On the Controller Synthesis for Finite-State Markov Decision Processes

Antonín Kučera* and Oldřich Stražovský**

Faculty of Informatics, Masaryk University,
Botanická 68a, 60200 Brno, Czech Republic.
{kucera,strazovsky}@fi.muni.cz

Abstract. We study the problem of effective controller synthesis for finite-state Markov decision processes (MDPs) and the class of properties definable in the logic PCTL extended with long-run average propositions. We show that the existence of such a controller is decidable, and we give an algorithm which computes the controller if it exists. We also address the issue of "controller robustness", i.e., the problem whether there is a controller which still guarantees the satisfaction of a given property when the probabilities in the considered MDP slightly deviate from their original values. From a practical point of view, this is an important aspect since the probabilities are often determined empirically and hence they are inherently imprecise. We show that the existence of robust controllers is also decidable, and that such controllers are effectively computable if they exist.

1 Introduction

The controller synthesis problem is one of the fundamental research topics in the area of system design. Loosely speaking, the task is to modify or limit some parts of a given system so that a given property is satisfied. The controller synthesis problem is well understood for discrete systems [11], and the scope of this study has recently been extended also to timed systems [2, 5] and probabilistic systems [1].

In this paper, we concentrate on a class of probabilistic systems that can be modelled by finite-state Markov decision processes. Intuitively, Markov decision processes (MDPs) are finite-state systems where each state has several outgoing transitions leading to probability distributions over states. Thus, Markov decision processes combine the paradigms of non-deterministic/probabilistic choice, and this combination turns out to be very useful in system modelling. Quantitative properties of MDPs can be defined only after resolving nondeterminism by assigning probabilities to the individual transitions. Similarly as in [1], we distinguish among four natural types of strategies for resolving nondeterminism, depending on whether

* Supported by the research center Institute for Theoretical Computer Science (ITI), project No. 1M0021620808.
** Supported by the Czech Science Foundation, grant No. 201/03/1161.

– the transition is chosen deterministically (D) or randomly (R);
– the choice does or does not depend on the sequence of previously visited states (Markovian (M) and history-dependent (H) strategies, respectively).

Thus, one obtains the four basic classes of MD, HD, MR, and HR strategies. In addition, we assume that the states of a given MDP are split into two disjoint subsets of *controllable* and *environmental* states, depending on whether the nondeterminism is resolved by a controller or by the environment, respectively. Hence, in our setting the controller synthesis problem is specified by choosing the type of strategy for controller and environment, and the class of properties that are to be achieved. The task is to find, for a given MDP and a given property, a controller strategy such that the property is satisfied for every strategy of the environment. In [1], it was shown that this problem is **NP**-complete for MD strategies and PCTL properties, and elementary for HD strategies and LTL properties.

For linear-time properties, the problem of finding a suitable controller strategy can also be formulated in the terms of stochastic games on graphs [12]. Controller and environment act as two players who resolve the non-deterministic choice in controllable and environmental states, resp., and thus produce a "play". The winning conditions are defined as certain properties of the produced play. In many cases, it turns out that the optimal strategies for both players are memoryless (i.e., Markovian in our terms). However, in the case of branching-time properties that are considered in this paper, optimal strategies are not necessarily memoryless and the four types of strategies mentioned above form a strict hierarchy [1].

Our contribution: In this paper we consider the controller synthesis problem for MR strategies and the class of properties definable in the logic PCTL extended with long-run average propositions defined in the style of [4]. The resulting logic is denoted PCTL+LAP. The long-run average propositions allow to specify long-run average properties such as the average service time, the average frequency of visits to a distinguished subset of states, etc. In the logic PCTL+LAP, one can express properties such as:

– the probability that the average service time for a request does not exceed 20 seconds is at least 98%;
– the system terminates with probability at least 80%, and at least 98% of runs have the property that the percentage of time spent in "dangerous" states does not exceed 3%.

A practical relevance of PCTL+LAP properties is obvious.

The controller synthesis problem for PCTL+LAP properties and MD strategies is trivially reducible to the satisfaction problem for finite-state Markov chains and PCTL+LAP properties. This is because there are only finitely many MD strategies for a given MDP, and hence one can try out all possibilities. For MR strategies, a more sophisticated approach is required because the total number of MR strategies is infinite (and in fact not countable). This is overcome by encoding the existence of a MR-controller in $(\mathbb{R}, +, *, \leq)$, the first-order theory

of reals, which is known to be decidable [10]. The encoding is not simple and includes several subtle tricks. Nevertheless, the size of the resulting formula is polynomial in the size of a given MDP and a given PCTL+LAP property, and the number of quantifier alternations is fixed. Hence, we obtain the **EXPTIME** upper complexity bound by applying the result of [6].

