Controller Synthesis and Verification for Markov Decision Processes with Qualitative Branching Time Objectives*

Tomáš Brázdil, Vojtěch Forejt, and Antonín Kučera

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

Abstract. We show that the controller synthesis and verification problems for Markov decision processes with qualitative PECTL* objectives are 2-**EXPTIME** complete. More precisely, the algorithms are *polynomial* in the size of a given Markov decision process and doubly exponential in the size of a given qualitative PECTL* formula. Moreover, we show that if a given qualitative PECTL* objective is achievable by *some* strategy, then it is also achievable by an effectively constructible *one-counter* strategy, where the associated complexity bounds are essentially the same as above. For the fragment of qualitative PCTL objectives, we obtain **EXPTIME** completeness and the algorithms are only singly exponential in the size of the formula.

1 Introduction

A *Markov decision process (MDP)* [16, 11] is a finite directed graph $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ where the vertices of V are partitioned into *non-deterministic* and *stochastic* subsets (denoted $V_\Box$ and $V_\bigcirc$, resp.), $E \subseteq V \times V$ is a set of edges, and $Prob$ assigns a fixed probability to every edge $(s, s') \in E$ where $s \in V_\bigcirc$ so that $\sum_{(s,s')\in E} Prob(s, s') = 1$ for every fixed $s \in V_\bigcirc$. Without restrictions, we assume that each vertex has at least one and at most two outgoing edges.

MDPs are used as a generic model for discrete systems where one can make *decisions* (by selecting successors in non-deterministic vertices) whose outcomes are *uncertain* (this is modeled by stochastic vertices). The application area of MDPs includes such diverse fields as ecology, chemistry, or economics. In this paper, we focus on more recent applications of MDPs in the area of computer systems (see, e.g., [18]). Here, non-deterministic vertices are used to model the environment, unpredictable users, process scheduler, etc. Stochastic vertices model stochastic features such as coin-tossing in randomized algorithms, bit-flips and other hardware errors whose probability is known empirically, probability distribution on input events, etc. There are two main problems studied in this area:

- *Controller synthesis.* The task is to construct a "controller" which selects appropriate successors at non-deterministic vertices so that a certain objective is achieved.

* Supported by the research center Institute for Theoretical Computer Science (ITI), project No. 1M0545.

- *Verification.* Here, we wonder whether a given objective is achieved for all "adversaries" that control the non-deterministic vertices. In other words, we want to know whether a given system behaves correctly in all environments, under all interleavings produced by a scheduler, etc.

Both "controller" and "adversary" are mathematically captured by the notion of *strategy*, i.e., a function which to every computational history $vs \in V^*V_\Box$ ending in a non-deterministic vertex assigns a probability distribution over the set of outgoing edges of s. General strategies are also referred to as HR strategies because the decision depends on the *history* of the current computation (H) and it is *randomized* (R). Strategies that always return a Dirac distribution are *deterministic* (D), and strategies which depend just on the currently visited vertex are *memoryless* (M). Thus, one can distinguish among HR, HD, MD, and MR strategies.

Since the original application field of MDPs was mainly economics and performance evaluation, there is a rich and mature mathematical theory of MDPs with *discounted* and *limit-average* objectives [16, 11]. In the context of computer systems, one is usually interested in objectives related to safety, liveness, fairness, etc. , and these can be naturally formalized as *temporal properties*. In particular, the subclass of *linear-time* properties (such as Büchi, parity, Rabin, Street, or Muller properties) is relatively well understood even in a more general framework of simple stochastic games [12, 19, 8, 6]. Another class of temporal objectives studied in the literature are *linear-time multi-objectives* [10, 7], which are Boolean combinations of linear-time objectives.

In this paper, we deal with a more general class of temporal properties that are specified as formulae of probabilistic branching-time logics PCTL, PCTL*, and PECTL* [13]. These logics are obtained from their non-probabilistic counterparts CTL, CTL*, and ECTL* (see, e.g., [9, 17]) by replacing the universal and existential path quantifiers with the probabilistic operator $\mathcal{P}^{\bowtie\varrho}$, where ϱ is a rational constant and $\bowtie$ is a comparison such as $\leq$ or $>$. Intuitively, the formula $\mathcal{P}^{\bowtie\varrho}\varphi$ says "the probability of all runs that satisfy φ is $\bowtie$-related to ϱ". If the bound ϱ is restricted just to 0 and 1, we obtain the *qualitative fragment* of a given logic. Controller synthesis for MDPs with branching-time objectives has been considered in [1] where it is shown that strategies for fairly simple qualitative PCTL objectives may require memory and/or randomization. Hence, the classes of MD, MR, HD, and HR strategies (see above) form a strict hierarchy. Moreover, in the same paper it is also proved that the controller synthesis problem for PCTL objectives is **NP**-complete for the subclass of MD strategies. A trivial consequence of this result is **coNP**-completeness of the verification problem for PCTL objectives and MD strategies. In [15], the subclass of MR strategies is examined, and it is proved that the controller synthesis problem for PCTL objectives and MR strategies is in **PSPACE** (the same holds for the verification problem). Some results about history-dependent strategies are presented in [3], where it is shown that controller synthesis for PCTL objectives and HD (and also HR) strategies is *highly undecidable* (in fact, this problem is complete for the Σ^1_1 level of the analytical hierarchy). In [3], it is also demonstrated that the controller synthesis and verification problems are **EXPTIME**-complete for HD/HR strategies and the fragment of PCTL that contains only the qualitative connectives $\mathcal{P}^{=1}\mathcal{F}$, $\mathcal{P}^{=1}\mathcal{G}$, and $\mathcal{P}^{>0}\mathcal{F}$. Moreover, it is shown that strategies for this type of objectives require only *finite memory*, and can be effectively constructed in exponential time. This study is continued

in [4] where the memory requirements for objectives of various fragments of qualitative PCTL are classified in a systematic way.

