# FI MU

**Faculty of Informatics**
**Masaryk University Brno**

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

**by**

**Tomáš Brázdil**
**Vojtěch Forejt**
**Antonín Kučera**

**FI MU Report Series** **FIMU-RS-2008-05**

 **April 2008**

Publications in the FI MU Report Series are in general accessible via WWW:

http://www.fi.muni.cz/reports/

Further information can be obtained by contacting:

Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic

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

Tomáš Brázdil
Faculty of Informatics, Masaryk University
brazdil@fi.muni.cz

Vojtěch Forejt
Faculty of Informatics, Masaryk University
forejt@fi.muni.cz

Antonín Kučera
Faculty of Informatics, Masaryk University
tony@fi.muni.cz

May 11, 2009

**Abstract**

We show that 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.

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

# 1 Introduction

A *Markov decision process (MDP)* [22, 17] is a finite directed graph $G = (V, E, (V_\square, V_\bigcirc), Prob)$ where the vertices of $V$ are partitioned into *non-deterministic* and *stochastic* subsets (denoted $V_\square$ 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., [25]). Here, non-deterministic vertices are used to model the environment, unpredictable users, process scheduler, etc., while stochastic vertices model 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.
- *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 $vs \in V^* V_\square$ 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[1] 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. In the controller synthesis problem, we usually want to find the simplest possible strategy (ideally MD) that achieves a given objective,

[1] A probability distribution is Dirac if it assigns 1 to exactly one element.

because this makes the controller easy to implement. In the verification problem, we want to know whether a given objective holds for unrestricted (i.e., HR) 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 [22, 17]. 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. Linear-time properties classify each run (i.e., an infinite path) as good or bad according to some criterion, and the associated *quantitative* (or *qualitative*) objective is to maximize the probability of good runs (or to make this probability equal to 1, respectively). There are many results concerning the complexity of algorithms for solving the corresponding controller synthesis and verification problems, and also results that classify the type of strategies that are needed to achieve a given objective. For a more detailed information, we refer to recent overviews such as [18, 26, 9, 7].

Another class of temporal objectives studied in the literature are *linear-time multi-objectives* [13, 8], which are Boolean combinations of linear-time objectives. Since there can be trade-offs among the individual linear-time objectives, i.e., satisfying property $P_1$ with high probability may necessitate satisfying property $P_2$ with low probability, the corresponding controller synthesis and verification problems are not directly reducible to the linear-time case. Strategies for linear-time multi-objectives may require both randomization and memory, even in the qualitative subcase [13]. The corresponding controller synthesis problem is solvable in time which is polynomial in the size of MDP and doubly exponential in the size of the objective.

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 even PECTL$^*$ [19]. These logics are obtained from their non-probabilistic counterparts CTL, CTL$^*$, and ECTL$^*$ (see, e.g., [10, 24]) by replacing the universal and existential path quantifiers with the probabilistic operator $\mathcal{P}^{\bowtie\rho}$, where $\rho$ is a rational constant and $\bowtie$ is a comparison such as $\leq$ or $>$. Intuitively, the formula $\mathcal{P}^{\bowtie\rho}\varphi$ says "the probability of all runs that satisfy $\varphi$ is $\bowtie$-related to $\rho$". If the probability bound $\rho$ 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 [21], 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 [4], 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 $\Sigma_1^1$ level of the analytical hierarchy). This result holds even for a fragment of PCTL where the set of modal connectives is restricted to $\mathcal{P}^{=1}\mathcal{F}$, $\mathcal{P}^{=1}\mathcal{G}$, $\mathcal{P}^{>0}\mathcal{F}$, and $\mathcal{P}^{=5/8}\mathcal{F}$. The role of the only *quantitative* connective $\mathcal{P}^{=5/8}\mathcal{F}$ is rather special[2], and one consequence of the results presented in this paper is that the proof *cannot* be completed without quantitative connectives. In [4], 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 [6] where the memory requirements for objectives of various fragments of qualitative PCTL are classified in a systematic way (it is noted already in [4] that strategies for qualitative PCTL objectives may require *infinite* memory).

**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 $G = (V, E, (V_\square, V_\bigcirc), Prob)$ be a MDP and let $(V_D, V_R)$ be a partitioning of $V_\square$ into the subsets of *Dirac* and *randomizing* vertices. An *HC strategy* is a HR strategy $\sigma$ 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 $V_D = V_\square$, respectively. Nevertheless, our solution covers also the cases when $\emptyset \neq V_D \neq V_\square$. Now we can formulate the main result of this paper.

[2]There is no "magic" in the number 5/8 itself—any rational constant strictly between 0 and 1 would suffice for purposes of this proof.

**Theorem 1.1.** *Let* $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ *be an MDP,* $(V_D, V_R)$ *a partitioning of* $V_\Box$*, and* $\varphi$ *a qualitative PECTL** *formula.*

- *The problem whether there is a HC strategy that achieves the objective* $\varphi$ *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* $\varphi$ *is achievable by some HC strategy, then it is also achievable by a* one-counter *strategy (see Definition 2.3). 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* $\varphi$ *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 and unification 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 [4].

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. Considering the complexity and expressive power of qualitative PECTL* objectives, it is somewhat surprising that a single non-negative integer counter suffices in all these cases. Although the above stated theorem itself does not give any explanation of *what* is actually counted in the counter and why, the proof *does* bring a good understanding of this issue. We try to give some basic intuition in Section 3. 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 (the corresponding controller synthesis algorithms are usually graph-theoretic).

From the practical point of view, the main point of our complexity analysis is the fact that both controller synthesis and verification problems for qualitative branching-time objectives are solvable in time which is *polynomial* in the number of vertices of a given MDP. An empirically confirmed fact (which is well-known in the model-checking community) is that the main limiting factor for effective analysis of computer systems is the size of the *model* (i.e., the size of the MDP), while specifications (temporal formulae) tend to be small. Hence, our results in fact show that the controller synthesis and verification problems for MDPs with qualitative branching-time objectives *are* tractable, and future software tools for automatic analysis of probabilistic systems can be equipped with this functionality.

# 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 $\Sigma$ is denoted $\Sigma^*$, and the set of all infinite words over $\Sigma$ is denoted $\Sigma^\omega$. 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 $\Sigma^+$ to denote the set $\Sigma^* \smallsetminus \{\varepsilon\}$ where $\varepsilon$ 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 $\omega$. Given a word (finite or infinite) over $\Sigma$, 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 $\sigma$-*field* over a set $\Omega$ is a set $\mathcal{F} \subseteq 2^\Omega$ that includes $\Omega$ and is closed under complement and countable union. A *probability space* is a triple $(\Omega, \mathcal{F}, \mathcal{P})$ where $\Omega$ is a set called *sample space*, $\mathcal{F}$ is a $\sigma$-field over $\Omega$ 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 2.1** (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* $\mathit{Prob}$ *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* $\sum_{s \to t} Prob(s \to t) = 1$ *(as usual, we write* $s \xrightarrow{x} t$ *instead of* $Prob(s \to t) = x$*).*

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 $1 \leq i < length(w)$. 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 $Run[M](w)$. When $M$ is clear from the context, we write $Run(w)$ instead of $Run[M](w)$.

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

**Definition 2.2** (Markov Decision Process)**.** *A* Markov decision process (MDP) *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 positive 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* $s \in V_\bigcirc$*. For technical convenience, we require that each vertex has at least one and at most two outgoing edges.*

Let $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ 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 $\sigma$ determines a unique Markov chain $G_\sigma$ 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 $\sigma$ 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 $\sigma$ 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 $\sigma$ is specified by a deterministic finite-state automaton $\mathcal{A}$ over the input alphabet $V$ (see, e.g., [20]), 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 2.3** (One counter automaton)**.** *A* one counter automaton *is a tuple* $\mathcal{C} = (Q, \Sigma, q_{in}, \delta^{=0}, \delta^{>0})$ *where* $Q$ *is a finite set of* control states*,* $\Sigma$ *is a finite* input alphabet*,* $q_{in} \in Q$ *is the* initial control state*, and*

$$\delta^{=0} : Q \times \Sigma \to Q \times \{0, 1\}, \quad \delta^{>0} : Q \times \Sigma \to Q \times \{0, 1, -1\}$$

*are* transition functions*. The set of* configurations *of* $\mathcal{C}$ *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 $\sigma$ for which there is a one-counter automaton $\mathcal{C} = (Q, V, q_{in}, \delta^{=0}, \delta^{>0})$ 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[3], $\sigma(vs)$ 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 $2^{\{1,\ldots,n\}}$ and the symbols assigned to transitions in the automaton are interpreted in a special way. These differences are not fundamental but technically convenient.

[3]In Definition 2.2, we require that each vertex has either one or two outgoing edges. Randomized vertices with just one outgoing edge are not interesting because every strategy has to assign probability 1 to the only available edge.

**Definition 2.4** (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, $\delta : Q \times 2^{\{1,\ldots,n\}} \to 2^Q$ *is the* transition function, *and* $A \subseteq Q$ *is the set of* accepting states. *A given infinite word* $w$ *over the alphabet* $2^{\{1,\ldots,n\}}$ *is* accepted *by* $\mathcal{B}$ *if there is an* accepting computation *of* $\mathcal{B}$ *for* $w$, *i.e., an infinite sequence of states* $q_0, q_1, \ldots$ *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* $\alpha_i \in 2^{\{1,\ldots,n\}}$ *such that* $q_{i+1} \in \delta(q_i, \alpha_i)$ *and* $\alpha_i \subseteq w(i)$. *The set of all infinite words over* $2^{\{1,\ldots,n\}}$ *accepted by* $\mathcal{B}$ *is denoted* $L(\mathcal{B})$.

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

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

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

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

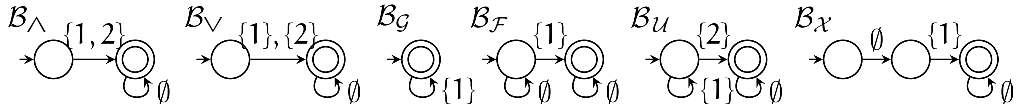
$$\begin{aligned}
s \models^{\nu} a &\quad \text{iff} \quad a \in \nu(s) \\
s \models^{\nu} \neg a &\quad \text{iff} \quad a \notin \nu(s) \\
s \models^{\nu} \mathcal{B}^{\bowtie\rho}(\varphi_1, \ldots, \varphi_n) &\quad \text{iff} \quad \mathcal{P}(\{w \in Run(s) \mid w[\varphi_1, \ldots, \varphi_n] \in L(\mathcal{B})\}) \bowtie \rho
\end{aligned}$$

Here $w[\varphi_1, \ldots, \varphi_n]$ is the infinite word over the alphabet $2^{\{1,\ldots,n\}}$ where $w[\varphi_1, \ldots, \varphi_n](i)$ is the set of all $1 \leq j \leq n$ such that $w(i) \models^{\nu} \varphi_j$. Let us note that the set of runs $\{w \in Run(s) \mid w[\varphi_1, \ldots, \varphi_n] \in L(\mathcal{B})\}$ 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 $s \models^{\nu} \varphi$ when $\nu$ 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[4], we show how to encode negation, conjunction, disjunction, and

[4] A reader familiar with the "standard" definition of Büchi automata can easily become confused by these examples. In this case, we recommend to read Definition 2.4 carefully.

temporal connectives $\mathcal{G}$, $\mathcal{F}$, $\mathcal{U}$ and $\mathcal{X}$. Negation can be encoded by simply changing a comparison; for example, formula $\mathcal{B}^{>0.3}\varphi$ is the negation of the formula $\mathcal{B}^{\leq 0.3}\varphi$. The other operators can be encoded using the automata from the following figure.




For example, the formula $\varphi_1 \wedge \mathcal{F}^{=1}\varphi_2$ is then a shortcut for $\mathcal{B}_\wedge^{=1}(\varphi_1, \mathcal{B}_\mathcal{F}^{=1}(\varphi_2))$. In our examples we stick to this more readable notation. Now the PCTL fragment of PECTL* is obtained by restricting the syntax to

$$\varphi \quad ::= \quad a \mid \neg a \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathcal{X}^{\bowtie\rho}\varphi \mid \varphi_1 \mathcal{U}^{\bowtie\rho}\varphi_2$$

We 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:

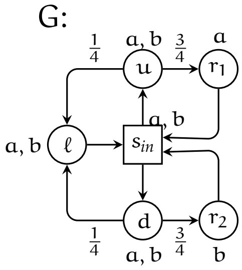
**Problem:** Controller synthesis for qualitative PECTL* objectives and HC strategies.

