Computing Quantiles in Markov Reward Models*

Michael Ummels[1] and Christel Baier[2]

[1] Institute of Transportation Systems, German Aerospace Center
michael.ummels@dlr.de
[2] Technische Universität Dresden
baier@tcs.inf.tu-dresden.de

Abstract. Probabilistic model checking mainly concentrates on techniques for reasoning about the probabilities of certain path properties or expected values of certain random variables. For the quantitative system analysis, however, there is also another type of interesting performance measure, namely *quantiles*. A typical quantile query takes as input a lower probability bound $p \in]0, 1]$ and a reachability property. The task is then to compute the minimal reward bound r such that with probability at least p the target set will be reached before the accumulated reward exceeds r. Quantiles are well-known from mathematical statistics, but to the best of our knowledge they have not been addressed by the model checking community so far.

In this paper, we study the complexity of quantile queries for until properties in discrete-time finite-state Markov decision processes with nonnegative rewards on states. We show that qualitative quantile queries can be evaluated in polynomial time and present an exponential algorithm for the evaluation of quantitative quantile queries. For the special case of Markov chains, we show that quantitative quantile queries can be evaluated in pseudo-polynomial time.

1 Introduction

Markov models with reward (or cost) functions are widely used for the quantitative system analysis. We focus here on the discrete-time or time-abstract case. Discrete-time Markov decision processes, MDPs for short, can be used, for instance, as an operational model for randomised distributed algorithms and rewards might serve to reason, e.g., about the size of the buffer of a communication channel or about the number of rounds that a leader election protocol might take until a leader has been elected.

Several authors considered variants of probabilistic computation tree logic (PCTL) [12,4] for specifying quantitative constraints on the behaviour of Markov

* This work was supported by the DFG project QuaOS and the collaborative research centre HAEC (SFB 912) funded by the DFG. This work was partly supported by the European Union Seventh Framework Programme under grant agreement no. 295261 (MEALS), the DFG/NWO project ROCKS and the cluster of excellence cfAED.

F. Pfenning (Ed.): FOSSACS 2013, LNCS 7794, pp. 353–368, 2013.

models with reward functions. Such extensions, briefly called PRCTL here, permit to specify constraints on the probabilities of reward-bounded reachability conditions, on the expected accumulated rewards until a certain set of target states is reached or expected instantaneous rewards after some fixed number of steps [7,6,9,1,15], or on long-run averages [8]. An example for a typical PRCTL formula with PCTL's probability operator and the reward-bounded until operator is the formula $\mathsf{P}_{>p}(a\ \mathsf{U}_{\leq r}\ b)$ where p is a lower probability bound in $[0,1[$ and r is an upper bound for the accumulated reward earned by path fragments that lead via states where a holds to a b-state. From a practical point of view, more important than checking whether a given PRCTL formula φ holds for (the initial state of) a Markov model $\mathcal{M}$ are PRCTL *queries* of the form $\mathsf{P}_{=?}\,\psi$ where the task is to calculate the (minimum or maximum) probability for the path formula ψ. Indeed, the standard PRCTL model checking algorithm checks whether a given formula $\mathsf{P}_{\bowtie p}\,\psi$ holds in $\mathcal{M}$ by evaluating the PRCTL query $\mathsf{P}_{=?}\,\psi$ and comparing the computed value q with the given probability bound p according to the comparison predicate $\bowtie$. The standard procedure for dealing with PRCTL formulas that refer to expected (instantaneous or accumulated) rewards relies on an analogous scheme; see e.g. [10]. An exception can be made for qualitative PRCTL properties $\mathsf{P}_{\bowtie p}\,\psi$ where the probability bound p is either 0 or 1, and the path formula ψ is a plain until formula without reward bound (or any ω-regular path property without reward constraints): in this case, a graph analysis suffices to check whether $\mathsf{P}_{\bowtie p}\,\psi$ holds for $\mathcal{M}$ [16,5].

In a common project with the operating system group of our department, we learned that a natural question for the systems community is to swap the given and unknown parameters in PRCTL queries and to ask for the computation of a *quantile* (see [2]). For instance, if $\mathcal{M}$ models a mutual exclusion protocol for competing processes $P_1,\ldots,P_n$ and rewards are used to represent the time spent by process P_i in its waiting location, then the *quantile query* $\mathsf{P}_{>0.9}(\mathit{wait}_i\,\mathsf{U}_{\leq ?}\,\mathit{crit}_i)$ asks for the minimal time bound r such that in all scenarios (i.e., under all schedulers) with probability greater than 0.9 process P_i will wait no longer than r time units before entering its critical section. For another example, suppose $\mathcal{M}$ models the management system of a service execution platform. Then the query $\mathsf{P}_{>0.98}(\mathit{true}\,\mathsf{U}_{\leq ?}\,\mathit{tasks\_completed})$ might ask for the minimal initial energy budget r that is required to ensure that even in the worst-case there is more than 98% chance to reach a state where all tasks have been completed successfully.

To the best of our knowledge, quantile queries have not yet been addressed directly in the model checking community. What is known from the literature is that for finite Markov chains with nonnegative rewards the task of checking whether a PRCTL formula $\mathsf{P}_{>p}(a\ \mathsf{U}_{\leq r}\ b)$ or $\mathsf{P}_{\geq p}(a\ \mathsf{U}_{\leq r}\ b)$ holds for some given state is NP-hard [14] when p and r are represented in binary. Since such a formula holds in state s if and only if the value of the corresponding quantile query at s is $\leq r$, this implies that evaluating quantile queries is also NP-hard.

The purpose of this paper is to study quantile queries for Markov decision processes with nonnegative rewards in more details. We consider quantile queries for reward-bounded until formulas in combination with the standard PRCTL

quantifier $\mathsf{P}_{\bowtie p}$ (in this paper denoted by $\forall\mathsf{P}_{\bowtie p}$), where universal quantification over all schedulers is inherent in the semantics, and its dual $\exists\mathsf{P}_{\bowtie p}$ that asks for the existence of some scheduler enjoying a certain property. By duality, our results carry over to reward-bounded release properties.

Contributions. First, we address *qualitative* quantile queries, i.e. quantile queries where the probability bound is either 0 or 1, and we show that such queries can be evaluated in strongly polynomial time. Our algorithm is surprisingly simple and does not rely on value iteration or linear programming techniques (as it is e.g. the case for extremal expected reachability times and stochastic shortest-paths problems in MDPs [9]). Instead, our algorithm relies on the greedy method and borrows ideas from Dijkstra's shortest-path algorithm. In particular, our algorithm can be used for checking PRCTL formulas of the form $\forall\mathsf{P}_{\bowtie p}(a\,\mathsf{U}_{\leq r}\,b)$ or $\exists\mathsf{P}_{\bowtie p}(a\,\mathsf{U}_{\leq r}\,b)$ with $p \in \{0,1\}$ in polynomial time. Previously, a polynomial-time algorithm was known only for the special case of MDPs where every loop contains a state with nonzero reward [13].

Second, we consider *quantitative* quantile queries. The standard way to compute the maximal or minimal probabilities for reward-bounded until properties, say $a\,\mathsf{U}_{\leq r}\,b$, relies on the iterative computation of the extremal probabilities $a\,\mathsf{U}_{\leq i}\,b$ for increasing reward bound i. We use here a reformulation of this computation scheme as a linear program whose size is polynomial in the number of states of $\mathcal{M}$ and the given reward bound r. The crux to derive from this linear program an algorithm for the evaluation of quantile queries is to provide a bound for the sought value, which is our second contribution. This bound then permits to perform a sequential search for the quantile, which yields an exponentially time-bounded algorithm for evaluating quantitative quantile queries. Finally, in the special case of Markov chains with integer rewards, we show that this algorithm can be improved to run in time polynomial in the size of the query, the size of the chain, and the largest reward, i.e. in *pseudo-polynomial* time.

Outline. The structure of the paper is as follows. Section 2 summarises the relevant concepts of Markov decision processes and briefly recalls the logic PRCTL. Quantile queries are introduced in Sect. 3. Our polynomial-time algorithms for qualitative quantile queries is presented in Sect. 4, whereas the quantitative case is addressed in Sect. 5. The paper ends with some concluding remarks in Sect. 6.