Another problem addressed in this paper is controller robustness [8]. Since the probabilities of events that are modelled in MDPs are often evaluated empirically, they are inherently imprecise. Hence, it is important to know whether the constructed controller still works if the probabilities in the considered MDP slightly deviate from their original values. We say that a controller is *ε-robust* if the property in question is still satisfied when probability distributions in the considered MDP change at most by ε in each component (here we do not allow for changing the probabilities from zero to non-zero (and vice versa), because this corresponds to changing from "impossible" to "possible"). Similarly, we can also wonder whether the constructed controller is "fragile" in the sense that it stops working if the computed strategy changes a little bit. We say that a controller is *δ-free* if every other controller obtained by changing the strategy by at most δ is again a correct controller. We show that the problem whether there is an ε-robust and δ-free controller for given MDP, PCTL+LAP property, and $\varepsilon, \delta \geq 0$, is in **EXPTIME**. Moreover, we also give an algorithm which effectively estimates the maximal achievable level of controller robustness for given MDP and PCTL+LAP property (i.e., we show how to compute the maximal ε, up to a given precision, such that there is an ε-robust controller for given MDP and PCTL+LAP property). Finally, we show how to construct an ε-robust controller for a given MDP and PCTL+LAP property, provided that an ε-robust and δ-free controller exists and $\delta > 0$.

2 Basic Definitions

We start by recalling basic notions of probability theory. A *σ-field* over a set X is a set $\mathcal{F} \subseteq 2^X$ that includes X and is closed under complement and countable union. A *measurable space* is a pair $(X, \mathcal{F})$ where X is a set called *sample space* and $\mathcal{F}$ is a σ-field over X. A measurable space $(X, \mathcal{F})$ is called *discrete* if $\mathcal{F} = 2^X$. A *probability measure* over measurable space $(X, \mathcal{F})$ is a function $\mathcal{P} : \mathcal{F} \to \mathbb{R}^{\geq 0}$ such that, for each countable collection $\{X_i\}_{i \in I}$ of pairwise disjoint elements of $\mathcal{F}$, $\mathcal{P}(\bigcup_{i \in I} X_i) = \sum_{i \in I} \mathcal{P}(X_i)$, and moreover $\mathcal{P}(X) = 1$. A *probabilistic space* is a triple $(X, \mathcal{F}, \mathcal{P})$ where $(X, \mathcal{F})$ is a measurable space and $\mathcal{P}$ is a probability measure over $(X, \mathcal{F})$. A probability measure over a discrete measurable space is called a *discrete measure*. We also refer to discrete measures as *distributions*. The set of all discrete measures over a measurable space $(X, 2^X)$ is denoted $Disc(X)$.

Markov decision processes. A *Markov decision process* (MDP) $\mathcal{M}$ is a triple (S, Act, P) where S is a finite or countably infinite set of *states*, Act is a finite set of *actions*, and $P : S \times Act \times S \to [0, 1]$ is a (total) *probabilistic function* such that for every $s \in S$ and every $a \in Act$ we have that $\sum_{t \in S} P(s, a, t) \in \{0, 1\}$. We

say that $a \in Act$ is *enabled* in $s \in S$ if $\sum_{t \in S} P(s,a,t) = 1$. The set of all actions that are enabled in a given $s \in S$ is denoted $Act(s)$. For technical convenience, we assume that each state $s \in S$ has at least one enabled action. We say that $\mathcal{M}$ is *finite* if S is finite. A *path* in $\mathcal{M}$ is a nonempty finite or infinite alternating sequence of states and actions $\pi = s_1 a_1 s_2 a_2 \dots a_{n-1} s_n$ or $\pi = s_1 a_1 s_2 a_2 \dots$ such that $P(s_i, a_i, s_{i+1}) > 0$ for all $1 \leq i < n$ or $i \in \mathbb{N}$, resp. The *length* (i.e., the number of actions) of a given π is denoted $|\pi|$, where $|\pi| = \infty$ if π is infinite. For every $1 \leq i \leq |\pi|+1$, the symbol $\pi(i)$ denotes the i-th state of π (which is s_i). A *run* is an infinite path. The sets of all finite paths and all runs of $\mathcal{M}$ are denoted $FPath$ and Run, respectively. Sometimes we write $FPath_{\mathcal{M}}$ and $Run_{\mathcal{M}}$ if $\mathcal{M}$ is not clear from the context. Similarly, the sets of all finite paths and runs that start in a given $s \in S$ are denoted $FPath(s)$ and $Run(s)$, respectively. For finite paths, $last(\pi) = \pi(|\pi|+1)$ denotes the last state of π.

For the rest of this section, we fix a MDP $\mathcal{M} = (S, Act, P)$.

Strategies, adversaries, and policies for MDPs. Let $S_0 \subseteq S$ be nonempty subset of *controllable* states. The states of $S \setminus S_0$ are *environmental*. A *strategy* is a function D that resolves nondeterminism for the controllable states of $\mathcal{M}$. Similarly as in [1], we distinguish among four basic types of strategies for $(\mathcal{M}, S_0)$, according to whether they are deterministic (D) or randomized (R), and Markovian (M) or history-dependent (H).

