Centre Fédéré en Vérification

Technical Report number 2007.82

Qualitative Logics and Equivalences for Probabilistic Systems

Luca de Alfaro, Krishnendu Chatterjee, Marco Faella, Axel Legay

This work was partially supported by a FRFC grant: 2.4530.02

http://www.ulb.ac.be/di/ssd/cfv

Qualitative Logics and Equivalences for Probabilistic Systems

Luca de Alfaro
University of California, Santa Cruz, USA

Krishnendu Chatterjee
University of California, Berkeley, USA

Marco Faella
Università di Napoli “Federico II”, Italy

Axel Legay
University of Liège, Belgium

Abstract

*We present Qualitative Randomized CTL (*QRCTL*), a qualitative version of pCTL, for specifying properties of Markov Decision Processes (MDPs).* QRCTL *formulas can express the fact that certain temporal properties hold with probability 0 or 1, but they do not distinguish other probabilities values.*

We present a symbolic, polynomial time model-checking algorithm for QRCTL *on MDPs. Then, we study the equivalence relation induced by* QRCTL*, called* qualitative equivalence*. We show that for finite* alternating *MDPs, where nondeterministic and probabilistic choice occur in different states, qualitative equivalence coincides with alternating bisimulation, and can thus be computed via efficient partition-refinement algorithms. Surprisingly, the result does not hold for* non-alternating *MDPs. Indeed, we show that no local partition refinement algorithm can compute qualitative equivalence on non-alternating MDPs. Finally, we consider* QRCTL**, that is the “star extension” of* QRCTL*. We show that* QRCTL *and* QRCTL* *induce the same qualitative equivalence on alternating MDPs, while on non-alternating MDPs, the equivalence arising from* QRCTL* *can be strictly finer. We also provide a full characterization of the relation between qualitative equivalence, bisimulation, and alternating bisimulation, according to whether the MDPs are finite, and to whether their transition relations are finite-branching.*

1 Introduction

Markov decision processes (MDPs) provide a model for systems exhibiting both probabilistic and nondeterministic behavior. MDPs were originally introduced to model and solve control problems for stochastic systems: there, nondeterminism represented the freedom in the choice of control action, while the probabilistic component of the behavior described the system's response to the control action [6]. MDPs were later adopted as models for concurrent probabilistic systems, probabilistic systems operating in open environments [25], and under-specified probabilistic systems [8, 12].

Given an MDP and a property of interest, we can ask two kinds of verification questions: *quantitative* and *qualitative* questions. Quantitative questions relate to the numerical value of the probability with which the property holds in the system; qualitative questions ask whether the property holds with probability 0 or 1. Example of quantitative questions include the computation of the maximal and minimal probabilities with which the MDP satisfies a safety, reachability, or in general, ω-regular property [8]; the corresponding qualitative questions asks whether said properties hold with probability 0 or 1.

While much recent work on probabilistic verification has focused on answering quantitative questions, the interest in qualitatitive verification questions predates the one in quantitative ones. Answering qualitative questions about MDPs is useful in a wide range of applications. In the analysis of randomized algorithms, it is natural to require that the correct behavior arises with probability 1, and not just with probability at least p for some $p < 1$. For instance, when analyzing a randomized embedded scheduler, we are interested in whether every thread progresses with probability 1 [13]. Such a qualitative question is much easier to study, and to justify, than its quantitative version; indeed, if we asked for a lower bound $p < 1$ for the probability of progress, the choice of p would need to be justified by an analysis of how much failure probability is acceptable in the final system, an analysis that is generally not easy to accomplish. For the same reason, the correctness of randomized disributed algorithms is often established with respect to qualitative, rather than quantitative, criteria (see, e.g., [24, 21, 28]). Furthermore, since qualitative answers can generally be computed more efficiently than quantitative ones, they are often used as a useful pre-processing step. For instance, when computing the maximal probability of reaching a set of target states T, it is convenient to first pre-compute the set of states $T_1 \supseteq T$ that can reach T

with probability 1, and then, compute the maximal probability of reaching T: this reduces the number of states where the quantitative question needs to be answered, and leads to more efficient algorithms [16]. Lastly, we remark that qualitative answers, unlike quantitative ones, are more robust to perturbations in the numerical values of transition probabilities in the MDP. Thus, whenever a system can be modeled only within some approximation, qualitative verification questions yield information about the system that is more robust with respect to modeling errors, and in many ways, more basic in nature.

In this paper, we provide logics for the specification of qualitative properties of Markov decision processes, along with model-checking algorithms for such logics, and we study the equivalence relations arising from such logics. Our starting point for the logics is provided by the probabilistic logics pCTL and pCTL* [19, 4, 8]. These logics are able to express bounds on the probability of events: the logic pCTL is derived from CTL by adding to its path quantifiers $\forall$ ("for all paths") and $\exists$ ("for at least one path") a probabilistic quantifier P. For a bound $q \in [0, 1]$, an inequality $\bowtie \in \{<, \leq, \geq, >\}$, and a path formula φ, the pCTL formula $\mathrm{P}_{\bowtie q}\varphi$ holds at a state if the path formula φ holds from that state with probability $\bowtie q$. The logic pCTL* is similarly derived from CTL*. In order to obtain logics for qualitative properties, we consider the subsets of pCTL and pCTL* where $\forall$, $\exists$ have been dropped, and where the bound q against which probabilities are compared can assume only the two values 0, 1. We call the resulting logics QRCTL and QRCTL*, for *Qualitative Randomized* CTL and CTL*. The logic QRCTL induces a relation over the state-space of an MDP: we write $s \approx^{>0} t$ if the states s, t satisfy the same QRCTL formulas; similarly, QRCTL* induces the relation $\approx^{>0}_*$. Informally, $s \approx^{>0} t$ holds if the set of properties that hold with probability 0, positive, and 1, at s and t coincide.

We provide symbolic model-checking algorithms for the logic QRCTL; these algorithms can be easily extended to QRCTL*, since in MDPs the verification of general temporal-logic properties can be reduced to reachability questions [11, 12]. As usual, the model-checking algorithms for QRCTL proceed by induction on the structure of a formula. The cases for some of the operators are known; for others, we give new algorithms, completing the picture of the symbolic algorithms required for QRCTL model checking.

We then proceed to study the equivalence relations that arise from QRCTL. For two states s and t of an MDP, we write $s \approx^{>0} t$ if the states s, t satisfy the same QRCTL formulas; similarly, QRCTL* induces the relation $\approx^{>0}_*$. Informally, $s \approx^{>0} t$ holds if the set of properties that hold with probability 0, positive, and 1, at s and t coincide. These relations are thus strictly coarser than standard probabilistic bisimulation [26], which relates states only when the precise probability values coincide. Other works ([18]) have introduced *distances* which quantify the difference in the probabilistic behavior of two MDPs. When the distance between s and t is zero, s and t are probabilistically bisimilar, and so they are also qualitatively bisimilar. Aside from that, the distance between two states is in general unrelated to the states being qualitatively equivalent or not.

The appeal of the relations $\approx^{>0}$ and $\approx^{>0}_*$ lies in their ability to relate implementations and specifications in a qualitative way, abstracting away from precise probability values. The relations, and their asymmetrical counterparts related to simulation, are particularly well-suited to the study of refinement and implementation of randomized algorithms, where the properties to be preserved are most often probability-1 properties. For instance, when implementing a randomized thread scheduler [13], the implementation needs to guarantee that each thread is scheduled infinitely often with probability 1; it is not important that the implementation realizes exactly the same probability of scheduling each thread as the specification. Our qualitative relations can also be used as a help to analyze qualitative properties of systems, similarly to how bisimulation reductions can help in verification. Given a system, the relations enable the construction of a minimized, qualitatively equivalent system, on which all qualitative questions about the original system can be answered. We will show that our qualitative equivalences are computable by efficient discrete graph-theoretic algorithms that do not refer to numerical computation.

We distinguish between *alternating* MDPs, where probabilistic and nondeterministic choices occur at different states, from the general case of *non-alternating* MDPs, where both choices can occur at the same state. Our first result is that on finite, alternating MDPs, the relation $\approx^{>0}$ coincides with alternating bisimulation [2] on the MDP regarded as a two-player game of probability vs. nondeterminism. This result enables the computation of $\approx^{>0}$ via the efficient partition-refinement algorithms developed for alternating bisimulation. We show that the correspondence between $\approx^{>0}$ and alternating bisimulation breaks down both for infinite MDPs, and for finite, but non-alternating, MDPs. Indeed, we show that on non-alternating MDPs, the relation $\approx^{>0}$ cannot be computed by any partition-refinement algorithm that is *local,* in the sense that partitions are refined by looking only at 1-neighbourhoods of states (the classical partition-refinement algorithms for simulation and bisimulation are local). These results are surprising. One is tempted to consider alternating and non-alternating MDPs as equivalent, since an non-alternating MDP can be translated into an alternating one by splitting its states into multiple alternating ones. The difference between the alternating and non-alternating models was already noted in [27] for strong and weak "precise" simulation, and in [5] for axiomatizations. Our results indicate that the difference between the alternating and non-alternating model is even more marked for $\approx^{>0}$, which is a local relation on alternating models, and a non-local relation in non-alternating ones.

More surprises follow when examining the roles of the $\bigcirc$ ("next") and $\mathcal{U}$ ("until") operators, and the distinction between QRCTL and QRCTL*. For CTL, it is known that the $\bigcirc$ operator alone suffices to characterize bisimulation; the $\mathcal{U}$ operator does not add distinguishing power. The same is true for QRCTL on finite, alternating MDPs. On the other hand, we show that for non-alternating, or infinite, MDPs, $\mathcal{U}$ adds distinguishing power to the logic. Similarly, the relations induced by QRCTL and QRCTL* coincide on finite, alternating MDPs, but QRCTL* has greater distinguishing power, and induces thus finer relations, on non-alternating or infinite MDPs.

In summary, we establish that on finite, alternating MDPs, qualitative equivalence can be computed efficiently, and enjoys many canonical properties. We also show that the situation becomes more complex as soon as infinite or non-alternating MDPs are considered. In all cases, we provide sharp boundaries for the classes of MDPs on which our statements apply, distinguishing also between finitely and infinitely-branching MDPs. Our results also indicate how the distinction between alternating and non-alternating MDPs, while often overlooked, is in fact of great importance where the logical properties of the MDPs are concerned.

2 Definitions

2.1 Markov Decision Processes

A probability distribution on a countable set X is a function $f : X \mapsto [0,1]$ such that $\sum_{x \in X} f(x) = 1$; we denote the set of al probability distributions on X by $\mathcal{D}(X)$. Given $f \in \mathcal{D}(X)$, we let $\mathit{Supp}(f) = \{x \in X \mid f(x) > 0\}$ to be the support of f. We consider a fixed set AP of atomic propositions, which includes the distinguished proposition *turn*. Given a set S, we denote S^+ (respectively S^ω) the set of finite (resp. infinite) sequences of elements of S.