2 Preliminaries

In the following, we assume a countably infinite set AP of *atomic propositions*. A Markov decision process (MDP) $\mathcal{M} = (S, Act, \gamma, \lambda, rew, \delta)$ with nonnegative rewards consists of a finite set S of states, a finite set Act of actions, a function $\gamma\colon S \to 2^{Act} \setminus \{\emptyset\}$ describing the set of *enabled actions* in each state, a labelling function $\lambda\colon S \to 2^{\mathsf{AP}}$, a reward function $rew\colon S \to \mathbb{R}^{\geq 0}$, and a transition function $\delta\colon S \times Act \times S \to [0,1]$ such that $\sum_{t\in S} \delta(s,\alpha,t) = 1$ for all $s \in S$ and $\alpha \in Act$. If the set Act of actions is just a singleton, we call $\mathcal{M}$ a Markov chain.

Given an MDP $\mathcal{M}$, we say that a state s of $\mathcal{M}$ is *absorbing* if $\delta(s, \alpha, s) = 1$ for all $\alpha \in \gamma(s)$. Moreover, for $a \in \mathsf{AP}$ we denote by $\lambda^{-1}(a)$ the set of states s such that $a \in \lambda(s)$, and for $x = s_0 s_1 \dots s_k \in S^*$ we denote by $rew(x)$ the accumulated reward after x, i.e. $rew(x) = \sum_{i=0}^{k} rew(s_i)$. Finally, we denote by $|\delta|$ the number of *nontrivial* transitions in $\mathcal{M}$, i.e. $|\delta| = |\{(s, \alpha, t) : \alpha \in \gamma(s) \text{ and } \delta(s, \alpha, t) > 0\}|$.

Schedulers are used to resolve the nondeterminism that arises from the possibility that more than one action might be enabled in a given state. Formally, a scheduler for $\mathcal{M}$ is a mapping $\sigma\colon S^+ \to Act$ such that $\sigma(xs) \in \gamma(s)$ for all $x \in S^*$ and $s \in S$. Such a scheduler σ is *memoryless* if $\sigma(xs) = \sigma(s)$ for all $x \in S^*$ and $s \in S$. Given a scheduler σ and an initial state $s = s_0$, there is a unique probability measure $\Pr_s^\sigma$ on the Borel σ-algebra over S^ω such that $\Pr_s^\sigma(s_0 s_1 \dots s_k \cdot S^\omega) = \prod_{i=0}^{k-1} \delta(s_i, \sigma(s_0 \dots s_i), s_{i+1})$; see [3].

Several logics have been introduced in order to reason about the probability measures $\Pr_s^\sigma$. In particular, the logics PCTL and PCTL* replace the path quantifiers of CTL and CTL* by a single probabilistic quantifier $\mathsf{P}_{\bowtie p}$, where $\bowtie \in \{<, \leq, \geq, >\}$ and $p \in [0, 1]$. In these logics, the formula $\varphi = \mathsf{P}_{\bowtie p}\, \psi$ holds in state s (written $s \models \varphi$) if under *all* schedulers σ the probability $\Pr_s^\sigma(\psi)$ of the path property ψ compares positively with p wrt. the comparison operator $\bowtie$, i.e. if $\Pr_s^\sigma(\psi) \bowtie \psi$. A dual existential quantifier $\exists\mathsf{P}_{\bowtie p}$ that asks for the existence of a scheduler can be introduced using the equivalence $\exists\mathsf{P}_{\bowtie p}\, \psi \equiv \neg\mathsf{P}_{\overline{\bowtie} p}\, \psi$, where $\overline{\bowtie}$ denotes the dual inequality. Since many properties of MDPs can be expressed more naturally using the $\exists\mathsf{P}$ quantifier, we consider this quantifier an equal citizen of the logic, and we denote the universal quantifier P by $\forall\mathsf{P}$ in order to stress its universal semantics.

In order to be able to reason about accumulated rewards, we amend the until operator U by a reward constraint of the form $\sim r$, where $\sim$ is a comparison operator and $r \in \mathbb{R} \cup \{\pm\infty\}$. Since we adopt the convention that a reward is earned upon *leaving* a state, a path $\pi = s_0 s_1 \dots$ fulfils the formula $\psi_1\, \mathsf{U}_{\sim r}\, \psi_2$ if there exists a point $k \in \mathbb{N}$ such that 1. $s_k s_{k+1} \dots \models \psi_2$, 2. $s_i s_{i+1} \dots \models \psi_1$ for all $i < k$, and 3. $rew(s_0 \dots s_{k-1}) \sim r$. Even though our logic is only a subset of the logics PRCTL and PRCTL* defined in [1], we use the same names for the extension of PCTL and PCTL* with the amended until operator. The following proposition states that extremal probabilities for PRCTL* are attainable. This follows, for instance, from the fact that PRCTL* can only describe ω-regular path properties.

Proposition 1. *Let $\mathcal{M}$ be an MDP and ψ a* PRCTL* *path formula. Then there exist schedulers σ^* and τ^* such that $\Pr_s^{\sigma^*}(\psi) = \sup_\sigma \Pr_s^\sigma(\psi)$ and $\Pr_s^{\tau^*}(\psi) = \inf_\tau \Pr_s^\sigma(\psi)$ for all states s of $\mathcal{M}$.*

3 Quantile Queries

A *quantile query* is of the form $\varphi = \forall\mathsf{P}_{\bowtie p}(a\, \mathsf{U}_{\leq ?}\, b)$ or $\varphi = \exists\mathsf{P}_{\bowtie p}(a\, \mathsf{U}_{\leq ?}\, b)$, where $a, b \in \mathsf{AP}$, $p \in [0, 1]$ and $\bowtie \in \{<, \leq, \geq, >\}$. We call queries of the former type *universal* and queries of the latter type *existential*. If $r \in \mathbb{R} \cup \{\pm\infty\}$, we write $\varphi[r]$ for the PRCTL formula that is obtained from φ by replacing ? with r.

Given an MDP $\mathcal{M}$ with rewards, evaluating φ on $\mathcal{M}$ amounts to computing, for each state s of $\mathcal{M}$, the least or the largest $r \in \mathbb{R}$ such that $s \models \varphi[r]$. Formally, if $\varphi = \forall \mathsf{P}_{\bowtie p}(a \,\mathsf{U}_{\leq ?}\, b)$ or $\varphi = \exists \mathsf{P}_{\bowtie p}(a \,\mathsf{U}_{\leq ?}\, b)$ then the *value* of a state s of $\mathcal{M}$ with respect to φ is $\mathrm{val}_\varphi^{\mathcal{M}}(s) \coloneqq \mathrm{opt}\{r \in \mathbb{R} : s \models \varphi[r]\}$, where opt = inf if $\bowtie \in \{\geq, >\}$ and opt = sup otherwise.[1] Depending on whether $\mathrm{val}_\varphi^{\mathcal{M}}(s)$ is defined as an infimum or a supremum, we call φ a *minimising* or a *maximising* query, respectively. In the following, we will omit the superscript $\mathcal{M}$ when the underlying MDP is clear from the context.

Given a query φ, we define the *dual query* to be the unique quantile query $\overline{\varphi}$ such that $\overline{\varphi}[r] \equiv \neg\varphi[r]$ for all $r \in \mathbb{R} \cup \{\pm\infty\}$. Hence, to form the dual of a query, one only needs to replace the quantifier $\forall \mathsf{P}_{\bowtie p}$ by $\exists \mathsf{P}_{\overline{\bowtie} p}$ and vice versa. For instance, the dual of $\forall \mathsf{P}_{<p}(a \,\mathsf{U}_{\leq ?}\, b)$ is $\exists \mathsf{P}_{\geq p}(a \,\mathsf{U}_{\leq ?}\, b)$. Note that the dual of a universal or minimising query is an existential or maximising query, respectively, and vice versa.

