value functions $\rho : S \to \{0,1\}$ just like in the μ-calculus.

Consider a formula of the form $\varphi = \Box\psi$. Its translation is $p(\varphi) = [\bigcirc p(\psi)]_{\geq 1}$. Suppose that $[\![p(\psi)]\!]^\rho_M = [\![\psi]\!]^\rho_K$. Then, $[\![\varphi]\!]^\rho_K(s) = 1$ if and only if for every successor s' of s we have $[\![\psi]\!]^\rho_K(s') = 1$. However, it follows that for every successor s' we have $[\![\psi]\!]^\rho_M(s') = 1$. Furthermore, as P is a probabilistic transition function it follows that $\sum_{(s,s')\in R} P(s,s') = 1$. Thus, $[\![p(\varphi)]\!]^\rho_M(s) = 1$ as well. The other direction is similar.

Consider a formula of the form $\varphi = \Diamond\psi$. Its translation is $p(\varphi) = [\bigcirc p(\psi)]_{>0}$. Suppose that $[\![p(\psi)]\!]^\rho_M = [\![\psi]\!]^\rho_K$. Then, $[\![\varphi]\!]^\rho_K(s) = 1$ if and only if there is a successor s' of s such that $[\![\psi]\!]^\rho_K(s') = 1$. However, it is then the case that $[\![\bigcirc p(\varphi)]\!]^\rho_M > 0$ and that $[\![\varphi]\!]^\rho_M = 1$.

As fixpoint operators range over qualitative functions only it follows that the two logics have the same least and greatest fixpoints. ◀

Proof of Lemma 3:

Proof. The proof proceeds by induction on the structure of the PCTL formula. For propositions and Boolean connectives the proof is immediate.

Consider the remaining types of formulas:

- Consider the PCTL formula $P_J(\bigcirc\psi)$ and its translation $[\bigcirc t(\psi)]_J$. By induction, $[\![t(\psi)]\!]^\rho_M = [\![\psi]\!]_M$.
  Then, for a state s the measure of paths that satisfy $\bigcirc\psi$ is exactly $\sum_{(s,s')\in R} P(s,s') \cdot [\![\psi]\!]_M(s)$. It follows that the semantics for the two formulas is the same.
- Consider the PCTL formula $\varphi = P_J(\psi_1 \mathcal{U} \psi_2)$ and its translation $[\mu X.t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)]_J$. Let $\psi = \psi_1 \mathcal{U} \psi_2$ and $t(\psi) = t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)$.
  Consider the following sequence of functions $f_i : S \to [0,1]$ that associates with a location $s \in S$ the probability of satisfying $\psi_1 \mathcal{U}^{\leq i} \psi_2$. That is, f_i is the probability of paths of length at most i satisfying that there is $j \leq i$ such that the j-th location satisfies ψ_2 and all locations before j satisfy ψ_1. Clearly, $f_0(s)$ is 1 iff $[\![\psi_2]\!]_M(s) = 1$. It is well known that for every location s we have $\mathsf{measure}_M(s, \psi_1 \mathcal{U} \psi_2) = \lim_{i\to\infty} f_i(s)$. We show that for every $i \geq 0$ we have $[\![\mu X.t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)]\!]^\rho_M \geq f_i$. Consider the case of f_0. As mentioned $f_0 = [\![\psi_2]\!]_M$. By definition $[\![t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)]\!]^\rho_M$ is at least $t(\psi_2)$. Hence, $[\![\mu X.t(\psi)]\!]^\rho_M \geq f_0$. Suppose that $[\![\mu X.t(\psi)]\!]^\rho_M \geq f_i$ and consider f_{i+1}. The following is true for f_{i+1}:

$$f_{i+1} = \begin{cases} 1 & \text{If } [\![\psi_2]\!]_M(s) = 1 \\ 0 & \text{If } [\![\psi_2]\!]_M(s) = 0 \text{ and } [\![\psi_1]\!]_M(s) = 0 \\ \sum_{(s,s')\in R} P(s,s') f_i(s') & \text{Otherwise} \end{cases}$$

  By induction assumption $[\![\mu X.t(\psi)]\!]^\rho_M \geq f_i$. By assumption, $[\![t(\psi_1)]\!]^\rho_M = [\![\psi_1]\!]_M$ and $[\![t(\psi_2)]\!]^\rho_M = [\![\psi_2]\!]_M$. Hence, for locations s such that $[\![\psi_1]\!]_M(s) = 1$ and $[\![\psi_2]\!]_M(s) = 0$ we have $[\![t(\psi)]\!]^\rho_M(s) \geq \sum_{(s,s')\in R} P(s,s')[\![t(\psi)]\!]^\rho_M(s')$. As $[\![t(\psi)]\!]^\rho_M(s') \geq f_i(s')$ we conclude that $[\![t(\psi)]\!]^\rho_M \geq f_{i+1}$.

It follows that for every $s \in S$ we have $[\![\mu X.t(\psi)]\!]^\rho_M(s) \geq \mathsf{measure}_M(s,\varphi)$.
However, for locations s such that $[\![\psi_1]\!]_M(s) = 1$ and $[\![\psi_2]\!]_M(s) = 0$ we have that $\mathsf{measure}_M(s,\varphi)$ is a fixpoint of the equation $\mathsf{measure}_M(s,\varphi) = \sum_{(s,s')\in R} P(s,s')\,\mathsf{measure}_M(s',\varphi)$.
By $[\![\mu X.t(\psi)]\!]^\rho_M$ being the least fixpoint it also follows that $[\![\mu X.t(\psi)]\!]^\rho_M \leq \mathsf{measure}_M(s,\varphi)$.
The probabilistic quantification added on top is the same in both logics.