A *Markov decision process* (MDP) $G = (S, A, \Gamma, \delta, [\cdot])$ consists of the following components:

- a countable set of states S;
- a finite set of actions A;
- an action assignment $\Gamma : S \mapsto 2^A \setminus \emptyset$, which associates with each state $s \in S$ the set $\Gamma(s)$ of actions that can be chosen at s;
- a transition function $\delta : S \times A \mapsto \mathcal{D}(S)$, which associates with each state s and action a a next-state probability distribution $\delta(s,a)$;
- a labeling function $[\cdot] : S \mapsto 2^{AP}$, which labels all $s \in S$ with the set $[s]$ of atomic propositions true at s.

For $s \in S$ and $a \in \Gamma(s)$, we let $\mathit{Dest}(s,a) = \mathit{Supp}(\delta(s,a))$ be the set of possible destinations when the action a is chosen at the state s. The MDP G is *finite* if the state space S is finite, and it is *finitely-branching* if for all $s \in S$ and $a \in \Gamma(s)$, the set $\mathit{Dest}(s,a)$ is finite. A *play* or *path* is an infinite sequence $\vec{\omega} = \langle s_0, s_1, \ldots \rangle \in S^\omega$ of states of the MDP. For $s \in S$ and $q \in AP$, we say that s is a q-state iff $q \in [s]$. We define an *edge relation* $E = \{(s,t) \in S \times S \mid \exists a \in \Gamma(s) \,.\, t \in \mathit{Dest}(s,a)\}$; for $s \in S$, we let $E(s) = \{t \mid (s,t) \in E\}$. An MDP G is a *Markov chain* if $|\Gamma(s)| = 1$ for all $s \in S$; in this case, for all $s, t \in S$ we write $\delta(s)(t)$ rather than $\delta(s,a)(t)$ for the unique $a \in \Gamma(s)$.

Interpretations. We interpret an MDP in two distinct ways: as a $1\,{}^1\!/_2$-player game, and as a 2-player game. In the $1\,{}^1\!/_2$-player interpretation, probabilistic choice is resolved probabilistically: at a state $s \in S$, player 1 chooses an action $a \in \Gamma(s)$, and the MDP moves to the successor state $t \in S$ with probability $\delta(s,a)(t)$. In the 2-player interpretation, we regard probabilistic choice as adversarial, and we treat the MDP as a game between player 1 and player p: at a state s, player 1 chooses an action $a \in \Gamma(s)$, and player p chooses a destination $t \in \mathit{Dest}(s,a)$. The $1\,{}^1\!/_2$-player interpretation is the classical one [17]. The 2-player interpretation will be used to relate the qualitative equivalence relations for the MDP, with the alternating relations of [2], and thereby derive algorithms for computing the qualitative equivalence relations.

Strategies. A *player-1 strategy* is a function $\sigma : S^+ \mapsto \mathcal{D}(A)$ that prescribes the probability distribution $\sigma(\vec{w})$ over actions to be played, given the past sequence $\vec{w} \in S^+$ of states visited in the play. We require that if $a \in \mathit{Supp}(\sigma(\vec{w} \cdot s))$, then $a \in \Gamma(s)$ for all $a \in A$, $s \in S$, and $\vec{w} \in S^*$. We denote by Σ the set of all player-1 strategies.

A *player-p strategy* is a function $\pi : S^+ \times A \mapsto \mathcal{D}(S)$. The strategy must be such that, for all $s \in S$, $\vec{w} \in S^*$, and $a \in \Gamma(s)$, we have that $\mathit{Supp}(\pi(\vec{w} \cdot s, a)) \subseteq \mathit{Supp}(\delta(s,a))$. Player p follows the strategy π if, whenever player 1 chooses move a after a history of play $\vec{w}$, she chooses the destination state with probability distribution $\pi(\vec{w}, a)$. Thus, in the 2-player interpretation, nondeterminism plays first, and probability second. We denote by Π the set of all player-p strategies.

The 2-player interpretation. In the 2-player interpretation, once a starting state $s \in S$ and two strategies $\sigma \in \Sigma$ and $\pi \in \Pi$ have been chosen, the game is reduced to an ordinary stochastic process, and it is possible to define the probabilities of *events*, where an *event* $\mathcal{A} \subseteq S^\omega$ is a measurable set of paths. We denote the probability of event $\mathcal{A}$, starting from

$s \in S$, under strategies $\sigma \in \Sigma$ and $\pi \in \Pi$ by $\Pr_s^{\sigma,\pi}(\mathcal{A})$. Given $s \in S$ and $\sigma \in \Sigma$, $\pi \in \Pi$, a play $\langle s_0, s_1, \ldots \rangle$ is *feasible* if for every $k \in \mathbb{N}$, there is $a \in \Gamma(s_k)$ such that $\sigma(s_0, s_1, \ldots, s_k)(a) > 0$ and $\pi(s_0, s_1, \ldots, s_k, a)(s_{k+1}) > 0$. We denote by $\mathrm{Outc}(s, \sigma, \pi) \subseteq S^\omega$ the set of feasible plays that start from s given strategies σ and π.

The $1{}^1\!/_2$-player interpretation. In the $1{}^1\!/_2$-player interpretation, we fix for player p the strategy π^* that chooses the next state with the distribution prescribed by δ. Precisely, for all $\vec{w} \in S^*$, $s \in S$, and $a \in \Gamma(s)$, we let $\pi^*(\vec{w} \cdot s, a) = \delta(s, a)$. We then write $\Pr_s^\sigma(\mathcal{A})$ and $\mathrm{Outc}(s, \sigma)$ instead of $\Pr_s^{\sigma,\pi^*}(\mathcal{A})$ and $\mathrm{Outc}(s, \sigma, \pi^*)$, respectively, to underline the fact that these probabilities and set of outcomes are functions only of the initial state and of the strategy of player 1.

Alternating MDPs. An *alternating MDP* (AMDP) is an MDP $G = (S, A, \Gamma, \delta, [\cdot])$ along with a partition (S_1, S_p) of S such that:

1. If $s \in S_1$, then *turn* $\in [s]$ and, for all $a \in \Gamma(s)$, there is $t \in S$ such that $\delta(s, a)(t) = 1$.
2. If $s \in S_p$, then *turn* $\notin [s]$ and $|\Gamma(s)| = 1$.

The states in S_1 are the player-1, or *nondeterministic* states, and the states in S_p are the player-p, or *probabilistic* states. The predicate *turn* ensures that the MDP is *visibly* alternating: the difference between player-1 and player-p states is obvious to the players, and we want it to be obvious to the logic too. Alternating MDPs can be represented more succinctly (and more intuitively) by providing, along with the partition (S_1, S_p) of S, the edge relation $E \subseteq S \times S$, and a probabilistic transition function $\tilde{\delta} : S_P \mapsto \mathcal{D}(S)$. The probabilistic transition function is defined, for $s \in S_p$, $t \in S$, and $a \in \Gamma(s)$, by $\tilde{\delta}(s)(t) = \delta(s, a)(t)$. A *non-alternating* MDP is a general (alternating or not) MDP.

We represent MDPs by graphs: vertices correspond to nodes, and each action a from a state s is drawn as a hyperedge from s to $Dest(s, a)$.

2.2 Logics

We consider two logics for the specification of MDP properties. The first, QRCTL*, is a logic that captures *qualitative* properties of MDPs, and is a qualitative version of pCTL* [19, 4, 8]. The logic is defined with respect to the classical, $1{}^1\!/_2$-player semantics of MDPs. The second logic, ATL*, is a game logic defined with respect to the 2-player semantics of MDPs as in [1].

Syntax. The syntax of both logics is given by defining the set of *path formulas* Φ and *state formulas* Ψ via the following inductive clauses:

$$\begin{array}{lcl} \Phi & ::= & \Psi \mid \Phi \vee \Phi \mid \neg\Phi \mid \bigcirc\Phi \mid \Phi\,\mathcal{U}\,\Phi \mid \Phi\,\mathcal{W}\,\Phi; \\ \Psi & ::= & q \mid \neg\Psi \mid \Psi_1 \vee \Psi_2 \mid PQ(\Phi) \mid \mathrm{tt}. \end{array}$$

where $q \in AP$ is an atomic proposition; tt is the boolean constant with value true, and PQ is a *path quantifier.* The operators $\mathcal{U}$, $\mathcal{W}$ and $\bigcirc$ are temporal operators. The logics ATL* and QRCTL* differ in the path quantifiers:

- The path quantifiers in QRCTL* are: $\exists^{all}$, $\forall^{all}$, $\exists^{some}$, $\forall^{some}$, $\exists^{1}$, $\forall^{1}$, $\exists^{>0}$ and $\forall^{>0}$.
- The path quantifiers in ATL* are: $\langle\!\langle 1 \rangle\!\rangle$, $\langle\!\langle p \rangle\!\rangle$, $\langle\!\langle 1, p \rangle\!\rangle$, $\langle\!\langle \emptyset \rangle\!\rangle$.

The fragments ATL of ATL* and QRCTL of QRCTL* consists of formulas where every temporal operator is immediately preceded by a path quantifier. In the following, when we refer to a "formula" of a logic, without specifying whether it is a state or path formula, we always mean a state formula. As usual, we define $\Box\varphi$ and $\Diamond\varphi$ to be abbreviations for $\varphi\mathcal{W}(\neg\mathrm{tt})$ and $\mathrm{tt}\,\mathcal{U}\varphi$, respectively.