Proposition 2. *Let $\mathcal{M}$ be an MDP and φ a quantile query. Then $\mathrm{val}_\varphi(s) = \mathrm{val}_{\overline{\varphi}}(s)$ for all states s of $\mathcal{M}$.*

Proof. Without loss of generality, assume that φ is a minimising query. Let $s \in S$, $v = \mathrm{val}_\varphi(s)$ and $v' = \mathrm{val}_{\overline{\varphi}}(s)$. On the one hand, for all $r < v$ we have $s \not\models \varphi[r]$, i.e. $s \models \overline{\varphi}[r]$, and therefore $v' \geq v$. On the other hand, since $\varphi[r]$ implies $\varphi[r']$ for $r' \geq r$, for all $r > v$ we have $s \models \varphi[r]$, i.e. $s \not\models \overline{\varphi}[r]$, and therefore also $v' \leq v$. □

Assume that we have computed the value $\mathrm{val}_\varphi(s)$ of a state s with respect to a quantile query φ. Then, for any $r \in \mathbb{R}$, to decide whether $s \models \varphi[r]$, we just need to compare r to $\mathrm{val}_\varphi(s)$.

Proposition 3. *Let $\mathcal{M}$ be an MDP, s a state of $\mathcal{M}$, φ a minimising or maximising quantile query, and $r \in \mathbb{R}$. Then $s \models \varphi[r]$ if and only if $\mathrm{val}_\varphi(s) \leq r$ or $\mathrm{val}_\varphi(s) > r$, respectively.*

Proof. First assume that $\varphi = Q(a \,\mathsf{U}_{\leq ?}\, b)$ is a minimizing query. Clearly, if $s \models \varphi[r]$, then $\mathrm{val}_\varphi(s) \leq r$. On the other hand, assume that $\mathrm{val}_\varphi(s) \leq r$ and denote by R the set of numbers $x \in \mathbb{R}$ of the form $x = \sum_{i=0}^{k} rew(s_i)$ for a finite sequence $s_0 s_1 \dots s_k$ of states. Since the set $\{x \in R : x \leq n\}$ is finite for all $n \in \mathbb{N}$, we can fix some $\varepsilon > 0$ such that $r + \delta \notin R$ for all $0 < \delta \leq \varepsilon$. Hence, the set of paths that fulfil $a \,\mathsf{U}_{\leq r}\, b$ agrees with the set of paths that fulfil $a \,\mathsf{U}_{\leq r+\varepsilon}\, b$. Since $\mathrm{val}_\varphi(s) < r + \varepsilon$ and φ is a minimising query, we know that $s \models \varphi[r + \varepsilon]$. Since replacing $r + \varepsilon$ by r does not affect the path property, this implies that $s \models \varphi[r]$. Finally, if φ is a maximising query, then $\overline{\varphi}$ is a minimising query, and $s \models \overline{\varphi}[r]$ if and only if $\mathrm{val}_{\overline{\varphi}}(s) = \mathrm{val}_\varphi(s) \leq r$, i.e. $s \models \varphi[r]$ if and only if $\mathrm{val}_\varphi(s) > r$. □

Proposition 3 does not hold when we allow r to take an infinite value. In fact, if φ is a minimizing query and $s \not\models \varphi[\infty]$, then $\mathrm{val}_\varphi(s) = \infty$. Analagously, if φ is a maximising query and $s \not\models \varphi[-\infty]$, then $\mathrm{val}_\varphi(s) = -\infty$.

To conclude this section, let us remark that queries using the reward-bounded *release* operator R can easily be accommodated in our framework. For instance, the query $\forall \mathsf{P}_{\geq p}(a \,\mathsf{R}_{\leq ?}\, b)$ is equivalent to the query $\forall \mathsf{P}_{\leq 1-p}(\neg a \,\mathsf{U}_{\leq ?}\, \neg b)$.

[1] As usual, we assume that $\inf \emptyset = \infty$ and $\sup \emptyset = -\infty$.

Algorithm 1. Solving qualitative queries of the form $Q(a \,\mathsf{U}_{\leq ?}\, b)$

Input: MDP $\mathcal{M} = (S, Act, \gamma, \lambda, rew, \delta)$, $\varphi = Q(a \,\mathsf{U}_{\leq ?}\, b)$

for each $s \in S$ **do**
  if $s \models b$ **then** $v(s) \leftarrow 0$ **else** $v(s) \leftarrow \infty$
$X \leftarrow \{s \in S : v(s) = 0\}$; $R \leftarrow \{0\}$
$Z \leftarrow \{s \in S : s \models a \wedge \neg b \text{ and } rew(s) = 0\}$
while $R \neq \emptyset$ **do**
  $r \leftarrow \min R$; $Y \leftarrow \{s \in X : v(s) \leq r\} \setminus Z$
  for each $s \in S \setminus X$ with $s \models a \wedge Q\,\mathsf{X}(Z \,\mathsf{U}\, Y)$ **do**
    $v(s) \leftarrow r + rew(s)$
    $X \leftarrow X \cup \{s\}$; $R \leftarrow R \cup \{v(s)\}$
  $R \leftarrow R \setminus \{r\}$
return v

4 Evaluating Qualitative Queries

In this section, we give a strongly polynomial-time algorithm for evaluating *qualitative queries*, i.e. queries where the probability bound p is either 0 or 1. Throughout this section, let $\mathcal{M} = (S, Act, \gamma, \lambda, rew, \delta)$ be an MDP with non-negative rewards. By Proposition 2, we can restrict to queries using one of the quantifiers $\forall\mathsf{P}_{>0}$, $\exists\mathsf{P}_{>0}$, $\forall\mathsf{P}_{=1}$ and $\exists\mathsf{P}_{=1}$. The following lemma allows to give a unified treatment of all cases. (X denotes the next-step operator).

Lemma 4. *The equivalence* $Q\,\mathsf{X}(a \,\mathsf{U}\, (\neg a \wedge \psi)) \equiv Q\,\mathsf{X}(a \,\mathsf{U}\, (\neg a \wedge Q\,\psi))$ *holds in* PRCTL* *for all* $Q \in \{\forall\mathsf{P}_{>0}, \exists\mathsf{P}_{>0}, \forall\mathsf{P}_{=1}, \exists\mathsf{P}_{=1}\}$, $a \in \mathsf{AP}$, *and all path formulas* ψ.

Algorithm 1 is our algorithm for computing the values of a quantile query where we look for an upper bound on the accumulated reward. The algorithm maintains a set X of states, a set R of real numbers, and a table v mapping states to non-negative real numbers or infinity. The algorithm works by discovering states with finite value repeatedly until only the states with infinite value remain. Whenever a new state is discovered, it is put into X and its value is put into R. In the initialisation phase, the algorithm discovers all states labelled with b, which have value 0. In every iteration of the main loop, new states are discovered by picking the least value r that has not been fully processed (i.e. the least element of R) and checking which undiscovered a-labelled states fulfil the PCTL* formula $Q\,\mathsf{X}(Z \,\mathsf{U}\, Y)$, where Y is the set of already discovered states whose value is at most r and Z is the set of states labelled with a but not with b and having reward 0. Any such newly discovered state s must have value $r + rew(s)$, and r can be deleted from R at the end of the current iteration. The termination of the algorithm follows from the fact that in every iteration of the main loop either the set X increases or it remains constant and one element is removed from R.

Lemma 5. *Let* $\mathcal{M}$ *be an MDP,* $\varphi = Q(a \,\mathsf{U}_{\leq ?}\, b)$ *a qualitative query, and let* v *be the result of Algorithm 1 on* $\mathcal{M}$ *and* φ. *Then* $v(s) = \mathrm{val}_\varphi(s)$ *for all states* s.