- Consider the PCTL formula $\varphi = P_J(\psi_1 \mathcal{W} \psi_2)$ and its translation $[\nu X.t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)]_J$. Let $\psi = \psi_1 \mathcal{W} \psi_2$ and $t(\psi) = t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)$.
Consider the following sequence of functions $f_i : S \to [0,1]$ that associates with a location $s \in S$ the probability of satisfying $\psi_1 \mathcal{W}^{\leq i} \psi_2$. That is, f_i is the probability of paths of length at most i satisfying either the first i locations satisfying ψ_1 or there is some $j \leq i$ such that the j-th location satisfies ψ_2 and all locations before j satisfy ψ_1. Clearly, $f_0(s)$ is 1 iff $[\![\psi_2]\!]_M(s) = 1$ or $[\![\psi_1]\!]_M(s) = 1$. It is well known that for every location s we have $\mathsf{measure}_M(s, \psi_1 \mathcal{W} \psi_2) = \lim_{i\to\infty} f_i(s)$. We show that for every $i \geq 0$ we have $[\![\mu X.t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)]\!]^\rho_M \leq f_i$. Consider the case of f_0. As mentioned $f_0 = \max([\![\psi_1]\!]_M, [\![\psi_2]\!]_M)$. By definition $[\![t(\psi_2) \vee (t(\psi_1) \wedge \bigcirc X)]\!]^\rho_M$ is at most $max(t(\psi_1), t(\psi_2))$. Hence, $[\![\mu X.t(\psi)]\!]^\rho_M \leq f_0$. Suppose that $[\![\mu X.t(\psi)]\!]^\rho_M \leq f_i$ and consider f_{i+1}. The following is true for f_{i+1}:
$$f_{i+1} = \begin{cases} 1 & \text{If } [\![\psi_2]\!]_M(s) = 1 \\ 0 & \text{If } [\![\psi_2]\!]_M(s) = 0 \text{ and } [\![\psi_1]\!]_M(s) = 0 \\ \sum_{(s,s')\in R} P(s,s')f_i(s') & \text{Otherwise} \end{cases}$$
By induction assumption $[\![\mu X.t(\psi)]\!]^\rho_M \leq f_i$. By assumption, $[\![t(\psi_1)]\!]^\rho_M = [\![\psi_1]\!]_M$ and $[\![t(\psi_2)]\!]^\rho_M = [\![\psi_2]\!]_M$. Hence, for locations s such that $[\![\psi_1]\!]_M(s) = 1$ and $[\![\psi_2]\!]_M(s) = 0$ we have $[\![t(\psi)]\!]^\rho_M(s) \leq \sum_{(s,s')\in R} P(s,s')[\![t(\psi)]\!]^\rho_M(s')$. As $[\![t(\psi)]\!]^\rho_M(s') \leq f_i(s')$ we conclude that $[\![t(\psi)]\!]^\rho_M \leq f_{i+1}$.
It follows that for every $s \in S$ we have $[\![\mu X.t(\psi)]\!]^\rho_M \leq \mathsf{measure}_M(s,\varphi)$.
However, for locations s such that $[\![\psi_1]\!]_M(s) = 1$ and $[\![\psi_2]\!]_M(s) = 0$ we have that $\mathsf{measure}_M(s,\varphi)$ is a fixpoint of the equation $\mathsf{measure}_M(s,\varphi) = \sum_{(s,s')\in R} P(s,s')\,\mathsf{measure}_M(s',\varphi)$.
By $[\![\mu X.t(\psi)]\!]^\rho_M$ being the greatest fixpoint it also follows that $[\![\mu X.t(\psi)]\!]^\rho_M \geq \mathsf{measure}_M(s,\varphi)$.
The probabilistic quantification added on top is the same in both logics.

◀

More details on the conversion of pctl* to μ^p-calculus:
We extend the conversion of PCTL to μ^p-calculus by a construction that takes a path formula, converts it to a *deterministic* parity word automaton, and in turn convert the automaton to a μ^p-calculus formula.

We start with a short definition of PCTL*. PCTL* formulas over a set AP are defined as state formulas (Φ) and path formulas (Ψ) as follows. As for PCTL, the set of bounds is $J = \{>, \geq\} \times [0,1]$.

$$\Phi ::= p_i \mid \neg p_i \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \mathcal{P}_J(\Psi) \qquad \Psi ::= \Phi \mid \bigcirc\Psi \mid \Psi\,\mathcal{U}\,\Psi \mid \Psi\,\mathcal{W}\,\Psi \mid \Psi \wedge \Psi \mid \Psi \vee \Psi$$

That is, we allow the full syntax of *negation normal form* LTL for defining path formulas. Notice that we allow negation only in front of propositions. As usual, negations can be pushed forward by considering well known equivalences. In particular, $\neg\mathcal{P}_J(\psi)$ is equivalent to $\mathcal{P}_{\overline{J}}(\neg\psi)$, where $\overline{>p} = {\geq}1-p$ and $\overline{\geq p} = {>}1-p$. State formulas are *formulas*. The semantics of PCTL* associates with every formula a set of states. We denote by $[\![\varphi]\!]_M$ the set of states of M that satisfy φ. For every path formula φ and state s of M, $\mathsf{measure}_M(s,\varphi)$ is the measure of paths starting in s that satisfy φ. The semantics and intuitions of PCTL* formulas are as usual, see [1].