- A *MD-strategy* is a function $D : S_0 \to Act$ such that $D(s) \in Act(s)$ for all states $s \in S_0$.
- A *MR-strategy* is a function $D : S_0 \to Disc(Act)$ such that $D(s) \in Disc(Act(s))$ for all states $s \in S_0$.
- A *HD-strategy* is a function $D : FPath \to Act$ such that $D(\pi) \in Act(last(\pi))$ for all finite paths $\pi \in FPath$ where $last(\pi) \in S_0$, otherwise $D(\pi) = \bot$.
- A *HR-strategy* is a function $D : FPath \to Disc(Act)$ such that $D(\pi) \in Disc(Act(last(\pi)))$ for all finite paths $\pi \in FPath$ where $last(\pi) \in S_0$, otherwise $D(\pi) = \bot$.

MD, MR, HD, and HR *adversaries* are defined in the same way as strategies of the corresponding type; the only difference is that adversaries range over environmental states. A *policy* is a pair $H = (D, E)$ where D is a strategy and E an adversary. Slightly abusing notation, we write $H(s)$ to denote either $D(s)$ or $E(s)$, depending on whether $s \in S_0$ or not, respectively.

Markov chains induced by policies. A *Markov chain* is a MDP with only one action, i.e., without nondeterminism. Formally, a Markov chain $\mathcal{MC}$ is a pair (S, P) where $(S, \{a\}, P)$ is a MDP. The (only) action a can safely be omitted, and so the probabilistic function is restricted to the set $S \times S$, and a path in $\mathcal{MC}$ is a (finite or infinite) sequence of states $s_1 s_2 s_3 \dots$.

Each $\pi \in FPath_{\mathcal{MC}}$ determines a *basic cylinder* $Run(\pi)$ which consists of all runs that start with π. To every $s \in S$ we associate the probabilistic space

$(Run(s), \mathcal{F}, \mathcal{P})$ where $\mathcal{F}$ is the σ-field generated by all basic cylinders $Run(\pi)$ where π starts with s (i.e., $\pi(1) = s$), and $\mathcal{P} : \mathcal{F} \to [0,1]$ is the unique probability measure such that $\mathcal{P}(Run(\pi)) = \Pi_{i=1}^{|\pi|} P(\pi(i), \pi(i+1))$ (if $|\pi| = 0$, we put $\mathcal{P}(Run(\pi)) = 1$).

Let $\mathcal{M} = (S, Act, P)$ be a MDP. Each policy H for $\mathcal{M}$ induces a Markov chain $\mathcal{MC}_H = (S_H, P_H)$ in the following way:

- If H is a Markovian (MD or MR) policy, then $S_H = S$.
- If H is a history-dependent (HD or HR) policy, then $S_H = FPath_{\mathcal{M}}$.

The function P_H is determined as follows:

- If H is a MD-policy, then $P_H(s_i, s_j) = P(s_i, H(s_i), s_j)$.
- If H is a MR-policy, then $P_H(s_i, s_j) = \Sigma_{a \in Act(s_i)} \mu(a).P(s_i, a, s_j)$ where $\mu = H(s_i)$.
- If H is a HD-policy, then $P_H(\pi, \pi') = P(last(\pi), H(\pi), s)$ if $\pi' = \pi.H(\pi).s$, and $P_H(\pi, \pi') = 0$ otherwise.
- If H is a HR-policy, then $P_H(\pi, \pi') = \mu(a).P(last(\pi), a, s)$ where $\mu = H(\pi)$, if $\pi' = \pi.a.s$, and $P_H(\pi, \pi') = 0$ otherwise.

The logics PCTL and PCTL+LAP. Let $Ap = \{p, q, \dots\}$ be a countably infinite set of *atomic propositions*. The syntax of PCTL *state* and *path* formulae is given by the following abstract syntax equations:

$$\Phi \ ::= \ \mathtt{tt} \ \mid \ p \ \mid \ \neg\Phi \ \mid \ \Phi_1 \wedge \Phi_2 \ \mid \ \mathcal{P}^{\sim\varrho}\varphi \qquad \varphi \ ::= \ \mathcal{X}\Phi \ \mid \ \Phi_1 \,\mathcal{U}\, \Phi_2$$

Here p ranges over Ap, $\varrho \in [0,1]$, and $\sim \in \{\leq, <, \geq, >\}$.

Let $\mathcal{MC} = (S, P)$ be a Markov chain, and let $\nu : Ap \to 2^S$ be a *valuation*. The semantics of PCTL is defined below. State formulae are interpreted over S, and path formulae are interpreted over Run.

$$
\begin{array}{lll}
s \models^\nu \mathtt{tt} & & \\
s \models^\nu p & \text{iff} & s \in \nu(p) \\
s \models^\nu \neg\Phi & \text{iff} & s \not\models^\nu \Phi \\
s \models^\nu \Phi_1 \wedge \Phi_2 & \text{iff} & s \models^\nu \Phi_1 \text{ and } s \models^\nu \Phi_2 \\
s \models^\nu \mathcal{P}^{\sim\varrho}\varphi & \text{iff} & \mathcal{P}(\{\pi \in Run(s) \mid \pi \models^\nu \varphi\}) \sim \varrho
\end{array}
$$

$$
\begin{array}{lll}
\pi \models^\nu \mathcal{X}\Phi & \text{iff} & \pi(2) \models^\nu \Phi \\
\pi \models^\nu \Phi_1 \,\mathcal{U}\, \Phi_2 & \text{iff} & \exists j \geq 1 : \pi(j) \models^\nu \Phi_2 \text{ and } \pi(i) \models^\nu \Phi_1 \text{ for all } 1 \leq i < j
\end{array}
$$