Our contribution. In this paper we solve the controller synthesis and verification problems for all qualitative PCTL and qualitative PECTL* objectives and history-dependent (i.e., HR and HD) strategies. For the sake of simplicity, we first unify HR and HD strategies into a single notion of *history-dependent combined (HC)* strategy. Let $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ be a MDP and let (V_D, V_R) be a partitioning of $V_\Box$ into the subsets of *Dirac* and *randomizing* vertices. A *HC strategy* is a HR strategy σ such that $\sigma(vs)$ is a Dirac distribution for every $vs \in V^*V_D$. Hence, HC strategies coincide with HR and HD strategies when $V_D = \emptyset$ and $V_D = V_\Box$, respectively. Nevertheless, our solution covers also the cases when $\emptyset \neq V_D \neq V_\Box$. Now we can formulate the main result of this paper.

Theorem 1. *Let $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ be a MDP, (V_D, V_R) a partitioning of $V_\Box$, and φ a qualitative PECTL* formula.*

- *The problem whether there is a HC strategy that achieves the objective φ is* 2-***EXPTIME****-complete. More precisely, the problem is solvable in time which is* polynomial *in $|G|$ and doubly exponential in $|\varphi|$. Since qualitative PECTL* objectives are closed under negation, the same complexity results hold for the verification problem.*
- *If the objective φ is achievable by some HC strategy, then it is also achievable by a* one-counter *strategy (see Definition 3). Moreover, the corresponding one-counter automaton can effectively be constructed in time which is polynomial in $|V|$, doubly exponential in $|\varphi|$, and singly exponential in bp, where bp is the number of bits of precision for the constants employed by Prob.*
- *In the special case when φ is a qualitative PCTL formula, the controller synthesis problem is* ***EXPTIME****-complete and the algorithms are only singly exponential in the size of the formula.*

This result gives a substantial generalization of the partial results discussed above and solves some of the major open questions formulated in these papers. In some sense, it complements the undecidability result for quantitative PCTL objectives given in [3].

The principal difficulty which requires new ideas and insights is that strategies for qualitative branching-time objectives need infinite memory in general. In Section 3 we give examples demonstrating this fact. Another difference from the previous work is that the precise values of probabilities that are employed by a given strategy *do influence* the (in)validity of qualitative PECTL* objectives. This is very different from qualitative linear-time (multi-)objectives whose (in)validity depends just on the information what edges have zero/positive probability.

Due to space constraints, we could not include all technical definitions and proofs. These can be found in the full version of this paper [5].

2 Definitions

In this section we recall basic definitions that are needed for understanding key results of this paper. For reader's convenience, we also repeat the definitions that appeared already in Section 1.

In the rest of this paper, $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Q}$, and $\mathbb{R}$ denote the set of positive integers, non-negative integers, rational numbers, and real numbers, respectively. We also use the standard notation for intervals of real numbers, writing, e.g., $(0, 1]$ to denote the set $\{x \in \mathbb{R} \mid 0 < x \leq 1\}$.

The set of all finite words over a given alphabet Σ is denoted Σ^*, and the set of all infinite words over Σ is denoted Σ^ω. Given two sets $K \subseteq \Sigma^*$ and $L \subseteq \Sigma^* \cup \Sigma^\omega$, we use $K \cdot L$ (or just KL) to denote the concatenation of K and L, i.e., $KL = \{ww' \mid w \in K, w' \in L\}$. We also use Σ^+ to denote the set $\Sigma^* \setminus \{\varepsilon\}$ where ε is the empty word. The length of a given $w \in \Sigma^* \cup \Sigma^\omega$ is denoted $\mathit{length}(w)$, where the length of an infinite word is ω. Given a word (finite or infinite) over Σ, the individual letters of w are denoted $w(0), w(1), \ldots$.

A *probability distribution* over a finite or countably infinite set X is a function $f : X \to [0, 1]$ such that $\sum_{x \in X} f(x) = 1$. A probability distribution is *Dirac* if it assigns 1 to exactly one element. A *σ-field* over a set Ω is a set $\mathcal{F} \subseteq 2^\Omega$ that includes Ω and is closed under complement and countable union. A *probability space* is a triple $(\Omega, \mathcal{F}, \mathcal{P})$ where Ω is a set called *sample space*, $\mathcal{F}$ is a σ-field over Ω whose elements are called *events*, and $\mathcal{P} : \mathcal{F} \to [0, 1]$ is a *probability measure* 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 $\mathcal{P}(\Omega)$=1.

Definition 1 (Markov Chain). *A* Markov chain *is a triple $M = (S, \to, \mathit{Prob})$ where S is a finite or countably infinite set of* states*, $\to \subseteq S \times S$ is a* transition relation*, and Prob is a function which to each transition $s \to t$ of M assigns its probability $\mathit{Prob}(s \to t) \in (0, 1]$ so that for every $s \in S$ we have $\sum_{s \to t} \mathit{Prob}(s \to t) = 1$ (as usual, we write $s \xrightarrow{x} t$ instead of $\mathit{Prob}(s \to t) = x$).*

A *path* in M is a finite or infinite word $w \in S^+ \cup S^\omega$ such that $w(i-1) \to w(i)$ for every $1 \leq i < \mathit{length}(w)$. A *run* in M is an infinite path in M. The set of all runs that start with a given finite path w is denoted $\mathit{Run}[M](w)$. When M is clear from the context, we write $\mathit{Run}(w)$ instead of $\mathit{Run}[M](w)$.