For our purposes here, a deterministic parity word automaton (DPW for short) is $\mathcal{D} = \langle \Sigma, D, \rho, d_0, c\rangle$, where Σ is a finite alphabet, D a finite set of states, $\rho : D \times \Sigma \to D$ is a deterministic transition function, $d_0 \in D$ is an initial state, and $c : D \to \mathbb{N}$ is a function associating a priority with each state of $\mathcal{D}$ (minimum parity is most significant). Consider a PCTL* path formula, where state formulas are restricted to propositions (notice that Boolean combinations of propositions can be expressed in the path fragment). Then, it is well known that we can construct DPW accepting all paths that satisfy the formula. Formally, we have the following.

▶ **Theorem 14.** [18, 16] *For every path formula ψ, where state formulas are restricted to propositions, we can construct a DPW D_ψ such that a path ψ satisfies ψ iff it is accepted by D_ψ.*

As before, we define a translation $t(\cdot)$ that given a PCTL* formula returns a μ^p-calculus formula. The difference from PCTL is in the handling of path formulas. For state formulas the translation is as for PCTL:

$$\begin{array}{lll} t(p_i) = p_i & t(\psi_1 \vee \psi_2) = t(\psi_1) \vee t(\psi_2) & t(\mathcal{P}_J(\phi)) = [t(\phi)]_J \\ t(\neg p_i) = \neg p_i & t(\psi_1 \wedge \psi_2) = t(\psi_2) \wedge t(\psi_2) & \end{array}$$

Notice that conjunction and disjunction above correspond only to disjunction and conjunction for state formulas. Consider a path formula ϕ.

Start with a path formula in which the only state formulas appearing are propositions. According to Theorem 14 there exists a DPW $\mathcal{D}_\phi = \langle 2^{AP}, D, \rho, d_0, c\rangle$ that accepts the language of ϕ. We translate $\mathcal{D}_\phi$ to a formula as follows. For a letter $\sigma \in \Sigma$ we define $t(\sigma)$ to be $(\bigwedge_{p\in\sigma} p) \wedge (\bigwedge_{p\notin\sigma} \neg p)$. For a state $d \in D$ we define the $t(d)$ to be $\bigvee_{\sigma\in\Sigma}(t(\sigma) \wedge \bigcirc X_{\rho(d,\sigma)})$. The formula for ϕ consists of a prefix of fixpoints and a suffix that corresponds to the transition of the DPW. For a priority i let $d_i^1, \ldots, d_i^{n_i}$ be an enumeration of the states of $\mathcal{D}$ such that $c(d) = i$. Let m be the maximal priority such that $m = c(d)$ for some $d \in D$. Consider a priority i, then the quantification of i is $t(i) = QX_{d_i^1} \cdots QX_{d_i^{n_i}}$, where Q is ν if i is even and μ if i is odd. Then, the prefix is $t(0) \cdots t(m)$. Overall, the formula for ϕ is $t(0) \cdots t(m).X_{d_0} \wedge (\bigwedge_{d\in D} X_d \to t(d))$. That is, bind all variables according to their parities and then enforce that the initial state holds and that transitions hold.

Consider now a path formula ϕ in which other state formulas appear. We notice that by our assumption the formula is converted to negation normal form. Thus, the state subformulas appearing in ϕ always appear positively. The translation proceeds as before except that the alphabet of the automaton D_ϕ now refers to satisfaction of state subformulas as well as to propositions. We change the encoding $t(\sigma)$ for such a letter to: $(\bigwedge_{p\in\sigma} p) \wedge (\bigwedge_{\neg p\notin\sigma} \neg p) \wedge (\bigwedge_{\psi\in\sigma} t(\psi))$. That is, for every state formula ψ appearing in the letter σ we add a conjunct that requires that $t(\psi)$ holds. As the level of alternation between state and path formulas in PCTL* is finite this construction is sufficient.

▶ **Lemma 15.** *For every Markov chain M and PCTL* formula φ we have $[\![\varphi]\!]_M = [\![t(\varphi)]\!]_M^\rho$.*

Proof. The proof proceeds by induction on the structure of the PCTL* formula. For propositions and Boolean connectives (working on state formulas) the proof is immediate. Furthermore, for all state subformulas it follows from the definition of μ^p-calculus that their semantics is either *true* or *false*, that is, in $0, 1$.

- Consider the PCTL* formula $P_J(\phi)$ and its translation $[t(\phi)]_J$. By induction, $[\![t(\phi)]\!]_M^\rho = [\![\phi]\!]_M$.

Then, for a state s the measure of paths that satisfy ϕ is exactly $[\![\phi]\!]_M(s)$. It follows that the semantics for the two formulas is the same.