The logic PCTL+LAP is obtained by extending PCTL with long-run average propositions (in the style of [4]). Intuitively, we aim at modelling systems which repeatedly service certain requests, and we are interested in measuring the average costs of servicing a request along an infinite run. The states where the individual services start are identified by (the validity of) a dedicated atomic proposition, and each service corresponds to a finite path between two consecutive occurrences of marked states.

Definition 1. *A* long-run average proposition *is a pair* $[p, f]$ *where* p *is an atomic proposition and* $f : S \to \mathbb{R}^{\geq 0}$ *a* reward function *that assigns to each* $s \in S$ *a reward* $f(s)$.

The reward assigned to a given $s \in S$ corresponds to some costs which are "paid" when s is visited. For example, $f(s)$ can be the expected average time spent in s, the amount of allocated memory, or simply a binary indicator specifying whether s is "good" or "bad". The proposition p is valid in exactly those states where a new service starts. Note that in this setup, a new service starts immediately after finishing the previous service. This is not a real restriction, because the states which precede/follow the actual service can be assigned zero reward.

The syntax of PCTL+LAP formulae is obtained by modifying the syntax of PCTL path formulae as follows:

$$\varphi \ ::= \ \mathcal{X}\Phi \ \mid \ \Phi_1 \,\mathcal{U}\, \Phi_2 \ \mid \ \xi \qquad\qquad \xi \ ::= \ [p, f]^{\sim b} \ \mid \ \neg\xi \ \mid \ \xi_1 \wedge \xi_2$$

Here $[p, f]$ ranges over long-run average propositions, $b \in \mathbb{R}^{\geq 0}$, and $\sim \in \{\leq, <, \geq, >\}$.

Let $\mathcal{MC} = (S, P)$ be a Markov chain, $[p, f]$ a long-run average proposition, and $\nu : Ap \to 2^S$ a valuation. Let $\pi \in \mathit{Run}$ be a run along which p holds infinitely often, and let $\pi(i_1), \pi(i_2), \ldots$ be the sequence of all states in π where p holds. Let $\pi[j]$ denote the subword $\pi(i_{j-1} + 1), \cdots, \pi(i_j)$ of π, where $i_0 = 0$. Hence, $\pi[j]$ is the subword of π consisting of all states in between the $j{-}1^{th}$ state satisfying p (not included) and the j^{th} state satisfying p (included). Intuitively, $\pi[j]$ corresponds to the j^{th} service. Slightly abusing notation, we use $f(\pi[j])$ to denote the total reward accumulated in $\pi[j]$, i.e., $f(\pi[j]) = \sum_{k=i_{j-1}+1}^{i_j} f(\pi(k))$. Now we define the average reward per service in π (with respect to $[p, f]$) as follows:

$$A[p, f](\pi) = \begin{cases} \lim_{n\to\infty} \frac{\sum_{j=1}^{n} f(\pi[j])}{n} & \text{if the limit exists;} \\ \bot & \text{otherwise.} \end{cases}$$

If $\pi \in \mathit{Run}$ contains only finitely many states satisfying p, we put $A[p, f](\pi) = \bot$. Now we define

$$\pi \models^{\nu} [p, f]^{\sim b} \quad \text{iff} \quad A[p, f](\pi) \neq \bot \text{ and } A[p, f](\pi) \sim b$$

The semantics of negation and conjunction of long-run average propositions is defined in the expected way.

3 Controller Synthesis

In this section we examine the controller synthesis problem for finite MDPs, PCTL+LAP properties, and MR policies.

Since the probabilities used in MDPs are often evaluated empirically (and hence inherently imprecise), it is important to analyze the extent to which a

given result about a given MDP is "robust" in the sense that its validity is not influenced by small probability fluctuations. This is formalized in our next definitions:

Definition 2. *Let $\mathcal{M} = (S, Act, P)$ be a MDP, and let $\varepsilon \in [0, 1]$. We say that a MDP $\mathcal{M}' = (S, Act, P')$ is an* ε-perturbation of $\mathcal{M}$ *if for all $(s, a, t) \in S \times Act \times S$ the following two conditions are satisfied:*

- $P(s, a, t) = 0$ *iff* $P'(s, a, t) = 0$,
- $|P(s, a, t) - P'(s, a, t)| \leq \varepsilon$.

Note that Definition 2 also applies to Markov chains.