When defining the semantics of probabilistic logics (see below), we need to measure the probability of certain sets of runs. Formally, to every $s \in S$ we associate the probability space $(\mathit{Run}(s), \mathcal{F}, \mathcal{P})$ where $\mathcal{F}$ is the σ-field generated by all *basic cylinders* $\mathit{Run}(w)$ where w is a finite path starting with s, and $\mathcal{P} : \mathcal{F} \to [0, 1]$ is the unique probability measure such that $\mathcal{P}(\mathit{Run}(w)) = \Pi_{i=1}^{\mathit{length}(w)-1} x_i$ where $w(i-1) \xrightarrow{x_i} w(i)$ for every $1 \leq i < \mathit{length}(w)$. If $\mathit{length}(w) = 1$, we put $\mathcal{P}(\mathit{Run}(w)) = 1$. Hence, only certain subsets of $\mathit{Run}(s)$ are $\mathcal{P}$-measurable, but in this paper we only deal with "safe" subsets that are guaranteed to be in $\mathcal{F}$.

Definition 2 (Markov Decision Process). *A* Markov decision process (MDP) *is a finite directed graph $G = (V, E, (V_\Box, V_\bigcirc), \mathit{Prob})$ where the vertices of V are partitioned into* non-deterministic *and* stochastic *subsets (denoted $V_\Box$ and $V_\bigcirc$, resp.), $E \subseteq V \times V$ is a set of edges, and Prob assigns a fixed positive probability to every edge $(s, s') \in E$ where $s \in V_\bigcirc$ so that $\sum_{(s,s') \in E} \mathit{Prob}(s, s') = 1$ for every fixed $s \in V_\bigcirc$. For technical convenience, we require that each vertex has at least one and at most two outgoing edges.*

Let $G = (V, E, (V_\Box, V_\bigcirc), \mathit{Prob})$ be a MDP. A *strategy* is a function which to every $vs \in V^* V_\Box$ assigns a probability distribution over the set of outgoing edges of s. Each

strategy σ determines a unique Markov chain G_σ where states are finite paths in G and $vs \xrightarrow{x} vss'$ iff either s is stochastic, $(s,s') \in E$, and $Prob((s,s')) = x$, or s is non-deterministic, $(s,s') \in E$, and x is the probability of (s,s') chosen by $\sigma(vs)$. General strategies are also called HR strategies, because they are *history-dependent (H)* and *randomized (R)*. We say that σ is *memoryless (M)* if $\sigma(vs)$ depends just on the last vertex s, and *deterministic* if $\sigma(vs)$ is a Dirac distribution. Thus, we obtain the classes of HR, HD, MR, and MD strategies. For the sake of clarity and uniformity of our presentation, we also introduce the notion of *history-dependent combined (HC)* strategy. Here we assume that the non-deterministic vertices of $V_\Box$ are split into two disjoint subsets V_D and V_R of *Dirac* and *randomizing* vertices. A HC strategy is a HR strategy σ such that $\sigma(vs)$ is a Dirac distribution for every $vs \in V^*V_D$. Hence, in the special case when $V_D = \emptyset$ (or $V_D = V_\Box$), every HC strategy is a HD strategy (or a HR strategy). A special type of history-dependent strategies are *finite-memory (F)* strategies. A finite-memory strategy σ is specified by a deterministic finite-state automaton $\mathcal{A}$ over the input alphabet V (see, e.g., [14]), where $\sigma(vs)$ depends just on the control state entered by $\mathcal{A}$ after reading the word vs. In this paper we also consider *one-counter* strategies which are specified by *one-counter automata*.

Definition 3 (One counter automaton). *A* one counter automaton *is a tuple* $C = (Q, \Sigma, q_{in}, \delta^{=0}, \delta^{>0})$ *where* Q *is a finite set of* control states*,* Σ *is a finite* input alphabet*,* $q_{in} \in Q$ *is the* initial state*, and* $\delta^{=0} : Q \times \Sigma \to Q \times \{0,1\}$, $\delta^{>0} : Q \times \Sigma \to Q \times \{0,1,-1\}$ *are* transition functions*. The set of* configurations *of* C *is* $Q \times \mathbb{N}_0$*. For every* $u \in \Sigma^+$ *we define a binary relation* $\overset{u}{\mapsto}$ *over configurations inductively as follows:*

- *for all* $a \in \Sigma$ *we put* $(q,c) \overset{a}{\mapsto} (q', c+i)$ *iff either* $c = 0$ *and* $\delta^{=0}(q,a) = (q', i)$*, or* $c > 0$ *and* $\delta^{>0}(q,a) = (q', i)$*;*
- $(q,c) \overset{au}{\mapsto} (q', c')$ *iff there is* (q'', c'') *such that* $(q,c) \overset{a}{\mapsto} (q'', c'')$ *and* $(q'', c'') \overset{u}{\mapsto} (q', c')$*.*

For every $u \in \Sigma^+$*, let* $q_u \in Q$ *and* $c_u \in \mathbb{N}_0$ *be the unique elements such that* $(q_{in}, 0) \overset{u}{\mapsto} (q_u, c_u)$*.*

Let $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ be a MDP and (V_D, V_R) a partitioning of $V_\Box$. A *one-counter strategy* is a HC strategy σ for which there is a one-counter automaton $C = (Q, V, q_{in}, \delta^{=0}, \delta^{>0})$ and a constant $k \in \mathbb{N}$ such that

- for every $vs \in V^*V_D$, $\sigma(vs)$ is a Dirac distribution that depends only on q_{vs} and the information whether c_{vs} is zero or not;
- for every $vs \in V^*V_R$ such that s has two outgoing edges, $\sigma(vs)$ is either a Dirac distribution or a distribution that assigns $k^{-c_{vs}}$ to one edge, and $1 - k^{-c_{vs}}$ to the other edge. The choice depends solely on q_{vs}.

Before presenting the definition of the logic PECTL*, we need to recall the notion of Büchi automaton. Our definition of Büchi automaton is somewhat nonstandard in the sense that we consider only special alphabets of the form $2^{\{1,\dots,n\}}$ and the symbols assigned to transitions in the automaton are interpreted in a special way. These differences are not fundamental but technically convenient.