- Consider a PCTL* path formula ϕ and its translation $t(\phi)$. By induction, for every state formula ψ, proposition p, or negated proposition $\neg p$ appearing in ϕ we have $[\![t(\psi)]\!]^{\rho}_M = [\![\psi]\!]_M$, $[\![t(p)]\!]^{\rho}_M = [\![p]\!]_M$, and $[\![t(\neg p)]\!]^{\rho}_M = [\![\neg p]\!]_M$. It follows that for every letter in the alphabet of the DPW for ϕ the value of this letter in a state of the Markov chain is 1 iff the letter holds over that state. Consider now a path $s_0, s_1, \ldots$ in the Markov chain. As the automaton D_ϕ for ϕ is deterministic, there exists a unique run $d_0, d_1, \ldots$ of D_ϕ on (the labels of $s_0, s_1, \ldots$).

  Consider the unwinding of the Markov chain from state s to an infinite tree and the labeling of this tree with the *unique* combined runs of D_ϕ on all the tree simultaneously. As D_ϕ is deterministic this is well defined. So D_ϕ in fact induces a function from a prefix of a computation from state s. Namely if $\pi = s_0, \ldots, s_n$, where $s = s_0$, then we identify $r(\pi)$ with the unique state d_n of D_ϕ such that forall $i \geq 1$ we have $d_i = \rho(d_{i_1}, L(s_i))$. We are more interested in the variable $X_{r(\pi)}$ appearing in $t(\phi)$.

  As the automaton is deterministic and as $t(\phi)$ includes X_{d_0} as a conjunct in the body of the formula, we can restrict attention to the values of $X_{r(\pi)}$ in state $last(\pi)$. We have to show that the values of these varialbes in the respective locations in the Markov tree correspond to the probability of the set of paths that satisfy the runs of the automaton starting from the respective state. We have to show that these values correspond to the respective fixpoints.

  It is clear that the value that associates with a set s and a variable X_d the measure of the sets of paths that are accepted by D_ϕ when started from d is a fixpoint of the equations arising from $t(\phi)$. The fixpoints satisfy the respective fixpoint type (least or greatest) following the correctness of the translation of the path formula to a deterministic parity word automaton in addition to the determinism of the alphabet.

◀

It may be possible to replace the deterministic automata in this construction by universal automata (see, e.g., work of Kupferman and Vardi on usage of universal automata in two-player games). Here we are mostly interested in the feasibility of translating PCTL* to μ^p-calculus and not in the complexity of solving PCTL* model checking through μ^p-calculus model checking. The latter may require to consider efficiency through the usage of universal automata. We leave this option for future work.

B Proof from Section 4

B.1 Alternation Depth

We adapt the notion of *alternation depth* [7] to our logic. Roughly speaking, the alternation depth of a formula is a measure of its complexity. Essentially, it is the largest number of μ

and ν alternations that appear in the formula. Formally, we have the following:

$$
\begin{aligned}
ad(p) &= 0 & ad(\varphi_1 \vee \varphi_2) &= \max(ad(\varphi_1), ad(\varphi_2))\\
ad(\neg p) &= 0 & ad(\varphi_1 \wedge \varphi_2) &= \max(ad(\varphi_1), ad(\varphi_2))\\
ad(X) &= 0 & &\\
ad(\bigcirc\varphi) &= ad(\varphi) & ad([\varphi]_J) &= ad(\varphi)
\end{aligned}
$$

$$ad(\mu X.\varphi(X)) = \max(1, ad(\varphi), 1 + ad(\nu Y.\psi)),$$

where, $\nu Y.\psi$ is a subformula of φ and X appears free in ψ.

$$ad(\nu X.\varphi(X)) = \max(1, ad(\varphi), 1 + ad(\mu Y.\psi)),$$

where, $\mu Y.\psi$ is a subformula of φ and X appears free in ψ.

Furthermore, let d be $ad(\varphi)$, with every subformula φ' of φ we can associate a *color* $c(\varphi')$ as follows. If $\varphi' = \nu X.\psi$ then $c(\varphi') = 2(d - ad(\varphi'))$. If $\varphi' = \mu X.\psi$ then $c(\varphi') = 2(d - ad(\varphi')) + 1$. For every other formula φ' we set $c(\varphi') = 2d - 1$. It follows that $c(\varphi')$ is in the range $[0..2d - 1]$.

B.2 Definition of Obligation Games

We recall the value definition in [3]. Let $G = (V, (V_0, V_1, V_p), E, \kappa, \mathcal{G})$ be a parity obligation game, where $\mathcal{G} = (c, O)$. The value of configuration v is determined by considering the following two-player parity game: $turn(G_v) = (V', (V_0', V_1'), E', c')$, where the components of $turn(G_v)$ are as follows:

- $V' = V^+ \times [0,1] \cup V^+ \times [0,1] \times \{\epsilon\} \cup V^+ \times \{f : V \to [0,1]\}$
- $V_0' = \{(w \cdot v, r) \mid O(v) \in \{>p, \bot\}\} \cup \{(w \cdot v, r, \epsilon)\}$
- $V_1' = V' - V_0'$
- $$
\begin{aligned}
E' \;=\; & \big\{((w\cdot v, r), (w\cdot v, f)) \,\big|\, v \in V_0,\ O(v) = \bot, \text{ and } max_{(v,v')\in E} f(v) > r\big\}\\
& \cup \big\{((w\cdot v, r), (w\cdot v, f)) \,\big|\, v \in V_1,\ O(v) = \bot, \text{ and } min_{(v,v')\in E} f(v) > r\big\}\\
& \cup \big\{((w\cdot v, r), (w\cdot v, f)) \,\big|\, v \in V_p,\ O(v) = \bot, \text{ and } \Sigma_{(v,v')\in E}\kappa(v,v')\cdot f(v) > r\big\}\\[1ex]
& \cup \big\{((w\cdot v, r), (w\cdot v, f)) \,\big|\, v \in V_0,\ O(v) = >r', \text{ and } max_{(v,v')\in E} f(v) > r'\big\}\\
& \cup \big\{((w\cdot v, r), (w\cdot v, f)) \,\big|\, v \in V_1,\ O(v) = >r', \text{ and } min_{(v,v')\in E} f(v) > r'\big\}\\
& \cup \big\{((w\cdot v, r), (w\cdot v, f)) \,\big|\, v \in V_p,\ O(v) = >r', \text{ and } \Sigma_{(v,v')\in E}\kappa(v,v')\cdot f(v) > r'\big\}\\[1ex]
& \cup \big\{((w\cdot v, r, \epsilon), (w\cdot v, f)) \,\big|\, v \in V_0 \text{ and } max_{(v,v')\in E} f(v) > r\big\}\\
& \cup \big\{((w\cdot v, r, \epsilon), (w\cdot v, f)) \,\big|\, v \in V_1 \text{ and } min_{(v,v')\in E} f(v) > r\big\}\\
& \cup \big\{((w\cdot v, r, \epsilon), (w\cdot v, f)) \,\big|\, v \in V_p \text{ and } \Sigma_{(v,v')\in E}\kappa(v,v')\cdot f(v) > r\big\}\\[1ex]
& \cup \{((w\cdot v, r), (w\cdot v, r'', \epsilon)) \mid O(v) = \geq r' \text{ and } r'' < r'\}\\
& \cup \{((w\cdot v, f), (w\cdot v', f(v'))) \mid f(v') > 0\}
\end{aligned}
$$
- $c'(w \cdot v) = c(v)$

Let W_0 denote the set of configurations from which player 0 wins in $turn(G_v)$. Then, $msr(w) = \sup\{r \mid (w, r) \in W_0\}$.

We are now ready to define the value of a configuration v in the obligation game G, denoted $val_0(v)$.

▶ **Definition 16.** $val_0(v) = msr(v)$

Notice that the successors of a configuration (v, r) for an obligation configuration v are independent of r and depend only on the obligation $O(v)$. It follows that either player 0 wins from (v, r) for every r or loses for every r. Thus, for every obligation configuration its value is in $\{0, 1\}$.

B.3 Proof of Theorem 6

Proof. Consider a Markov chain M, a location s, a formula φ, and the game $G_{M,\varphi}$ arising from the combination of M and φ. The value of a configuration (s,ψ) in $G_{M,\varphi}$ is determined using a game $turn(G_{(s,\psi)})$ obtained from $G_{M,\varphi}$ as above.

We show that for every configuration v of $G_{M,\varphi}$ the relation between it and its successors satisfies the mathematical relation in definition of the μ^p-calculus. For example, the value of a configuration $(s,\psi_1 \wedge \psi_2)$ is the minimum of the values of (s,ψ_1) and (s,ψ_2). We are interested only in configurations for which the value of v is non-zero. At the second stage we show that the values are actually equivalent to the mathematical definition. Consider the configuration $v=(s,\psi)$:

- If ψ is a proposition p then $val(s,p)$ is 1 if $s \in L(p)$ and 0 otherwise.
- If ψ is a negation of a proposition $\neg p$ then $val(s,\neg p)$ is 1 if $s \notin L(p)$ and 0 otherwise.
- If ψ is a conjunction $\psi=\psi_1 \wedge \psi_2$, then (s,ψ) has two successors (s,ψ_1) and (s,ψ_2) in $G_{M,\varphi}$. Consider the value r such that player 0 wins from $((s,\psi),r')$ for every $r'<r$ in $turn(G_{(s,\psi)})$. By construction, the successors of $((s,\psi),r')$ are $((s,\psi),f)$, where f is a function from $\{(s,\psi_1),(s,\psi_2)\}$ to $[0,1]$ such that and $\min_i\{f(s,\psi_i)\}>r'$.
  It follows that for both (s,ψ_1) and (s,ψ_2) the set of values $\{r''\}$ such that player 0 wins from $((s,\psi_i),r'')$ satisfies $\sup\{r''\} \geq r$. Thus, $val(s,\psi) \leq \min_i\{val(s,\psi_i)\}$.
  A similar argument shows that $val(s,\psi) \geq \min_i\{val(s,\psi_i)\}$.
- The proof for the case that ψ is $\psi_1 \vee \psi_2$ is similar.
- If ψ is $\Box\psi_1$, then (s,ψ) has successors (s',ψ_1) for every successor s' of s in $G_{M,\varphi}$. Consider the value r such that player 0 wins from $((s,\psi),r')$ for every $r'<r$ in $turn(G_{(s,\psi)})$. By construction, the successors of $((s,\psi),r')$ are $((s,\psi),f)$, where f is a function from $\{(s',\psi_1) \mid (s,s') \in R\}$ to $[0,1]$ such that $\min_{s'}\{f(s',\psi_1)\}>r'$.
  It follows that for every s' such that $(s,s') \in R$ we have that the set of values $\{r''\}$ such that player 0 wins from $((s',\psi_1),r'')$ satisfies $\sup\{r''\} \geq r$. Thus, $val(s,\psi) \leq \min_{s'}\{val(s',\psi_1)\}$.
  A similar argument shows that $val(s,\psi) \geq \min_i\{val(s,\psi_i)\}$.
- The proof for the case that ψ is $\Diamond\psi_1$ is similar.
- If ψ is $\bigcirc\psi_1$, then (s,ψ) is a probabilistic configuration in $G_{M,\varphi}$. For every successor s' of s we have that (s',ψ_1) is a successor of (s,ψ) and that the probability of this transition is $\kappa(s,s')$. Consider the value r such that player 0 wins from $((s,\psi),r')$ for every $r'<r$ in $turn(G_{(s,\psi)})$. By construction, the successors of $((s,\psi),r')$ are $((s,\psi),f)$, where f is a function from $\{(s',\psi_1) \mid (s,s') \in R\}$ to $[0,1]$ such that $\sum_{s'} \kappa(s,s') \cdot f(s',\psi_1) > r'$.
  It follows that if $\sum_{s'} \kappa(s,s') \cdot val(s',\psi_1)$ is not $val(s,\psi)$ we can arrive at a contradiction.
- If ψ is $[\psi_1]_J$, then (s,ψ) is an obligation configuration with the successor (s,ψ_1) in $G_{M,\varphi}$. We consider two different cases according to J.
  - Consider the case that $J={>}r$.
    By construction, for every $r' \in (0,1]$ the successor of $((s,\psi),r')$ in $turn(G_{(s,\psi)})$ is $((s,\psi),f)$, where f is a function from (s,ψ_1) to $[0,1]$ such that $f(s,\psi_1)>r$.
    It follows that the set of values r'' such that player 0 wins from (s,ψ_1) satisfies $\sup\{r''\}>r$. Thus, $val(s,\psi_1)>r$ showing that $val(s,\psi)$ is 1.
  - Consider the case that $J={\geq}r$.
    By construction, for every $r' \in (0,1]$ the successors of $((s,\psi),r')$ in $turn(G_{(s,\psi)})$ are of the form $((s,\psi),r'',\epsilon)$ for some $r''<r$. Furthermore, $((s,\psi),r')$ is a player 1 configuration. It follows that for every $r''<r$ we have that player 0 wins from