Proof. We first prove that $s \models \varphi[v(s)]$ for all states s with $v(s) < \infty$. Hence, v is an upper bound on val_φ. We prove this by induction on the number of iterations the while loop has performed before assigning a finite value to $v(s)$. Note that this is the same iteration when s is put into X and that $v(s)$ never changes afterwards. If s is put into X before the first iteration, then $s \models b$ and therefore also $s \models \varphi[0] = \varphi[v(s)]$. Now assume that the while loop has already completed i iterations and is about to add s to X in the current iteration; let X, r and Y be as at the beginning of this iteration (after r and Y have been assigned, but before any new state is added to X). By the induction hypothesis, $t \models \varphi[r]$ for all $t \in Y$. Since s is added to X, we have that $s \models a \wedge Q\,\mathsf{X}(Z\,\mathsf{U}\,Y)$. Using Lemma 4 and some basic PRCTL* laws, we can conclude that $s \models \varphi[v(s)]$ as follows:

$$
\begin{aligned}
& s \models a \wedge Q\,\mathsf{X}(Z\,\mathsf{U}\,Y) \\
\implies\ & s \models a \wedge Q\,\mathsf{X}(Z\,\mathsf{U}\,(\neg Z \wedge Q(a\,\mathsf{U}_{\leq r}\,b))) \\
\implies\ & s \models a \wedge Q\,\mathsf{X}(Z\,\mathsf{U}\,(\neg Z \wedge (a\,\mathsf{U}_{\leq r}\,b))) \\
\implies\ & s \models a \wedge Q\,\mathsf{X}(a\,\mathsf{U}_{\leq r}\,b) \\
\implies\ & s \models Q(a\,\mathsf{U}_{\leq r+rew(s)}\,b) \\
\implies\ & s \models \varphi[v(s)]
\end{aligned}
$$

To complete the proof, we need to show that v is also a lower bound on val_φ. We define a strict partial order $\prec$ on states by setting $s \prec t$ if one of the following conditions holds:

1. $s \models b$ and $t \not\models b$,
2. $\mathrm{val}_\varphi(s) < \mathrm{val}_\varphi(t)$, or
3. $\mathrm{val}_\varphi(s) = \mathrm{val}_\varphi(t)$ and $rew(s) > rew(t)$.

Towards a contradiction, assume that the set C of states s with $\mathrm{val}_\varphi(s) < v(s)$ is non-empty, and pick a state $s \in C$ that is minimal with respect to $\prec$ (in particular, $\mathrm{val}_\varphi(s) < \infty$). Since $s \models \varphi[\infty]$ and the algorithm correctly sets $v(s)$ to 0 if $s \models b$, we know that $s \models a \wedge \neg b$ and $\mathrm{val}_\varphi(s) \geq rew(s)$. Moreover, by Proposition 3, $s \models \varphi[\mathrm{val}_\varphi(s)]$. Let T be the set of all states $t \in S \setminus Z$ such that $\mathrm{val}_\varphi(t) + rew(s) \leq \mathrm{val}_\varphi(s)$, i.e. $t \models \varphi[\mathrm{val}_\varphi(s) - rew(s)]$. Note that $T \neq \emptyset$ (because every state labelled with b is in T) and that $t \prec s$ for all $t \in T$. Since s is a minimal counter-example, we know that $v(t) \leq \mathrm{val}_\varphi(t) < \infty$ for all $t \in T$. Consequently, after some number of iterations of the while loop all elements of T have been added to X and the numbers $v(t)$ have been added to R. Since R is empty upon termination, in a following iteration we have that $r = \max\{v(t) : t \in T\}$ and that $T \subseteq Y$. Let $x \coloneqq \mathrm{val}_\varphi(s) - rew(s)$. Using Lemma 4 and some basic PRCTL* laws, we can conclude that $s \models Q\,\mathsf{X}(Z\,\mathsf{U}\,Y)$ as follows:

$$
\begin{aligned}
& s \models \neg b \wedge \varphi[\mathrm{val}_\varphi(s)] \\
\implies\ & s \models Q(\neg b \wedge (a\,\mathsf{U}_{\leq x+rew(s)}\,b)) \\
\implies\ & s \models Q\,\mathsf{X}(a\,\mathsf{U}_{\leq x}\,b) \\
\implies\ & s \models Q\,\mathsf{X}(Z\,\mathsf{U}\,(\neg Z \wedge (a\,\mathsf{U}_{\leq x}\,b))) \\
\implies\ & s \models Q\,\mathsf{X}(Z\,\mathsf{U}\,(\neg Z \wedge Q(a\,\mathsf{U}_{\leq x}\,b)))
\end{aligned}
$$

$$\implies \quad s \models Q\,\mathsf{X}(Z \,\mathsf{U}\, T)$$
$$\implies \quad s \models Q\,\mathsf{X}(Z \,\mathsf{U}\, Y)$$

Since also $s \models a$, this means that s is added to X no later than in the current iteration. Hence, $v(s) \leq r + rew(s) \leq \mathrm{val}_\varphi(s)$, which contradicts our assumption that $s \in C$. □

Theorem 6. *Qualitative queries of the form $Q(a \,\mathsf{U}_{\leq ?}\, b)$ can be evaluated in strongly polynomial time.*

Proof. By Lemma 5, Algorithm 1 can be used to compute the values of $Q(a \,\mathsf{U}_{\leq ?}\, b)$. During the execution of the algorithm, the running time of one iteration of the while loop is dominated by computing the set of states that fulfil the PCTL* formula $Q\,\mathsf{X}(Z \,\mathsf{U}\, Y)$, which can be done in time $\mathrm{O}(|\delta|)$ for $Q \in \{\forall\mathsf{P}_{>0}, \exists\mathsf{P}_{>0}, \forall\mathsf{P}_{=1}\}$ and in time $\mathrm{O}(|S| \cdot |\delta|)$ for $Q = \exists\mathsf{P}_{=1}$ (see [3, Chapter 10]). In each iteration of the while loop, one element of R is removed, and the number of elements that are put into R in total is bounded by the number of states in the given MDP. Hence, the number of iterations is also bounded by the number of states, and the algorithm runs in time $\mathrm{O}(|S| \cdot |\delta|)$ or $\mathrm{O}(|S|^2 \cdot |\delta|)$, depending on Q. Finally, since the only arithmetic operation used by the algorithm is addition, the algorithm is strongly polynomial. □

Of course, queries of the form $\exists\mathsf{P}_{>0}(a \,\mathsf{U}_{\leq ?}\, b)$ can actually be evaluated in time $\mathrm{O}(|S|^2 + |\delta|)$ using Dijkstra's algorithm since the value of a state with respect to such a query is just the weight of a shortest path from s via a-labeled states to a b-labelled state.

Algorithm 1 also gives us a useful upper bound on the value of a state with respect to a qualitative query.

Proposition 7. *Let $\mathcal{M}$ be an MDP, $\varphi = Q(a \,\mathsf{U}_{\leq ?}\, b)$ a qualitative quantile query, $n = |\lambda^{-1}(a)|$, and $c = \max\{rew(s) : s \in \lambda^{-1}(a)\}$. Then $\mathrm{val}_\varphi(s) \leq nc$ for all states s with $\mathrm{val}_\varphi(s) < \infty$.*

Proof. By induction on the number of iterations Algorithm 1 performs before assigning a finite number to $v(s)$. □

Finally, let us remark that our algorithm can be extended to handle queries of the form $Q(a \,\mathsf{U}_{>?}\, b)$, where a *lower bound* on the accumulated reward is sought. To this end, the initialisation step has to be extended to identify states with value $-\infty$ and the rule for discovering new states has to be modified slightly. We invite the reader to make the necessary modifications and to verify the correctness of the resulting algorithm. This proves that the fragment of PRCTL with probability thresholds 0 and 1 and without reward constraints of the form $= r$ can be model-checked in polynomial time. Previously, a polynomial-time algorithm was only known for the special case where the models are restricted to MDPs in which every loop contains a state with nonzero reward [13].

5 Evaluating Quantitative Queries

In the following, we assume that all state rewards are natural numbers. This does not limit the applicability of our results since any MDP $\mathcal{M}$ with non-negative rational numbers as state rewards can be converted efficiently to an MDP $\mathcal{M}'$ with natural rewards by multiplying all state rewards with the least common multiple K of all denominators occurring in state rewards. It follows that $\mathrm{val}_\varphi^{\mathcal{M}'}(s) = K \cdot \mathrm{val}_\varphi^{\mathcal{M}}(s)$ for any quantile query φ and any state s of $\mathcal{M}$, so in order to evaluate a quantile query on $\mathcal{M}$ we can evaluate it on $\mathcal{M}'$ and divide by K. Throughout this section, we also assume that any transition probability and any probability threshold p occurring in a quantile query is rational. Finally, we define the *size* of an MDP $\mathcal{M} = (S, Act, \gamma, \lambda, rew, \delta)$ to be $|M| := \sum_{s \in S} \|rew(s)\| + \sum_{(s,\alpha,t) \in \delta, \alpha \in \gamma(s)} \|\delta(s,\alpha,t)\|$, where $\|x\|$ denotes the length of the binary representation of x.