Semantics. For a play $\vec{\omega} = \langle s_0, s_1, \ldots \rangle$ we denote by $\vec{\omega}[i]$ the play starting from the i-th state of $\vec{\omega}$, i.e., $\vec{\omega}[i] = \langle s_i, s_{i+1}, \ldots \rangle$. The semantics for atomic propositions and boolean connectives of formulas are the standard ones. The semantics for the path formulas is defined as follows, for path formulas φ, φ_1, φ_2:

$$\begin{array}{c} \vec{\omega} \models \bigcirc\varphi \text{ iff } \vec{\omega}[1] \models \varphi \\ \vec{\omega} \models \varphi_1\,\mathcal{U}\varphi_2 \text{ iff } \exists j \in \mathbb{N}.\ \vec{\omega}[j] \models \varphi_2 \text{ and} \\ \forall 0 \le i < j.\ \vec{\omega}[i] \models \varphi_1 \\ \vec{\omega} \models \varphi_1\mathcal{W}\varphi_2 \text{ iff } \forall j \in \mathbb{N}.\ \vec{\omega}[j] \models \varphi_1 \text{ or } \big(\exists j \in \mathbb{N}.\ \vec{\omega}[j] \models \varphi_2 \\ \text{and } \forall 0 \le i \le j.\ \vec{\omega}[i] \models \varphi_1\big). \end{array}$$

Notice that it holds $\neg(\varphi_1\,\mathcal{U}\varphi_2) \equiv (\neg\varphi_2)\mathcal{W}(\neg\varphi_1)$. Given a path formula φ we denote by $[\![\varphi]\!] = \{\vec{\omega} \mid \vec{\omega} \models \varphi\}$ the set of plays that satisfy φ. The semantics of the path quantifiers of ATL* and QRCTL* is defined as follows:

$$\begin{array}{ll} s \models \exists^{all}(\varphi) & \text{iff } \exists\sigma \in \Sigma.\ \mathrm{Outc}(s, \sigma) \subseteq [\![\varphi]\!] \\ s \models \forall^{all}(\varphi) & \text{iff } \forall\sigma \in \Sigma.\ \mathrm{Outc}(s, \sigma) \subseteq [\![\varphi]\!] \\[4pt] s \models \exists^{1}(\varphi) & \text{iff } \exists\sigma \in \Sigma.\ \Pr_s^\sigma([\![\varphi]\!]) = 1 \\ s \models \forall^{1}(\varphi) & \text{iff } \forall\sigma \in \Sigma.\ \Pr_s^\sigma([\![\varphi]\!]) = 1 \\[4pt] s \models \exists^{>0}(\varphi) & \text{iff } \exists\sigma \in \Sigma.\ \Pr_s^\sigma([\![\varphi]\!]) > 0 \\ s \models \forall^{>0}(\varphi) & \text{iff } \forall\sigma \in \Sigma.\ \Pr_s^\sigma([\![\varphi]\!]) > 0 \\[4pt] s \models \exists^{some}(\varphi) & \text{iff } \exists\sigma \in \Sigma.\ \mathrm{Outc}(s, \sigma) \cap [\![\varphi]\!] \neq \emptyset \\ s \models \forall^{some}(\varphi) & \text{iff } \forall\sigma \in \Sigma.\ \mathrm{Outc}(s, \sigma) \cap [\![\varphi]\!] \neq \emptyset \\[4pt] s \models \langle\!\langle 1 \rangle\!\rangle(\varphi) & \text{iff } \exists\sigma \in \Sigma.\forall\pi \in \Pi.\mathrm{Outc}(s, \sigma, \pi) \subseteq [\![\varphi]\!] \\ s \models \langle\!\langle p \rangle\!\rangle(\varphi) & \text{iff } \exists\pi \in \Pi.\forall\sigma \in \Sigma.\mathrm{Outc}(s, \sigma, \pi) \subseteq [\![\varphi]\!] \\ s \models \langle\!\langle 1, p \rangle\!\rangle(\varphi) & \text{iff } \exists\sigma \in \Sigma.\exists\pi \in \Pi.\mathrm{Outc}(s, \sigma, \pi) \subseteq [\![\varphi]\!] \\ s \models \langle\!\langle \emptyset \rangle\!\rangle(\varphi) & \text{iff } \forall\sigma \in \Sigma.\forall\pi \in \Pi.\mathrm{Outc}(s, \sigma, \pi) \subseteq [\![\varphi]\!]. \end{array}$$

Given an ATL* or QRCTL* formula φ and an MDP $G = (S, A, \Gamma, \delta, [\cdot])$, we denote by $[\![\varphi]\!]_G = \{s \in S \mid s \models \varphi\}$ the set of states that satisfy φ, and we omit the subscript

G when obvious from the context. For all states s and all formulas φ of QRCTL, the following dualities hold:

$$\begin{aligned} s \models \exists^{all}(\varphi) &\text{ iff } s \models \neg(\forall^{some}(\neg\varphi)) \\ s \models \exists^{some}(\varphi) &\text{ iff } s \models \neg(\forall^{all}(\neg\varphi)) \\ s \models \exists^{>0}(\varphi) &\text{ iff } s \models \neg(\forall^{1}(\neg\varphi)) \\ s \models \exists^{1}(\varphi) &\text{ iff } s \models \neg(\forall^{>0}(\neg\varphi)). \end{aligned} \quad (1)$$

2.3 Equivalence Relations

Given an MDP $G = (S, A, \Gamma, \delta, [\cdot])$, we consider the equivalence relations induced over its state space by various syntactic subsets of the logics QRCTL and ATL. Define the following fragments of QRCTL:

- QRCTL$^{>0}$ is the syntactic fragment of QRCTL containing only the path quantifiers $\exists^{>0}$ and $\forall^{>0}$;
- QRCTL^{all} is the syntactic fragment of QRCTL containing only the path quantifiers $\exists^{all}$ and $\forall^{all}$.

Note that, because of the dualities (1), we do not need to consider the fragments for $\forall^{all}$, $\exists^{all}$, $\forall^{some}$, $\exists^{some}$. The relations induced by QRCTL$^{>0}$ and QRCTL^{all} provide us with a notion of *qualitative* equivalence between states.

$$\approx^{>0} = \{(s,s') \in S \times S \mid \forall \psi \in \text{QRCTL}^{>0},\ s \models \psi \text{ iff } s' \models \psi\}$$
$$\approx^{all} = \{(s,s') \in S \times S \mid \forall \psi \in \text{QRCTL}^{all},\ s \models \psi \text{ iff } s' \models \psi\}.$$

We denote by $\approx^{>0,\bigcirc}$ be the equivalence relation defined by QRCTL$^{>0}$, with $\bigcirc$ as the only temporal operator. We also define the equivalences $\approx_*^{>0}$ and $\approx_*^{all}$ as the QRCTL*-version of $\approx^{>0}$ and $\approx^{all}$, respectively.

The syntactic subset of ATL which uses only the path quantifiers $\langle\langle 1, p\rangle\rangle$ and $\langle\langle \emptyset \rangle\rangle$ induces the usual notion of bisimulation [23]: indeed, the quantifier $\langle\langle 1, p\rangle\rangle$ corresponds to the quantifier $\exists$ of CTL [10]. The syntactic subset of ATL which uses only the path quantifiers $\langle\langle 1\rangle\rangle$ and $\langle\langle p\rangle\rangle$ induces *alternating bisimulation* [2]: in fact, the quantifiers $\langle\langle 1\rangle\rangle$ and $\langle\langle p\rangle\rangle$ directly correspond to the player-1 and player-p quantifiers of ATL [1].

$$\begin{aligned} \approx_{1p} = \{(s,s') \in S \times S \mid &\text{ for all ATL formulas } \psi \text{ with} \\ &\langle\langle 1,p\rangle\rangle, \langle\langle \emptyset\rangle\rangle \text{ as path quantifiers, } s \models \psi \text{ iff } s' \models \psi\}. \end{aligned}$$

$$\begin{aligned} \approx_{1} = \{(s,s') \in S \times S \mid &\text{ for all ATL formulas } \psi \text{ with} \\ &\langle\langle 1\rangle\rangle, \langle\langle p\rangle\rangle \text{ as path quantifiers, } s \models \varphi \text{ iff } s' \models \varphi\}. \end{aligned}$$

$$\begin{aligned} \approx_{1,1p} = \{(s,s') \in S \times S \mid &\text{ for all ATL formulas } \psi, \\ &s \models \varphi \text{ iff } s' \models \varphi\}. \end{aligned}$$

The relations $\approx_{1p}$, $\approx_1$, and $\approx_{1,1p}$ can be computed via well-known partition-refinement algorithms [23, 2].

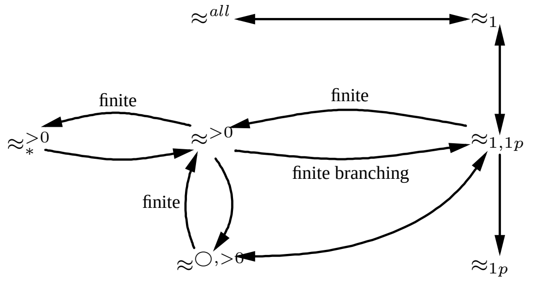


Figure 1. Relationship between equivalence relations for AMDPs.

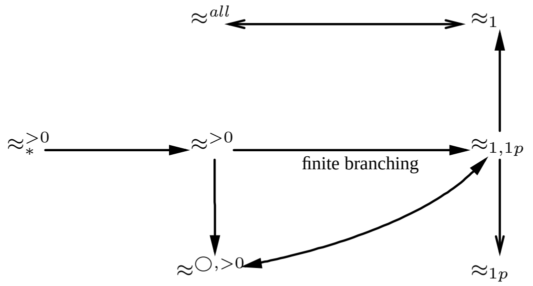


Figure 2. Relationship between equivalence relations for MDPs.

Figure 1 (respectively Figure 2) summarizes the relationships between different equivalence relations on alternating-based MDPs (resp. general MDPs) that we will show in this paper. An arrow from relation A to relation B indicates that A implies B, i.e. that A is finer than B.

3 Model Checking QRCTL

In order to characterize the equivalence relations for QRCTL, it is useful to present first the algorithms for QRCTL model checking. As usual, we present only the algorithms for formulas containing one path quantifier, as nested formulas can be model-checked by recursively iterating the algorithms. As a consequence of dualities (1), we need to provide algorithms only for the operators $\exists\bigcirc$, $\exists\mathcal{U}$, and $\exists\mathcal{W}$, and for the modalities *all*, *1*, > 0, and *some*. The algorithms use the following predecessor operators, for $X, Y \subseteq S$:

$$\begin{aligned} \mathit{Pre}(X) &= \{s \in S \mid \exists a \in \Gamma(s) \,.\, \mathit{Dest}(s,a) \cap X \neq \emptyset\} \\ \mathit{Cpre}(X) &= \{s \in S \mid \exists a \in \Gamma(s) \,.\, \mathit{Dest}(s,a) \subseteq X\} \\ \mathit{Apre}(Y,X) &= \{s \in S \mid \exists a \in \Gamma(s) \,.\, \mathit{Dest}(s,a) \subseteq Y \wedge \\ &\qquad \mathit{Dest}(s,a) \cap X \neq \emptyset\}. \end{aligned}$$

The operators *Pre* and *Cpre* are classical; the operator *Apre* is from [14]. We write the algorithms in μ-calculus notation [20]. Given an MDP $G = (S, A, \Gamma, \delta, [\cdot])$, the interpretation $\llbracket \psi \rrbracket$ of a μ-calculus formula φ is a subset of states; For a propositional symbol $q \in AP$, we have $\llbracket q \rrbracket = \{s \in S \mid q \in [s]\}$ and $\llbracket \neg q \rrbracket = \{s \in S \mid q \notin [s]\}$. The following result directly leads to model-checking algorithms for QRCTL.