$((s,\psi),r'',\epsilon)$. Every configuration $((s,\psi),r'',\epsilon)$ has a successors of the form $((s,\psi),f)$, where f is a function from (s,ψ_1) to $[0,1]$ such that $f(s,\psi_1) > r''$.
It follows that the set of values $\{r'''\}$ such that player 0 wins from (s,ψ_1) satisfies $\sup\{r''\} \geq r$. Thus, $val(s,\psi_1) \geq r$ showing that $val(s,\psi)$ is 1.

- If ψ is $\sigma X.\psi_1$, then (s,ψ) has a unique successor (s,ψ_1) in $G_{M,\varphi}$. Consider the value r such that player 0 wins from $((s,\psi_1),r')$ for every $r' < r$ in $turn(G_{(s,\psi)})$. By construction, the successors of $((s,\psi),r')$ are $((s,\psi),f)$, where f is a function from $\{(s,\psi_1)\}$ to $[0,1]$ such that $f(s,\psi_1) > r'$.
It follows that $val(s,\psi) = val(s,\psi_1)$.
- If ψ is X, where $\sigma X.\psi_1$ is the fixpoint bounding X, then by an argument similar to the above we show that $val(s,X) = val(s,\sigma X.\psi_1)$.

It follows that the values in the game satisfy the same relation as in the mathematical definition.

We now have to show that the values are the same.

We prove this by induction on the alternation depth of the formula. If the alternation depth is 0, then the claim follows by simple induction on the structure of the formula from propositions up. Consider a formula φ of alternation depth 1. It follows that φ is of the form $\sigma X.\psi$, where for subformulas that do not depend on X we can assume that the claim holds by induction on their structure.

Consider first the case where $\sigma = \mu$. That is, the formula φ is a least fixed point. Then, by the Knaster-Tarski theorem, the value of φ is the limit of the following functions defined for ordinals α and limit ordinals β.

$$\rho_0 = \lambda s.0$$
$$\rho_{\alpha_1} = [\![\psi]\!]_M^{\rho_\alpha}$$
$$\rho_\beta = \lambda s.\sup_{\alpha<\beta} [\![\psi]\!]_M^{\rho_\alpha}(s)$$

We consider all configurations of the game of the form $(w \cdot v, r)$ where $v = (s, \mu X.\psi)$ for some location s. We show that for every value of the approximation we can find such configurations that match the approximation of the fixpoint. This roughly corresponds to the number of times the fixpoint unfolds below a certain occurrence $(w \cdot v, r)$. As the game $turn(G_{(s,\psi)})$ is winning for player 0 we know that on every infinite path in $turn(G_{(s,\psi)})$ there are a finite number of unfoldings of the fixpoint. However, $turn(G_{(s,\psi)})$ may have configurations under which there is infinite (uncountable, actually) branching. Thus, below some occurrences of the fixpoint there could be no limit to the number of unfoldings of the fixpoint on a single path.

We do know that for the first ordinal γ such that $\rho_\gamma = \rho_{\gamma+1}$ we have that $[\![\varphi]\!]_M^\rho = [\![\varphi]\!]_M^{\rho_\gamma}$. We show a similar result for the game.