5.1 Existential Queries

In order to solve queries of the form $\exists \mathsf{P}_{\geq p}(a \,\mathsf{U}_{\leq ?}\, b)$ or $\exists \mathsf{P}_{>p}(a \,\mathsf{U}_{\leq ?}\, b)$, we first show how to compute the *maximal* probabilities for fulfilling the path formula $a \,\mathsf{U}_{\leq r}\, b$ when we are given the reward bound r. Given an MDP $\mathcal{M}$, $a, b \in \mathsf{AP}$ and $r \in \mathbb{N}$, consider the following linear program over the variables $x_{s,i}$ for $s \in S$ and $i \in \{0, 1, \ldots, r\}$:

$$
\begin{array}{ll}
\text{Minimise } \sum x_{s,i} \text{ subject to} & \\
x_{s,i} \geq 0 & \text{for all } s \in S \text{ and } i \leq r, \\
x_{s,i} = 1 & \text{for all } s \in \lambda^{-1}(b) \text{ and } i \leq r, \\
x_{s,i} \geq \sum_{t \in S} \delta(s,\alpha,t) \cdot x_{t,i-rew(s)} & \\
& \text{for all } s \in \lambda^{-1}(a),\ \alpha \in Act \text{ and } rew(s) \leq i \leq r.
\end{array}
$$

This linear program is of size $r \cdot |\mathcal{M}|$, and it can be shown that setting $x_{i,s}$ to $\max_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq i}\, b)$ yields the optimal solution. Hence, we can compute the numbers $\max_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq i}\, b)$ in time $\mathsf{poly}(r \cdot |\mathcal{M}|)$.

Our algorithm for computing the value of a state s wrt. a query of the form $\exists \mathsf{P}_{>p}(a \,\mathsf{U}_{\leq ?}\, b)$ just computes the numbers $\max_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq i}\, b)$ for increasing i and stops as soon as this probability exceeds p. However, in order to make this algorithm work and to show that it does not take too much time, we need a bound on the value of s provided this value is not infinite. Such a bound can be derived from the following lemma, which resembles a result by Hansen et al., who gave a bound on the convergence rate of *value iteration* in *concurrent reachability games* [11]. Our proof is technically more involved though, since we have to deal with paths that from some point onwards do not earn any more rewards.

Lemma 8. *Let $\mathcal{M}$ be an MDP where the denominator of each transition probability is at most m, and let $n = |\lambda^{-1}(a)|$, $c = \max\{rew(s) : s \in \lambda^{-1}(a)\}$ and $r = kncm^{-n}$ for some $k \in \mathbb{N}^+$. Then $\max_\sigma \Pr_s^\sigma(a \,\mathsf{U}\, b) < \max_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq r}\, b) + \mathrm{e}^{-k}$ for all $s \in S$.*

Proof. Without loss of generality, assume that all b-labelled states are absorbing. Let us call a state s of $\mathcal{M}$ *dead* if $s \models \forall \mathsf{P}_{=0}(a \,\mathsf{U}\, b)$, and denote by D the set of dead states. Note that $s \in D$ for all states s with $s \models \neg a \wedge \neg b$. Finally, let τ be a memoryless scheduler such that $\Pr_s^\tau(a \,\mathsf{U}\, b) = \max_\sigma \Pr_s^\sigma(a \,\mathsf{U}\, b)$ for all states s, and denote by Z the set of all states s with $s \models a \wedge \neg b$ and $\mathit{rew}(s) = 0$. By the definition of D and Z, we have that $\Pr_s^\tau(a \,\mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z) \wedge a \,\mathsf{U}\, b) = 0$ for all $s \in S$. Moreover, if s is not dead, then there must be a simple path from s to a b-labelled state via a-labelled states in the Markov chain induced by τ. Since any a-labelled state has reward at most c, this implies that $\Pr_s^\tau(a \,\mathsf{U}_{\leq nc}\, b) \geq m^{-n}$ for all non-dead states s. Now let ψ be the path formula $b \vee D \vee \mathsf{G}Z$. We claim that $\Pr_s^\tau(\neg(a \,\mathsf{U}_{\leq r}\, \psi)) < \mathrm{e}^{-k}$ for all states s. To prove this, let $s \in S$. We first show that $\Pr_s^\tau(a \mathsf{U}_{\leq i+nc} \psi \mid \neg(a \mathsf{U}_{\leq i} \psi)) \geq m^{-n}$ for all $i \in \mathbb{N}$ with $\Pr_s^\tau(a \mathsf{U}_{\leq i} \psi) < 1$. Let X be the set of sequences $xt \in S^* \cdot S$ such that $xt \in \{s \in S \setminus D : s \models a \wedge \neg b\}^*$, $\mathit{rew}(x) \leq i$ and $\mathit{rew}(xt) > i$. It is easy to see that the set $\{xt \cdot S^\omega : xt \in X\}$ is a partition of the set of infinite sequences over S that violate $a \,\mathsf{U}_{\leq i}\, \psi$. Using the fact that τ is memoryless, we can conclude that

$$
\begin{aligned}
& \Pr_s^\tau(a \,\mathsf{U}_{\leq i+nc}\, \psi \mid \neg(a \,\mathsf{U}_{\leq i}\, \psi)) \\
\geq\ & \Pr_s^\tau(a \,\mathsf{U}_{\leq i+nc}\, b \mid \neg(a \,\mathsf{U}_{\leq i}\, \psi)) \\
=\ & \Pr_s^\tau(a \,\mathsf{U}_{\leq i+nc}\, b \cap X \cdot S^\omega) / \Pr_s^\tau(X \cdot S^\omega) \\
=\ & \sum_{xt \in X} \Pr_s^\tau(a \,\mathsf{U}_{\leq i+nc}\, b \cap xt \cdot S^\omega) / \Pr_s^\tau(X \cdot S^\omega) \\
=\ & \sum_{xt \in X} \Pr_t^\tau(a \,\mathsf{U}_{\leq i-\mathit{rew}(x)+nc}\, b) \cdot \Pr_s^\tau(xt \cdot S^\omega) / \Pr_s^\tau(X \cdot S^\omega) \\
\geq\ & \sum_{xt \in X} \Pr_t^\tau(a \,\mathsf{U}_{\leq nc}\, b) \cdot \Pr_s^\tau(xt \cdot S^\omega) / \Pr_s^\tau(X \cdot S^\omega) \\
\geq\ & \sum_{xt \in X} m^{-n} \cdot \Pr_s^\tau(xt \cdot S^\omega) / \Pr_s^\tau(X \cdot S^\omega) \\
=\ & m^{-n}.
\end{aligned}
$$

Now, applying this inequality successively, we get that $\Pr_s^\tau(\neg(a \,\mathsf{U}_{\leq r}\, \psi)) \leq (1 - m^{-n})^{\frac{r}{nc}} = (1 - m^{-n})^{km^n} < \mathrm{e}^{-k}$. Finally,

$$
\begin{aligned}
\Pr_s^\tau(a \,\mathsf{U}\, b) &= \Pr_s^\tau(a \,\mathsf{U}\, b \wedge \neg(a \,\mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z))) \\
&\leq \Pr_s^\tau(\neg(a \,\mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z))) \\
&\leq \Pr_s^\tau(\neg(a \,\mathsf{U}_{\leq r}\, \psi) \vee (a \,\mathsf{U}_{\leq r}\, b)) \\
&\leq \Pr_s^\tau(\neg(a \,\mathsf{U}_{\leq r}\, \psi)) + \Pr_s^\tau(a \,\mathsf{U}_{\leq r}\, b) \\
&< \mathrm{e}^{-k} + \max_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq r}\, b)
\end{aligned}
$$

for all $s \in S$. Since $\Pr_s^\tau(a \,\mathsf{U}\, b) = \max_\sigma \Pr_s^\sigma(a \,\mathsf{U}\, b)$, this inequality proves the lemma. □

Given an MDP $\mathcal{M}$ and $a, b \in \mathsf{AP}$, we denote by $\tilde{\mathcal{M}}$ the MDP that arises from $\mathcal{M}$ by performing the following transformation:

1. In each state s, remove all actions α with $\sum_{t \in S} \delta(s, \alpha, t) \cdot \max_\sigma \Pr_t^\sigma(a \,\mathsf{U}\, b) < \max_\sigma \Pr_s^\sigma(a \,\mathsf{U}\, b)$ from the set $\gamma(s)$ of enabled actions.
2. Label all states s such that $s \models \mathsf{P}_{=0}(a \,\mathsf{U}\, b)$ with b.