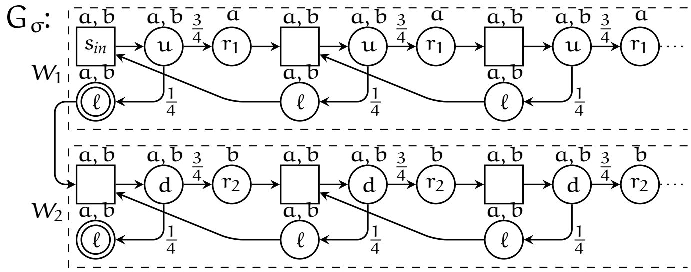
**Instance:** A MDP $G = (V, E, (V_\Box, V_\bigcirc), \mathit{Prob})$, a partition $(V_D, V_R)$ of $V_\Box$, $s_{in} \in V$, $\nu : V \to 2^{Ap}$, and a qualitative PECTL* formula $\varphi$. (The $\nu$ is extended to all $vs \in V^*V$ by stipulating $\nu(vs) = \nu(s)$.)

**Question:** Is there a HC strategy $\sigma$ such that $s_{in} \models^\nu \varphi$ in $G_\sigma$?

Our solution of the problem (see Theorem 1.1) is based on one central idea underpinned by many technically involved observations which "make it work". Roughly speaking, a given objective $\varphi$ 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 $\sigma$ that achieves the original objective $\varphi$.

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 $\varphi_a$ and $\varphi_b$, that are achievable by memoryless strategies $\sigma_u$ and $\sigma_d$ 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 $G_{\sigma_u}$, and similarly $s_{in} \models \varphi_b$, $s_{in} \not\models \varphi_a$ in $G_{\sigma_d}$. Hence, none of these two strategies achieves the objective $\varphi$ (in fact, one can easily show that $\varphi$ is not achievable by any *finite-memory* strategy). Now we show how to combine the strategies $\sigma_u$ and $\sigma_d$ into a single one-counter strategy $\sigma$ such that $s_{in} \models \varphi$ in $G_\sigma$.

Let us start with an informal description of the strategy $\sigma$. During the whole play, the *mode* of $\sigma$ is either $\sigma_u$ or $\sigma_d$, which means that $\sigma$ makes the same decision as $\sigma_u$ or $\sigma_d$, respectively. Initially, the mode of $\sigma$ is $\sigma_u$, 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 $\ell$ decrements the counter, and each visit to $r_1$ or $r_2$ increments the counter.

Obviously, $\sigma$ is a one-counter strategy. However, it is not so obvious why it works. The structure[5] of the play $G_\sigma$ is indicated in the figure above, where the initial state is labeled $s_{in}$. The play $G_\sigma$ 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 $\sigma_u/\sigma_d$ 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

[5]Let us note that the actual graph of $G_\sigma$ is an infinite tree obtained by *unfolding* the structure shown in the figure. However, this does not influence our arguments.

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 $G_\sigma$. 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 page 11. Hence, $s_{in} \models \varphi$ in $G_\sigma$ as needed.

The main idea of "combining" the constructed finite-memory strategies $\sigma_1, \ldots, \sigma_n$ into a single one-counter strategy $\sigma$ 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,

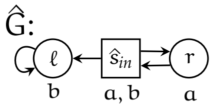
- in general, the "sub-objectives" do *not* correspond to subformulae of $\varphi$. They depend both on a given $\varphi$ and a given $G$;

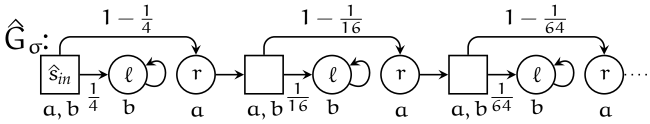
- the events counted in the counter are not just individual visits to selected vertices;

- 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 $\kappa$ 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, $\kappa$ 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 $\hat{\sigma}(\nu\hat{s}_{in})$ assigns $4^{-c(\nu\hat{s}_{in})}$ to $\hat{s}_{in} \to \ell$ and $1 - 4^{-c(\nu\hat{s}_{in})}$ to $\hat{s}_{in} \to r$, where $c(\nu\hat{s}_{in})$ is the number of occurrences of $\hat{s}_{in}$ in $\nu\hat{s}_{in}$. 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 $\hat{G}_{\hat{\sigma}}$.

## 3.1 A Proof of the Result

Due to space constraints, we cannot give a full proof of Theorem 1.1 in the main body of the paper. 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. The presented notes should also provide basic "guidelines" for reading the full technical exposition given in the appendix. 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, $\Sigma$ 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[6]. 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 $\gamma$ is *accepting* if $\inf(\gamma) \in A$, where $\inf(\gamma)$ is the set of all control states that occur infinitely often in $\gamma$.

**Definition 3.1** (Consistency objective). *Let* $G = (V, E, (V_\Box, V_\bigcirc), \mathit{Prob})$ *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* $Q$ *such that 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 $\sigma$ be a HC strategy, and let $G_\sigma^{s_{in}}$ be the play $G_\sigma$ restricted to states that are reachable from $s_{in}$ in $G_\sigma$. For every state $vs$ of $G_\sigma^{s_{in}}$ and every $q \in Q$, let $\mathit{Acc}(vs, q)$ 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 $\rho$, we write $vs \models_\sigma \mathit{Acc}^{\bowtie\rho}(q)$ if $\mathcal{P}(\mathit{Acc}(vs, q)) \bowtie \rho$ in $G_\sigma$.

A HC strategy $\sigma$ 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 $\bowtie\rho \in \{=1, >0\}$ we have that if $q \in Q_{\bowtie\rho} \cap L(s)$, then $vs \models \mathit{Acc}^{\bowtie\rho}(q)$.

**Phase (1).** Let $G = (V, E, (V_\Box, V_\bigcirc), \mathit{Prob})$ be a MDP, $(V_D, V_R)$ a partition of $V_\Box$, $s_{in} \in V$, $\nu : V \to 2^{Ap}$ a valuation, and $\varphi$ a qualitative PECTL* formula. We construct a MDP $G' = (V', E', (V'_\Box, V'_\bigcirc), \mathit{Prob}')$, 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 HC strategy $\sigma$ where $s_{in} \models^\nu \varphi$ in $G_\sigma$ implies the existence of a HC strategy $\pi$ 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 [6] and proceeds as follows. First, all Büchi automata that occur in $\varphi$ 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 $\varphi'$. Let $M^{>0}$ and $M^{=1}$ be the sets of

[6]For the purposes of a complexity analysis, we represent $A$ in a special way that allows to perform complementation in polynomial time. Formal definition is presented in the beginning of Appendix A.

all deterministic Muller automata that appear in $\varphi'$ 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 $\varphi'$" are to be satisfied in a given vertex. The structure of $G'$ itself does not guarantee that the commitments chosen by a strategy are fulfilled. This is done by the automaton $\mathcal{M}$ and the condition that $\nu s \models \mathit{Acc}^{\bowtie\rho}(q)$ for all $q \in Q_{\bowtie\rho} \cap L(s)$. (Intuitively, this condition says that the play $G'^{s'_{in}}_{\pi}$ 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 two steps:

(a) We solve the special case when the strategy is *strictly randomizing*.

(b) We reduce the general (unrestricted) case to the special case of (a).

Now we describe the two steps in more detail. Let $G = (V, E, (V_\Box, V_\bigcirc), \mathit{Prob})$ 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 $\mathcal{M} = (Q, V, \delta, A)$.

In step (a), we concentrate on the 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 $\sigma$ such that $\sigma(\nu s)$ 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 3.2.** *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* $\xi$ *in* $G|X$ *satisfying the following conditions:*

*1.* $s \models_\xi \mathit{Acc}^{=1}(q)$;

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 $\xi$ of Definition 3.2 (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[7]:

**Proposition 3.3.** *The consistency objective* $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ *is achievable by a srHC strategy* $\sigma$ *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 (b) is relatively simple (compared to 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 presended in [2], where the techniques for encoding the necessary properties in qualitative PECTL* formulae were developed. The **EXPTIME** lower bound for qualitative PCTL has been established already in [4].

[7]The proposition holds when several technical assumptions are imposed. These assumptions do not cause any loss of generality and are presented at the beginning of Appendix B.2.

# 4 Future Work

There are at least two natural directions for future work. First, we may wonder whether similar results can be achieved in a more general setting of $2\frac{1}{2}$-player games, where the non-deterministic vertices are split into two subsets controlled by two players with opposite objectives. Another possible generalization is to consider classes of MDPs with infinitely many states. Some of the recent results achieved for probabilistic extensions of pushdown automata [11, 12, 5, 3] (or, equivalently, recursive state machines [15, 16, 14]) indicate that this is not completely hopeless, at least in some restricted cases.

# 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. Reachability in recursive Markov decision processes. In *Proceedings of CONCUR 2006*, vol. 4137 of *LNCS*, pp. 358–374. Springer, 2006.

[4] 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.

[5] T. Brázdil, J. Esparza, and A. Kučera. Analysis and prediction of the long-run behavior of probabilistic sequential programs with recursion. In *Proceedings of FOCS 2005*, pp. 521–530. IEEE, 2005.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] E.A. Emerson. Temporal and modal logic. *Handbook of TCS*, B:995–1072, 1991.

[11] J. Esparza, A. Kučera, and R. Mayr. Model-checking probabilistic pushdown automata. In *Proceedings of LICS 2004*, pp. 12–21. IEEE, 2004.

[12] J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: Expectations and variances. In *Proceedings of LICS 2005*, pp. 117–126. IEEE, 2005.

[13] 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.

[14] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proceedings of STACS 2005*, vol. 3404 of *LNCS*, pp. 340–352. Springer, 2005.

[15] K. Etessami and M. Yannakakis. Efficient qualitative analysis of classes of recursive Markov decision processes and simple stochastic games. In *Proceedings of STACS 2006*, vol. 3884 of *LNCS*, pp. 634–645. Springer, 2006.

[16] K. Etessami and M. Yannakakis. Recursive concurrent stochastic games. In *Proceedings of ICALP 2006*, vol. 4052 of *LNCS*, pp. 324–335. Springer, 2006.

[17] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1996.

[18] E. Grädel. Positional determinacy of infinite games. In *Proceedings of STACS 2004*, vol. 2996 of *LNCS*, pp. 4–18. Springer, 2004.

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

[20] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[21] 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.

[22] M.L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[23] S. Safra. *Complexity of Automata on Infinite Objects*. PhD thesis, 1989.

[24] C. Stirling. Modal and temporal logics. *Handbook of Logic in Comp. Sci.*, 2:477–563, 1992.

[25] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of FOCS'85*, pp. 327–338. IEEE, 1985.

[26] I. Walukiewicz. A landscape with games in the background. In *Proceedings of LICS 2004*, pp. 356–366. IEEE, 2004.

# Appendix

## A Phase (1)

Let $G = (V, E, (V_\Box, V_\bigcirc), \mathit{Prob})$ be a MDP, $(V_D, V_R)$ a partition of $V_\Box$, $s_{in} \in V$, $\nu : V \to 2^{Ap}$ a valuation, and $\psi$ a qualitative PECTL* formula. We construct a MDP $G' = (V', E', (V'_\Box, V'_\bigcirc), \mathit{Prob}')$, 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 HC strategy $\sigma$ where $s_{in} \models^\nu \psi$ in $G_\sigma$ implies the existence of a HC strategy $\pi$ that achieves the objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ in ${G'}_\pi^{s'_{in}}$, and vice versa.

Given an infinite sequence $\alpha = a_0, a_1, \ldots$ we use $\alpha_i$ and $\alpha^i$ to denote the sequences $a_i, a_{i+1}, \ldots$ and $a_0, a_1, \ldots, a_i$, respectively. Given a finite sequence $\beta = b_0, b_1 \ldots, b_n$ we define $\mathit{last}(\beta) = b_n$. We also write $L_{>0}(s)$ and $L_{=1}(s)$ instead of $L(s) \cap Q_{>0}$ and $L(s) \cap Q_{=1}$, respectively.

First, we transform the formula $\psi$ to an equivalent formula $\varphi$ such that each subformula of $\varphi$ of the form $\mathcal{B}^{\bowtie\rho}(\psi_1, \ldots, \psi_n)$ satisfies the following:

1. The automaton $\mathcal{B}$ is a deterministic Muller automaton[8] over the alphabet $2^{\{1,\ldots,n\}}$ with a set of states $Q_\mathcal{B}$, an initial state, and the set of accepting sets $\mathcal{A}_\mathcal{B}$ such that the following holds:

   - the size of $Q_\mathcal{B}$ is exponential in $|\psi|$;
   - there is a set $\mathit{Acc}_\mathcal{B} \subseteq 2^{Q_\mathcal{B}} \times 2^{Q_\mathcal{B}}$ such that $|\mathit{Acc}_\mathcal{B}|$ depends polynomially on $|\psi|$ and $\mathcal{A}_\mathcal{B}$ is either equal to

   $$\{A \subseteq Q_\mathcal{B} \mid \exists (X, Y) \in \mathit{Acc}_\mathcal{B} : A \cap X \neq \emptyset, A \cap Y = \emptyset\}$$

   or to

   $$2^{Q_\mathcal{B}} \setminus \{A \subseteq Q_\mathcal{B} \mid \exists (X, Y) \in \mathit{Acc}_\mathcal{B} : A \cap X \neq \emptyset, A \cap Y = \emptyset\}$$

   (i.e., $\mathcal{B}$ is in fact either a Rabin automaton, or a Streett automaton).