Definition 4 (Büchi automaton). *A* Büchi automaton of arity $n \in \mathbb{N}$ *is a tuple* $\mathcal{B} = (Q, q_{in}, \delta, A)$*, where* Q *is a finite set of* control states*,* $q_{in} \in Q$ *is the* initial state*,* $\delta : Q \times 2^{\{1,\dots,n\}} \to 2^Q$ *is a* transition function*, and* $A \subseteq Q$ *is a set of* accepting states*. A*

given infinite word w over the alphabet $2^{\{1,\dots,n\}}$ is accepted *by $\mathcal{B}$ if there is an* accepting computation *for w, i.e., an infinite sequence of states $q_0, q_1, \dots$ such that $q_0 = q_{in}$, $q_j \in A$ for infinitely many $j \in \mathbb{N}_0$, and for all $i \in \mathbb{N}_0$ there is $\alpha_i \in 2^{\{1,\dots,n\}}$ such that $q_{i+1} \in \delta(q_i, \alpha_i)$ and $\alpha_i \subseteq w(i)$. The set of all infinite words accepted by $\mathcal{B}$ is denoted $L(\mathcal{B})$.*

Let $Ap = \{a, b, c, \dots\}$ be a countably infinite set of *atomic propositions*. The syntax of PECTL* formulae is defined by the following abstract syntax equation:

$$\varphi \quad ::= \quad a \mid \neg a \mid \mathcal{P}^{\bowtie \varrho}\mathcal{B}(\varphi_1, \dots, \varphi_n)$$

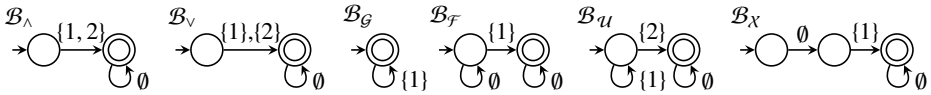
Here a ranges over Ap, $\bowtie$ is a comparison (i.e., $\bowtie \in \{<, >, \leq, \geq, =\}$), ϱ is a rational constant, $n \in \mathbb{N}$, and the $\mathcal{B}$ in $\mathcal{B}(\varphi_1, \dots, \varphi_n)$ is a Büchi automaton of arity n. The *qualitative fragment* of PECTL* is obtained by restricting ϱ to 0 and 1. For simplicity, from now on we write $\mathcal{B}^{\bowtie \varrho}(\varphi_1, \dots, \varphi_n)$ instead of $\mathcal{P}^{\bowtie \varrho}\mathcal{B}(\varphi_1, \dots, \varphi_n)$.

Let $M = (S, \to, Prob)$ be a Markov chain, and let $\eta : S \to 2^{Ap}$ be a *valuation*. The validity of PECTL* formulae in the states of M is defined inductively as follows: $s \models^\eta a$ iff $a \in \eta(s)$, $s \models^\eta \neg a$ iff $a \notin \eta(s)$, and

$$s \models^\eta \mathcal{B}^{\bowtie \varrho}(\varphi_1, \dots, \varphi_n) \text{ iff } \mathcal{P}(\{w \in Run(s) \mid w[\varphi_1, \dots, \varphi_n] \in L(\mathcal{B})\}) \bowtie \varrho$$

Here $w[\varphi_1, \dots, \varphi_n]$ is the infinite word over the alphabet $2^{\{1,\dots,n\}}$ where $w[\varphi_1, \dots, \varphi_n](i)$ is the set of all $1 \leq j \leq n$ such that $w(i) \models^\eta \varphi_j$. Let us note that the set of runs $\{w \in Run(s) \mid w[\varphi_1, \dots, \varphi_n] \in L(\mathcal{B})\}$ is indeed $\mathcal{P}$-measurable in the above introduced probability space $(Run(s), \mathcal{F}, \mathcal{P})$, and hence the definition of PECTL* semantics makes sense for all PECTL* formulae. In the rest of this paper, we often write $s \models \varphi$ instead of $s \models^\eta \varphi$ when η is clear from the context.

The syntax of PECTL* is rather terse and does not include conventional temporal operators such as $\mathcal{G}$ and $\mathcal{F}$. This is convenient for our purposes (proofs become simpler), but the intuition about the actual expressiveness of PECTL* and its sublogics is lost. As a little compensation, we show how to encode conjunction, disjunction, and temporal connectives $\mathcal{G}, \mathcal{F}, \mathcal{U}$ and $\mathcal{X}$ (the negation of φ corresponds to $\mathcal{B}^{=0}_{\wedge}(\varphi, \varphi)$).



For example, the formula $\varphi_1 \wedge \mathcal{F}^{=1}\varphi_2$ is then a shortcut for $\mathcal{B}^{=1}_{\wedge}(\varphi_1, \mathcal{B}^{=1}_{\mathcal{F}}(\varphi_2))$, and in our examples we stick to this simpler notation. The PCTL fragment of PECTL* is obtained by restricting the syntax to $\varphi ::= a \mid \neg a \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathcal{X}^{\bowtie \varrho}\varphi \mid \varphi_1 \mathcal{U}^{\bowtie \varrho} \varphi_2$. We also write $a \Rightarrow \varphi$ instead of $\neg a \vee \varphi$.