Theorem 1 *For atomic propositions q and r, and for all MDPs, the following equalities hold:*

$$\llbracket \exists^{all} \bigcirc q \rrbracket = \llbracket \exists^{1} \bigcirc q \rrbracket = Cpre(\llbracket q \rrbracket) \tag{2}$$

$$\llbracket \exists^{>0} \bigcirc q \rrbracket = \llbracket \exists^{some} \bigcirc q \rrbracket = Pre(\llbracket q \rrbracket) \tag{3}$$

$$\llbracket \exists^{all} q \mathcal{U} r \rrbracket = \mu X.(\llbracket r \rrbracket \cup (\llbracket q \rrbracket \cap Cpre(X))) \tag{4}$$

$$\llbracket \exists^{>0} q \mathcal{U} r \rrbracket = \llbracket \exists^{some} q \mathcal{U} r \rrbracket = \mu X.(\llbracket r \rrbracket \cup (\llbracket q \rrbracket \cap Pre(X))) \tag{5}$$

$$\llbracket \exists^{all} q \mathcal{W} r \rrbracket = \llbracket \exists^{1} q \mathcal{W} r \rrbracket = \nu Y.(\llbracket r \rrbracket \cup (\llbracket q \rrbracket \cap Cpre(Y))) \tag{6}$$

$$\llbracket \exists^{some} q \mathcal{W} r \rrbracket = \nu Y.(\llbracket r \rrbracket \cup (\llbracket q \rrbracket \cap Pre(Y))) \tag{7}$$

If the MDP is finite, the following equalities also hold:

$$\llbracket \exists^{1} q \mathcal{U} r \rrbracket = \nu Y \,.\, \mu X \,.\, (\llbracket r \rrbracket \cup (\llbracket q \rrbracket \cap Apre(Y, X))) \tag{8}$$

$$\llbracket \exists^{>0} q \mathcal{W} r \rrbracket \;=\; \llbracket \exists^{>0} q \mathcal{U} (r \vee \exists^{all} \Box q) \rrbracket. \tag{9}$$

Proof. The formulas for the $\exists^{all}$ modality are derived by noting that, since the formula must hold for *all* resolutions of probabilistic choice, the model-checking problem is equivalent to a game problem, where nondeterminism plays against probability. Formula (5) follows by noting that $s \models \exists^{>0} q \mathcal{U} r$ iff there is a path in (S, E) from s to a r-state, and all states of the path, except possibly the last, are q-states. Formula (8) is from [14]. Formula (6) follow by adapting the classical formulas for $\Box q$ to $q \mathcal{W} r$. Formula (9) can be understood as follows. A *closed component* is a subset of states $T \subseteq S$ such that, for all $s \in T$, there is at least one $a \in \Gamma(s)$ such that $Dest(s, a) \subseteq T$. Using the relation $q \mathcal{W} r \equiv (q \mathcal{U} r) \vee \Box q$ [22], we have for $s \in S$ that $s \models \exists^{>0} q \mathcal{W} r$ iff (i) $s \models \exists^{>0} q \mathcal{U} r$, or (ii) there is a closed component T composed only of q-states, and a path $s_0, s_1, \ldots, s_n$ in (S, E) composed of q-states, with $s_0 = s$ and $s_n \in T$ (see, e.g., [12]). The formula (9) encodes the disjunctions (i) and (ii). ■

Note that, even though (9) is not a μ-calculus formula, it can be readily translated into μ-calculus via (5) and (6). Also observe the μ-calculus formulas corresponding to QRCTL are either alternation free or contain one quantifier alternation between the μ and ν operator. Thus, from the complexity of evaluating μ-calculus formulas we obtain the following result.

Theorem 2 *Given a finite MDP $G = (S, A, \Gamma, \delta, [\cdot])$ and a QRCTL formula Φ, the set $\llbracket \Phi \rrbracket_G$ can be computed in $O(|S| \cdot |\delta| \cdot \ell)$ time, where $|\delta| = \sum_{s \in S} \sum_{a \in \Gamma(s)} |Dest(s, a)|$ and ℓ denotes the length of Φ.*

4 Relationship between QRCTL and ATL Equivalences

In this section, we compare the relations induced by QRCTL and ATL. These comparisons will then be used in Section 5 to derive algorithms to compute $\approx^{all}$ and $\approx^{>0}$.

We first compare $\approx^{all}$ with the relations induced by ATL. As a first result, we show that the relations induced by ATL coincide on alternating MDPs (AMDPs). This result follows from the fact that the turn is visible to the logic.

Proposition 1 *On AMDPs, we have $\approx_1 = \approx_{1p}$.*

Proof. Since turn is observable (via the truth-value of the predicate *turn*), both $\approx_1$ and $\approx_{1p}$ can relate only states where the same player (1 or p) can choose the next move. Based on this observation, the equality of the relations can be proved straightforwardly by induction. ■

Corollary 1 *On AMDPs, we have $\approx_{1,1p} = \approx_1 = \approx_{1p}$.*

Another easy result is that $\approx^{all}$ and $\approx_1$ coincide. This follows by comparing the μ-calculus formulas of Theorem 1 with the algorithms for ATL model-checking [1]. This enables the computation of $\approx^{all}$ via the algorithms for alternating bisimulation [2].

Proposition 2 *For all MDPs, $\approx^{all} = \approx_1$.*

Next, we examine the relationship between $\approx^{>0}$ and $\approx_{1,1p}$. On finitely-branching MDPs, $\approx^{>0}$ is finer than $\approx_{1,1p}$; the result cannot be extended to infinitely-branching MDPs.

Theorem 3 *The following assertions hold:*

1. *On finitely-branching MDPs we have $\approx^{>0} \subseteq \approx_{1,1p}$.*
2. *There is an infinitely-branching AMDP on which $\approx^{>0} \not\subseteq \approx_{1,1p}$.*

Proof. *Part 1.* For $n > 0$, we consider the n-step approximation $\approx_{1,1p}^n$ of $\approx_{1,1p}$: if $s \approx_{1,1p}^n t$, this means that s and t are $(1, 1p)$-bisimilar for n steps. In finite MDPs, we have $\approx_{1,1p} = \approx_{1,1p}^n$ for $n = |S|$; in finitely-branching MDPs, we have $\approx_{1,1p} = \cap_{n=0}^{\infty} \approx_{1,1p}^n$; and this does not extend to MDPs that are not finitely-branching. We define a sequence $\Psi_0, \Psi_1, \Psi_2, \ldots$ of sets of formulas such that, for all $s, t \in S$, we have $s \approx_{1,1p}^n t$ iff s and t satisfy the same formulas in Ψ_n. To this end, given a finite set Ψ of formulas, we denote with BoolC(Ψ) the set of all formulas that consist in disjunctions of conjunctions of formulas in $\{\psi, \neg\psi \mid \psi \in \Psi\}$. We assume that each conjunction (resp. disjunction) in BoolC(Ψ) does not contain repeated elements, so that from the finiteness of Ψ follows the one of

BoolC(Ψ). We let $\Psi_0 = \text{BoolC}(AP)$ and, for $k \geq 0$, we let $\Psi_{k+1} = \text{BoolC}(\Psi_k \cup \{\exists^{>0}\bigcirc\psi, \exists^{all}\bigcirc\psi \mid \psi \in \Psi_k\})$. From these definitions, it is easy to prove the first assertion

Part 2. Consider a Markov chain with state space $S = \mathbb{N} \cup \{s, s'\}$, with only one predicate symbol q, such that $[0] = \{q\}$, and $[t] = \emptyset$ for all $t \in S \setminus \{0\}$. There is a transition from s to every $i \in \mathbb{N}$ with probability $1/2^{i+1}$. There is a transition from s' to s' with probability $1/2$, and from s' to every $i \in \mathbb{N}$ with probability $1/2^{i+2}$. There is a transition from $i \in \mathbb{N}$ with $i > 0$ to every state in $\{j \in \mathbb{N} \mid j < i\}$, with uniform probability. There is a deterministic transition from 0 to itself. Since this is a Markov chain, the two path quantifiers $\exists$ and $\forall$ are equivalent, and we need only consider formulas of the form $\exists^{>0}$ and $\exists^{1}$. By induction on the length of a QRCTL formula φ, we can then show that φ cannot distinguish between states in the set $\{i \in \mathbb{N} \mid i > |\varphi|\} \cup \{s, s'\}$. Hence, $s \approx^{>0} s'$. On the other hand, we have $s \not\approx_{1,1p} s'$, since $s \not\models \langle\langle p \rangle\rangle \Box q$ and $s' \models \langle\langle p \rangle\rangle \Box q$. ∎

To obtain a partial converse of this theorem, we need to relate the semantics of QRCTL and ATL. We do this with a sequence of three lemmas. The first lemma follows by comparing the model-checking algorithms given by Theorem 1 with the algorithms for ATL [1].

Lemma 1 *For all atomic propositions q, r, and for all MDPs, we have:*

$$\begin{aligned}
[\![\exists^{all}\bigcirc q]\!] = [\![\exists^{1}\bigcirc q]\!] &= [\![\langle\langle 1 \rangle\rangle \bigcirc q]\!] \\
[\![\exists^{>0}\bigcirc q]\!] = [\![\exists^{some}\bigcirc q]\!] &= [\![\langle\langle 1,p \rangle\rangle \bigcirc q]\!] \\
[\![\exists^{>0} q\,\mathcal{U} r]\!] = [\![\exists^{some} q\,\mathcal{U} r]\!] &= [\![\langle\langle 1,p \rangle\rangle q\,\mathcal{U} r]\!] \\
[\![\exists^{all} q\,\mathcal{U} r]\!] &= [\![\langle\langle 1 \rangle\rangle q\,\mathcal{U} r]\!] \qquad (10) \\
[\![\exists^{all} q\mathcal{W} r]\!] = [\![\exists^{1} q\mathcal{W} r]\!] &= [\![\langle\langle 1 \rangle\rangle q\mathcal{W} r]\!] \\
[\![\exists^{some} q\mathcal{W} r]\!] &= [\![\langle\langle 1,p \rangle\rangle q\mathcal{W} r]\!].
\end{aligned}$$

Considering finite MDPs only, from the preceding lemma together with (9) we obtain the following result.

Lemma 2 *For finite MDPs, and for all atomic propositions q, r, we have*

$$[\![\exists^{>0} q\mathcal{W} r]\!] = [\![\langle\langle 1,p \rangle\rangle \big(q\,\mathcal{U}(r \vee \langle\langle 1 \rangle\rangle \Box q)\big)]\!]. \qquad (11)$$

For finite MDPs, Lemmas 1 and 2 enable us to translate into ATL all combinations of path quantifiers and temporal operators of QRCTL, except for formulas of the type $\exists^{1}\,\mathcal{U}$. These latter formulas can be model-checked using the μ-calculus expression (8): to obtain a translation into ATL, which will be given in proof of Theorem 4, we first translate into ATL the operator $Apre$. To this end, for ATL formulas φ, ψ, define

$$F_{Apre}(\varphi, \psi) = (\langle\langle 1 \rangle\rangle \bigcirc \psi) \vee \big(\langle\langle \emptyset \rangle\rangle \bigcirc \varphi \wedge \langle\langle p \rangle\rangle \bigcirc \psi\big).$$

Lemma 3 *For AMDPs, and for all* ATL *formulas φ, ψ, we have $[\![F_{Apre}(\varphi, \psi)]\!] = Apre([\![\varphi]\!], [\![\psi]\!])$.*

Note that the lemma holds only for alternating MDPs: indeed, we will show that, on non-alternating MDPs, the operator $Apre$ is not translatable into ATL.