   The formula $\psi$ can be transformed to the above form using results of [23]. This form of the acceptance condition is crucial in the complexity analysis (see Appendix A.1 and Appendix B).

[8]The definition of semantics of qualitative PECTL* can easily be adapted to deal with Muller automata instead of Büchi automata if we equip Muller automata with an initial state.

2. For arbitrary state $s$ of an arbitrary Markov chain and for arbitrary valuation $\nu$ we have that if $q \stackrel{X}{\to} q'$ is a transition of $\mathcal{B}$ such that $s \models^{\nu} \bigwedge_{i \in X} \psi_i$, then $s \not\models^{\nu} \bigvee_{j \in \{1,\dots,n\} \setminus X} \psi_j$.

   Note that each subformula $\mathcal{B}^{\bowtie\rho}(\psi_1, \dots, \psi_n)$ which does not have this property, can be replaced with a formula of the form $\bar{\mathcal{B}}^{\bowtie\rho}(\psi_1, \dots, \psi_n, \neg\psi_1, \dots, \neg\psi_n)$ where $\bar{\mathcal{B}}$ has the same set of states, the same initial state, and the same set of accepting sets as $\mathcal{B}$ and transitions of $\bar{\mathcal{B}}$ are defined as follows: $q \stackrel{X' \cup X''}{\to} q'$ in $\bar{\mathcal{B}}$ for $X' \subseteq \{1, \dots, n\}$ and $X'' \subseteq \{n+1, \dots, 2n\}$ if and only if $X'' = \{n + j \mid j \in \{1, \dots, n\} \setminus X'\}$ and there is $X \subseteq X'$ such that $q \stackrel{X}{\to} q'$ in $\mathcal{B}$.

3. We have that $\bowtie\rho$ is either of the form $=1$, or $>0$. The formula can be transformed to this form using the dualities $\mathcal{B}^{<1}(\psi_1, \dots, \psi_n) \equiv \bar{\mathcal{B}}^{>0}(\psi_1, \dots, \psi_n)$ and $\mathcal{B}^{=0}(\psi_1, \dots, \psi_n) \equiv \bar{\mathcal{B}}^{=1}(\psi_1, \dots, \psi_n)$ where $\bar{B}$ is obtained by switching accepting and nonaccepting sets of states (observe that the special form of accepting sets described in 2. allows complementation in polynomial time).

In what follows we define the decision process $G'$. The vertices of $G'$ will contain information about states of $\mathcal{M}$, so first we have to define the sets $Q$, $Q_{=1}$ and $Q_{>0}$. To $Q$ we put all states of all Muller automata that occur in $\varphi$, and to $Q_{>0}$ we put states of all Muller automata $\mathcal{B}$ such that a subformula of the form $\mathcal{B}^{>0}(\psi_1, \dots, \psi_n)$ occurs in $\varphi$. We put $Q_{=1} = Q \setminus Q_{>0}$. By $\mathtt{Lit}$ we denote the set of all literals (i.e., atomic propositions and their negations) that occur in $\varphi$ and by $\mathtt{Lit}(s)$ where $s \in V$ we denote the set of literals satisfied in $s$.

Let $\psi$ be a subformula of $\varphi$ of the form $\mathcal{B}^{\bowtie\rho}(\psi_1, \dots, \psi_n)$. We denote $\mathtt{Init}(\psi)$ the initial state of the automaton $\mathcal{B}$. Given a state $q$ of $\mathcal{B}$ we use $\mathtt{Form}(q)$ to denote the formula $\mathcal{B}_q^{\bowtie\rho}(\psi_1, \dots, \psi_n)$ where $\mathcal{B}_q$ is obtained from $\mathcal{B}$ by changing its initial state to $q$. Given a transition $q \stackrel{X}{\to} q'$ of $\mathcal{B}$ we denote $\mathtt{Start}(q \stackrel{X}{\to} q')$ the smallest subset of $Q \cup \mathtt{Lit}$ which for all $i \in X$ contains either $\psi_i$, or $\mathtt{Init}(\psi_i)$, depending on whether $\psi_i$ is a literal, or not.

The process $G' = (V', E', (V'_{\Box}, V'_{\bigcirc}), \mathtt{Prob}')$ is constructed as follows. The set $V'$ consists of the following vertices.

- $(s, A)^f$ for $s \in V$ and $A \subseteq Q$;
- $(s, \mathcal{D})^g$ for $s \in V$ and $\mathcal{D} \subseteq \{(t, A) | (s, t) \in E \text{ and } A \subseteq Q\}$ where for each $(s, t) \in E$ the set $\mathcal{D}$ contains exactly one element with $t$ in the first component.

The set $E'$ consists of the following transitions[9].

- $((s,A)^f,(s,\mathcal{D})^g)$ iff for every $q \in A$ there is $q'$ such that all of the following holds.
  - $q \xrightarrow{X} q'$ and $\mathtt{Start}(q \xrightarrow{X} q') \subseteq A \cup \mathtt{Lit}(s)$;
  - if $q \in Q_{=1}$, then $q' \in \bigcap_{(t,B)\in\mathcal{D}} B$;
  - if $q \in Q_{>0}$, then $q' \in \bigcup_{(t,B)\in\mathcal{D}} B$;
- $((s,\mathcal{D})^g,(t,A)^f)$ iff $(t,A) \in \mathcal{D}$;
- $((s,A)^f,\mathtt{Dead})$ for all $(s,A)^f$;
- $(s'_{in},(s_{in},A)^f)$ iff $\mathtt{Init}(\varphi) \in A$;
- $(\mathtt{Dead},\mathtt{Dead})$.

The set $V'_\bigcirc$ consists of all vertices of the form $(s,\mathcal{D})^g$ where $s \in V_\bigcirc$, and we define $V'_\Box = V' \setminus V'_\bigcirc$. Finally, we define $\mathit{Prob}'((s,\mathcal{D})^g,(t,A)^f) = \mathit{Prob}(s,t)$ for all $(s,\mathcal{D})^g \in V'_\bigcirc$ and $((s,\mathcal{D})^g,(t,A)^f) \in E'$.

The set $V'_R$ contains all vertices $(s,\mathcal{D})^g$ where $s \in V_R$ and the set $V'_D$ contains all other vertices from $V'_\Box$. Now we define the automaton $\mathcal{M} = (Q, V', \delta, \mathcal{A})$. The set $Q$ has already been defined above. Given a state $q \in Q$, we denote $\mathcal{B}[q]$ the unique automaton which occurs in $\varphi$ and contains the state $q$. The transition function $\delta$ is defined as follows:

- for all $q \in Q$ and $(s,\mathcal{D})^g \in V'$ we define $\delta(q,(s,\mathcal{D})^g) = q$;
- for all $q, q' \in Q$ and $(s,A)^f \in V'$ we put $\delta(q,(s,A)^f) = q'$ if and only if there is exactly one $X$ such that $q \xrightarrow{X} q'$ in $\mathcal{B}[q]$ and $\mathtt{Start}(q \xrightarrow{X} q') \subseteq \mathtt{Lit}(s) \cup A$.

The set $\mathcal{A}$ consists of all accepting sets of all automata occurring in $\varphi$. Finally we define the labeling $L$ as follows:

- for $(s,A)^f \in V'$ we define $L((s,A)^f) = A$;
- for $(s,\mathcal{D})^g \in V'$ we define $L((s,\mathcal{D})^g) = (Q_{>0} \cap \bigcup_{(t,B)\in\mathcal{D}} B) \cup (Q_{=1} \cap \bigcap_{(t,B)\in\mathcal{D}} B)$.

The rest of this section is devoted to the proof of the following lemma.

[9] Note that here we do not require each vertex of $V'_\Box$ to have at most two outgoing edges (as opposed to Definition 2.2). In Lemma B.1 we show that the game $G'$ can be turned back into a game that obeys the definition.

**Lemma A.1.** *There is a HC strategy* $\sigma$ *in* $\mathsf{G}$ *such that* $s_{in} \models_\sigma \varphi$ *iff there is a consistent*[10] *HC strategy* $\pi$ *in* $\mathsf{G}'$.

In this proof we use the following notation: Given a state $q \in Q$, we denote $\psi[q]$ the subformula of $\varphi$ of the form $\mathcal{B}^{\bowtie\rho}(\psi_1, \dots, \psi_n)$ such that $q$ is a state of $\mathcal{B}$. Similarly as above the automaton $\mathcal{B}$ is denoted $\mathcal{B}[q]$.

### A.0.1 From $\pi$ to $\sigma$.

Let $\pi$ be a consistent HC strategy in $\mathsf{G}'$. We construct a HC strategy $\sigma$ in $\mathsf{G}$ and show that $s_{in} \models_\sigma \varphi$. In what follows, states of $\mathsf{G}'_\pi$ of the form $u\cdot(s, A)^f$ and $u\cdot(s, \mathcal{D})^g$ are called $f$-states and $g$-states, respectively. Let $R$ be the set of all $f$-states reachable from $s'_{in}$ in $\mathsf{G}'_\pi$. We define a function $\Lambda : R \to V^+$ as follows.

- $\Lambda(s'_{in} \cdot (s_{in}, A)^f) = s_{in}$
- $\Lambda(u \cdot (s, \mathcal{D})^g \cdot (t, A)^f) = \Lambda(u) \cdot t$

It follows immediately from the definition of $\mathsf{G}'$ and $V'_R$ that $\Lambda$ is injective. Let $us \in \Lambda(R)$ and let $\Lambda^{-1}(us) \cdot (s, \mathcal{D})^g$ be the unique successor of $\Lambda^{-1}(us)$ in $\mathsf{G}'_\pi$. We define

$$\sigma(us)(s, t) = \pi(\Lambda^{-1}(us) \cdot (s, \mathcal{D})^g)((s, \mathcal{D})^g, (t, A)^f)$$

The following lemma can straightforwardly be proved by induction.

**Lemma A.2.** *Let* $u \in \Lambda(R)$. *Then* $u \xrightarrow{x} us$ *in* $\mathsf{G}_\sigma$ *iff* $\Lambda^{-1}(u) \xrightarrow{1} v \xrightarrow{x} v'$ *where* $\Lambda(v') = us$.

Now we prove that $s_{in} \models_\sigma \varphi$. We show that if $q \in L(u)$ for an $f$-state $u$ reachable from $s'_{in}$, then $\Lambda(u) \models_\sigma \mathtt{Form}(q)$. The rest follows from the fact that $\mathtt{Init}(\varphi) \in L(\Lambda^{-1}(s_{in}))$.

Let us fix an $f$-state $u$ and $q \in L(u)$. Let us impose some linear ordering $<$ on $Q$ such that $q' < q$ whenever $\psi[q']$ is a subformula of $\psi[q]$. Suppose that for all $f$-states $v$ and all $q' \in Q$ such that $q' < q$ we have $\Lambda(v) \models_\sigma \mathtt{Form}(q')$ whenever $q' \in L(v)$.

We analyze only the case for $q \in Q_{>0}$ here. The case for $q \in Q_{=1}$ is similar. Because $\pi$ is consistent and $q \in L_{>0}(u)$, there is a set $U \subseteq \mathit{Run}[\mathsf{G}'_\pi](u)$ such that $\mathcal{P}(U) > 0$ and for every $\omega \in U$ the word $\mathit{last}(\omega(0))\mathit{last}(\omega(1)) \cdots$ is accepted by $\mathcal{M}$ initiated in $q$. Let $U'$ be

[10] If a consistency objective is clear from the context, we shall write that a strategy is consistent to denote that the strategy achieves the consistency objective.

the set of all sequences of the form $\Lambda(\omega(0)), \Lambda(\omega(2)), \ldots, \Lambda(\omega(2i)), \ldots$ where $\omega \in U$. It follows from Lemma A.2 that each sequence of $U'$ is a run in $G_\sigma$ and $\mathcal{P}(U') = \mathcal{P}(U) > 0$.

Let us denote $\mathsf{Form}(q) = \mathcal{B}^{>0}(\psi_1, \ldots, \psi_n)$. We show that $\mathcal{B}(\psi_1, \ldots, \psi_n)$ accepts all runs[11] of $U'$. Let $\omega' = \Lambda(\omega(0)), \Lambda(\omega(2)), \ldots$ be a run of $U'$. Let us denote $last(\omega'(i)) = (s_i, A_i)^f$. Let $p_0, p_1, p_2, \ldots$ be the unique accepting computation of $\mathcal{M}$ on $last(\omega(0))last(\omega(1))\cdots$. It follows from definition of $G'$ and $\mathcal{M}$ that for every $i \geq 0$ we have $p_{2i} \in L(\omega(i)) = A_i$ and there is exactly one transition $p_{2i} \xrightarrow{X} p_{2(i+1)}$ of $\mathcal{B}$ such that $\mathsf{Start}(p_{2i} \xrightarrow{X} p_{2(i+1)}) \subseteq A_i \cup L(s)$. However, $L(\omega'(i)) = A_i$ and thus $\omega'(i) \models_\sigma \psi_k$ for all $k \in X$ by induction. By our assumptions about $\varphi$, we have that $\omega'(i) \not\models_\sigma \psi_\ell$ for all $\ell \in \{1, \ldots, n\} \setminus X$. Hence, $\mathcal{B}(\psi_1, \ldots, \psi_n)$ accepts all runs of $U'$. It follows that $\Lambda(u) \models_\sigma \mathsf{Form}(q)$.