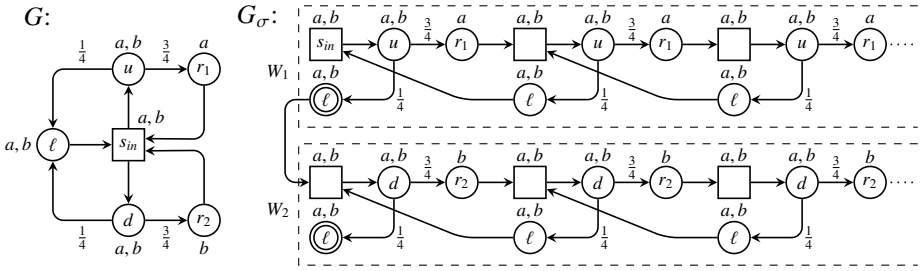
3 The Result

As we have already noted, qualitative PECTL* formulae are closed under negation, and hence it suffices to consider only the controller synthesis problem (a solution for the verification problem is then obtained as a trivial corollary). Formally, the controller synthesis problem for qualitative PECTL* objectives and HC strategies is specified as follows:

Problem: Controller synthesis for qualitative PECTL* objectives and HC strategies.
Instance: A MDP $G = (V, E, (V_\Box, V_\bigcirc), Prob)$, a partition (V_D, V_R) of $V_\Box$, $s_{in} \in V$, $\eta : V \to 2^{Ap}$, and a qualitative PECTL* formula φ. (The η is extended to all $vs \in V^*V$ by stipulating $\eta(vs) = \eta(s)$.)
Question: Is there a HC strategy σ such that $s_{in} \models^\eta \varphi$ in G_σ ?

Our solution of the problem (see Theorem 1) is based on one central idea underpinned by many technically involved observations which "make it work". Roughly speaking, a given objective φ is first split into finitely many "sub-objectives" $\varphi_1, \ldots, \varphi_n$ that are achievable by effectively constructible *finite-memory* strategies $\sigma_1, \ldots, \sigma_n$. Then, the finite-memory strategies $\sigma_1, \ldots, \sigma_n$ are combined into a single one-counter strategy σ that achieves the original objective φ.

Let us illustrate this idea on a concrete example. Consider the MDP G of the following figure, where s_{in} is Dirac.



The winning objective is the formula $\varphi \equiv \varphi_a \wedge \varphi_b$, where $\varphi_a \equiv \mathcal{G}^{=1}(a \Rightarrow \mathcal{G}^{>0}a)$ and $\varphi_b \equiv \mathcal{G}^{=1}(b \Rightarrow \mathcal{G}^{>0}b)$. The validity of a, b in the vertices of G is also indicated in the figure. In this case, the "sub-objectives" are the formulae φ_a and φ_b, that are achievable by memoryless strategies σ_u and σ_d that always select the transitions $s_{in} \to u$ and $s_{in} \to d$, respectively. Obviously, $s_{in} \models \varphi_a$, $s_{in} \not\models \varphi_b$ in G_{σ_u}, and similarly $s_{in} \models \varphi_b$, $s_{in} \not\models \varphi_a$ in G_{σ_d}. Hence, none of these two strategies achieves the objective φ (in fact, one can easily show that φ is not achievable by any *finite-memory* strategy). Now we show how to combine the strategies σ_u and σ_d into a single one-counter strategy σ such that $s_{in} \models \varphi$ in G_σ.

Let us start with an informal description of the strategy σ. During the whole play, the *mode* of σ is either σ_u or σ_d, which means that σ makes the same decision as σ_u or σ_d, respectively. Initially, the mode of σ is σ_u, and the counter is initialized to 1. If (and only if) the counter reaches zero, the current mode is switched to the other mode, and the counter is set to 1 again. This keeps happening ad infinitum. During the play, the counter is modified as follows: each visit to ℓ decrements the counter, and each visit to r_1 or r_2 increments the counter.

Obviously, σ is a one-counter strategy. However, it is not so obvious why it works. The structure of the play G_σ is indicated in the figure above, where the initial state is labeled s_{in} (the actual graph of G_σ is an infinite tree obtained by *unfolding* the graph shown in the figure). The play G_σ closely resembles an "infinite sequence" $W_1, W_2, \ldots$ of one-dimensional random walks. In each W_i, the probability of going right is $\frac{3}{4}$, the probability of going left is $\frac{1}{4}$, and whenever the "left end" is entered (i.e., the counter

becomes zero), the next random walk W_{i+1} in the sequence is started. All W_i, where i is odd/even, correspond to the σ_u/σ_d mode. In the above figure, only W_1 and W_2 are shown, and their "left ends" are indicated by double circles. By applying standard results about one-dimensional random walks, we can conclude that for every state s of every W_i that is not a "left end", the probability of reaching the "left end" of W_i from s is strictly less than one. Now it suffices to realize the following:

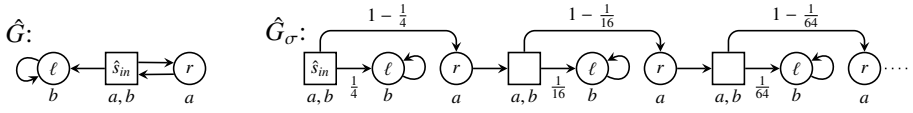
- Let s be a state of W_i, where i is odd. Then $s \models \mathcal{G}^{>0}a$ in G_σ. This is because all states of W_i satisfy a, and the probability of reaching the "left end" of W_i from s is strictly less than one. For the same reason, all states of W_i, where i is even, satisfy the formula $\mathcal{G}^{>0}b$.
- Let s be a state of W_i, where i is odd, such that $s \models b$. Then $s \models \mathcal{G}^{>0}b$. This is because there is a *finite* path to a state s' in W_{i+1} along which b holds (this path leads through the "left end" of W_i). Since $s' \models \mathcal{G}^{>0}b$ (as justified in the previous item), we obtain that $s \models \mathcal{G}^{>0}b$. For the same reason, for every state s of every W_i such that i is even and $s \models a$ we have that $s \models \mathcal{G}^{>0}a$.

Both claims can easily be verified by inspecting the figure on the previous page. Hence, $s_{in} \models \varphi$ in G_σ as needed.