Using these lemmas, we can show that on finite AMDPs, we have $\approx_{1,1p} \subseteq \approx^{>0}$. This result is tight: we cannot relax the assumption that the MDP is finite, nor the assumption that it is alternating.

Theorem 4 *The following assertions hold:*

1. *On finite AMDPs, we have $\approx_{1,1p} \subseteq \approx^{>0}$.*

2. *There is a finite MDP on which $\approx_{1,1p} \not\subseteq \approx^{>0}$.*

3. *There is an infinite, but finitely-branching, AMDP on which $\approx_{1,1p} \not\subseteq \approx^{>0}$.*

Proof. *Part 1.* We prove that on a finite, alternating MDP, the counterpositive holds: if $s \not\approx^{>0} t$, then $s \not\approx_{1,1p} t$. Let s and t be two states such that $s \not\approx^{>0} t$. Then, there must be a formula φ in QRCTL$^{>0}$ that distinguishes s from t. From this formula, we derive a formula $f(\varphi)$ in ATL that distinguishes s from t.

We proceed by structural induction on φ, starting from the inner part of the formula and replacing successive parts that are in the scope of a path quantifier by their ATL version. The cases where φ is an atomic proposition, or a boolean combination of formulas are trivial. Using (1), we reduce QRCTL$^{>0}$-formulas that involve a $\forall$ operator to formulas that only involve the $\exists$ operator. Lemma 1 provides translations for all such formulas, except those of type $\exists^{1}(\varphi\,\mathcal{U}\psi)$. For instance, (10) leads to $f(\exists^{>0}\varphi\,\mathcal{U}\psi) = \langle\langle 1,p \rangle\rangle f(\varphi)\,\mathcal{U} f(\psi)$. In order to translate a formula of the form $\gamma = \exists^{1}(\varphi\,\mathcal{U}\psi)$, we translate the evaluation of the nested μ-calculus formula (8) into the evaluation of a nested ATL formula, as follows. Define the set of formulas $\{\alpha_{i,j} \mid 0 \leq i, j \leq n\}$, where $n = |S|$ is the number of states of the AMDP, via the following clauses:

$$\begin{aligned}
\forall i \in [0..n] : &\quad \alpha_{i,0} = \text{ff} \\
\forall j \in [1..n] : &\quad \alpha_{0,j} = \text{tt} \\
\forall i \in [1..n]\,.\,\forall j \in [0..n-1] : & \\
&\alpha_{i,j+1} = f(\psi) \vee \big(f(\varphi) \wedge F_{Apre}(\alpha_{i-1,n}, \alpha_{i,j})\big).
\end{aligned}$$

From Lemma 3, the above set of formulas encodes the iterative evaluation of the nested fixpoint (8), so that we have $[\![\alpha_{n,n}]\!] = [\![\gamma]\!]$, and we can define $f(\gamma) = \alpha_{n,n}$. This concludes the translation.

Part 2. Consider the MDP shown in Figure 3. The states s and t are such that $(s, t) \in \approx_{1,1p}$. However, $s \models \exists^{1}(\Diamond q)$ (consider the strategy that plays always a), whereas $t \not\models \exists^{1}(\Diamond q)$.

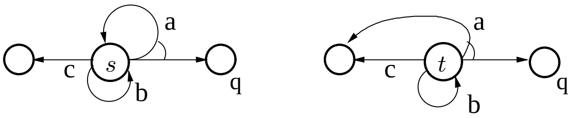


Figure 3. States s and t cannot be distinguished by ATL, but are distinguished by $\exists^1 \Diamond q$.

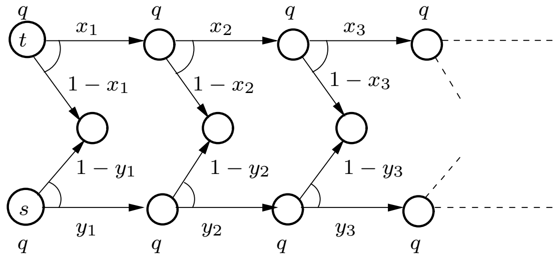


Figure 4. An infinite Markov chain on which $\approx_{1,1p} \not\subseteq \approx^{>0}$, where x_i's and y_i's represent the probabilities that the corresponding edge is taken.

Part 3. Consider the infinite AMDP shown in Figure 4. All states are probabilistic states, i.e. $S_1 = \emptyset$. For all $i > 0$, we set $x_i = \frac{1}{2}$ and $y_i = 2^{-\frac{1}{2^i}}$, so that $\prod_{i>0} x_i = 0$ and $\prod_{i>0} y_i = \frac{1}{2}$. It is easy to see that $s \approx_{1,1p} t$. However, $s \models \exists^{>0}(\Box q)$ and $t \not\models \exists^{>0}(\Box q)$. ■

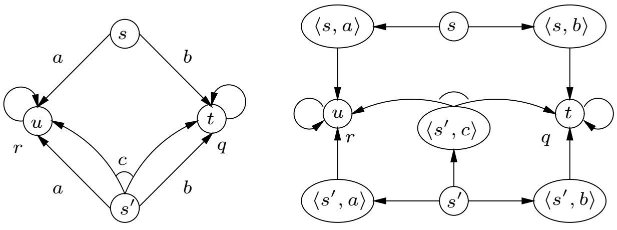
The example in Figure 3 also shows that on non-alternating MDPs, unlike on alternating ones (see Lemma 3), the Apre operator cannot be encoded in ATL. If we were able to encode Apre in ATL, by proceeding as in the proof of the first assertion, given two states s, t with $s \not\approx^{>0} t$, we could construct an ATL formula distinguishing s from t.

As a corollary of Theorems 3 and 4, we have that on finite, alternating MDPs, the equivalences induced by ATL and QRCTL coincide. Thus the discrete graph theoretic algorithms to compute equivalences for ATL can be used to compute the QRCTL equivalences for finite AMDPs.

Corollary 2 *For finite AMDPs, we have* $\approx^{>0} = \approx_{1,1p}$.

5 Computing QRCTL Equivalences

In this section, we take advantage of the results obtained in Section 4 to derive algorithms to compute $\approx^{>0}$ and $\approx^{all}$ for AMDPs. We also provide an algorithm to compute those relations on non-alternating MDPs.



(a) A non-alternating MDP.

(b) An alternating MDP.

Figure 5. MDPs illustrating how separating nondeterministic and probabilistic choice does not help to compute $\approx^{>0}$.

5.1 Alternating MDPs

Corollary 2 immediately provides an algorithm for the computation of the QRCTL equivalences on AMDPs, via the computation of the ATL equivalences (interpreting nondeterminism and probability as the two players). In particular, the partition-refinement algorithms presented in [1] can be directly applied to the problem. This yields the following result.

Theorem 5 *The two problems of computing* $\approx^{>0}$*, and computing* $\approx^{all}$*, on finite AMDPs are PTIME-complete.*

Proof. The result follows from Corollary 2, and from the PTIME-completeness of ATL model checking [1]. ■

5.2 Non-Alternating MDPs

For the general case of *non-alternating* MDPs, on the other hand, the situation is not nearly as simple. First, let us dispel the belief that, in order to compute $\approx^{>0}$ on a non-alternating MDP, we can convert the MDP into an alternating one, compute $\approx^{>0}$ via $\approx_{1,1p}$ (using Corollary 2) on the alternating one, and then somehow obtain $\approx^{>0}$ on the original non-alternating MDP. The following example shows that this, in general, is not possible.

Example 1 Consider the MDP depicted in Figure 5(a), where the set of predicates is $AP = \{q, r\}$. We have $s \approx^{>0} s'$. Indeed, the only difference between s and s' is that at state s' the action c is available: since c is a convex combination of a and b, s and s' are probabilistically bisimilar in the sense of [26], and thus also related by $\approx^{>0}$. We transform this MDP into an alternating one by adding, for each state s each $a \in \Gamma(s)$, a state $\langle s, a\rangle$ which represents the decision of choosing a at s; the result is depicted in Figure 5(b). In this AMDP, however, the state $\langle s', c\rangle$ has no equivalent, as it satisfies both $\exists^{>0}\bigcirc q$ and $\exists^{>0}\bigcirc r$. Therefore, on this AMDP we have $s \not\approx^{>0} s'$, as witnessed by the formula $\exists^{all}\bigcirc((\exists^{>0}\bigcirc q) \wedge (\exists^{>0}\bigcirc r))$. ∎

As the example illustrates, the problem is that once nondeterminism and probability are separated into different states, the distinguishing power of $\approx^{>0}$ increases, so that computing $\approx_{1,1p}$ on the resulting alternating MDP does not help to compute $\approx^{>0}$ on the original non-alternating one.

Failure of local partition refinement. Simulation and bisimulation relations can be computed via partition refinement algorithms that consider, at each step, the *1-neighbourhood* of each state: that is, the set of states reachable from a given state in one step [23]. We call such algorithms *1-neighbourhood partition refinements.* Here, we show a general result: no 1-neighbourhood partition refinement algorithm exists for $\approx^{>0}$ on non-alternating MDPs.

We make this notion precise as follows. Consider an MDP $G = (S, A, \Gamma, \delta, [\cdot])$, together with an equivalence relation $\sim$ on S. Intuitively, two states are 1-neighbourhood isomorphic up to $\sim$ if their 1-step future looks identical, up to the equivalence $\sim$. Formally, we say that two states $s, t \in S$ are *1-neighbourhood isomorphic up to* $\sim$, written $s \overset{1}{\sim} t$, iff $s \sim t$, and if there is a bijection R between $E(s)$ and $E(t)$, and a bijection $\hat{R}$ between $\Gamma(s)$ and $\Gamma(t)$, which preserve $\sim$ and the transition probabilities. Precisely, we require that:

- if $s' \in E(s)$ and $t' \in E(t)$ with $s' \, R \, t'$, then $s' \sim t'$;
- if $a \in \Gamma(s)$ and $b \in \Gamma(t)$ with $a \, \hat{R} \, b$, then for all $s' \in E(s)$ and $t' \in E(t)$ with $s' \, R \, t'$, we have $\delta(s,a)(s') = \delta(t,b)(t')$.

Let *PartS* be the set of equivalence relations on S. A *partition refinement operator* $f : \mathit{PartS} \mapsto \mathit{PartS}$ is an operator such that, for all $\sim \in \mathit{PartS}$, we have $f(\sim) \subseteq \sim$. We say that a partition operator *computes* a relation $\approx$ if we have $\approx = \lim_{n\to\infty} f^n(\sim_{\mathit{pred}})$, where $s \sim_{\mathit{pred}} t$ iff $[s] = [t]$.