**From $\sigma$ to $\pi$.** Let $\sigma$ be a HC strategy in G such that $s_{in} \models_\sigma \varphi$. Let R be a set of states reachable from $s_{in}$ in $G_\sigma$. We define a strategy $\pi$ and a function $\Lambda : R \to (V')^*$ as follows. We put $\pi(s'_{in})(s'_{in}, (s_{in}, A)^f) = 1$ and $\Lambda(s_{in}) = s'_{in} \cdot (s_{in}, A)^f$ where $A$ consists of all $q \in Q$ such that $s_{in} \models_\sigma \mathsf{Form}(q)$. Given $u \in R$ such that $\mathsf{last}(\Lambda(u)) = (s, A)^f$ we define $\pi(\Lambda(u))((s, A)^f, (s, \mathcal{D})^g) = 1$ where

- $((s, A)^f, (s, \mathcal{D})^g) \in E'$
- for each successor $ut$ of $u$ in $G_\sigma$ there is some $(t, B) \in \mathcal{D}$;
- $B$ consists of all $q \in Q$ such that $ut \models \mathsf{Form}(q)$.

We define $\Lambda(ut) = \Lambda(u) \cdot (s, \mathcal{D})^g \cdot (t, B)^f$ for $(t, B) \in \mathcal{D}$. If $s \in V_\Box$, then we define $\pi(\Lambda(u) \cdot (s, \mathcal{D})^g)((s, \mathcal{D})^g, (t, A)^f) = \sigma(us)(s, t)$ for $(t, A) \in \mathcal{D}$

It is straightforward to show that $\pi$ is indeed a HC strategy in $G'$ and that $\Lambda$ is injective.

**Lemma A.3.** *Let $u \in R$. Then $u \xrightarrow{x} us$ in $G_\sigma$ iff $\Lambda(u) \xrightarrow{1} v \xrightarrow{x} v'$ where $\Lambda^{-1}(v') = us$.*

For the sake of our proof we need to prove the following technical lemma.

**Lemma A.4.** *Let $\mathcal{T} = (S, \to, \mathsf{Prob})$ be a Markov chain, $s \in S$ its state, $\nu$ a valuation and $\psi$ a qualitative PECTL* formula of the form $\mathcal{B}^{>0}(\psi_1, \ldots, \psi_n)$ such that $s \models \psi$. Let $Y \subseteq Run(s)$ be the set of runs accepted by $\mathcal{B}(\psi_1, \ldots, \psi_n)$ such that for all $\omega \in Y$ and $i \geq 0$ we have $\omega(i) \models \mathsf{Form}(p_i)$ where $p_0, p_1, \ldots$ is the accepting computation of $\mathcal{B}(\psi_1, \ldots, \psi_n)$ on $\omega$. Then $\mathcal{P}(Y) > 0$.*

[11] We say that $\mathcal{B}(\psi_1, \ldots, \psi_n)$ accepts a run $\omega$ if there is a word $w$ accepted by $\mathcal{B}$ such that for all $1 \leq i \leq n$ and $\ell \geq 0$ we have $i \in w(\ell)$ iff $\psi_i \models \omega(\ell)$.

*Proof.* For every $\omega \in Run(s)$ we denote $\gamma_\omega = p_0, p_1, \ldots$ the unique computation of $\mathcal{B}(\psi_1, \ldots, \psi_n)$ on $\omega$. Let $W \subseteq Run(s)$ be the set of all runs accepted by $\mathcal{B}(\psi_1, \ldots, \psi_n)$ and let $U = W \setminus Y$. Suppose that $\mathcal{P}(Y) = 0$, i.e., that $\mathcal{P}(U|W) = 1$. For all $\omega \in U$ there is $i_\omega \geq 0$ such that $\omega(i_\omega) \not\models \mathsf{Form}(\gamma_\omega(i_\omega))$.

Let $U_\omega$ denote the set of all runs from $Run(\omega^{i_\omega})$ accepted by $\mathcal{B}(\psi_1, \ldots, \psi_n)$. Observe that $\mathcal{P}(U_\omega) = \mathcal{P}(Run(\omega^{i_\omega})) \cdot c$ where $c$ is the probability of all runs of $Run(\omega(i_\omega))$ accepted by $\mathcal{B}(\psi_1, \ldots, \psi_n)$ initiated in $\gamma_\omega(i_\omega)$. Because $\omega(i_\omega) \not\models \mathsf{Form}(\gamma_\omega(i_\omega))$ we get $c = 0$, and hence $\mathcal{P}(U_\omega) = 0$.

We have $U \subseteq \bigcup_{\omega \in U} U_\omega$ and thus $\mathcal{P}(U) \leq \bigcup_{\omega \in U} U_\omega$. Since $\mathcal{P}(\bigcup_{\omega \in U} U_\omega) \leq \sum_{\omega \in U} \mathcal{P}(U_\omega) = 0$, we have $\mathcal{P}(U) = 0$ and $\mathcal{P}(Y) = 0$, which contradicts $\mathcal{P}(W) = \mathcal{P}(U \cup Y) > 0$. $\square$

Now let $u \in R$ and let $q \in L_{>0}(\Lambda(u))$. We show that $\Lambda(u) \models_\pi \mathsf{Acc}^{>0}(q)$. Let us denote $\mathsf{Form}(q) = \mathcal{B}^{>0}(\psi_1, \ldots, \psi_n)$. Given a run $\omega \in Run(u)$ we denote $\gamma_\omega = p_0, p_1, \ldots$ the unique computation of $\mathcal{B}(\psi_1, \ldots, \psi_n)$ on $\omega$ such that $p_0 = q$.

Because $q \in L(\Lambda(u))$, we have $u \models_\sigma \mathsf{Form}(q)$. Hence, by Lemma A.4 there is a set of runs $Y \subseteq Run(u)$ such that $\mathcal{P}(Y) > 0$ which satisfies the following: for all $\omega \in Y$ and all $i \geq 0$ we have $\omega(i) \models_\sigma \mathsf{Form}(\gamma(i))$. By Lemma A.3 there is a unique run $\Lambda(\omega) \in Run[G'_\pi](\Lambda(u))$ such that $\Lambda(\omega)(2i) = \Lambda(\omega(i))$ for all $i \geq 0$. It follows from Lemma A.3 that $\mathcal{P}(\{\Lambda(\omega) \mid \omega \in Y\}) = \mathcal{P}(Y) > 0$.

By definition $\gamma_\omega(i) \in L(\Lambda(\omega)(2i))$ for all $i \geq 0$ because $\gamma_\omega(0) = q \in L(\Lambda(u)) = L(\Lambda(\omega)(0))$. Let us assume that $\Lambda(\omega)(2i) = \nu \cdot (s, A)^f$. We show that $\delta(\gamma(i), (s, A)^f) = \gamma(i+1)$ (remember that $\delta$ is the transition function of $\mathcal{M}$). Let $\delta_\mathcal{B}$ be the transition function of $\mathcal{B}$. Observe that there is $X$ such that $\delta_\mathcal{B}(\gamma(i), X) = \gamma(i+1)$ and $\omega(i) \models \mathsf{Form}(\psi_j)$ for all $j \in X$. It follows that $\mathsf{Start}(\gamma(i) \xrightarrow{X} \gamma(i+1)) \subseteq A \cup L(\omega(i))$, and thus $\delta(\gamma(i), (s, A)^f) = \gamma(i+1)$.

It remains to show that $\Lambda(u) \cdot (s, \mathcal{D})^g \models_\sigma \mathsf{Acc}^{\bowtie\rho}(q)$ whenever $q \in L_{\bowtie\rho}(\Lambda(u) \cdot (s, \mathcal{D})^g)$ and $\Lambda(u) \cdot (s, \mathcal{D})^g$ is the unique successor of $\Lambda(u)$. The proof is similar to the proof for f-states and is omitted here.

## A.1 Complexity Analysis

Here we use the notation from 1. in the beginning of this proof. The size of $G'$ is in $|G| \cdot 2^{p(|\psi|)}$ where $p$ is a polynomial. The size of $Q$ is in $2^{p(\psi)}$ (if $\psi$ is a qualitative PCTL formula, then the size of $Q$ is polynomial in the size of $\psi$).

Now let us denote $\mathsf{Rab}$ the set of all Muller automata $\mathcal{B}$ that occur in $\varphi$ that satisfy

$$\mathcal{A}_{\mathcal{B}} = \{A \subseteq Q_{\mathcal{B}} \mid \exists (X, Y) \in Acc_{\mathcal{B}} : A \cap X \neq \emptyset, A \cap Y = \emptyset\}$$

Let $\mathsf{Saf}$ be the set of all Muller automata that occur in $\varphi$ and are not in $\mathsf{Rab}$. Let $R$ be the union of all sets of states of all automata of $\mathsf{Rab}$. Let $S$ be the union of all sets of states of all automata of $\mathsf{Saf}$. Let $\{(A_1, B_1), \ldots, (A_n, B_n)\} \subseteq 2^R \times 2^R$ be the union of all sets $Acc_{\mathcal{B}}$ for all $\mathcal{B} \in \mathsf{Rab}$. Let $\{(C_1, D_1), \ldots, (C_m, D_m)\} \subseteq 2^S \times 2^S$ be the union of all sets $Acc_{\mathcal{B}}$ for all $\mathcal{B} \in \mathsf{Saf}$. It is easy to show that

$$\mathcal{A} = \{A \subseteq R \mid \exists i : A \cap A_i \neq \emptyset, A \cap B_i = \emptyset\} \cup (2^S \setminus \{A \subseteq S \mid \exists i : A \cap C_i \neq \emptyset, A \cap D_i = \emptyset\})$$

where the numbers $n$ and $m$ are polynomial in both $|G|$ and $|\psi|$.

# B Phase (2)

## B.1 Technical assumptions

To make the proofs in this section simpler, we impose several technical assumptions on MDPs and Muller automata. This section defines these assumptions and argues that they don't cause any loss of generality.

Let $G = (V, E, (V_\Box, V_\bigcirc), \mathit{Prob})$ 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 $\mathcal{M} = (Q, V, \delta, A)$.

First, we assume that each vertex $s \in V$ has exactly two outgoing transitions. From the following lemma we have that this is no restriction.

**Lemma B.1.** *There is a game $G' = (V', E', (V'_\Box, V'_\bigcirc), Prob')$ in which each vertex has exactly two successors, $(V'_D, V'_R)$ a partition of $V'$ and a consistency objective $(\mathcal{M}', (Q_{>0}, Q_{=1}), L)$ where $\mathcal{M}' = (Q', V', \delta', A)$ such that the following holds: There is a HC (or srHC) consistent*[12] *strategy in $G$ iff there is HC (or srHC) consistent strategy in $G'$. Moreover, the size of $G'$ and $\mathcal{M}'$ is polynomial in the size of $G$ and $\mathcal{M}$, respectively.*

*Proof.* We can suppose that there are no $s \in V_\bigcirc$ with exactly one successor (if there is such $s$, it can be moved to $V_\Box$). We put $V' = V \cup \{s_1, s_2, \ldots, s_{n-1} \mid \forall s \in V_\Box$ where $s$ has $n > 2$ successors$\}$ (note that the set $V'_\Box$ in game $G'$ from Appendix A can contain vertices with more than 2 successors). The set $E'$ contains the following transitions:

[12] Similarly to the previous section, we shall write that a strategy is consistent if it achieves a consistency objective that is clear from the context.

- $(s,t)$ for all $(s,t) \in E$ where either $s \in V_{\bigcirc}$, or $s$ is a vertex from $V_{\Box}$ with precisely two successors;
- if $s \in V_{\Box}$ has exactly one successor $t$, then we put $(s,t)$ and $(s, s_X)$ to $E'$;
- if $s \in V_{\Box}$ has $n$ successors where $n > 2$, then to $E'$ we put transitions $(s, s_2)$, $(s, t_1)$, $(s_{n-1}, t_n)$ and transitions $(s_i, s_{i+1})$ for $2 \leq i < n-1$ and $(s_i, t_i)$ for $2 \leq i < n$;
- transitions $s_X \to s_X$, $s_X \to s_Y$, $s_Y \to s_Y$ and $s_Y \to s_X$.

We define $\mathrm{Prob}' = \mathrm{Prob}$. The set $V'_R$ contains vertices $s \in V_{\Box} \cap V_R$ that have more than one successor, and also all newly created vertices $s_2, s_3, \ldots, s_{n-1}$ for $s \in V_R$. The set $V'_D$ contains all vertices from $V'_{\Box} \setminus V'_R$.