The following lemma, whose proof is rather technical, allows us to reduce the query $\exists \mathsf{P}_{\geq p}(a \,\mathsf{U}_{\leq ?}\, b)$ to the qualitative query $\exists \mathsf{P}_{=1}(a \,\mathsf{U}_{\leq ?}\, b)$ in the special case that p equals the optimal probability of fulfilling $a \,\mathsf{U}\, b$.

Lemma 9. *Let $\mathcal{M}$ be an MDP, $\varphi = \exists \mathsf{P}_{\geq p}(a \,\mathsf{U}_{\leq ?}\, b)$ and $\tilde{\varphi} = \exists \mathsf{P}_{=1}(a \,\mathsf{U}_{\leq ?}\, b)$. Then $\mathrm{val}_\varphi^{\mathcal{M}}(s) = \mathrm{val}_{\tilde{\varphi}}^{\tilde{\mathcal{M}}}(s)$ for all states s of $\mathcal{M}$ with $p = \max_\sigma \Pr_s^\sigma(a \,\mathsf{U}\, b)$.*

With the help of Lemmas 8 and 9, we can devise an upper bound for the value of any query whose value is finite.

Lemma 10. *Let $\mathcal{M}$ be an MDP where the denominator of each transition probability is at most m, $\varphi = \exists \mathsf{P}_{\rhd p}(a \,\mathsf{U}_{\leq ?}\, b)$ for $\rhd \in \{\geq, >\}$, $n = |\lambda^{-1}(a)|$, $c = \max\{rew(s) : s \in \lambda^{-1}(a)\}$, $s \in S$, and $q = \max_\sigma \Pr_s^\sigma(a \,\mathsf{U}\, b)$. Then at least one of the following statements holds:*

1. *$p \geq q$ and $\mathrm{val}_\varphi(s) = \infty$.*
2. *$p = q$, $\rhd = \geq$ and $\mathrm{val}_\varphi(s) \leq nc$.*
3. *$p < q$ and $\mathrm{val}_\varphi(s) \leq kncm^n$, where $k = \max\{-\lfloor \ln(q-p) \rfloor, 1\}$.*

Proof. Clearly, if either $\rhd = >$ and $p \geq q$ or $\rhd = \geq$ and $p > q$, then $\mathrm{val}_\varphi(s) = \infty$, and 1. holds. Now assume that $p = q$ and $\rhd = \geq$. By Lemma 9, we have that $\mathrm{val}_\varphi^{\mathcal{M}}(s) = \mathrm{val}_{\tilde{\varphi}}^{\tilde{\mathcal{M}}}(s)$. Hence, if $\mathrm{val}_{\tilde{\varphi}}^{\tilde{\mathcal{M}}}(s) = \infty$, then 1. holds. On the other hand, if $\mathrm{val}_{\tilde{\varphi}}^{\tilde{\mathcal{M}}}(s) < \infty$, then Proposition 7 gives us that $\mathrm{val}_{\tilde{\varphi}}^{\tilde{\mathcal{M}}}(s) \leq nc$, and 2. holds. Finally, if $p < q$, then let $r := kncm^n$. By Lemma 8, we have that $\max_\sigma \Pr_s(a \,\mathsf{U}_{\leq r}\, b) > q - e^{-k} \geq q - e^{\lfloor \ln(q-p) \rfloor} \geq q - (q-p) = p$, i.e. $s \models \exists \mathsf{P}_{\rhd p}(a \,\mathsf{U}_{\leq r}\, b)$. Hence, $\mathrm{val}_\varphi(s) \leq r$, and 3. holds. □

It follows from Lemma 10 that we can compute the value of a state s wrt. a query φ of the form $\exists \mathsf{P}_{>p}(a \,\mathsf{U}_{\leq ?}\, b)$ as follows: First compute the maximal probability q of fulfilling $a \,\mathsf{U}\, b$ from s, which can be done in polynomial time. If $p \geq q$, we know that the value of s wrt. φ must be infinite. Otherwise, $\mathrm{val}_\varphi(s) \leq r := kncm^n$, where $k = \max\{-\lfloor \ln(q-p) \rfloor, 1\}$, and we can find the least i such that $\max_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq i}\, b) > p$ by computing $\max_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq i}\, b)$ for all $i \in \{0, 1, \ldots, r\}$, which can be done in time $\mathsf{poly}(r \cdot |\mathcal{M}|)$. Since r is exponential in the number of states of the given MDP $\mathcal{M}$, the running time of this algorithm is exponential in the size of $\mathcal{M}$. If φ is of the form $\exists \mathsf{P}_{\geq p}(a \,\mathsf{U}_{\leq ?}\, b)$, the algorithm is similar, but in the case that $p = q$, we compute $\max_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq i}\, b)$ for all $i \in \{0, 1, \ldots, nc\}$ in order to determine whether the value is infinite or one of these numbers i.

Theorem 11. *Queries of the form $\exists \mathsf{P}_{\geq p}(a \,\mathsf{U}_{\leq ?}\, b)$ or $\exists \mathsf{P}_{>p}(a \,\mathsf{U}_{\leq ?}\, b)$ can be evaluated in exponential time.*

5.2 Universal Queries

In order to solve queries of the form $\forall \mathsf{P}_{>p}(a \,\mathsf{U}_{\leq ?}\, b)$, we first show how to compute the *minimal* probabilities for fulfilling the path formula $a \,\mathsf{U}_{\leq r}\, b$ when we are given the reward bound r. Given an MDP $\mathcal{M}$, $a, b \in \mathsf{AP}$ and $r \in \mathbb{N}$, consider the following linear program over the variables $x_{s,i}$ for $s \in S$ and $i \in \{0, 1, \ldots, r\}$:

$$
\begin{array}{ll}
\text{Maximise } \sum x_{s,i} \text{ subject to} & \\
x_{s,i} \leq 1 & \text{for all } s \in S \text{ and } i \leq r, \\
x_{s,i} = 0 & \text{for all } s \in S \text{ with } s \not\models \forall \mathsf{P}_{>0}(a \,\mathsf{U}_{\leq i}\, b) \text{ and } i \leq r, \\
x_{s,i} \leq \sum_{t \in S} \delta(s, \alpha, t) \cdot x_{t, i - rew(s)} & \\
& \text{for all } s \in S \setminus \lambda^{-1}(b),\ \alpha \in Act \text{ and } rew(s) \leq i \leq r.
\end{array}
$$

This program is of size $r \cdot |\mathcal{M}|$, and it can be shown that setting $x_{i,s}$ to $\min_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq i}\, b)$ yields the optimal solution. Since the set of states s with $s \models \forall \mathsf{P}_{>0}(a \,\mathsf{U}_{\leq i}\, b)$ can be computed in polynomial time (Theorem 6), this means that we can compute the numbers $\min_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq i}\, b)$ in time $\mathsf{poly}(r \cdot |\mathcal{M}|)$. The following lemma is the analogue of Lemma 8 for minimal probabilities.