We say that a partition refinement operator f is *1-neighbourhood* if it refines an equivalence relation $\sim$ on the basis of the 1-neighbourhood of the states, treating in the same fashion states whose 1-neighbourhoods are isomorphic up to $\sim$. Precisely, f is *1-neighbourhood* if, for all $\sim \in \mathit{PartS}$ and for all $s, s', t, t' \in S$ with $s \overset{1}{\sim} s'$, $t \overset{1}{\sim} t'$, we have either $(s,t), (s',t') \in f(\sim)$, or $(s,t), (s',t') \notin f(\sim)$. We can now state the non-existence of 1-neighbourhood refinement operators for $\approx^{>0}$ as follows.

Theorem 6 *There is no 1-neighbourhood partition refinement operator which computes $\approx^{>0}$ on all MDPs.*

To give an algorithm for the computation of $\approx^{>0}$, given two sets of states C_1 and C_2, let:

$$U(C_1, C_2) = \{\vec{\omega} = \langle s_0, s_1, \ldots\rangle \mid \exists j \geq 0 \,.\, s_j \in C_2 \text{ and } \forall\, 0 \leq i < j \,.\, s_i \in C_1\}$$

$$\mathit{EU}^1(C_1, C_2) = \{s \in S \mid \exists \sigma \in \Sigma.\, \Pr_s^\sigma(U(C_1, C_2)) = 1\}.$$

Intuitively, if $C_1 = [\![\varphi_1]\!]$ and $C_2 = [\![\varphi_2]\!]$ for two QRCTL formulas φ_1 and φ_2, then $\mathit{EU}^1(C_1, C_2)$ is $[\![\exists^1(\varphi_1 \,\mathcal{U}\, \varphi_2)]\!]$. We say that an equivalence relation $\simeq$ is $1, p, \mathit{EU}$*-stable* if, for all unions C_1, C_2 of equivalence classes with respect to $\simeq$, and for all $s, t \in S$ with $s \simeq t$, we have:

1. $s \in \mathit{Pre}(C_1)$ iff $t \in \mathit{Pre}(C_1)$;
2. $s \in \mathit{Cpre}(C_1)$ iff $t \in \mathit{Cpre}(C_1)$;
3. $s \in \mathit{EU}^1(C_1, C_2)$ iff $t \in \mathit{EU}^1(C_1, C_2)$.

Let $\approx_{1,1p}^{\mathit{EU}}$ be the coarsest equivalence relation that is $1, p, \mathit{EU}$-stable. We show that $\approx_{1,1p}^{\mathit{EU}}$ coincides with $\approx^{>0}$.

Theorem 7 *For all finite MDPs, we have $\approx_{1,1p}^{\mathit{EU}} = \approx^{>0}$.*

The following theorem provides an upper bound for the complexity of computing $\approx^{>0}$ on MDPs. The PTIME-completeness of ordinary simulation [3] provides a lower bound, but no tight lower bound is known.

Theorem 8 *The problem of deciding whether $s \approx^{>0} t$ for two states s and t of an MDP is in co-NP.*

6 The Roles of Until, Wait-For and Linear Time Nesting

In this section we study the roles of the until and the wait-for operator, and the relationship between the equivalences induced by QRCTL and QRCTL*.

It is well known that in the standard branching logics CTL and CTL*, as well as in ATL, the next-time operator $\bigcirc$ is the only temporal operator needed for characterizing bisimulation. For QRCTL, this is not the case: the operators $\mathcal{U}$ and $\mathcal{W}$ can increase the distinguishing power of the logics, as the following theorem indicates.

Theorem 9 *The following assertions hold:*

1. *For all MDPs, we have $\approx^{>0,\bigcirc} = \approx_{1,1p}$.*
2. *For all MDPs, we have $\approx^{>0} \subseteq \approx^{>0,\bigcirc}$.*
3. *For finite AMDPs, we have $\approx^{>0,\bigcirc} = \approx^{>0}$.*
4. *There is a finitely-branching, infinite AMDP on which $\approx^{>0,\bigcirc} \not\subseteq \approx^{>0}$.*
5. *There is a finite, (non-alternating) MDP on which $\approx^{>0,\bigcirc} \not\subseteq \approx^{>0}$.*

The logics CTL and CTL* induce the same equivalence, namely, bisimulation. Similarly, ATL and ATL* both induce alternating bisimulation. We show here that QRCTL and QRCTL* induce the same equivalences on finite, alternating MDPs, but we show that for infinite, or non-alternating, MDPs, QRCTL* induces finer relations than QRCTL. These results are summarized by the following theorem.

Theorem 10 *The following assertions hold:*

1. *For all MDPs, we have* $\approx_*^{>0} \subseteq \approx^{>0}$.
2. *For all finite AMDPs, we have* $\approx_*^{>0} = \approx^{>0}$.
3. *There is an infinite AMDP on which* $\approx^{>0} \not\subseteq \approx_*^{>0}$.
4. *There is a finite MDP on which* $\approx^{>0} \not\subseteq \approx_*^{>0}$.

We do not provide an algorithm for computing $\approx_*^{>0}$ on non-alternating MDPs. One of the common uses of equivalences is to reduce a system before applying other verification or analysis techniques. As the complexity of computing $\approx^{>0}$ on non-alternating MDPs is already high (see Theorem 8), reducing a system with respect to $\approx_*^{>0}$ would most likely not be useful in practice.

References

[1] R. Alur, T. Henzinger, and O. Kupferman. Alternating time temporal logic. *J. ACM*, 49:672–713, 2002.

[2] R. Alur, T. Henzinger, O. Kupferman, and M. Vardi. Alternating refinement relations. In *CONCUR 98: Concurrency Theory. 9th Int. Conf.*, volume 1466 of *Lect. Notes in Comp. Sci.*, pages 163–178. Springer-Verlag, 1998.

[3] C. Álvarez, J. L. Balcázar, J. Gabarró, and M. Sántha. Parallel complexity in the design and analysis of concurrent systems. In *PARLE '91: Proc. on Parallel architectures and languages Europe*. Springer-Verlag, 1991.

[4] A. Aziz, V. Singhal, F. Balarin, R. Brayton, and A. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In *Computer Aided Verification*, volume 939 of *Lect. Notes in Comp. Sci.* Springer-Verlag, 1995.

[5] E. Bandini and R. Segala. Axiomatizations for probabilistic bisimulation. In *Proc. 28th Int. Colloq. Aut. Lang. Prog.*, volume 2076 of *Lect. Notes in Comp. Sci.*, pages 370–381. Springer-Verlag, 2001.

[6] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995. Volumes I and II.

[7] G. Bhat and R. Cleaveland. Efficient model checking via the equational μ-calculus. In *Proc. 11th IEEE Symp. Logic in Comp. Sci.*, pages 304–312, 1996.

[8] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Found. of Software Tech. and Theor. Comp. Sci.*, volume 1026 of *Lect. Notes in Comp. Sci.*, pages 499–513. Springer-Verlag, 1995.

[9] K. Chatterjee, L. de Alfaro, and T. Henzinger. Trading memory for randomness. In *QEST 04*. IEEE Computer Society Press, 2004.

[10] E. Clarke and E. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proc. Workshop on Logic of Programs*, volume 131 of *Lect. Notes in Comp. Sci.*, pages 52–71. Springer-Verlag, 1981.

[11] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.

[12] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997. Technical Report STAN-CS-TR-98-1601.

[13] L. de Alfaro, M. Faella, R. Majumdar, and V. Raman. Code-aware resource management. In *EMSOFT 05: ACM Conference on Embedded Software*, Lect. Notes in Comp. Sci. Springer-Verlag, 2005.

[14] L. de Alfaro, T. Henzinger, and O. Kupferman. Concurrent reachability games. In *Proc. 39th IEEE Symp. Found. of Comp. Sci.*, pages 564–575. IEEE Computer Society Press, 1998.

[15] L. de Alfaro, T. Henzinger, and R. Majumdar. From verification to control: Dynamic programs for omega-regular objectives. In *Proc. 16th IEEE Symp. Logic in Comp. Sci.*, pages 279–290. IEEE Press, 2001.

[16] L. de Alfaro, M. Kwiatkowska, G. Norman, D. Parker, and R. Segala. Symbolic model checking of concurrent probabilistic processes using MTBDDs and the Kronecker representation. In *TACAS: Tools and Algorithms for the Construction and Analysis of Systems*, volume 1785 of *Lect. Notes in Comp. Sci.*, pages 395–410. Springer-Verlag, 2000.

[17] C. Derman. *Finite State Markovian Decision Processes*. Academic Press, 1970.

[18] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled markov systems. In *CONCUR'99: Concurrency Theory. 10th Int. Conf.*, volume 1664 of *Lect. Notes in Comp. Sci.*, pages 258–273. Springer, 1999.

[19] H. Hansson and B. Jonsson. A logic for reasoning about time and probability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[20] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354, 1983.

[21] M. Kwiatkowska, G. Norman, and D. Parker. Verifying randomized distributed algorithms with prism. In *Workshop on Advances in Verification (WAVE'00)*, 2000.

[22] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, New York, 1991.

[23] R. Milner. Operational and algebraic semantics of concurrent processes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 1202–1242. Elsevier Science Publishers (North-Holland), Amsterdam, 1990.

[24] A. Pogosyants, R. Segala, and N. Lynch. Verification of the randomized consensus algorithm of Aspnes and Herlihy: a case study. *Distributed Computing*, 13(3):155–186, July 2000.

[25] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, 1995. Technical Report MIT/LCS/TR-676.

[26] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In *CONCUR'94: Concurrency Theory. 5th Int. Conf.*, volume 836 of *Lect. Notes in Comp. Sci.*, pages 481–496. Springer-Verlag, 1994.

[27] R. Segala and A. Turrini. Comparative analysis of bisimulation relations on alternating and non-alternating probabilistic models. In *QEST 05*. IEEE, 2005.

[28] M. Stoelinga. Fun with FireWire: Experiments with verifying the IEEE1394 root contention protocol. In *Formal Aspects of Computing*, 2002.

[29] M. Vardi and P. Wolper. Automata theoretic techniques for modal logics of programs. *J. Comp. Sys. Sci.*, 32:183–221, 1986.