Definition 3. *Let $\mathcal{M} = (S, Act, P)$ be a MDP, $\varepsilon \in [0, 1]$, $s_i \in S$, and Prop some property of s_i. We say that Prop is* ε-robust *if for every MDP $\mathcal{M}'$ which is an ε-perturbation of $\mathcal{M}$ we have that if $s_i \models Prop$ in $\mathcal{M}$, then $s_i \models Prop$ in $\mathcal{M}'$.*

Examples of 1-robust properties are qualitative LTL and qualitative PCTL properties of states in finite Markov chains, whose (in)validity depends just on the "topology" of a given chain [3]. On the other hand, the property of "being bisimilar to a given state" (here we consider a probabilistic variant of bisimilarity [9]) is generally 0-robust, because even a very small change in probability distribution can spoil the bisimilarity relation.

In a similar fashion we also define a δ-perturbation of a randomized strategy.

Definition 4. *Let $\mathcal{M} = (S, Act, P)$ be a MDP, $S_0 \subseteq S$ a nonempty set of controllable states, D a randomized (i.e., MR or HR) strategy, and $\delta \in [0, 1]$. We say that a strategy D' is a* δ-perturbation of D *if D' is of the same type as D and for all $a \in Act$:*

- *MR case: for all $s \in S_0$: $|D(s)(a) - D'(s)(a)| \leq \delta$ and $D(s)(a) = 0 \Leftrightarrow D'(s)(a) = 0$*
- *HR case: for all $\pi \in FPath$ where $last(\pi) \in S_0$: $|D(\pi)(a) - D'(\pi)(a)| \leq \delta$ and $D(\pi)(a) = 0 \Leftrightarrow D'(\pi)(a) = 0$*

Let $\mathcal{M} = (S, Act, P)$ be a MDP, $S_0 \subseteq S$ a nonempty set of controllable states, $s_i \in S$, and *Prop* some property of s_i. Let $T \in \{\mathrm{MD}, \mathrm{MR}, \mathrm{HD}, \mathrm{HR}\}$. A *$T$-controller* for $\mathcal{M}$ and *Prop* is a T-strategy D such that $s_i \models Prop$ in $\mathcal{MC}_{(D,E)}$ for every T-environment E. We say that the controller D is

- ε-robust for a given $\varepsilon \in [0, 1]$ if the property "D is a controller for $\mathcal{M}$ and *Prop*" is ε-robust. In other words, D is a valid controller for *Prop* even if the probabilities in $\mathcal{M}$ are slightly (i.e., at most by ε) changed.
- ε-robust and δ-free for given $\varepsilon, \delta \in [0, 1]$ if every D' which is a δ-perturbation of D is an ε-robust controller for $\mathcal{M}$ and *Prop*.

In the rest of this section we consider the problem of MR-controller synthesis for a given MDP $\mathcal{M} = (S, Act, P)$, a set of controllable states $S_0 \subseteq S$, a state $s_i \in S$, a PCTL+LAP formula φ, and a valuation ν. For notation simplification, we do not list these elements in our theorems explicitly, although they are always a part of a problem instance.

Theorem 5. *Let $\varepsilon, \delta \in [0,1]$. The problem whether there is an ε-robust and δ-free MR-controller is in **EXPTIME**.*

Proof. We construct a closed formula of $(\mathbb{R}, *, +, \leq)$ which is valid iff an ε-robust and δ-free MR-controller exists. The formula has the following structure:

$$\exists D\ \forall D'\ (D'\ \delta\text{-pert. of } D) \Rightarrow (\forall E\ \forall P'\ (P'\ \varepsilon\text{-pert. of } P) \Rightarrow (\exists Y\ (Y_\varphi^{s_i}{=}1)))$$

Intuitively, the formula says "there is an MR-strategy D such that for every strategy D', which is a δ-perturbation of D, every environment E, and every chain (an ε-perturbation of $\mathcal{M}$) with probabilities P', there is a consistent validity assumption Y (which declares each subformula of φ to be either true or false in every state of S) such that Y sets the formula φ to true in the state s_i". Now we describe these parts in greater detail.

Let $X_a^s, X_a'^s$ be fresh first-order variables for all $s \in S_0$ and $a \in Act(s)$. These variables are used to encode the strategies D, D'. Intuitively, X_a^s and $X_a'^s$ carry the probability of choosing the action a in the state s in D and D', respectively. The

$$\exists D\ \forall D'\ (D'\ \delta\text{-pert. of } D)$$

part can then be implemented as follows:

$$\exists \{X_a^s \mid s \in S_0, a \in Act(s)\} : \bigwedge_{X_a^s} (0 \leq X_a^s \leq 1) \wedge \bigwedge_{s \in S_0} \Big(\sum_{a \in Act(s)} X_a^s = 1 \Big) \wedge$$
$$\forall \{X_a'^s \mid s \in S_0, a \in Act(s)\} : \Big(\bigwedge_{X_a'^s} (0 \leq X_a'^s \leq 1) \wedge \bigwedge_{s \in S_0} \Big(\sum_{a \in Act(s)} X_a'^s = 1 \Big) \wedge$$
$$\bigwedge_{X_a^s} ((X_a^s = 0 \Leftrightarrow X_a'^s = 0) \wedge (|X_a^s - X_a'^s| \leq \delta))$$