We put $Q' = Q \cup \{q\}$ where $q$ is a fresh state and we create $\delta'$ by extending $\delta$ with transitions $\delta'(s,q) = q$ for all states $q \in Q'$ and vertices $s \in V'_{\Box} \setminus V_{\Box}$. The labeling $L'$ is defined by

- $L'(s_X) = L'(s_Y) = \{q\}$ for the state $q \in Q' \setminus Q$
- $L'(s) = L(s)$ for all vertices $s \in V_{\Box}$
- $L'(s_i) = \{q' \mid \delta(s,q) = q'$ for some $q \in Q\}$ for all vertices $s_i$ that were added for vertex $s \in V_{\Box}$ with more than two successors.

It is a mere technicality to show that each strategy in $G$ has it's counterpart in $G'$ (and vice versa). $\Box$

To obtain optimal complexity estimates we assume that the automaton $\mathcal{M}$ has the following special form: $Q$ is a disjoint union of two non-empty sets $R$ and $S$, and there are no transitions from $R$ to $S$ and vice versa. The set $\mathcal{A}$ of accepting sets is described by two collections $\{(A_1,B_1),\ldots,(A_n,B_n)\} \subseteq 2^R \times 2^R$ and $\{(C_1,D_1),\ldots,(C_m,D_m)\} \subseteq 2^S \times 2^S$ in such a way that

$$\mathcal{A} = \{A \subseteq R \mid \exists i : A \cap A_i \neq \emptyset, A \cap B_i = \emptyset\} \cup (2^S \setminus \{A \subseteq S \mid \exists i : A \cap C_i \neq \emptyset, A \cap D_i = \emptyset\})$$

## B.2 Step (a)

Let $G = (V, E, (V_{\Box}, V_{\bigcirc}), \mathit{Prob})$ 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 $\mathcal{M} = (Q, V, \delta, A)$. In this section we show how to solve the general consistency problem for srHC strategies.

To further simplify the consistency problem we impose the following assumptions on the MDP G and the automaton $\mathcal{M}$ (these are the assumptions under which the Proposition 3.3 holds):

- For all $s \in V$ and all states $q, q' \in Q$ satisfying $q \in L(s)$ and $q \xrightarrow{s} q'$ we assume that
  1. if either $q \in Q_{=1}$, or $s \in V_D$, then for all $(s,t) \in E$ we have that $q' \in L(t)$;
  2. if $q \in Q_{>0}$ and $s \in V_{\bigcirc} \cup V_R$, then for some $(s,t) \in E$ we have that $q' \in L(t)$.
- We have that $L_{>0}(s) \neq \emptyset$ for all $s \in V$.

In the rest of this section we prove the Proposition 3.3 that gives an alternative characterization of existence of a consistent srHC strategy in terms of the sets $EP(X)$. It follows from the proof that the problem of existence of a consistent srHC strategy is decidable in time $(|G| \cdot |\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|nm)}$ and that the existence of a consistent srHC strategy implies existence of a one-counter consistent srHC strategy computable in time $(|G| \cdot |\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|nm)}$. The following lemma says that we can compute the set $EP(X)$ for a given X.

**Lemma B.2.** *The problem whether $(s_0, q_0) \in EP(X)$ is decidable in time $(|G| \cdot |\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|nm)}$. Moreover, if $(s_0, q_0) \in EP(X)$, then there is a FD strategy $\zeta$ witnessing that $(s_0, q_0) \in EP(X)$ which is computable in time $(|G| \cdot |\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|nm)}$.*

*Proof.* We define a new Markov decision process $G = (V', E', (V'_{\square}, V'_{\bigcirc}), \mathit{Prob}')$, a partition $(V'_D, V'_R)$ of $V'_{\square}$ and a consistency objective $(\mathcal{M}', (Q'_{>0}, Q'_{=1}), L)$ such that there is a consistent HC strategy in $G'$ if $(s_0, q_0) \in EP(X)$. We put $V' = X \times Q_{>0}$ and define transitions in $G'$ as follows: $((s,p),(t,p')) \in E'$ iff $s, t \in X$, $(s,t) \in E$ and $p' = \delta(p,s)$. We define $\mathit{Prob}'((s,p),(t,\delta(p,s))) = \mathit{Prob}(s,t)$. We put $V'_{\square} = (V_{\square} \cap X) \times Q_{>0}$.

We define a Muller automaton $\mathcal{M}' = (Q', V', \delta', \mathcal{A}')$ where $Q' = Q \cup \{\mathrm{Accept}\}$ and transitions are defined as follows: Given $p \in Q$ we define

$$\delta'(p,(s,p')) = \begin{cases} \mathrm{Accept} & \text{if } p \in Q_{>0} \text{ and } p \neq p' \text{ and } s \in V_{\bigcirc} \cup V_R; \\ \delta(p,s) & \text{otherwise.} \end{cases}$$

We define $\delta'(\mathrm{Accept},(s,p)) = \mathrm{Accept}$ and $\mathcal{A}' = \mathcal{A} \cup \{\{\mathrm{Accept}\}\}$ and $L'((s,p)) = L(s) \cup \{\mathrm{Accept}\}$ for all $(s,p) \in V'$. We define $Q'_{=1} = Q'$ and $Q'_{>0} = \emptyset$ and $s'_{in} = (s_0, q_0)$.

Let $V'_R = \emptyset$ and $V'_D = V'_{\square}$ and let us assume that there is a HC strategy in $G'$ that achieves the objective $(\mathcal{M}', (Q_{>0}, Q_{=1}), L)$ from vertex $s'_{in}$. Then, as we show in Section

B.2.2, there is also an FD consistent strategy $\zeta$ computable in time $(|G| \cdot |\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|nm)}$. We define a strategy $\xi$ as follows: Given $s_0 \cdots s_n \in V^*V_\Box$ we put $\xi(s_0 \cdots s_n)(s_n, t) = \zeta((s_0, p_0) \cdots (s_n, p_n))((s_n, p_n), (t, q))$ where $p_0, \ldots, p_n, q$ is the unique computation of $\mathcal{M}$ on $s_0 \cdots s_n$ such that $p_0 = q_0$. Then $\xi$ is an FD strategy which witnesses that $(s_0, q_0) \in EP(X)$. Clearly the strategy $\xi$ can be computed in time $(|G| \cdot |\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|nm)}$.

Note also that using the same equation one can define a consistent HC strategy $\zeta$ in $G'$ using a HD strategy $\xi$ which witnesses that $(s_0, q_0) \in EP(X)$. $\square$

### B.2.1 Proof of Proposition 3.3: Right to Left

Throughout this proof we assume that there is an (arbitrary) ordering on the set of states of $\mathcal{M}$. Given a set $A$ of states of $\mathcal{M}$ we denote $\min(A) \in A \cup \{\bot\}$ the least state of $A$ in the ordering (here $\min(\emptyset) = \bot$). Similarly we impose an arbitrary ordering on the set of vertices of $G$.

**Proposition B.3.** *Let $\mathcal{T}$ be a probabilistic tree and let $\mathcal{T}'$ be a tree obtained by cutting off some subtrees of $\mathcal{T}$. Let $A \subseteq$ IPath$[\mathcal{T}'](v)$ be a measurable set. Then the probability of $A$ in $\mathcal{T}'$ is equal to the probability of $A$ in $\mathcal{T}$.*

**Constructing the strategy $\sigma$:** Let $(s, q) \in EP(X)$ and let $\zeta$ be an FD strategy in $G|X$ witnessing that $(s, q) \in EP(X)$ (see Definition 3.2 and Lemma B.2).

We define

$$D_0 = \{vt \in V^+V \mid \mathit{last}(v) \in V_R \text{ and } \zeta(v)(\mathit{last}(v), t) = 0\}$$

We show that there is a probabilistic tree $\mathcal{E}_{(s,q)}$ obtained from $G^s_\zeta$ by adding states of $D_0$ and by cutting off some subtrees such that $\mathcal{E}_{(s,q)}$ has the following properties:

1. The probability of *IPath*$[\mathcal{E}_{(s,q)}](s)$ is non-zero (this implies that $s$ satisfies $\mathrm{Acc}^{>0}(q)$ in $\mathcal{E}_{(s,q)}$.)

2. If $p \in L_{>0}(v)$ for a state $v$ of $\mathcal{E}_{(s,q)}$, then either there is a path $\omega$ in $\mathcal{E}_{(s,q)}$ from $v$ to a leaf of $\mathcal{E}_{(s,q)}$ such that $\mathrm{Comp}(p, \omega) \neq \bot$, or $v$ satisfies $\mathrm{Acc}^{>0}(p)$ in $\mathcal{E}_{(s,q)}$.

3. If $p \in L_{=1}(v)$ for a state $v$ of $\mathcal{E}_{(s,q)}$, then for almost all $\omega \in$ *IPath*$[\mathcal{E}_{(s,q)}](v)$ the computation $\mathrm{Comp}(p, \omega)$ is accepting.

Later we glue the trees $\mathcal{E}_{(s,q)}$ together and obtain a tree which corresponds to a consistent strategy.

In order to define $\mathcal{E}_{(s,q)}$ we label states of $G^s_\xi$ with values of a counter (similarly to the example from page 11) which simulates asymmetric one-dimensional random walk. Then we add states of $D_0$ while modifying probabilities of transitions to obtain Markov chain similar to the Markov chain $\hat{G}_{\hat{\sigma}}$ from page 12. Finally, we cut off subtrees rooted either in states of $D_0$, or in states labeled with 0. Properties of random walk ensure that in the resulting tree a leaf is reached with a probability less than one. To ensure the properties 2. and 3., the labeling with values of the counter has to take into account the labeling L. Hence, we start by defining a labeling I which labels each state $v$ with a subset of $L(v)$ and a labeling J which picks a state from $I(v)$. Consequently we use these labelings to define the labeling $C : T \to \mathbb{Z}$ which labels states with values of the counter.

Let us define the labelings $I : T \to 2^{Q_{>0}}$ and $J : T \to Q_{>0} \cup \{\perp\}$ as follows: Define $I(s) = L_{>0}(s)$ and $J(s) = \min(L_{>0}(s))$. Let $v \in T$ be a state of $G^s_\xi$ and let us assume that $I(v)$ and $J(v)$ have already been defined. If $I(v) = \emptyset$, then for all successors $vt$ of $v$ we define $I(vt) = L_{>0}(t)$ and $J(vt) = \min(L_{>0}(t))$. Otherwise, let us denote

$$A = \{\delta(p, \mathit{last}(v)) \mid p \in I(v)\} \neq \emptyset$$

If $\mathit{last}(v) \in V_\Box$, then for all successors $vt$ of $v$ we define $I(vt) = A$ and $J(vt) = \delta(J(v), \mathit{last}(v))$. If $\mathit{last}(v) \in V_\bigcirc$ and $vt_1, vt_2$ are the two successors of $v$ such that $t_1$ is less than $t_2$ in the fixed ordering on V, then we define $I(vt_1) = A \cap L_{>0}(t_1)$ and $I(vt_2) = (A \cap L_{>0}(t_2)) \backslash I(vt_1)$, and for $i \in \{1, 2\}$ we define

$$J(vt_i) = \begin{cases} \delta(J(v), \mathit{last}(v)) & \text{if } \delta(J(v), \mathit{last}(v)) \in I(vt_i); \\ \min(I(vt_i)) & \text{otherwise.} \end{cases}$$

Observe that if $\mathit{last}(v) \in V_\bigcirc$, then $I(vt_1) \cap I(vt_2) = \emptyset$, and moreover, the successor $t_1$ is in some sense privileged: All states of the form $\delta(p, \mathit{last}(v))$, where $p \in I(v)$, that are in $L_{>0}(t_1)$ are also in $I(t_1)$. Hence also $\delta(J(v), \mathit{last}(v))$ is either in $I(vt_1)$, or $I(vt_2)$, but not in both (and whether $\delta(J(v), \mathit{last}(v))$ is put to $I(vt_1)$, or to $I(vt_2)$ depends only on $J(v)$ and $\mathit{last}(v)$). Also observe that $|I(vt_i)| \leq |I(v)|$ whenever $I(v)$ is non-empty. These properties of I and J are essential in ensuring the condition 2. (see Lemma B.5).