The main idea of "combining" the constructed finite-memory strategies $\sigma_1, \ldots, \sigma_n$ into a single one-counter strategy σ is illustrated quite well by the above example. One basically "rotates" among the strategies $\sigma_1, \ldots, \sigma_n$ ad infinitum. Of course, some issues are (over)simplified in this example. In particular,

- in general, the "sub-objectives" do *not* correspond to subformulae of φ. They depend both on a given φ and a given G;
- the events counted in the counter are not just individual visits to selected vertices;
- the individual random walks obtained by "rotating" the modes $\sigma_1, \ldots, \sigma_n$ do not form an infinite sequence but an infinite tree;
- in the previous example, the only way how to leave a given W_i is to pass through its "left end". In general, each state of a given W_i can have a transition which "leaves" W_i. However, these transitions have progressively smaller and smaller probabilities so that the probability of "staying within" W_i remains positive.

Note that the last item explains why the definition of one-counter strategy admits the use of "exponentially small" probabilities that depend on the current counter value (the one-counter strategy defined in the above example only tested the counter for zero). To demonstrate that the use of "exponentially small" probabilities is unavoidable, consider the MDP $\hat{G}$ of the following figure, where $\hat{s}_{in}$ is randomizing.



Let $\hat{\varphi} \equiv \mathcal{G}^{>0}(a \wedge (b \Rightarrow \mathcal{G}^{>0}b))$. We claim that every HC strategy κ which achieves the objective $\hat{\varphi}$ must satisfy the following: Let K be the set of all probabilities that are assigned to the edge $\hat{s}_{in} \to \ell$ in the play $\hat{G}_\kappa$. Then all elements of K are positive and $\inf(K) = 0$, otherwise the formula $\hat{\varphi} \equiv \mathcal{G}^{>0}(a \wedge (b \Rightarrow \mathcal{G}^{>0}b))$ would not hold. Hence,

κ must inevitably assign "smaller and smaller" positive probability to the edge $\hat{s}_{in} \to \ell$. This is achievable by a one-counter strategy $\hat{\sigma}$ where $\hat{\sigma}(v\hat{s}_{in})$ assigns $4^{-c(v\hat{s}_{in})}$ to $\hat{s}_{in} \to \ell$ and $1 - 4^{-c(v\hat{s}_{in})}$ to $\hat{s}_{in} \to r$, where $c(v\hat{s}_{in})$ is the number of occurrences of $\hat{s}_{in}$ in $v\hat{s}_{in}$. The play $\hat{G}_{\hat{\sigma}}$ is also shown in the above figure. It is easy to see that $\hat{s}_{in} \models \mathcal{G}^{>0}(a \wedge (b \Rightarrow \mathcal{G}^{>0} b))$ in $\hat{G}_{\hat{\sigma}}$.

A Formal Proof of the Result. Due to space constraints, we cannot give a full proof of Theorem 1 (it can be found in [5]). Here we only outline the structure of our proof, identify the milestones, and try to "map" the vague notions introduced earlier to precise technical definitions. Roughly speaking, our proof has two major phases.

(1) The controller synthesis problem for qualitative PECTL* objectives and HC strategies is reduced to the controller synthesis problem for "consistency objectives" and HC strategies. The "consistency objectives" are technically simpler than PECTL* objectives, and they in fact represent the very core of the whole problem.
(2) The controller synthesis problem for consistency objectives and HC strategies is solved.

The most important insights are concentrated in Phase (2). Our complexity results are based on a careful analysis of the individual steps which constitute Phase (1) and (2). Since all of our constructions are effective, one can also effectively construct the strategy for the original PECTL* objective by taking the strategy for the constructed consistency objective and modifying it accordingly.

We start by a formal definition of consistency objectives. First, we need to recall the notion of a *deterministic Muller automaton*, which is a tuple $\mathcal{M} = (Q, \Sigma, \delta, A)$ where Q is a finite set of control states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to Q$ is a transition function (which is extended to the elements of $Q \times \Sigma^*$ in the standard way), and $A \subseteq 2^Q$ is a set of accepting sets. A *computation* of $\mathcal{M}$ on $w \in \Sigma^\omega$ initiated in $q \in Q$ is the (unique) infinite sequence of control states $\gamma = q_0, q_1, \ldots$ such that $q_0 = q$ and $\delta(q_i, w(i)) = q_{i+1}$ for all $i \in \mathbb{N}_0$. A computation γ is *accepting* if $\inf(\gamma) \in A$, where $\inf(\gamma)$ is the set of all control states that occur infinitely often in γ.

Definition 5 (Consistency objective). *Let $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ be a MDP, $s_{in} \in V$ an initial vertex, and (V_D, V_R) a partition of $V_\Box$. A* consistency objective *is a triple $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$, where $\mathcal{M} = (Q, V, \delta, A)$ is a deterministic Muller automaton over the alphabet V, $(Q_{>0}, Q_{=1})$ is a partition of Q s.t. for all $q \in Q_{>0}$, $q' \in Q_{=1}$ and $w \in V^*$ we have that $\delta(q, w) \in Q_{>0}$ and $\delta(q', w) \in Q_{=1}$, and $L : V \to 2^Q$ is a labeling.*

Let σ be a HC strategy, and let $G_\sigma^{s_{in}}$ be the play G_σ restricted to states that are reachable from s_{in} in G_σ. For every state vs of $G_\sigma^{s_{in}}$ and every $q \in Q$, let $Acc(vs, q)$ be the set of all runs $v_0 s_0, v_1 s_1, \ldots$ initiated in vs such that for every $i \in \mathbb{N}_0$ we have that $\delta(q, s_0 \cdots s_i) \in L(s_{i+1})$ and the computation of $\mathcal{M}$ on $s_0 s_1 \cdots$ initiated in q is accepting. For every comparison $\bowtie$ and every rational constant ϱ, we write $vs \models_\sigma Acc^{\bowtie\varrho}(q)$ if $\mathcal{P}(Acc(vs, q)) \bowtie \varrho$ in G_σ. A HC strategy σ achieves the consistency objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ if for every state $vs \in V^*V$ of the play $G_\sigma^{s_{in}}$, every $q \in Q$, and every $\bowtie\varrho \in \{=1, >0\}$ we have that if $q \in Q_{\bowtie\varrho} \cap L(s)$, then $vs \models Acc^{\bowtie\varrho}(q)$.