Similarly,

- for all $s \in S \setminus S_0$ and $a \in Act(s)$ we fix fresh first-order variables $X_a'^s$ that encode the environment E (from a certain point on, we do not need to distinguish between the probabilities chosen by D' and E);
- for all $s, t \in S$ and $a \in Act(s)$ we fix a fresh variable $P_a^{s,t}$ that encodes the corresponding probability of P';
- for every $\phi \in cl(\varphi)$ (here $cl(\varphi)$ is the set of all subformulas of φ) and every $s \in S$ we fix a variable Y_ϕ^s that carries either 1 or 0, depending on whether s satisfies ϕ or not, respectively. As we shall see, the value of Y_ϕ^s is first "guessed" and then "verified".

The $\forall E\ \forall P'\ (P'\ \varepsilon\text{-pert. of } P) \Rightarrow (\exists Y\ (Y_\varphi^{s_i}{=}1))$ part can now be implemented as follows:

$$\forall\{X'^s_a \mid s \in S \setminus S_0, a \in Act(s)\} : \bigwedge_{X'^s_a} (0 \le X'^s_a \le 1) \wedge \bigwedge_{s \in S \setminus S_0} \Big(\sum_{a \in Act(s)} X'^s_a = 1 \Big) \Rightarrow$$
$$\forall\{P^{s,t}_a \mid s,t \in S, a \in Act(s)\} :$$
$$\bigwedge_{P^{s,t}_a} ((P(s,a,t) = 0 \Leftrightarrow P^{s,t}_a = 0) \wedge (|P(s,a,t) - P^{s,t}_a| \le \varepsilon)) \Rightarrow$$
$$\exists\{Y^s_\phi \mid \phi \in cl(\varphi), s \in S\} :$$
$$\bigwedge_{Y^s_\phi} ((Y^s_\phi = 0 \vee Y^s_\phi = 1) \wedge (Y^s_\phi = 1 \Leftrightarrow \psi^s_\phi)) \wedge (Y^{s_i}_\varphi = 1)$$

The tricky part of the construction is the formula ψ^s_ϕ, which is defined inductively on the structure of ϕ. Intuitively, ψ^s_ϕ says that s satisfies ϕ, where we assume that this has already been achieved for all subformulae of ϕ (hence, by justifying all steps in our inductive definition we also yield a correctness proof for our construction):

- $\phi \equiv p$. If $s \in \nu(p)$, then $\psi^s_\phi \equiv \mathtt{tt}$, otherwise $\psi^s_\phi \equiv \mathtt{ff}$.
- $\phi \equiv \neg\phi'$. Then $\psi^s_\phi \equiv (Y^s_{\phi'} = 0)$.
- $\phi \equiv \phi_1 \wedge \phi_2$. Then $\psi^s_\phi \equiv (Y^s_{\phi_1} = 1) \wedge (Y^s_{\phi_2} = 1)$.
- $\phi \equiv \mathcal{P}^{\sim\varrho}\mathcal{X}\phi'$. Then $\psi^s_\phi \equiv \Big(\sum_{a \in Act(s), t \in S} X'^s_a \cdot P^{s,t}_a \cdot Y^t_{\phi'} \Big) \sim \varrho$.

The case when $\phi \equiv \mathcal{P}^{\sim\varrho}\phi_1 \,\mathcal{U}\, \phi_2$ is slightly more complicated. The probabilities $\{Z^r \mid r \in S\}$, where Z^r is the probability that a run initiated in r satisfies the path formula $\phi_1 \,\mathcal{U}\, \phi_2$, form the least solution (in the interval $[0,1]$) of a system of recursive linear equations constructed as follows (where Z^r should be seen as "unknowns"; cf. [7, 3]):

- if $Y^r_{\phi_2} = 1$, we put $Z^r = 1$;
- if $Y^r_{\phi_1} = 0$ and $Y^r_{\phi_2} = 0$, we put $Z^r = 0$;
- if $Y^r_{\phi_1} = 1$ and $Y^r_{\phi_2} = 0$, we put $Z^r = \Big(\sum_{a \in Act(s), t \in S} X'^r_a \cdot P^{r,t}_a \cdot Z^t \Big)$.

So, the formula ψ^s_ϕ for the case when $\phi \equiv \mathcal{P}^{\sim\varrho}\phi_1 \,\mathcal{U}\, \phi_2$ looks as follows:

$$\exists\{Z^r \mid r \in S\} : \bigwedge_{r \in S} (0 \le Z^r \le 1) \;\wedge\; \{Z^r\} \text{ is a solution} \;\wedge\; Z^s \sim \varrho \;\wedge$$
$$\Big(\forall\{Z'^r \mid r \in S\} : \big(\bigwedge_{r \in S} (0 \le Z'^r \le 1) \wedge \{Z'^r\} \text{ is solution } \big) \Rightarrow \big(\bigwedge_{r \in S} Z^r \le Z'^r \big) \Big)$$

Here "$\{Z^r\}$ is a solution" means that the variables $\{Z^r\}$ satisfy the above system of recursive linear equations, which can be easily encoded in $(\mathbb{R}, +, *, \le)$.