Suppose that there is no configuration of the fixpoint that does not have no unfoldings below it. Then, we can find an infinite path that visits infinitely many unfoldings. Clearly, this is a contradiction. It follows that for occurrences of the fixpoint that have no unfolding below thatm we have $val(s,\varphi) = \rho_1(s)$.

Assume by induction that we have identified occurrences of the fixpoint that match ρ_α. That is, configurations $(w \cdot v, r)$, where $v = (s,\varphi)$, for which $val(w \cdot v, r) = \rho_\alpha$.

Consider now a successor ordinal $\alpha + 1$. Suppose that there are no configurations that have at most one unfolding to a configuration that is an α configuration. Then, we claim that the computation is finished. Suppose otherwise, then every configuration depends on an extra unfolding before reaching an α configuration. Clearly, this leads to an infinite number of unfoldings, which contradicts the win of player 0 in the game. Thus, for a configuration that has at most one unfolding until reaching an α configuration it's value can be shown by induction on the structure of the formula to be equivalent to that of $\rho_{\alpha+1}$.

Consider a limit ordinal β. By the part considering the successor ordinals there are configurations depending on ρ_α for every $\alpha < \beta$, otherwise, the approximation stops as explained above. There has to be an unfolding of the fixpoint that depends on all of them and nothing else. Otherwise, as before, we can construct an infinite path with infinitely many unfoldings of the fixpoint. By definition, a dependency on infinitely many unfoldings (on different paths) can occur only if there is a configuration of the form $(s, [\psi']_J)$, where $J = {\geq} r'$ for some r'. It follows that according to the approximation ρ_β the value of $(s, [\psi']_J)$ is computed and some unfolding of the fixpoint that depends on it.

The case of a greatest fixpoint of alternation depth 1 is proven similarly.

We now proceed by induction on the alternation depth of the formula. Consider a formula $\mu X.\varphi$ where the alternation depth of $\mu X.\varphi$ is greater than 1. The proof proceeds as above, however, this time, between every increase of the ordinal, we can consider parts of the game where the game goes into configurations of lower alternation depth. By the induction hypothesis, for such configurations the value in the game is the value of the mathematical approximation. The rest of the proof is the same.

◀

C Proof from Section 5

Proof of Lemma 9:

Proof. We show that for every configuration $[\![\mu X.\psi_R]\!]^\rho_{M_G} = val_0(v)$.

We can show that $val_0(v)$ is a fixpoint of ψ_R. Indeed, consider a configuration v. If $v \in T$ then clearly, $val_0(v) = [\![\mu X.\psi_R]\!]$. Otherwise, if $v \in V_0$ then there is a successor v' such that $(v, v') \in E$ and $val_0(v) = val_0(v')$. The same holds for ψ_R where the value of ψ_R is the maximum of the value of ψ_R over the successors of v. The case of player 1 and probabilistic configurations is similar.

It remains to show that the value is the least fixpoint. It is well known that the following sequence of value functions approximates the value in the game (from below) and converges to the value [6]. Let $val_0^0(v) = 1$ if $v \in T$ and 0 otherwise. Let

$$val_0^{i+1}(v) = \begin{cases} \max_{(v,v')\in E}(val_0^i(v')) & \text{If } v \in V_0 \\ \min_{(v,v')\in E}(val_0^i(v')) & \text{If } v \in V_1 \\ \sum_{(v,v')\in E} \kappa(v,v') val_0^i(v') & \text{If } v \in V_p \end{cases}$$

It is simple to see that for every iteration we have $val_0^i \leq [\![\mu X.\psi_R]\!]$. It follows that the two are equivalent. ◀

D Proofs from Section 6

D.1 Value in finite parity obligation games

We recall the definition of value in finite parity obligation games [3]. Let $G = (V, (V_0, V_1, V_p), E, \kappa, \mathcal{G})$, where $\mathcal{G} = (O, c)$, be a finite parity obligation game. Let k be the maximal priority appearing in c. Let $\mathcal{O}$ denote the set of configurations $v \in V$ such that $O(v) \neq \bot$. Let $\mathcal{N}$ denote the set of non-obligation configurations, that is $\mathcal{N} = V \setminus \mathcal{O}$. For a set of configurations W, let $W_{\geq i}$ denote the subset of W of configurations with priorities at least i according to c. Let W_i denote the subset of W of configurations of priority exactly i. Finally, let α denote the set of infinite paths of configurations that have the minimal priority appearing in them infinitely often is even.

A *dependency* for $v \in \mathcal{O}$ is either $C_v = \bot$ or $C_v \subseteq (\mathcal{O} \times [0..k])$. That is, C_v is either *undefined* or a (possibly empty) set of pairs of obligation configurations annotated by priorities. A *game dependency* is a set $\{C_v\}_{v \in \mathcal{O}}$ of dependencies. A game dependency is *good* if the following conditions hold:

- If for some $v' \in \mathcal{O}$ we have $(v', i) \in C_v$ then $C_{v'} \neq \bot$.
- For every infinite sequence $(v_0, i_0), (v_1, i_1), \ldots$ such that for every j we have $(v_{j+1}, i_{j+1}) \in C_{v_j}$ we have that the minimal priority occurring infinitely often in $i_0, i_1, \ldots$ is even.
- For every $v \in \mathcal{O}$ such that $C_v \neq \bot$ we have $val_0(G', v)J$, where $O(v) = J$ and G' is the turn-based stochastic game with the goal:

$$\bigcup_{(v',i)\in C_v} \left(\begin{array}{ll} (\mathcal{N}^*_{\geq i} \cdot \mathcal{N}_i \cdot \mathcal{N}^*_{\geq i} \cdot (\mathcal{O}_{\geq i} \cap \{v'\}) \cdot V^\omega) & \cup \\ (\mathcal{N}^*_{\geq i} \cdot (\mathcal{O}_i \cap \{v'\}) \cdot V^\omega) & \cup \\ (c \cap \mathcal{N}^\omega) & \end{array} \right)$$