Appendix

A Proof of Theorem 2

Proof of Theorem 2. We first consider the computation of $Pre(X)$, $Cpre(X)$, and $Apre(Y, X)$ for $X, Y \subseteq S$. To decide whether $s \in Pre(X)$ we check if there exists $a \in \Gamma(s)$ such that $Dest(s, a) \cap X \neq \emptyset$. Similarly, to decide whether $s \in Cpre(X)$ (resp. $Apre(Y, X)$) we check if there exists $a \in \Gamma(s)$ such that $Dest(s, a) \subseteq X$ (resp. $Dest(s, a) \subseteq Y$ and $Dest(s, a) \cap X \neq \emptyset$). It follows that given sets X and Y, and the sets $Pre(X)$, $Cpre(X)$, and $Apre(Y, X)$ can be computed in time $O(\sum_{s \in S} \sum_{a \in A} |Dest(s, a)|)$. Given a formula Φ in QRCTL, with all its sub-formulas are evaluated, it follows from Theorem 1 that the computation of $\llbracket \Phi \rrbracket$ can be obtained by computing a μ-calculus formula of constant length with atmost one quantifier alternation of μ and ν. Using the monotonicity property of Pre, $Cpre$ and $Apre$, and the computation of Pre, $Cpre$ and $Apre$, it follows that each inner iteration of the μ-calculus formula can be computed in time $O(\sum_{s \in S} \sum_{a \in A} |Dest(s, a)|)$. Since the outer iteration of the μ-calculus formula converges in $|S|$ iterations, it follows that the $\llbracket \Phi \rrbracket$ can be computed in time $O(S \cdot \sum_{s \in S} \sum_{a \in A} |Dest(s, a)|)$. By a bottom up algorithm that evaluates sub-formulas of a formula first, we obtain the desired bound of the algorithm. ∎

B Proof of Theorem 6

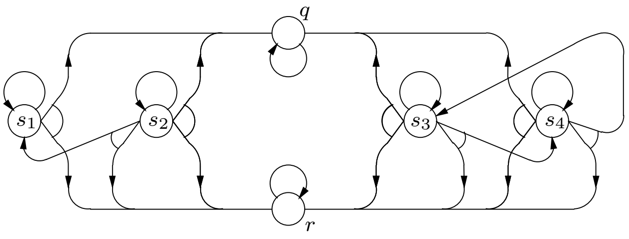


Figure 6. MDP showing the lack of 1-neighbourhood refinement operators.

Proof of Theorem 6. Consider the states s_1, s_2, s_3, s_4 of the MDP depicted in Figure 6, and take $\sim = \sim_{pred}$. Let f be any 1-neighbourhood partition refinement operator. From $s_1 \sim s_2 \sim s_3 \sim s_4$, we can see that $s_2 \overset{1}{\sim} s_3 \overset{1}{\sim} s_4$. Hence, considering the pairs (s_1, s_2), (s_1, s_3), and (s_1, s_4) in the definition of 1-neighbourhood partition refinement operator, we have that f must compute a relation $\sim' = f(\sim)$ satisfying one of the following two cases:

1. $s_1 \not\sim' s_2$ and $s_1 \not\sim' s_3$ and $s_1 \not\sim' s_4$,
2. $s_1 \sim' s_2$ and $s_1 \sim' s_3$ and $s_1 \sim' s_4$.

In the first case, the partition refinement terminates with a relation $\sim''$ such that $s_1 \not\sim'' s_2$. This is incorrect, since we can prove by induction on the length of QRCTL$^{>0}$ formulas that no such formula distinguishes s_1 from s_2, so that $s_1 \approx^{>0} s_2$. In the second case, the partition refinement terminates with a relation $\sim''$ such that $s_1 \sim'' s_3$. This is also incorrect, since the formula $\exists^{1} \Diamond r$ is a witness to $s_1 \not\approx^{>0} s_3$. We conclude that a 1-neighbourhood partition refinement operator cannot compute $\approx^{>0}$. ∎

C Proof of Theorem 7

Proof of Theorem 7. We prove containment in the two directions.

$\approx^{EU}_{1,1p} \subseteq \approx^{>0}$. This statement is equivalent to saying that for all formulas φ in QRCTL$^{>0}$, $[\![\varphi]\!]$ is the union of classes in $S/\approx^{EU}_{1,1p}$. Let s and t be two states such that $s \not\approx^{>0} t$, and let φ be a formula from QRCTL$^{>0}$ such that $s \models \varphi$ and $t \not\models \varphi$. We show by structural induction on φ that $s \not\approx^{EU}_{1,1p} t$. The cases where φ is a proposition, or the boolean combination of formulas are trivial. All other cases follow as in the proof of the first part of Theorem 4, except for the case $\varphi = \exists^{1}(\varphi_1 \mathcal{U} \varphi_2)$. For $\varphi = \exists^{1}(\varphi_1 \mathcal{U} \varphi_2)$, we have $s \in EU^{1}([\![\varphi_1]\!], [\![\varphi_2]\!])$, while $t \notin EU^{1}([\![\varphi_1]\!], [\![\varphi_2]\!])$. By inductive hypothesis, we can assume that $[\![\varphi_1]\!]$ and $[\![\varphi_2]\!]$ are unions of classes in $S/\approx^{EU}_{1,1p}$. So, $(s,t) \notin \approx^{EU}_{1,1p}$.

$\approx^{>0} \subseteq \approx^{EU}_{1,1p}$. The proof follows the same idea of the proof of the first part of Theorem 3. The only modification needed is in the inductive definition of the set of formulas: we take here $\Psi_{k+1} = \text{BoolC}(\Psi_k \cup \{\exists^{>0}\bigcirc\psi, \exists^{all}\bigcirc\psi, \exists^{1}\psi\mathcal{U}\psi' \mid \psi, \psi' \in \Psi_k\})$. ∎

D Proof of Theorem 8

Proof of Theorem 8. We show that the problem of deciding $s \not\approx^{>0} t$ is in NP. To this end, we have to show that there is a certificate for $s \not\approx^{>0} t$ that has polynomial size, and is polynomially checkable. Consider the usual partition-refinement method for computing $\approx_{1,1p}$ [23, 2]. The method starts with an equivalence relation $\simeq$ that reflects propositional equivalence. There are at most $m = |S|$ partition refinements. At each partition refinement, some state-pairs are removed from $\simeq$. A certificate for the removal of a pair from $\simeq$ is simply a *Cpre* or *Pre* or EU^{1} operator, along with a union of equivalence classes; it is thus of size polynomial in m. Since at most m^2 pairs can be removed from $\simeq$, the total size of these state-pair removal certificates is polynomial in m. This yields a polynomial-size and polynomially-checkable certificate for $s \not\approx^{>0} t$. ∎

E Proof of Theorem 9

Proof of Theorem 9. Part 1. The inclusion $\approx^{>0,\bigcirc} \subseteq \approx_{1,1p}$ follows from the fact that formulas used in the first part of the proof of Theorem 3 make use only of the $\bigcirc$ temporal operator, and from $\approx_{1,1p} = \approx^{\bigcirc}_{1,1p}$. To prove the inclusion $\approx_{1,1p} \subseteq \approx^{>0,\bigcirc}$, consider two states $s, t \in S$ such that $s \not\approx^{>0,\bigcirc} t$. Then, there is a QRCTL$^{>0}$ formula φ that distinguishes them. From this formula we derive an ATL formula $f(\varphi)$ that also distinguishes them. We proceed by structural induction. The result is obvious for boolean operators and atomic propositions. The cases $\varphi = \exists^{1}\bigcirc\varphi_1$ and $\varphi = \exists^{>0}\bigcirc\varphi_1$ are an easy consequence of Lemma 1.

Part 2. Immediate, as the set of QRCTL$^{>0}$ formulas without $\mathcal{U}$ and $\mathcal{W}$ is a subset of the set of all QRCTL$^{>0}$ formulas.

Part 3. The result is derived as follows: $\approx^{>0,\bigcirc} \subseteq \approx_{1,1p} = \approx_1 = \approx^{>0}$. The inclusion $\approx^{>0,\bigcirc} \subseteq \approx_{1,1p}$ is a consequence of Part 1. The equality $\approx_{1,1p} = \approx_1$ follows from Corollary 1. The equality $\approx_1 = \approx^{>0}$ follows by combining Theorems 3 and 4.

Part 4. The result follows by considering again the infinite AMDP of Figure 4. Reasoning as in the proof of Theorem 4, it holds $(s,t) \in \approx^{>0,\bigcirc}$, but $(s,t) \notin \approx^{>0}$: indeed, note that $s \models \exists^{>0}(\Box q)$ and $t \not\models \exists^{>0}(\Box q)$.

Part 5. The result is a consequence of Theorem 4, Part 2, and of the present theorem, Part 1: the same MDP used to show $\approx_{1,1p} \not\subseteq \approx^{>0}$, depicted in Figure 3, also shows $\approx^{>0,\bigcirc} \not\subseteq \approx^{>0}$. ∎

F Proof of Theorem 10

Before presenting the proof of this result, it is useful to recall some facts about Rabin automata, Markov decision processes, and probabilistic verification.

Rabin automata and temporal logic. An *infinite-word automaton over AP* is a tuple $A = (L, L_{\mathit{init}}, \ulcorner\cdot\urcorner, \Delta)$, where L is a finite set of locations, $L_{\mathit{init}} \subseteq L$ is the set of initial locations, $\ulcorner\cdot\urcorner : L \mapsto 2^{AP}$ is a labeling function that associates with each location $l \in L$ the set $\ulcorner l \urcorner \subseteq AP$ of predicates that are true at l, and $\Delta : L \mapsto 2^{L}$ is the transition relation. The automaton A is deterministic if the following conditions hold:

- for all $\eta \subseteq AP$, there is a unique $l \in L_{\mathit{init}}$ with $\ulcorner l \urcorner = \eta$;
- for all $l \in L$ and all $\eta \subseteq AP$, there is $l' \in \Delta(l)$ with $\ulcorner l' \urcorner = \eta$;
- for all $l, l', l'' \in L$, we have that $l', l'' \in \Delta(l)$ and $l' \neq l''$ implies $\ulcorner l' \urcorner \neq \ulcorner l'' \urcorner$.

The set of paths of A is *Paths*$(A) = \{l_0, l_1, l_2, \ldots \mid l_0 \in L_{\mathit{init}} \wedge \forall k \geq 0 \,.\, l_{k+1} \in \Delta(l_k)\}$.

A *Rabin acceptance condition* over a set L is a set of pairs $F = \{(P_1, R_1), (P_2, R_2), \ldots, (P_m, R_m)\}$ where, for $1 \leq i \leq m$, we have $P_i, R_i \subseteq L$. The acceptance condition F defines a set of paths over L. For a path $\tau = s_0, s_1, s_2, \ldots \in L^\omega$, we define $\mathrm{Inf}(\tau)$ to be the set of locations that occur infinitely often along τ. We define *Paths*$(F) = \{\tau \in L^\omega \mid \exists i \in [1..m] \,.\, (\mathrm{Inf}(\tau) \subseteq P_i \wedge \mathrm{Inf}(\tau) \cap R_i \neq \emptyset)\}$. A *Rabin automaton* (A, F) is an infinite-word automaton A with set of locations L, together with a Rabin acceptance condition F on L; we associate with it the set of paths *Paths*$(A, F) =$ *Paths*$(A) \cap$*Paths*(F).