Now we define a function $C : T \to \mathbb{Z}$. Let $p = max\{p' \mid s \xrightarrow{p'} s' \text{ in } G\}$ and let $\lambda$ be a number[13] such that $p^\lambda < \frac{1}{8}$. Let us denote $size = |V| \cdot |Q| \cdot |\xi|$ where $|\xi|$ is the size of the FD strategy $\xi$. Let us define $C(s) = 1$. Let $v$ be a state of $G^s_\xi$ such that C has already

[13] Note that size of $\lambda$ can be singly exponential in number of bits of precision of $p$. The role of $\lambda$ will be explained later.

been defined for all ancestors of $v$. Let $v_0, \ldots, v_{n+1}$ be a (unique) path in $G^s_\xi$ such that $C(v_1) = \cdots = C(v_n)$ and $v_{n+1} = v$ and either $C(v_0) \neq C(v_1)$, or $v_0 = s$. If $\mathit{last}(v) \in V_\bigcirc$ and $n > \lambda \cdot \mathit{size}$, then we define $C(v) = C(v_n) + 1$. If $\mathit{last}(v) \in V_\bigcirc$ and $n \leq \lambda \cdot \mathit{size}$ and there is a tuple $1 < i_1 < i_2 < \cdots < i_{\lambda-1} \leq n$ such that $\mathit{last}(v_1) = \mathit{last}(v_{i_1}) = \mathit{last}(v_{i_2}) = \cdots = \mathit{last}(v_{i_{\lambda-1}}) \in V_\bigcirc$ and $J(v_1) = J(v_{i_1}) = J(v_{i_2}) = \cdots = J(v_{i_{\lambda-1}})$, then we define

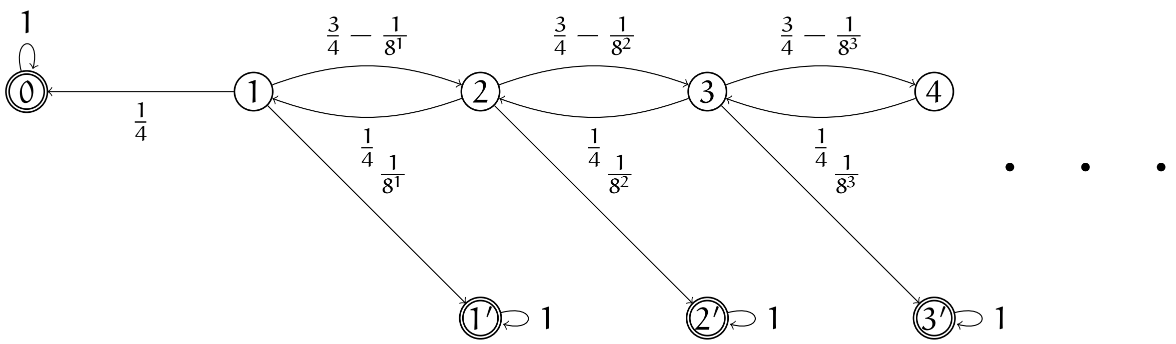
$$C(v) = \begin{cases} C(v_n) - 1 & \text{if } \mathit{last}(v_2) = \mathit{last}(v_{i_1+1}) = \mathit{last}(v_{i_2+1}) = \cdots = \mathit{last}(v_{i_{\lambda-1}+1}); \\ C(v_n) + 1 & \text{otherwise.} \end{cases}$$

Otherwise, if there is no state from $V^*V_\bigcirc$ reachable from $v$ in $G_\zeta$, then we define $C(v) = C(v_n) + 1$. Otherwise, we define $C(v) = C(v_n)$. Let us denote $C_0 = \{v \in T \mid C(v) = 0 \text{ and } C(u) \neq 0 \text{ for all predecessors } u \text{ of } v\}$.

Now we add states of $D_0$ to the probabilistic tree $G^s_\zeta$ (recall that each Markov chain can be treated as a probabilistic tree). We add the set $D_0$ to the set of states of $G^s_\zeta$ and modify the probabilities of edges as follows. For every state $vt \in V^*V_D$ the probabilities of outgoing edges remain intact. For every state $vt \in V^*V_R$ we put a probability of $(vt, vtt')$ equal to $(8 \cdot \lambda \cdot \mathit{size})^{-C(vt)}$ and $1 - (8 \cdot \lambda \cdot \mathit{size})^{-C(vt)}$ for $t' \in V \setminus D_0$ and $t' \in D_0$, respectively. The tree $\mathcal{E}_{(s,q)}$ is now obtained by cutting of all subtrees rooted in states of $C_0 \cup D_0$. Let us denote $E_{(s,q)}$ the set of states of $\mathcal{E}_{(s,q)}$.

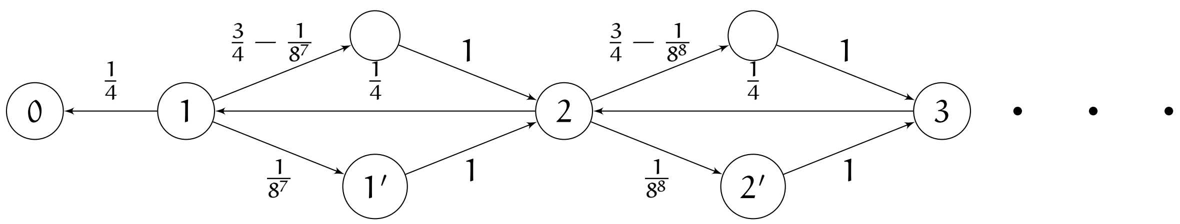
**Lemma B.4.** *The probability of reaching* $C_0 \cup D_0$ *from* $s$ *in* $\mathcal{E}_{(s,q)}$ *is at most* $\frac{2}{3}$.

*Sketch.* Let us have the following Markov chain $\mathcal{M}_1$:




We prove that the probability of reaching the double circled states $A = \{0, 1', 2', \ldots\}$ from 1 is strictly less than one.

Let us consider the following Markov chain $\mathcal{M}_2$.




Clearly, the probability of reaching $\{0, 1', 2', \ldots\}$ from 1 in $\mathcal{M}_1$ is greater than 0 iff the probability of reching $A := \{0, 1', 2', \ldots\}$ from 1 in $\mathcal{M}_2$ is greater than 0. Hence, in the rest of this proof we consider only $\mathcal{M}_2$.

Let $Z_1, Z_2, \ldots$ be a sequence of random variables defined on $Run(0)$ as follows: Given $w \in Run(0)$ we put $Z_k(w) = w(2k)$. Observe that for $k \geq 1$ and arbitrary states $m, n_1, \ldots, n_{k-1}$ we have

$$\mathcal{P}(Z_{k+1} = m+1 \mid Z_k = m, Z_{k-1} = n_{k-1}, \ldots Z_1 = n_1) = \mathcal{P}(Z_{k+1} = m+1 \mid Z_k = m) = \frac{3}{4}$$

$$\mathcal{P}(Z_{k+1} = m-1 \mid Z_k = m, Z_{k-1} = n_{k-1}, \ldots Z_1 = n_1) = \mathcal{P}(Z_{k+1} = m-1 \mid Z_k = m) = \frac{1}{4}$$

Hence, $Z_1, Z_2, \ldots$ is a random walk on $\mathbb{N}_0$ with an absorbing barrier in 0. This means that the probability of reaching 0 from 1 is equal to $\frac{2}{3}$.

In the following text, we use $I(n)$ to denote the set of all runs initiated in $n$ that never enter $n-1$. Note that $\mathcal{P}(I(n)) = \frac{2}{3}$ for all $n \geq 1$.

For each run $w \in I(n)$ and $\ell \geq 0$, we define $\ell$-th strict minimum $m_\ell(w)$ to be the number $j \geq 1$ such that

- $Z_j(w) = Z_1(w) + \ell = n + \ell$;
- for all $i > j$ holds $Z_i(w) > Z_1(w) + \ell$.

If there is no such j, we define $m_\ell(w) = \infty$. However, it is easy to show that for almost all $w \in I(n)$ we have that $m_\ell(w) < \infty$ for all $\ell$. Hence, we may safely assume that *all* runs of $I(n)$ satisfy $m_\ell(w) < \infty$ for all $\ell$.

For every run $w \in Run(n)$ and every $i \geq 0$ we denote $X_i(w) = w(i)$. This defines a sequence of random variables $X_1, X_2, \ldots$.

Now we come to the central notion of this proof. Let R be the set of all runs $w \in Run(1)$ such that for every $\ell \geq 0$ holds $m_{\ell+1}(w) - m_\ell(w) \leq 2 \cdot (\ell+1)^2$. Let N be the set of all runs $w \in Run(1)$ such that for every $\ell \geq 0$ and $m_\ell(w) \leq k \leq m_\ell(w) + 2 \cdot (\ell+1)^2$ holds $X_k \notin A$. We prove the following:

1. $\mathcal{P}(R \mid I(1)) \geq \frac{1}{2}$;

2. $\mathcal{P}(N \mid I(1)) \geq \frac{3}{4}$

Putting 1. and 2. together we obtain that

$$\mathcal{P}(\bigwedge_{i=0}^{\infty} X_i \notin A \mid I(1)) \geq \mathcal{P}(R \cap N \mid I(1)) \geq \frac{1}{4}$$

**Proof of 1.** We proceed as follows:

(a) Let us denote $F(n) = Run(n) \setminus I(n)$. For every run $w \in F(n)$ we define $K_n(w) = \min\{m \mid Z_m(w) = n-1\}$. For every $n \geq 1$, let $e_n := E(K_n \mid F(n))$. We prove that $e_n = 2$ for every $n \geq 1$.

(b) For every $\ell \geq 0$, let $E_\ell := E(m_{\ell+1} - m_\ell \mid I(1))$. Using (a), we prove that $E_\ell = 2$.

(c) We prove that $\mathcal{P}(R) \geq \frac{1}{2}$ using (b), Markov inquality applied to $m_{\ell+1} - m_\ell$, and the fact that $\prod_{n=1}^{\infty}(1 - \frac{1}{n^2}) = \frac{1}{2}$ (see below).

We start with (a). For every $n$ we denote $R(n)$ the set of all finite paths of the form $k_1 \dots k_i$ satysfying $k_1 = n$, $k_i = n-1$, and $k_r \geq n$ for all $1 \leq r < i$.

$$\begin{aligned}
e_n &= \frac{1}{\mathcal{P}(F(n))} \sum_{v \in R(n)} |v| \cdot \mathcal{P}(v) \\
&= 3 \cdot \big(\frac{1}{4} + \frac{3}{4} \cdot \sum_{v \in R(n+1)} \sum_{u \in R(n)} (1 + |v| + |u|)\mathcal{P}(v)\mathcal{P}(u)\big) \\
&= 3 \cdot \Big(\frac{1}{4} + \frac{3}{4} \cdot \big( \sum_{v \in R(n+1)} \sum_{u \in R(n)} \mathcal{P}(v)\mathcal{P}(u) + \sum_{v \in R(n+1)} \sum_{u \in R(n)} |v|\mathcal{P}(v)\mathcal{P}(u) + \\
&\quad + \sum_{v \in R(n+1)} \sum_{u \in R(n)} |u|\mathcal{P}(v)\mathcal{P}(u)\big)\Big) \\
&= 3 \cdot \Big(\frac{1}{4} + \frac{3}{4} \cdot \big(\mathcal{P}(F(n+1))\mathcal{P}(F(n)) + \mathcal{P}(F(n)) \sum_{v \in R(n+1)} |v|\mathcal{P}(v) + \mathcal{P}(F(n+1)) \sum_{u \in R(n)} |u|\mathcal{P}(u)\big)\Big) \\
&= 3 \cdot \Big(\frac{1}{4} + \frac{3}{4} \cdot \big(\frac{1}{3}\frac{1}{3} + \frac{1}{3}\frac{1}{3}e_{n+1} + \frac{1}{3}\frac{1}{3}e_n\big)\Big) \\
&= \frac{3}{4} + \frac{1}{4} \cdot \big(1 + 2e_n\big)
\end{aligned}$$

Then (a) follows from the fact that $e_n = 2$ is the unique solution of $e_n = \frac{3}{4} + \frac{1}{4} \cdot \big(1 + 2e_n\big)$.