Phase (1). Let $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ be a MDP, (V_D, V_R) a partition of $V_\Box$, $s_{in} \in V$, $\eta : V \to 2^{Ap}$ a valuation, and φ a qualitative PECTL* formula. We construct a MDP

$G' = (V', E', (V'_\Box, V'_\bigcirc), Prob')$, a partitioning (V'_D, V'_R), a vertex $s'_{in} \in V$, and a consistency objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ such that the existence of a HC strategy σ where $s_{in} \models^\eta \varphi$ in G_σ implies the existence of a HC strategy π that achieves the objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ in ${G'}_\pi^{s'_{in}}$, and vice versa. The size of G' is polynomial in $|G|$ and exponential in $|\varphi|$.

The construction is partly based on ideas of [4] and proceeds as follows. First, all Büchi automata that occur in φ are replaced with equivalent deterministic Muller automata. The resulting formula is further modified so that all probability bounds take the form ">0" or "=1" (to achieve that, some of the deterministic Muller automata may be complemented). Thus, we obtain a formula φ'. Let $M^{>0}$ and $M^{=1}$ be the sets of all deterministic Muller automata that appear in φ' with the probability bound >0 and =1, respectively. The automaton $\mathcal{M}$ is essentially the disjoint union of all automata in $M^{>0}$ and $M^{=1}$. The sets $Q_{>0}$ and $Q_{=1}$ are unions of sets of control states of all Muller automata in $M^{>0}$ and $M^{=1}$, respectively. The tricky part is the construction of G'. Intuitively, the MDP G' is the same as G, but several instances of Muller automata from $M^{>0} \cup M^{=1}$ are simulated "on the fly". Moreover, some "guessing" vertices are added so that a strategy can decide what "subformulae of φ'" are to be satisfied in a given vertex. The structure of G' itself does not guarantee that the commitments chosen by the strategy are fulfilled. This is done by the automaton $\mathcal{M}$ and the condition that $vs \models Acc^{\bowtie q}(q)$ for all $q \in Q_{\bowtie q} \cap L(s)$. (Intuitively, this condition says that the play ${G'}_\pi^{s'_{in}}$ is "consistent" with the commitments chosen in the guessing vertices.)

Phase (2). The controller synthesis problem for consistency objectives and HC strategies is solved in three steps:

(a) We solve the special case when the set $Q_{>0}$ (see Definition 5) is empty.
(b) We solve the special case when the strategy is *strictly randomizing* (see below), using the result of (a).
(c) We reduce the general (unrestricted) case to the special case of (b).

Now we describe the three steps in more detail. Let $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ be a MDP, $s_{in} \in V$ an initial vertex, (V_D, V_R) a partition of $V_\Box$, and $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ a consistency objective, where $\mathcal{M} = (Q, V, \delta, A)$.

As for step (a), the key insight is the following observation (the proposition holds under the non-restrictive assumption that for all $s, t \in V$ such that $(s, t) \in E$ and for all $p \in Q_{=1}$ such that $p \in L(s)$ we have $\delta(p, s) \in L(t)$):

Proposition 1. *Let us assume that $Q_{>0} = \emptyset$. Then the objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ is achievable by* some *HC strategy iff there is a HC strategy σ such that for every state vs of $G_\sigma^{s_{in}}$, every $p \in L(s) \cap Q_{=1}$, and almost all runs $v_0 s_0, v_1 s_1, \ldots$ initiated in vs there are $k \in \mathbb{N}_0$, $q \in Q$, and $X \in A$ such that $\delta(p, s_0 \cdots s_{k-1}) = q$ and almost all runs $\hat{v}_0 \hat{s}_0, \hat{v}_1 \hat{s}_1, \ldots$ initiated in $v_k s_k$ satisfy the following conditions: $\delta(q, \hat{s}_0 \cdots \hat{s}_j) \in X$ for every $j \in \mathbb{N}_0$, and for every $r \in X$ there are infinitely many $j \in \mathbb{N}_0$ such that $\delta(q, \hat{s}_0 \cdots \hat{s}_j) = r$.*

In other words, if $Q_{>0} = \emptyset$, then the objective is achievable by a strategy which simply "guesses" an appropriate moment and an appropriate $X \in A$, and then it suffices to verify that the guess was correct, i.e., almost all simulated computations of $\mathcal{M}$ visit

only the states of X and each of them is visited infinitely often. This can be effectively implemented by a qualitative Büchi objective, and hence we can rely on the existing algorithms (see Section 1). At this point, there is no need for infinite memory.

In step (b), we concentrate on another special case where both $Q_{>0}$ and $Q_{=1}$ may be non-empty, but the set of strategies is restricted to *strictly randomizing HC (srHC)* strategies. A srHC strategy is a HC strategy σ such that $\sigma(vs)$ assigns a positive probability to *all* outgoing edges whenever $s \in V_R$. This is perhaps the most demanding part of the whole construction, where we formalize the notion of "sub-objective" mentioned earlier, invent the technique of "rotating" the finite-memory strategies for the individual "sub-objectives", etc. The main technical ingredient is the notion of *entry point*.