Given a set of predicates AP, a *trace* $\rho \in (2^{AP})^\omega$ *over* AP is an infinite sequences of interpretations of AP; we indicate with *Traces*$(AP) = (2^{AP})^\omega$ the set of all traces over AP. A Rabin automaton (A, F) with $A = (L, L_{\mathit{init}}, \ulcorner\cdot\urcorner, \Delta)$ induces the set of traces *Traces*$(AP) = \{\ulcorner l_0 \urcorner, \ulcorner l_1 \urcorner, \ulcorner l_2 \urcorner, \ldots \mid l_0, l_1, l_2, \ldots \in$ *Paths*$(A, F)\}$. An LTL formula φ over the set of propositions AP induces the set of traces *Paths*$(AP) \subseteq$ *Traces*(AP), defined as usual (see, e.g., [22]. From [29] it is known that for an LTL formula φ we can construct a deterministic Rabin automaton (A, F) such that *Traces*$(A, F) =$ *Traces*(φ).

We can now proceed to prove Theorem 10.

Proof of Theorem 10. The first assertion is obvious. For the other assertions, we proceed as follows.

Assertion 2. Consider a finite, alternating MDP $G = (S, A, \Gamma, \delta, [\cdot])$. Since QRCTL is a fragment of QRCTL*, it follows that $\approx^{>0}_{*} \subseteq \approx^{>0}$. To prove $\approx^{>0} (G) \subseteq \approx^{>0}_{*} (G)$ we will show that if there exists a QRCTL* formula that distinguishes two states s and t, then there also exists a QRCTL formula that distinguishes s and t. We focus on formulas of the type $\exists^{>0}\varphi$ and $\exists^{1}\varphi$, where φ is an LTL formula. The generalization to the complete logic follows by structural induction and duality. Thus, assume that there are two states $s^*, t^* \in S$ and $\alpha \in \{1, > 0\}$ such that $s^* \models \exists^\alpha \varphi$ and $t^* \not\models \exists^\alpha \varphi$. Let (A, F) be a deterministic Rabin automaton such that *Traces*$(A, F) =$ *Traces*(φ), and assume that $A = (L, L_{\mathit{init}}, \ulcorner\cdot\urcorner, \Delta)$ and $F = \{(P_1, R_1), \ldots, (P_m, R_m)\}$. Let $G' = G \times A = (S', A, \Gamma', \delta', [\cdot]')$ be the MDP resulting from forming the usual synchronous product of G and A. In detail, we have:

- $S' = \{(s, l) \in S \times L \mid [s] = \ulcorner l \urcorner\}$;
- $\Gamma'(s, l) = \Gamma(s)$ for all $(s, l) \in S'$;
- for all $(s_1, l_1), (s_2, l_2) \in S'$ and $a \in A$, we have $\delta'((s_1, l_1), a)(s_2, l_2) = \delta(s_1, a)(s_2)$ if $l_2 \in \Delta(l_1)$, and $\delta'((s_1, l_1), a)(s_2, l_2) = 0$ otherwise;
- $[(s, l)] = \ulcorner l \urcorner$, for all $(s, l) \in S'$.

Let F' be the Rabin accepting condition of G', defined by $F' = \{(P'_1, R'_1), \ldots, (P'_m, R'_m)\}$, where each $P'_i, R'_i \subseteq S'$ is defined as follows: $P'_i = \{(s, l) \mid l \in P_i\}$ and $R'_i = \{(s, l) \mid l \in R_i\}$. For every $s \in S$, denote with $l_{\mathit{init}}(s)$ the unique $l \in L_{\mathit{init}}$ such that $[s] = \ulcorner l \urcorner$. Using the results of [12, 14, 9] on the model-checking of MDPs with respect to probabilistic temporal-logic properties, we can construct μ-calculus formulas to distinguish $(s^*, l_{\mathit{init}}(s^*))$ and $(t^*, l_{\mathit{init}}(t^*))$. Define, first of all, the following abbreviations:

$$\hat{\psi}^{all} = \bigcup_{i=1}^{m} \nu Y.\, \mu X. \Big[P'_i \cap \big(\mathit{Cpre}(X) \cup (R'_i \cap \mathit{Cpre}(Y))\big)\Big]$$

$$\hat{\psi}^{1} = \bigcup_{i=1}^{m} \nu Y\, .\mu X. \Big[P'_i \cap \big(\mathit{Apre}(Y, X) \cup (R'_i \cap \mathit{Cpre}(Y))\big)\Big]$$

$$\hat{\psi}^{some} = \bigcup_{i=1}^{m} \nu Y.\, \mu X. \Big[P'_i \cap \big(\mathit{Pre}(X) \cup (R'_i \cap \mathit{Pre}(Y))\big)\Big].$$

On the basis of the above formulas, define:

$$\psi^{all} = \mu W \,.\, \big(\hat{\psi}^{all} \cup \mathit{Cpre}(W)\big)$$

$$\psi^{1} = \nu Z \,.\, \mu W \,.\, \big(\mathit{Apre}(Z, W) \cup \hat{\psi}^{1}\big)$$

$$\psi^{>0} = \mu W \,.\, \big(\hat{\psi}^{1} \cup \mathit{Pre}(W)\big)$$

$$\psi^{some} = \mu W \,.\, \big(\hat{\psi}^{some} \cup \mathit{Pre}(W)\big).$$

For $\alpha \in \{\mathit{all}, \mathit{1}, > 0, \mathit{some}\}$ and $s \in S$, we have:

$$(s, l_{\mathit{init}}(s)) \in [\![\psi^\alpha]\!]_{G'} \quad \textit{iff} \quad s \models \exists^\alpha \varphi$$

so that, in particular, $(s^*, l_{\mathit{init}}(s^*)) \in [\![\psi^\alpha]\!]_{G'}$ and $(t^*, l_{\mathit{init}}(t^*)) \notin [\![\psi^\alpha]\!]_{G'}$. Hence, the formula ψ^α is a μ-calculus witness, on G', of the distinction between s^* and t^*. We now show how to transform ψ^α, first into a μ-calculus formula to be evaluated on G, and then into a QRCTL formula to be evaluated on G. this will show that $s^* \not\approx^{>0} t^*$, as required.

To obtain a μ-calculus formula on G, from ψ^α we construct a μ-calculus formula γ^α with the following property: for all $s \in S$, we have $s \in [\![\gamma^\alpha]\!]_G$ iff $(s, l_{\mathit{init}}(s)) \in [\![\psi^\alpha]\!]_{G'}$. The idea, taken from [15], is as follows.

First, ψ^α can be rewritten in *equational form* [7], as a sequence of blocks $B'_1, \ldots, B'_k$, where B'_1 is the innermost block and B'_k the outermost block. Each block B'_j, for $1 \leq j \leq k$, has the form $v_j = \lambda e_j$, where $\lambda \in \{\mu, \nu\}$, and where e_j is an expression not containing μ, ν, in which all the occurrences of the variables $v_1, \ldots, v_k$ have positive polarity [7]; the output variable is v_k.

From this formula, we obtain another formula γ^α, also in equational form, with sets of variables $\{v_i^l \mid 1 \leq i \leq k \wedge l \in L\} \cup \{v_{k+1}\}$. Formula γ^α simulates on G the evaluation of ψ^α on G': for each variable v_i, with $1 \leq i \leq k$, formula γ^α contains the set of variables $\{v_i^l \mid l \in L\}$, where the value of v_i at location $l \in L$ is encoded as the value of v_i^l at s. The formula ψ consists of the blocks $B_1, \ldots, B_k$, plus an additional block B_{k+1}. For $1 \leq i \leq k$, the block B_i contains the equations for the variables $\{v_i^l \mid l \in L\}$. The equation for v_i^l is obtained from the equation for v_i as follows:

- replace each variable v_i on the left-hand side with the variable v_i^l;
- replace P_j (resp. R_j), for $1 \leq j \leq m$, with S if $l \in P_j$ (resp. $l \in R_j$), and with $\emptyset$ if $l \not\in P_j$ (resp. $l \not\in R_j$);
- replace $\mathit{Cpre}(v_h)$, for variable $1 \leq h \leq k$, with $\mathit{Cpre}(\bigcup_{l' \in \Delta(l)} v_h^{l'})$;
- intersect the right-hand side with $\bigcap_{q \in \ulcorner l \urcorner} q \cap \bigcap_{q \in AP \setminus \ulcorner l \urcorner} \neg q$.

The block B_{k+1} consists of only one equation $v_{k+1} = \bigcup_{l \in L_{\mathit{init}}} v_k^l$, and can be either a μ or a ν-block. The output variable is v_{k+1}.

The result of the above transformation is a μ-calculus formula γ^α on G containing only the operators Cpre and Apre. By (2) and Lemma 3, both operators can be encoded in QRCTL. Then, proceeding as in the first part of the proof of Theorem 4, we can "unroll" the computation of the fixpoints of the μ-calculus formulas, since we know that each fixpoint converges in at most $|S|$ iterations. The result of these two transformations is a QRCTL formula λ^α, such that $s^* \models \lambda^\alpha$ and $t^* \not\models \lambda^\alpha$, as required.

Assertion 3. Consider the Markov chain G with state space $S = (\{1\} \times \mathbb{N}) \cup (\{2\} \times \mathbb{N}) \cup \{0\}$. For $n \geq 0$, we have:

$$\begin{aligned} \delta(\langle 1, n\rangle)(\langle 1, n+1\rangle) &= \exp(-1/2^n) \\ \delta(\langle 1, n\rangle)(0) &= 1 - \exp(-1/2^n) \\ \delta(\langle 2, n\rangle)(\langle 2, n+1\rangle) = \delta(\langle 2, n\rangle)(0) &= 1/2. \end{aligned}$$

We take $AP = \{q\}$, and we ask that the predicate q be true at all states of the form $\langle 1, 2n\rangle$ and $\langle 2, 2n\rangle$, for all $n \geq 0$. Then, by induction on the structure of a QRCTL formula, it is not hard to see that $\langle 1, 0\rangle \approx^{>0} \langle 2, 0\rangle$. On the other hand, we have $\langle 1, 0\rangle \models \exists^{>0}\Box\Diamond q$ and $\langle 2, 0\rangle \not\models \exists^{>0}\Box\Diamond q$.

Assertion 4. Consider the MDP depicted in Figure 7. By induction on the structure of a QRCTL formula, it is not hard to see that $s \approx^{>0} s'$. On the other hand, for $\varphi = \exists^{1}(\Diamond q \wedge \Box\exists^{>0}\bigcirc q)$ we have $s \models \varphi$, $s' \not\models \varphi$. ∎

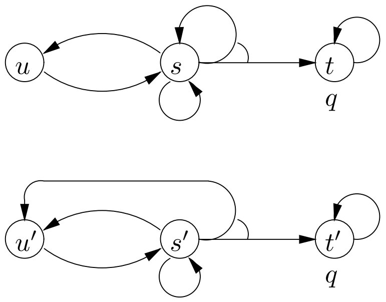


Figure 7. An MDP where $s \approx^{>0} s'$ and $s \not\approx^{>0}_{*} s'$.