Lemma 12. *Let $\mathcal{M}$ be an MDP where the denominator of each transition probability is at most m, and let $n = |\lambda^{-1}(a)|$, $c = \max\{rew(s) : s \in \lambda^{-1}(a)\}$ and $r = kncm^{-n}$ for some $k \in \mathbb{N}^+$. Then $\min_\sigma \Pr_s^\sigma(a \,\mathsf{U}\, b) < \min_\sigma \Pr_s^\sigma(a \,\mathsf{U}_{\leq r}\, b) + \mathrm{e}^{-k}$ for all $s \in S$.*

Proof. Without loss of generality, assume that all b-labelled states are absorbing. Let us call a state s of $\mathcal{M}$ *dull* if $s \models \exists \mathsf{P}_{=0}(a \,\mathsf{U}\, b)$, and denote by D the set of dull states. Note that $s \in D$ for all states s with $s \models \neg a \wedge \neg b$. If s is not dull, then it is easy to see that, for any scheduler σ, the probability of reaching a b-labelled state from s in *at most n steps* (while seeing only a-labelled states before reaching a b-labelled state) is at least m^{-n}. Since any a-labelled state has reward at most c, we get that $\Pr_s^\sigma(a \,\mathsf{U}_{\leq nc}\, b) \geq m^{-n}$ for all non-dull states s and all schedulers σ. In the following, denote by Z the set $\{s \in S : s \models a \wedge \neg b \text{ and } rew(s) = 0\}$, and let ψ be the path formula $b \vee D \vee \mathsf{G}Z$. In the same way as in the proof of Lemma 8, we can infer that $\Pr_s^\sigma(\neg(a \,\mathsf{U}_{\leq r}\, \psi)) < \mathrm{e}^{-k}$ for all states s and all schedulers σ. Now fix a scheduler τ that minimises $\Pr_s^\tau(a \,\mathsf{U}_{\leq r}\, b)$ for all $s \in S$ and a scheduler σ such that $\Pr_s^\sigma(a \,\mathsf{U}\, b) = 0$ for all $s \in D$. From τ and σ, we devise another scheduler τ^* by setting

$$
\tau^*(x) = \begin{cases} \tau(x) & \text{if } x \in (S \setminus D)^*, \\ \sigma(x_2) & \text{if } x = x_1 \cdot x_2 \text{ where } x_1 \in (S \setminus D)^* \text{ and } x_2 \in D \cdot S^*. \end{cases}
$$

Note that $\Pr_s^{\tau^*}(a \,\mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z) \wedge a \,\mathsf{U}\, b) = 0$ and $\Pr_s^{\tau^*}(a \,\mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z)) = \Pr_s^\tau(a \,\mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z))$ for all $s \in S$. Hence,

$$\begin{aligned}
\Pr_s^{\tau^*}(a \mathsf{U}\, b) &= \Pr_s^{\tau^*}(a \mathsf{U}\, b \wedge \neg(a \mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z))) \\
&\leq \Pr_s^{\tau^*}(\neg(a \mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z))) \\
&= \Pr_s^{\tau}(\neg(a \mathsf{U}_{\leq r}\, (D \vee \mathsf{G}Z))) \\
&\leq \Pr_s^{\tau}(\neg(a \mathsf{U}_{\leq r}\, \psi) \vee (a \mathsf{U}_{\leq r}\, b)) \\
&\leq \Pr_s^{\tau}(\neg(a \mathsf{U}_{\leq r}\, \psi)) + \Pr_s^{\tau}(a \mathsf{U}_{\leq r}\, b) \\
&< e^{-k} + \Pr_s^{\tau}(a \mathsf{U}_{\leq r}\, b) \\
&= e^{-k} + \min_\sigma \Pr_s^{\sigma}(a \mathsf{U}_{\leq r}\, b)
\end{aligned}$$

for all $s \in S$. Since $\min_\sigma \Pr_s^\sigma(a \mathsf{U}\, b) \leq \Pr_s^{\tau^*}(a \mathsf{U}\, b)$, this inequality proves the lemma. □

With the help of Lemma 12, we can devise an upper bound for the value of a query of the form $\forall \mathsf{P}_{>p}(a \mathsf{U}_{\leq ?}\, b)$ in case this value is finite.

Lemma 13. *Let $\mathcal{M}$ be an MDP where the denominator of each transition probability is $\leq m$, $\varphi = \forall \mathsf{P}_{>p}(a \mathsf{U}_{\leq ?}\, b)$, $n = |\lambda^{-1}(a)|$, $c = \max\{rew(s) : s \in \lambda^{-1}(a)\}$, $s \in S$, and $q = \min_\sigma \Pr_s^\sigma(a \mathsf{U}\, b)$. Then one of the following statements holds:*

1. *$p \geq q$ and $\mathrm{val}_\varphi(s) = \infty$.*
2. *$p < q$ and $\mathrm{val}_\varphi(s) \leq kncm^n$, where $k = \max\{-\lfloor \ln(q-p) \rfloor, 1\}$.*

Proof. Clearly, if $p \geq q$, then $\mathrm{val}_\varphi(s) = \infty$, and 1. holds. On the other hand, if $p < q$, then let $r \coloneqq kncm^n$. By Lemma 12, we have that $\min_\sigma \Pr_s(a \mathsf{U}_{\leq r}\, b) > q - e^{-k} \geq q - e^{\lfloor \ln(q-p) \rfloor} \geq q - (q-p) = p$, i.e. $s \models \forall \mathsf{P}_{>p}(a \mathsf{U}_{\leq r}\, b)$. Hence, $\mathrm{val}_\varphi(s) \leq r$, and 3. holds. □

As in the last section, Lemma 13 can be used to derive an exponential algorithm for computing the value of a state wrt. a query of the form $\forall \mathsf{P}_{>p}(a \mathsf{U}_{\leq ?}\, b)$.

Theorem 14. *Queries of the form $\forall \mathsf{P}_{>p}(a \mathsf{U}_{\leq ?} b)$ can be evaluated in exponential time.*

Regarding queries of the form $\forall \mathsf{P}_{\geq p}(a \mathsf{U}_{\leq ?}\, b)$, we can compute the value of a state s whenever the probability $\min_\sigma \Pr_s^\sigma(a \mathsf{U}\, b)$ differs from p using the same algorithm. However, in the case that $p = \min_\sigma \Pr_s^\sigma(a \mathsf{U}\, b)$ it is not clear how to bound the value of s. As the following example shows, the analogous bound of nc for existential queries from Lemma 10 does not apply in this case.

Example 15. Consider the MDP depicted in Fig. 1, where $Act = \{\flat, \natural\}$ and $q \in [0, 1[$ is an arbitrary probability. A state's reward is depicted in its bottom half, and a transition from s to t labelled with α, p indicates that $\delta(s, \alpha, t) = p$. Only transitions from non-absorbing states with nonzero probability and corresponding to enabled actions are shown. Assuming that every state is labelled with a but only s_3 and s_5 are labelled with b, it is easy to see that $\min_\sigma \Pr_{s_0}^\sigma(a \mathsf{U} b) = \frac{1}{2}$. Moreover, a quick calculation reveals that the value of state s_0 with respect to the query $\forall \mathsf{P}_{\geq 1/2}(a \mathsf{U}_{\leq ?}\, b)$ equals $-\lfloor 1/\log_2 q \rfloor$. Since q can be chosen arbitrarily close to 1, this value can be made arbitrarily high.

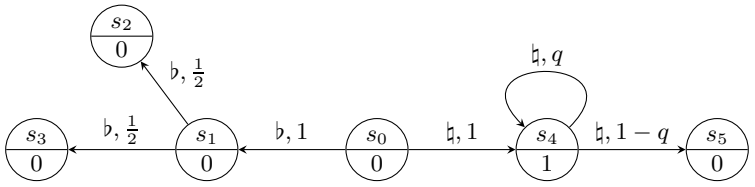

Fig. 1. An MDP with nonnegative rewards

5.3 A Pseudo-polynomial Algorithm for Markov Chains

In this section, we give a *pseudo-polynomial* algorithm for evaluating quantile queries of the form $\mathsf{P}_{\rhd p}(a \,\mathsf{U}_{\leq ?}\, b)$ on Markov chains. (Note that the quantifiers $\exists\mathsf{P}$ and $\forall\mathsf{P}$ coincide for Markov chains.) More precisely, our algorithm runs in time $\mathsf{poly}(c \cdot |\mathcal{M}| \cdot \|p\|)$ if c is the largest reward in $\mathcal{M}$. As an important special case, our algorithm runs in polynomial time on Markov chains where each state has reward 0 or 1.