Definition 6. *A set $X \subseteq V$ is* closed *if each $s \in X$ has at least one immediate successor in X, and every $s \in X$ which is stochastic or randomizing has all immediate successors in X. Each closed X determines a* sub-MDP *$G|X$ which is obtained from G by restricting the set of vertices to X.*

Let X be a closed set. An entry point *for X is a pair $(s, q) \in X \times Q_{>0}$ for which there is a HD strategy ξ in $G|X$ satisfying the following conditions:*

1. *$s \models_\xi Acc^{=1}(q)$;*
2. *for every state vt of $(G|X)^s_\xi$ and every $p \in L(t) \cap Q_{=1}$ we have that $vt \models_\xi Acc^{=1}(p)$;*
3. *for all states vt of $(G|X)^s_\xi$ and all $p \in L(t) \cap Q_{>0}$ we have the following: if there is no state of $V^*V_\bigcirc$ reachable from vt in $(G|X)^s_\xi$, then either $wt \models_\xi Acc^{=1}(p)$, or there is a finite path $v_0t_0, \ldots, v_kt_k$ initiated in vt such that $t_k \in V_R$ and t_k has two outgoing edges $(t_k, r_1), (t_k, r_2) \in E$ such that $\xi(v_kt_k)$ selects the edge (t_k, r_1) and $\delta(p, t_0 \cdots t_k) \in L(r_2) \cap Q_{>0}$.*

Intuitively, entry points correspond to the finitely many "sub-objectives" discussed earlier. The next step is to show that the set of all entry points for a given closed set X can be effectively computed in time which is polynomial in $|G|$ and exponential in $|Q|$. Further, we show that for each entry point (s, q) one can effectively construct a *finite-memory deterministic* strategy $\xi(s, q)$ which has the same properties as the HD strategy ξ of Definition 6 (this is what we meant by "achieving a sub-objective"). Here we use the results of step (a). Technically, the key observation of step (b) is the following proposition (this proposition holds under some technical assumptions that are not listed explicitly here).

Proposition 2. *The consistency objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ is achievable by a srHC strategy σ iff there is a closed $X \subseteq V$ such that $s_{in} \in X$ and for all $s_0 \in X$ and $q_0 \in L(s_0) \cap Q_{>0}$ there is finite sequence $(s_0, q_0), \ldots, (s_n, q_n)$ such that $(s_i, s_{i+1}) \in E$, $q_i \in L(s_i)$ and $\delta(q_i, s_i) = q_{i+1}$ for all $0 \leq i < n$, and (s_n, q_n) is an entry point for X.*

Both directions of the proof require effort, and the "if" part can safely be declared as difficult. This is where we introduce the counter and "rotate" the $\xi(s, q)$ strategies for the individual entry points to obtain a srHC strategy that achieves the objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$. This part is highly non-trivial and relies on many subtle observations. Nevertheless, the whole construction is effective and admits a detailed complexity analysis.

Step (c) is relatively simple (compared to step (a) and particularly step (b)). The 2-**EXPTIME** lower bound for qualitative PECTL* objectives also requires a proof (the

bound does not follow from the previous work). Here we use a standard technique for simulating an exponentially bounded alternating Turing machine, employing some ideas presented in [2]. The **EXPTIME** lower bound for qualitative PCTL has been established already in [3].

References

1. C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski. Controller synthesis for probabilistic systems. In *Proceedings of IFIP TCS'2004*, pp. 493–506. Kluwer, 2004.
2. T. Brázdil, V. Brožek, and V. Forejt. Branching-time model-checking of probabilistic pushdown automata. In *Proceedings of INFINITY'2007*, pp. 24–33, 2007.
3. T. Brázdil, V. Brožek, V. Forejt, and A. Kučera. Stochastic games with branching-time winning objectives. In *Proceedings of LICS 2006*, pp. 349–358. IEEE, 2006.
4. T. Brázdil and V. Forejt. Strategy synthesis for Markov decision processes and branching-time logics. In *Proceedings of CONCUR 2007*, vol. 4703 of *LNCS*, pp. 428–444. Springer, 2007.
5. T. Brázdil, V. Forejt, and A. Kučera. Controller synthesis and verification for Markov decision processes with qualitative branching time objectives. Technical report FIMU-RS-2008-05, Faculty of Informatics, Masaryk University, 2008.
6. K. Chatterjee, L. de Alfaro, and T. Henzinger. Trading memory for randomness. In *Proceedings of 2nd Int. Conf. on Quantitative Evaluation of Systems (QEST'04)*, pp. 206–217. IEEE, 2004.
7. K. Chatterjee, R. Majumdar, and T. Henzinger. Markov decision processes with multiple objectives. In *Proceedings of STACS 2006*, vol. 3884 of *LNCS*, pp. 325–336. Springer, 2006.
8. L. de Alfaro. Quantitative verification and control via the mu-calculus. In *Proceedings of CONCUR 2003*, vol. 2761 of *LNCS*, pp. 102–126. Springer, 2003.
9. E.A. Emerson. Temporal and modal logic. *Handbook of TCS*, B:995–1072, 1991.
10. K. Etessami, M. Kwiatkowska, M. Vardi, and M. Yannakakis. Multi-objective model checking of Markov decision processes. In *Proceedings of TACAS 2007*, vol. 4424 of *LNCS*, pp. 50–65. Springer, 2007.
11. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1996.
12. E. Grädel. Positional determinacy of infinite games. In *Proceedings of STACS 2004*, vol. 2996 of *LNCS*, pp. 4–18. Springer, 2004.
13. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
14. J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
15. A. Kučera and O. Stražovský. On the controller synthesis for finite-state Markov decision processes. In *Proceedings of FST&TCS 2005*, vol. 3821 of *LNCS*, pp. 541–552. Springer, 2005.
16. M.L. Puterman. *Markov Decision Processes*. Wiley, 1994.
17. C. Stirling. Modal and temporal logics. *Handbook of Logic in Comp. Sci.*, 2:477–563, 1992.
18. M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of FOCS'85*, pp. 327–338. IEEE, 1985.
19. I. Walukiewicz. A landscape with games in the background. In *Proceedings of LICS 2004*, pp. 356–366. IEEE, 2004.