▶ **Theorem 17.** [3] *For every configuration v, $val_0(G, v) = r$ iff there is a good game dependency $\{C_{v'}\}_{v' \in \mathcal{O}}$ such that $val(G', v) = r$, where G' is obtained from G by considering the goal*

$$(\mathcal{N}^* \cdot \{v : C_v \neq \bot\} \cdot V^\omega) \cup (\alpha \cap \mathcal{N}^\omega)$$

D.2 Proof of Theorem 11

Proof. It is possible to see that the sets S_i induce a good dependency. Consider the satisfaction of a set S_i corresponding to the formula $\sigma X.\psi$. It promises that the PCTL formula obtained by replacing X_i in ψ by c_i is satisfied. Restrict attention to the witness that shows satisfaction of this formula. Consider now an obligation appearing inside this witness. Clearly, this obligation is met. If some variable $X_{i'}$ is nested within this obligation then the obligation could depend on other obligations that are nested within $\sigma X_{i'}\psi'$. For every obligation we add to its dependency the set of obligation configurations on which it depends. It is possible to see that the value requirement for the dependencies holds. We have to show that every cycle of within the dependencies satisfies the parity condition. However, such a cycle arises from the cycles between the different sets S_i. However, as sets S_i for least-fixed points start from $\emptyset$ it follows that every cycle within dependencies must visit a smaller even priority.

In the other direction, consider the game $G_{M,\varphi}$ and a good dependency for it. By the structure of the μ-PCTL formula that we can think of as a tree with back edges, it follows that there is exactly one way to reach from one obligation configuration to another. It follows that the dependency of one configuration cannot depend on multiple ways to reach another configuration. It follows that we can remove from the definition of a dependency the annotation of configurations with the minimal priority that needs to be met on the way. Then, the definition of dependency just becomes the set of other obligations that the configuration depends on. It follows that a similar algorithm where we search for the maximal fixpoints of sets of obligation configurations of even priority and minimal fixpoints of sets of obligation configurations of odd priority would exactly find good dependencies. Translating the dependency from the set of obligation configurations and the locations where they are satisfied to the set of fixpoint configurations with the locations they are satisfied shows that the above algorithm is correct.

◀

D.3 Proof of Theorem 12

Proof. The proof proceeds by induction on $ad(\phi)$. Let $k \geq 1$ be the constant such that PCTL model checking of a formula ϕ on a structure M can be computed in time $O(|M|^k \cdot |\phi|)$.

If $ad(\phi) = 0$, it is direct to see that all the base cases take time $O(|M|^k \cdot |\varphi|)$.

If $ad(\phi) = 1$, then all the quantified subformulas are quantified either with μ or ν. Let us first consider the case when we have a unique quantifier, thus $\phi = \mu X_i.\phi_i$ (the proof for ν is similar) where $\phi_i[X_i \leftarrow c_i]$ is a PCTL formula, let us denote by $T_{eval}(\phi)$ the time that $eval(\phi)$ takes. We have that:

$$T_{eval}(\mu X_i.\phi_i) \leq c \cdot |S| \cdot (|S| + (|M|^k \cdot |\phi|))$$

since the loop of lines 18-21 is executed at most $|S|$ times, and model checking the PCTL formula ϕ_i takes $|M|^k \cdot |\phi|$, thus we have that $T_{eval}(\mu X_i.\phi_i) \in O((|M|^k \cdot |\phi|)^2)$. Now, we also note that every other operator (that is, $\wedge$, $\vee$ and $\neg$) can be calculated in time $O((|M|^k \cdot |\phi|)^2)$. Now, suppose that we have a formula with several μ quantifiers. Noting that, in Algorithm 1 each set S_i is calculated once, and their initialization is made only once per each outermost quantifier, we have that:

$$T_{eval}(\phi) \leq c \cdot |\phi| \cdot (|M|^k \cdot |\phi|)^2 + c' \cdot |\phi| \cdot |S| \leq c'' \cdot (|M|^k \cdot |\phi|^{\frac{3}{2}})^2$$

Now, for the inductive case, we assume $ad(\phi) = n+1$, and suppose that $\phi = \nu X_i.\phi_i$ (the other cases can be treated similarly), by induction we have $T_{eval}(\phi_i) \in O((|\phi_i|^{\frac{3}{2}} \cdot |M|^k)^{ad(\phi)})$, thus:

$$T_{eval}(\phi) \leq |\phi| \cdot |S| + |S| \cdot (|S| + (|\phi|^{\frac{3}{2}} \cdot |M|^k)^{ad(\phi)})$$

since we initialized at most $|\phi|$ S_i's and the loop is executed at most $|S|$ times. Then we obtain: $T_{eval}(\phi) \in O((|M|^k \cdot |\phi|^{\frac{3}{2}})^{ad(\phi)+1})$ ◀