Now we prove (b). For every $n$ we denote $Q(n)$ the set of all finite paths of the form $k_1 \dots k_i$ satysfying $k_1 = n$, $k_{i-1} = n$, $k_i = n+1$, and $k_r \geq n$ for all $1 \leq r < i$. We denote

$R(1,n)$ the set of all paths from 1 to $n$. We have

$$
\begin{aligned}
E_\ell &= E(m_{\ell+1} - m_\ell \mid I(1)) \\
&= \frac{1}{\mathcal{P}(I(1))} \sum_{v \in R(1,\ell)} \mathcal{P}(v) \sum_{u \in Q(\ell)} |u| \mathcal{P}(u) \mathcal{P}\big(I(\ell+1)\big) \\
&= \frac{\mathcal{P}(R(1,\ell))}{\mathcal{P}(R(1,\ell))\mathcal{P}(I(\ell))} \sum_{u \in Q(\ell)} |u| \mathcal{P}(u) \mathcal{P}\big(I(\ell+1)\big) \\
&= \frac{1}{\mathcal{P}(I(\ell))} \sum_{u \in Q(\ell)} |u| \mathcal{P}(u) \mathcal{P}\big(I(\ell+1)\big)
\end{aligned}
$$

Also

$$
\frac{1}{\mathcal{P}(I(\ell))} \sum_{u \in Q(\ell)} \mathcal{P}(u) \mathcal{P}\big(I(\ell+1)\big) = \frac{1}{\mathcal{P}(I(\ell))} \mathcal{P}(I(\ell)) = 1
$$

Hence

$$
\begin{aligned}
E_\ell &= \frac{1}{\mathcal{P}(I(\ell))} \sum_{v \in Q(\ell)} |v| \cdot \mathcal{P}(v) \mathcal{P}\big(I(\ell+1)\big) \\
&= \frac{1}{\mathcal{P}(I(\ell))} \Big( \frac{3}{4} \cdot \mathcal{P}\big(I(\ell+1)\big) + \frac{3}{4} \cdot \sum_{v \in R(\ell+1)} \sum_{u \in Q(\ell)} (1 + |v| + |u|) \mathcal{P}(v) \mathcal{P}(u) \mathcal{P}\big(I(\ell+1)\big)\Big) \\
&= \frac{3}{4} + \frac{3}{4} \cdot \Big( \sum_{v \in R(\ell+1)} \mathcal{P}(v) \cdot \frac{1}{\mathcal{P}(I(\ell))} \sum_{u \in Q(\ell)} \mathcal{P}(u) \mathcal{P}\big(I(\ell+1)\big) \\
&+ \sum_{v \in R(\ell+1)} |v| \mathcal{P}(v) \cdot \frac{1}{\mathcal{P}(I(\ell))} \sum_{u \in Q(\ell)} \mathcal{P}(u) \mathcal{P}\big(I(\ell+1)\big) \\
&+ \sum_{v \in R(\ell+1)} \mathcal{P}(v) \cdot \frac{1}{\mathcal{P}(I(\ell))} \sum_{u \in Q(\ell)} |u| \mathcal{P}(u) \mathcal{P}\big(I(\ell+1)\big)\Big) \\
&= \frac{3}{4} + \frac{3}{4} \cdot \Big( \mathcal{P}(F(\ell+1)) + \frac{1}{3} e_{\ell+1} + \mathcal{P}(F(\ell+1)) E_\ell \Big) \\
&= \frac{3}{4} + \frac{3}{4} \cdot \Big( \frac{1}{3} + \frac{1}{3} e_{\ell+1} + \frac{1}{3} E_\ell \Big) \\
&= \frac{3}{4} + \frac{1}{4} \cdot \big(3 + E_\ell\big) \\
&= \frac{3}{2} + \frac{1}{4} E_\ell
\end{aligned}
$$

Then (b) follows from the fact that $E_\ell = 2$ is the unique solution of $E_\ell = \frac{3}{2} + \frac{1}{4} E_\ell$.

Finally, we prove (c). Let us denote $M_\ell = m_{\ell+1} - m_\ell$. First, we need the following

$$\begin{aligned}
\mathcal{P}(\bigwedge_{\ell=1}^{n} M_\ell \leq k_\ell \mid I(1)) &= \frac{1}{\mathcal{P}(I(1))}\sum_{\ell=1}^{n} \sum_{\nu_\ell \in Q(\ell), |\nu_\ell| \leq k_\ell} \mathcal{P}(\nu_1)\cdots\mathcal{P}(\nu_n)\mathcal{P}(I(\ell)) \\
&= \prod_{\ell=1}^{n} \sum_{\nu_\ell \in Q(\ell), |\nu_\ell| \leq k_\ell} \mathcal{P}(\nu_\ell) \\
&= \prod_{\ell=1}^{n} \sum_{\nu_\ell \in Q(\ell), |\nu_\ell| \leq k_\ell} \frac{\mathcal{P}(R(1,\ell))}{\mathcal{P}(R(1,\ell)) \cdot \mathcal{P}(I(\ell))}\mathcal{P}(\nu_\ell)\mathcal{P}(I(\ell)) \\
&= \prod_{\ell=1}^{n} \frac{1}{\mathcal{P}(I(1))} \sum_{\nu_\ell \in Q(\ell), |\nu_\ell| \leq k_\ell} \mathcal{P}(R(1,\ell))\mathcal{P}(\nu_\ell)\mathcal{P}(I(\ell)) \\
&= \prod_{\ell=1}^{n} \mathcal{P}(M_\ell \leq k_\ell \mid I(1))
\end{aligned}$$

In particular,

$$\mathcal{P}(\bigwedge_{\ell=1}^{\infty} M_\ell \leq 2(\ell+1)^2 \mid I(1)) = \prod_{\ell=1}^{\infty} \mathcal{P}(M_\ell \leq 2(\ell+1)^2 \mid I(1))$$

We finish the proof of 1. by showing that

$$\prod_{\ell=1}^{\infty} \mathcal{P}(M_\ell \leq 2(\ell+1)^2 \mid I(1)) \geq \prod_{n=2}^{\infty}(1 - \frac{1}{n^2}) = \frac{1}{2}$$

The inequality follows directly from Markov inequality because $E_\ell = E(M_\ell \mid I(1)) = 2$ and thus

$$\mathcal{P}(M_\ell \leq 2(\ell+1)^2 \mid I(1)) = 1 - \mathcal{P}(M_\ell > 2(\ell+1)^2 \mid I(1)) \geq 1 - \frac{2}{2 \cdot (k+1)^2} = 1 - \frac{1}{(k+1)^2}$$

The fact that $\prod_{n=2}^{\infty}(1 - \frac{1}{n^2}) = \frac{1}{2}$ can be proved as follows: Clearly,

$$\prod_{n=2}^{\infty}(1 - \frac{1}{n^2}) = \frac{1}{2}$$

iff

$$\ln \prod_{n=2}^{\infty}(1 - \frac{1}{n^2}) = \ln \frac{1}{2}$$

We have

$$
\begin{aligned}
\ln \prod_{n=2}^{\infty}(1-\frac{1}{n^2}) &= \sum_{n=2}^{\infty} \ln(1-\frac{1}{n^2}) && (1)\\
&= \sum_{n=2}^{\infty} \ln(\frac{n^2-1}{n^2}) && (2)\\
&= \sum_{n=2}^{\infty} \ln(\frac{(n+1)(n-1)}{n^2}) && (3)\\
&= \sum_{n=2}^{\infty} \big(\ln(n+1)+\ln(n-1)-\ln n^2\big) && (4)\\
&= \sum_{n=2}^{\infty} \big(\ln(n+1)+\ln(n-1)-2\ln n\big) && (5)\\
&= \ln 1-\ln 2+\ln 2+\sum_{n=2}^{\infty} \big(\ln(n+1)-\ln n\big) && (6)\\
&= \ln 1-\ln 2 && (7)\\
&= \ln \frac{1}{2} && (8)
\end{aligned}
$$

Here, (1) follows from the continuity of ln. To see that 5 follows from the preceding term, see that $\ln(n-1)$ in the $n$+1-th summand is eliminated by half of $2\ln n$ of $n$-th summand. To see that (6) follows from (5), observe that $\ln(n+1)$ in the $n$-th summand is eliminated by the remaining $-\ln n$ in $n$+1-th summad.

**Proof of 2.** For every $\ell \geq 1$ and every $k \geq 1$ we have

$$\mathcal{P}(X_{m_\ell+k} \in A \mid I(1)) = \mathcal{P}(X_k \in A \mid I(\ell)) \leq \frac{\mathcal{P}(X_k \in A)}{\mathcal{P}(I(\ell))} \leq \frac{3}{2}\frac{1}{8^{\ell+7}}$$

Thus

$$
\begin{aligned}
\mathcal{P}(\bigvee_{\ell=1}^{\infty}\bigvee_{k=1}^{4\cdot(\ell+1)^2} X_{m_\ell+k} \in A \mid I(1)) &\leq \sum_{\ell=1}^{\infty}\sum_{k=1}^{4\cdot(\ell+1)^2} \mathcal{P}(X_{m_\ell+k} \in A \mid I(1))\\
&\leq \sum_{\ell=1}^{\infty} \frac{3}{2}\frac{1}{8^{\ell+7}} 4\cdot(\ell+1)^2\\
&= \sum_{\ell=1}^{\infty} \frac{(\ell+1)^2}{8^{\ell+6}}\\
&\leq \sum_{\ell=1}^{\infty} \frac{1}{4^{\ell+6}}\frac{(\ell+1)^2}{2^{\ell+6}}\\
&\leq \sum_{\ell=1}^{\infty} \frac{1}{4^{\ell+6}} \leq \frac{1}{4}
\end{aligned}
$$

Here the last inequality follows from the fact that $(\ell+1)^2 < 2^{\ell+6}$ for all $\ell \geq 1$. Finally,

$$\mathcal{P}(\mathsf{N} \mid \mathrm{I}(1)) = \mathcal{P}(\bigwedge_{\ell=1}^{\infty} \bigwedge_{k=1}^{4\cdot(\ell+1)^2} X_{m_\ell+k} \notin A \mid \mathrm{I}(1)) = 1 - \mathcal{P}(\bigvee_{\ell=1}^{\infty} \bigvee_{k=1}^{4\cdot(\ell+1)^2} X_{m_\ell+k} \in A \mid \mathrm{I}(1)) \geq \frac{3}{4}$$

We first define a Markov chain $\mathcal{S} = (S, \to, P)$, in which probability of reaching given states is at most $\frac{2}{3}$. Then we show that there is a correspondence among finite paths of $\mathcal{S}$ and $\mathcal{E}_{(s,q)}$, from which the lemma follows.

The set S contains vertices $a_i$ for all $i > \mathbb{N}_0$ and $b_i$ for all $i > \mathbb{N}$. For all $i \in \mathbb{N}$ there are transitions $a_i \xrightarrow{\frac{1}{8^i}} b_i$, $a_i \xrightarrow{\frac{1}{4}} a_{i-1}$, $a_i \xrightarrow{\frac{3}{4}-\frac{1}{8^i}} a_{i+1}$, $b_i \xrightarrow{1} b_i$ and $a_0 \xrightarrow{1} a_0$. Using standard results of probability theory one can show that the probability of reaching the set $\{b_i \mid i \in \mathbb{N}\} \cup \{a_0\}$ is at most $\frac{2}{3}$.

Let R be a set of states $v_0$ of $\mathcal{E}_{(s,q)}$ such that either $v_0 = s$, or $v_0 = v't$ where $C(v') \neq C(v_0)$. For each $v_0 \in R$ we define sets $\mathrm{Next}_{>}(v_0), \mathrm{Next}_{<}(v_0) \subseteq R$ and $B(v_0) \subseteq D_0$ as follows. $\mathrm{Next}_{\bowtie}(v_0)$ where $\bowtie \in \{<, >\}$ is a set of all states $v_n \in R$ for which $C(v_n) \bowtie C(v_0)$ and there is a finite path $v_0, v_1, \ldots, v_n$ in $\mathcal{E}_{(s,q)}$ such that $C(v_0) = C(v_i)$ for all $1 \leq i < n$. The set $B(v_0)$ is a set of all states $v_n \in D_0$ for which there is a finite path $v_0, v_1, \ldots, v_n$ in $\mathcal{E}_{(s,q)}$ such that $C(v_0) = C(v_i)$ for all $1 \leq i < n$.

We define a function $\Theta : R \cup D_0 \to S$ by $\Theta(v) = a_i$ if $C(v) = i$ and $v \notin D_0$, and $\Theta(v) = b_i$ if $C(v) = i$ and $v \in D_0$. In particular, $\Theta(E_{(s,q)} \cap (C_0 \cup D_0)) = \{a_0\} \cup \{b_i \mid i \in \mathbb{N}\}$. The function $\Theta$ reveals the correspondence between $\mathcal{E}_{(s,q)}$ and $\mathcal{S}$: to each path from $v \in R \cup D_0$ to $v' \in R \cup D_0$ corresponds a unique path from $\Theta(v)$ to $\Theta(v')$ in $\mathcal{S}$ (note that this correspondence is not injective). Observe that for each state $v \in R$ the probability of reaching $B(v)$ and $\mathrm{Next}_{<}(v)$ is at most $8^{-C(v)}$ and $\frac{1}{4}$, respectively (here the role of the number $\lambda$ from the definition of C is crucial). Now it can be shown that the probability of reaching $C_0 \cup D_0$ from s can not be greater than the probability of reaching $\{a_0\} \cup \{b_i \mid i \in \mathbb{N}\}$ from $a_1$. □

Now we show that $\mathcal{E}_{(s,q)}$ satisfies the properties 1. – 3. The property 1. follows immediately from Lemma B.4. The property 2. follows from the following lemma.

**Lemma B.5.** *For every $v \in E_{(s,q)}$ and every $p \in L_{>0}(v)$ there is a finite path $\omega$ in $\mathcal{E}_{(s,q)}$ initiated in $v$ such that $\mathrm{Comp}(p, \omega) \neq \perp$ and last$(\omega)$ is either a leaf of $\mathcal{E}_{(s,q)}$, or last$(\omega)$ satisfies $\mathrm{Acc}^{>0}(\mathit{last}(\mathrm{Comp}(p, \omega)))$ in $\mathcal{E}_{(s,q)}$.*

*Proof.* If there is a path $\omega$ in $G^s_\xi$ initiated in $v$ such that $\mathrm{Comp}(p, \omega) \neq \perp$ and $\mathit{last}(\omega) \in D_0$, then we are done. In what follows we assume that this case does not occur.