Finally, we analyze the most complicated case when $\phi \equiv \mathcal{P}^{\sim\varrho}[p,f]^{\approx b}$. In order to check long-run average propositions, we need to analyze the structure of the Markov chain induced by the current values of the X'^r_a variables and find bottom strongly connected components (BSCC) of this chain.

We start by computing the probabilities $Prob^t_r$ of reaching the state t from the state r. The set $\{Prob^t_r \mid r,t \in S\}$ forms the least solution (in the interval $[0,1]$) of the following system of recursive linear equations, where $Prob^t_r$ should be interpreted as "unknowns":

– if $r = t$, we put $Prob_r^t = 1$;
– if $r \neq t$, we put $Prob_r^t = \sum_{u \in S} \left(\sum_{a \in Act(r)} X_a^{\prime r} \cdot P_a^{r,u} \right) \cdot Prob_u^t$.

So, the formula which "computes" all $Prob_r^t$ looks as follows:

$$\exists \{Prob_r^t \mid r,t \in S\} : \bigwedge_{r,t \in S} (0 \leq Prob_r^t \leq 1) \wedge \{Prob_r^t\} \text{ is solution } \wedge$$
$$\Big(\forall \{{Prob'}_r^t \mid r,t \in S\} : (\bigwedge_{r,t \in S} (0 \leq {Prob'}_r^t \leq 1) \wedge \{{Prob'}_r^t\} \text{ is solution }) \Rightarrow$$
$$(\bigwedge_{r,t \in S} Prob_r^t \leq {Prob'}_r^t) \Big)$$

Now we introduce predicates $SCC_{r,t}$ and $BSCC_r$, where $SCC_{r,t}$ means that r,t are in the same strongly connected component, and $BSCC_r$ means that r is in a bottom strongly connected component.

$$SCC_{r,t} ::= (Prob_r^t > 0 \wedge Prob_t^r > 0)$$
$$BSCC_r ::= \bigwedge_{t \in S} (Prob_r^t > 0 \Rightarrow Prob_t^r > 0)$$

The next step is to compute the (unique) invariant distribution for each $BSCC$. Recall that the invariant distribution in a finite strongly connected Markov chain is the (unique) vector Inv of numbers from $[0,1]$ such that the sum of all components in Inv is equal to 1 and $Inv * T = Inv$ where T is the transition matrix of the considered Markov chain.

For each BSCC (represented by a given $t \in S$), the following formula "computes" its unique invariant distribution $\{Inv_r^t \mid r,t \in S\}$. More precisely, Inv_r^t is either zero (if r does not belong to the BSCC represented by t), or equals the value of the invariant distribution in r (otherwise). We also need to ensure that the representative t is chosen uniquely, i.e., the values of all $Inv_r^{t'}$, where t' is in the same SCC as t, is zero:

$$\exists \{Inv_r^t \mid r,t \in S\} :$$
$$\bigwedge_{r,t \in S} \Big(\quad (0 \leq Inv_r^t \leq 1) \wedge ((\neg BSCC_r \vee \neg BSCC_t \vee \neg SCC_{r,t}) \Rightarrow Inv_r^t = 0)$$
$$\wedge ((BSCC_r \wedge BSCC_t \wedge SCC_{r,t}) \Rightarrow$$
$$Inv_r^t = \sum_{u \in S} (Inv_u^t \cdot \sum_{a \in Act(u)} X_a^{\prime u} \cdot P_a^{u,r})) \Big) \wedge$$
$$\bigwedge_{t \in S} \Big(BSCC_t \Rightarrow \Big(\sum_{r \in S} Inv_r^t = 1 \wedge \bigwedge_{t' \in S, t' \neq t} (SCC_{t,t'} \Rightarrow \sum_{r \in S} Inv_r^{t'} = 0) \Big) \vee$$
$$\Big(\sum_{r \in S} Inv_r^t = 0 \wedge \bigvee_{t' \in S, t' \neq t} (SCC_{t,t'} \wedge \sum_{r \in S} Inv_r^{t'} = 1) \Big) \Big)$$

According to ergodic theorem, almost all runs (i.e., with probability one) end up in some BSCC, and then "behave" according to the corresponding invariant distribution (i.e., the "percentage of visits" to each state is given by the invariant

distribution). From this one can deduce that the average reward per service is the same for almost all runs that hit a given BSCC. Hence, for each $t \in S$ we can "compute" a value Rew_t which is equal to 1 iff

- t represents some BSCC and
- at least one state in this BSCC satisfies p (and hence p is satisfied infinitely often in almost all runs that hit this BSCC) and
- the average reward per service associated with this BSCC is "good" with respect to the long-run average proposition $[p, f]^{\approx b}$.