Our polynomial-time algorithm relies on the following equations for computing the probability of the event $a \,\mathsf{U}_{=i}\, b$ in a Markov chain with rewards 0 and 1. Given such a Markov chain $\mathcal{M}$ and $a \in \mathsf{AP}$, we denote by Z the set of states s such that $\mathit{rew}(s) = 0$ and $s \models a \wedge \neg b$. Then the following equations hold for all $s \in S$, $a, b \in \mathsf{AP}$ and $r \in \mathbb{N}$:

- $\Pr_s(a \,\mathsf{U}_{=0}\, b) = \Pr_s(Z \,\mathsf{U}\, b)$,
- $\Pr_s(a \,\mathsf{U}_{=2r}\, b) = \sum_{t \in S \setminus Z} \Pr_s(a \,\mathsf{U}_{=r}\, \{t\}) \cdot \Pr_t(a \,\mathsf{U}_{=r}\, b)$,
- $\Pr_s(a \,\mathsf{U}_{=2r+1}\, b) = \sum_{t \in \lambda^{-1}(a) \setminus Z} \sum_{u \in S} \Pr_s(a \,\mathsf{U}_{=r}\, \{t\}) \cdot \delta(t,u) \cdot \Pr_u(a \,\mathsf{U}_{=r}\, b)$,

Using these equations, we can compute the numbers $\Pr_s(a \mathsf{U}_{=r} b)$ along the binary representation of r in time $\mathrm{O}(\mathsf{poly}(|\mathcal{M}|) \cdot \log r)$ for Markov chains with rewards 0 and 1 (see also [12]). Since any Markov chain $\mathcal{M}$ with rewards $0, 1, \ldots, c$ can easily be transformed into an equivalent Markov chain of size $c \cdot |\mathcal{M}|$ with rewards 0 and 1, the same numbers can be computed in time $\mathrm{O}(\mathsf{poly}(c \cdot |\mathcal{M}|) \cdot \log r)$ for general Markov chains. Finally, we can compute the numbers $\Pr_s(a \,\mathsf{U}_{\leq r}\, b)$ in the same time by first applying the following operations to each b-labelled state s: Make s absorbing, add a to $\lambda(s)$, and set $\mathit{rew}(s) = 1$; in the resulting Markov chain each state s fulfils $\Pr_s(a \,\mathsf{U}_{\leq r}\, b) = \Pr_s(a \,\mathsf{U}_{=r}\, b)$.

Now let $\varphi = \mathsf{P}_{\rhd p}(a \,\mathsf{U}_{\leq ?}\, b)$. Our algorithm for evaluating φ at state s of a Markov chain $\mathcal{M}$ is essentially the same algorithm as for MDPs. Hence, we first compute the probability $q := \Pr_s(a \,\mathsf{U}\, b)$. If either $p > q$ or $p = q$ and $\rhd = >$, then $\mathrm{val}_\varphi(s) = \infty$, by Lemma 10. If $p < q$, then the same lemma entails that $\mathrm{val}_\varphi(s) \leq r := kncm^n$, where $n = |\lambda^{-1}(a)|$, m is the least denominator of any transition probability, and $k = \max\{-\lfloor \ln(q-p) \rfloor, 1\} \leq \mathsf{poly}(\mathcal{M}) + \|p\|$. Hence, we can determine $\mathrm{val}_\varphi(s)$ using an ordinary binary search in time $\mathrm{O}(\mathsf{poly}(c \cdot |\mathcal{M}|) \cdot \log^2 r) = \mathrm{O}(\mathsf{poly}(c \cdot |\mathcal{M}| \cdot \|p\|))$. Finally, the same method can be applied if $p = q$ and $\rhd = \geq$ since Lemma 10 tells us that $\mathrm{val}_\varphi(s) \leq nc$ in this case.

Theorem 16. *Queries of the form* $\mathsf{P}_{\geq p}(a \mathsf{U}_{\leq ?} b)$ *or* $\mathsf{P}_{>p}(a \mathsf{U}_{\leq ?} b)$ *can be evaluated in pseudo-polynomial time on Markov chains.*

6 Conclusions

Although many researchers presented algorithms and several sophisticated techniques for the PCTL model checking problem and to solve PCTL and PRCTL queries, the class of quantile-based queries has not yet been addressed in the model checking community. In this paper, we presented algorithms for qualitative and quantitative quantile queries of the form $\mathsf{P}_{\bowtie p}(a\,\mathsf{U}_{\leq ?}\,b)$ and their duals $\exists\mathsf{P}_{\bowtie p}(a\,\mathsf{U}_{\leq ?}\,b)$. We established a polynomial algorithms for the qualitative case and exponential algorithms for all but one of the quantitative cases. Although the algorithms for the quantitative cases rely on a simple search algorithm for the quantile, the crucial feature is the bound we presented in Lemmas 8 and 12. These bounds might be interesting also for other purposes. There are several open problems to be studied in future work. First, the precise complexity of quantitative quantile queries is unknown and more efficient algorithms might exist, despite the NP-hardness shown in [14]. Second, we concentrated here on reward-bounded until properties, and by duality our results also apply to reward-bounded release properties. But quantile queries can also be derived from other PCTL-like formulas, such as formulas reasoning about expected rewards, e.g. in combination with step bounds.

Acknowledgments. We would like to thank Manuela Berg, Joachim Klein, Sascha Klüppelholz and Dominik Wojtczak for helpful discussions and the anonymous reviewers for their valuable remarks and suggestions.

References

1. Andova, S., Hermanns, H., Katoen, J.-P.: Discrete-Time Rewards Model-Checked. In: Larsen, K.G., Niebert, P. (eds.) FORMATS 2003. LNCS, vol. 2791, pp. 88–104. Springer, Heidelberg (2004)
2. Baier, C., Daum, M., Engel, B., Härtig, H., Klein, J., Klüppelholz, S., Märcker, S., Tews, H., Völp, M.: Waiting for Locks: How Long Does It Usually Take? In: Stoelinga, M., Pinger, R. (eds.) FMICS 2012. LNCS, vol. 7437, pp. 47–62. Springer, Heidelberg (2012)
3. Baier, C., Katoen, J.-P.: Principles of Model Checking. MIT Press (2008)
4. Bianco, A., de Alfaro, L.: Model Checking of Probabilistic and Nondeterministic Systems. In: Thiagarajan, P.S. (ed.) FSTTCS 1995. LNCS, vol. 1026, pp. 499–513. Springer, Heidelberg (1995)
5. Courcoubetis, C.A., Yannakakis, M.: The complexity of probabilistic verification. Journal of the ACM 42(4), 857–907 (1995)
6. de Alfaro, L.: Formal Verification of Probabilistic Systems. PhD thesis, Stanford University (1997)
7. de Alfaro, L.: Temporal Logics for the Specification of Performance and Reliability. In: Reischuk, R., Morvan, M. (eds.) STACS 1997. LNCS, vol. 1200, pp. 165–176. Springer, Heidelberg (1997)
8. de Alfaro, L.: How to specify and verify the long-run average behavior of probabilistic systems. In: Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science, LICS, pp. 454–465. IEEE Press (1998)

9. de Alfaro, L.: Computing Minimum and Maximum Reachability Times in Probabilistic Systems. In: Baeten, J.C.M., Mauw, S. (eds.) CONCUR 1999. LNCS, vol. 1664, pp. 66–81. Springer, Heidelberg (1999)
10. Forejt, V., Kwiatkowska, M., Norman, G., Parker, D.: Automated Verification Techniques for Probabilistic Systems. In: Bernardo, M., Issarny, V. (eds.) SFM 2011. LNCS, vol. 6659, pp. 53–113. Springer, Heidelberg (2011)
11. Hansen, K.A., Ibsen-Jensen, R., Miltersen, P.B.: The Complexity of Solving Reachability Games Using Value and Strategy Iteration. In: Kulikov, A., Vereshchagin, N. (eds.) CSR 2011. LNCS, vol. 6651, pp. 77–90. Springer, Heidelberg (2011)
12. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing 6(5), 512–535 (1994)
13. Jurdziński, M., Sproston, J., Laroussinie, F.: Model checking probabilistic timed automata with one or two clocks. Logical Methods in Computer Science 4(3) (2008)
14. Laroussinie, F., Sproston, J.: Model Checking Durational Probabilistic Systems. In: Sassone, V. (ed.) FOSSACS 2005. LNCS, vol. 3441, pp. 140–154. Springer, Heidelberg (2005)
15. Pekergin, N., Younès, S.: Stochastic Model Checking with Stochastic Comparison. In: Bravetti, M., Kloul, L., Zavattaro, G. (eds.) EPEW/WS-FM 2005. LNCS, vol. 3670, pp. 109–123. Springer, Heidelberg (2005)
16. Vardi, M.: Automatic verification of probabilistic concurrent finite-state programs. In: Proceedings of the 26th IEEE Symposium on Foundations of Computer Science, FOCS, pp. 327–338. IEEE Press (1985)