Note that the average reward per service can be computed as the ratio between the average reward per state and the percentage of visits to states where the service starts. Thus, we obtain the formula

$$\exists\{Rew_t \mid t \in S\} : \bigwedge_{t \in S} (Rew_t = 0 \vee Rew_t = 1) \wedge \left(Rew_t = 1 \Leftrightarrow \Big(\sum_{r \in S} Inv_r^t \cdot Y_p^r > 0 \Big) \wedge \left(\frac{\sum_{r \in S} Inv_r^t \cdot f(r)}{\sum_{r \in S} Inv_r^t \cdot Y_p^r} \approx b \right) \right)$$

Finally, the formula ψ_ϕ^s "checks" whether the "good" BSCCs are reachable with a suitable probability:

$$\psi_\phi^s ::= \Big(\sum_{t \in S} Prob_s^t \cdot Rew_t \Big) \sim \varrho$$

Although the whole construction is technically complicated, none of the above considered subcases leads to an exponential blowup. Hence, we can conclude that the size of the resulting formula is *polynomial* in the size of our instance. Moreover, a closer look reveals that the quantifiers are alternated only to a fixed depth. Hence, our theorem follows by applying the result of [6]. □

The technique used in the proof of Theorem 5 can easily be adapted to prove the following:

Theorem 6. *For every $\varepsilon \in [0, 1]$, if there is an ε-robust MR-controller which is δ-free for some $\delta > 0$, then an ε-robust MR-controller is effectively constructible.*

Proof. First, realize that the problem whether there is an ε-robust MR-controller which is δ-free for some $\delta > 0$ is in **EXPTIME**. We use the formula constructed in the proof of Theorem 5, where the constant δ is now treated as first-order variable, and the whole formula is prefixed by "$\exists \delta > 0$". If the answer is positive (i.e., there is a controller with a non-zero freedom), one can effectively find some δ' for which there is an ε-robust and δ'-free controller by trying smaller and smaller δ'. As soon as we have such a δ', there are only finitely many candidates for a suitable MR-strategy D. Intuitively, we divide the interval $[0, 1]$ into finitely many pieces of length δ', and from each such subinterval we test only one value. This suffices because the controller we are looking for is δ'-free. More precisely, we successively try to set each of the variable $\{X_a^s\}$ to values

$$\left\{ \frac{n}{|Act(s)|} + m\delta' \text{ where } n, m \in \mathbb{Z}, 0 \leq n \leq |Act(s)|, -\left\lceil \frac{1}{\delta'} \right\rceil \leq m \leq \left\lceil \frac{1}{\delta'} \right\rceil \right\}$$

so that $0 \leq X_a^s \leq 1$ and $\sum_{a \in Act(s)} X_a^s = 1$ for each $s \in S$. For each choice we check if it works (using the formula of Theorem 5 where the $\{X_a^s\}$ variables are replaced with their chosen values and δ is set to zero). One of these finitely many options is guaranteed to work, and hence a controller is eventually found. □

Similarly, we can also approximate the maximal ε for which there is an ε-robust MR-controller (this maximal ε is denoted ε_m):

Theorem 7. *For a given $\theta > 0$, one can effectively compute a rational number κ such that $|\kappa - \varepsilon_m| \leq \theta$.*

Since our algorithm for computing an ε-robust MR-controller works only if there is at least one such controller with a non-zero freedom, it makes sense to ask what is the maximal ε for which there is an ε-robust MR-controller with a non-zero freedom. Let us denote this maximal ε by ε'_m.

Theorem 8. *For a given $\theta > 0$, one can effectively compute a rational number κ such that $|\kappa - \varepsilon'_m| \leq \theta$.*

References

1. C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski. Controller synthesis for probabilistic systems. In *Proceedings of IFIP TCS'2004*. Kluwer, 2004.
2. P. Bouyer, D. D'Souza, P. Madhusudan, and A. Petit. Timed control with partial observability. In *Proceedings of CAV 2003*, vol. 2725 of *LNCS*, pp. 180–192. Springer, 2003.
3. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *JACM*, 42(4):857–907, 1995.
4. L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *Proceedings of LICS'98*, pp. 454–465. IEEE, 1998.
5. L. de Alfaro, M. Faella, T. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In *Proceedings of CONCUR 2003*, vol. 2761 of *LNCS*, pp. 144–158. Springer, 2003.
6. D. Grigoriev. Complexity of deciding Tarski algebra. *Journal of Symbolic Computation*, 5(1–2):65–108, 1988.
7. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
8. A. Nilim and L. El Ghaoui. Robustness in markov decision problems with uncertain transition matrices. In *Proceedings of NIPS 2003*. MIT Press, 2003.
9. R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. *NJC*, 2(2):250–273, 1995.
10. A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. Univ. of California Press, Berkeley, 1951.
11. W. Thomas. Infinite games and verification. In *Proceedings of CAV 2003*, vol. 2725 of *LNCS*, pp. 58–64. Springer, 2003.
12. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *TCS*, 158(1&2):343–359, 1996.