Assume that $p \in I(\nu)$. We proceed by induction on $|I(\nu)|$. Let $\omega$ be the unique infinite path in $G^{s}_{\xi}$ initiated in $\nu$ such that for all $i \geq 0$ we have that $J(\omega(i)) = \mathrm{Comp}(J(\nu), \omega)(i)$. Note that for all $i, j \geq 0$ such that $\mathit{last}(\omega(i)) = \mathit{last}(\omega(j)) \in V_{\bigcirc}$ and $J(\omega(i)) = J(\omega(j))$ we have $\mathit{last}(\omega(i+1)) = \mathit{last}(\omega(j+1))$.

We claim that either there are only finitely many occurrences of vertices of $V_{\bigcirc}$ in $\omega$, or $\lim_{j\to\infty} C(\omega(j)) = -\infty$. Indeed, assume that there are infinitely many occurrences of vertices of $V_{\bigcirc}$ in $\omega$. Because $\xi$ is an FD strategy, we obtain that there are $0 \leq k, \ell \leq \mathit{size}$ such that for all $j \geq k$ we have $\mathit{last}(\omega(j)) = \mathit{last}(\omega(j+\ell))$ and $J(\omega(j)) = J(\omega(j+\ell))$ and $\mathit{last}(\omega(j+1)) = \mathit{last}(\omega(j+\ell+1))$. Hence, whenever for some $j \geq k$ we have that $\mathit{last}(\omega(j)) \in V_{\bigcirc}$, then there are numbers $0 < j_1 < j_2 < \cdots < j_\lambda \leq \lambda \cdot \mathit{size}$ such that $\mathit{last}(\omega(j)) = \mathit{last}(\omega(j_1)) = \mathit{last}(\omega(j_2)) = \cdots = \mathit{last}(\omega(j_\lambda))$ and $J(\omega(j)) = J(\omega(j_1)) = J(\omega(j_2)) = \cdots = J(\omega(j_\lambda))$ and $\mathit{last}(\omega(j+1)) = \mathit{last}(\omega(j_1+1)) = \mathit{last}(\omega(j_2+1)) = \cdots = \mathit{last}(\omega(j_\lambda+1))$. It follows that for all $j \geq k$ holds $C(\omega(j)) > C(\omega(j+\lambda\cdot\mathit{size}))$.

Assume that $\mathrm{Comp}(p, \omega) \neq \bot$. If there are infinitely many vertices of $V_{\bigcirc}$ in $\omega$, then $\lim_{j\to\infty} C(\omega(j)) = -\infty$ and we are done. If there are only finitely many vertices of $V_R \cup V_{\bigcirc}$ in $\omega$, then $\mathcal{P}(\{\omega\}) > 0$ and $\mathrm{Comp}(p, \omega)$ is accepting by Definition 3.2 (3.) and we are done. Assume that there are finitely many vertices of $V_{\bigcirc}$ in $\omega$ and infinitely many vertices of $V_R$ in $\omega$. Let $i \geq 0$ be a number such that for all $j \geq i$ we have that $\omega(j) \notin V^*V_{\bigcirc}$. Let us denote $q = \mathit{LS}(p, \omega^i)$. By our assumption, for all $j \geq i$ we have $\omega(j) \notin D_0$ and also if $\omega(j) \in V^*V_R$ and $\omega(j)t$ is a successor of $\omega(j)$ distinct from $\omega(j+1)$, then $\delta(\mathrm{Comp}(p,\omega)(j), \mathit{last}(\omega(j))) \notin L_{>0}(t)$ (because $\omega(j)t \in D_0$ by definition). It is easy to show that $\mathcal{P}(\{\omega\}) > 0$. However, $\mathit{last}(\omega(i)) \in X$ and thus there is a strategy $\xi'$ such that $\omega(i) \models_{\xi'} \mathrm{Acc}^{>0}(q)$ because $(\mathit{last}(\omega(i)), q) \in \mathrm{EP}(X)$. By the above arguments we have that $\omega$ is the only run of $\mathit{Run}[G_{\xi'}](\omega(i))$ such that $\mathrm{Comp}(q, \omega) \neq \bot$. Hence, $\mathrm{Comp}(q, \omega)$ has to be accepting.

Let us assume that $\mathrm{Comp}(p, \omega) = \bot$. Let $k$ be the least number such that $\mathrm{Comp}(p, \omega^{k+1}) = \bot$. Let $\nu'$ be the successor of $\omega(k)$ distinct from $\omega(k+1)$. Note that $|I(\nu')| < |I(\omega(k)| \leq I(\nu)$ (in particular, $\delta(J(\omega(k)), \mathit{last}(\omega(k))) \notin I(\nu')$ by definition) and that $\mathrm{Comp}(p, \omega^k \cdot \nu') \neq \bot$. Let us denote $p' = \mathit{LS}(p, \omega^k \cdot \nu')$. Because $|I(\nu')| < |I(\nu)|$, we may apply induction hypothesis and conclude that the proposition holds for $\nu'$ and $p'$. However, then clearly it holds also for $\nu$ and $p$.

Now assume that $p \notin I(\nu)$. Similarly as above, we proceed by induction on $|I(\nu)|$. If $|I(\nu)| = 0$, then there is a successor $\nu'$ of $\nu$ such that $\delta(p, \mathit{last}(\nu)) \in I(\nu')$, and we can apply the above argument to $\nu'$ and $\delta(p, \mathit{last}(\nu))$ and obtain the result. Now assume that

$|I(v)| > 0$. Let $\omega$ be the unique infinite path in $G^s_\xi$ initiated in $v$ such that for all $i \geq 0$ we have that $J(\omega(i)) = \mathsf{Comp}(J(v), \omega)(i)$. If $\mathsf{Comp}(p, \omega) \neq \bot$, then we can use the same argumentation as above. Assume that $\mathsf{Comp}(p, \omega) = \bot$. Let $k$ be the least number such that $\mathsf{Comp}(p, \omega^{k+1}) = \bot$. Let $v'$ be the successor of $\omega(k)$ distinct from $\omega(k+1)$. Then $|I(v')| < |I(v)|$ and we may apply induction to $v'$ and $LS(p, \omega^k \cdot v')$ and obtain the result. $\square$

The following lemma is an immediate consequence of Proposition B.3.

**Lemma B.6.** *For every $v \in E_{(s,q)}$ and every $p \in L_{=1}(v)$, almost all $\omega \in IPath[\mathcal{E}_{(s,q)}](v)$ satisfy that $\mathsf{Comp}(p, \omega)$ is accepting.*

Now we show how the trees $\mathcal{E}_{(s,q)}$ can be combined together to an infinite tree that is generated by a consistent strategy.

In what follows we use the following concept to simplify our notation. Let $\mathcal{T} = (T, P)$ be a probabilistic tree rooted in $s \in V$, let $I : T \to 2^{Q_{>0}}$, and let $A \subseteq Q_{>0}$ be a nonempty set. We say that $I$ is a *labeling of $\mathcal{T}$ determined by* $A$ iff $I(s) = A$ and for all states $v \in T$ and all successors $w$ of $v$ in $\mathcal{T}$ we have that

$$I(w) = \begin{cases} \delta(I(v), \mathit{last}(v)) \cap L_{>0}(w) & \text{if } \delta(I(v), \mathit{last}(v)) \cap L_{>0}(w) \neq \emptyset; \\ L_{>0}(w) & \text{otherwise.} \end{cases}$$

where $\delta(I(v), \mathit{last}(v)) = \{\delta(p, \mathit{last}(v)) \mid p \in I(v)\}$.

For every $s_0 \in X$ and let $q_0 \in L_{>0}(s_0)$ we fix a sequence $s_0, \ldots, s_n$ of vertices of $X$ and a sequence $q_0, \ldots, q_n$ of states of $\mathcal{M}$ such that $(s_n, q_n) \in EP(X)$ and for all $0 < i \leq n$ we have that $(s_{i-1}, s_i) \in E$ and $\delta(q_{i-1}, s_{i-1}) = q_i$ and $q_i \in L(s_i)$. We denote $ep(s_0, q_0) = s_0 \cdots s_n$ and $es(s_0, q_0) = q_n$. We define a probabilistic tree $\mathcal{T}_{(s_0,q_0)} = (T_{(s_0,q_0)}, P_{(s_0,q_0)})$ where

$$T_{(s_0,q_0)} = \{s_0 \cdots s_i \mid 0 \leq i \leq n\} \cup \{s_0 \cdots s_i t \mid 0 \leq i < n, s_i \in V_\bigcirc \cup V_R, (s_i, t) \in E\}$$

and

$$P_{(s_0,q_0)}(s_0 \cdots s_i, s_0 \cdots s_i t) = \begin{cases} \mathit{Prob}(s_i, t) & \text{if } s_i \in V_\bigcirc; \\ \frac{1}{2} & \text{if } s_i \in V_R; \\ 1 & \text{otherwise.} \end{cases}$$

We define a sequence of probabilistic trees $\mathcal{T}^n = (T^n, P^n)$ and labelings $I^n : T^n \to 2^{Q_{>0}}$ as follows: We define $\mathcal{T}^0 = (\{s_{in}\}, \emptyset)$ and $I^0(s_{in}) = L_{>0}(s_{in})$. Now let us assume that

$\mathcal{T}^{n-1}$ and $I^{n-1}$ have already been defined. Let $u$ be a leaf of $\mathcal{T}^{n-1}$ of minimal distance from the root. Let $q = \min(I^{n-1}(u))$, let $v = ep(\mathit{last}(u), q)$, and let $p = es(\mathit{last}(u), q)$. We define

$$\mathcal{T}^n = \mathcal{T}^{n-1} \odot_u (\mathcal{T}_{(\mathit{last}(u),q)} \odot_v \mathcal{E}_{(\mathit{last}(v),p)})$$

Let $K : T_{(\mathit{last}(u),q)} \to 2^{Q_{>0}}$ be the labeling of $\mathcal{T}_{(\mathit{last}(u),q)}$ determined by $I^{n-1}(u)\setminus\{q\}$. We define a labeling $I'$ obtained from $K$ by changing the value for $\mathit{last}(u)$ to $I^{n-1}(u)$. Now let $I'' : E_{(\mathit{last}(v),p)} \to 2^{Q_{>0}}$ be the labeling of $\mathcal{E}_{(\mathit{last}(v),p)}$ determined by $I'(v)$. Now labelings $I^{n-1}$, $I'$ and $I''$ induce a labeling $I^n$ of $\mathcal{T}^n = \mathcal{T}^{n-1} \odot_u (\mathcal{T}_{(\mathit{last}(u),q)} \odot_v \mathcal{E}_{(\mathit{last}(v),p)})$ in an obvious way.

Let us define $\mathcal{T} = (\bigcup_{n=0}^{\infty} T^n, \bigcup_{n=0}^{\infty} P^n)$ and $I = \bigcup_{n=0}^{\infty} I^n$. Observe that there is a srHC strategy $\sigma$ such that $\mathcal{T} = G_{\sigma}^{s_{in}}$.

**Lemma B.7.** *If $w$ is a state of $\mathcal{T}$ and $r \in L_{>0}(w)$, then $w$ satisfies $\mathrm{Acc}^{>0}(r)$.*

*Proof.* Assume that $w$ is an inner state of $\mathcal{T}^n = \mathcal{T}^{n-1} \odot_u (\mathcal{T}_{(\mathit{last}(u),q)} \odot_v \mathcal{E}_{(\mathit{last}(v),p)})$ but is not an inner state of $\mathcal{T}^{n-1}$.

Let us first assume that $r \in I(w)$. We proceed by induction on $|I(w)|$. We distinguish two cases:

1a. $w = u$ : If $r = q$, then we are done due to Lemma B.4, Lemma B.3, and definition of $\mathcal{T}_{(\mathit{last}(u),q)} \odot_v \mathcal{E}_{(\mathit{last}(v),p)}$. Assume that $r \neq q$. Then $|I(w)| \geq 2$ and there is $t$ such that $\delta(r, \mathit{last}(w)) \in I(wt)$ and by definition $|I(wt)| < |I(w)|$. We may apply the induction hypothesis and conclude that $wt$ satisfies $\mathrm{Acc}^{>0}(\delta(r, \mathit{last}(w)))$, which implies that $w$ satisfies $\mathrm{Acc}^{>0}(r)$.

2a. $w \neq u$ : Then by Lemma B.5 and the definition of $\mathcal{T}_{(\mathit{last}(u),q)} \odot_v \mathcal{E}_{(\mathit{last}(v),p)}$ there is a finite path $\omega$ initiated in $w$ such that $\mathrm{Comp}(r,\omega) \neq \perp$ and $|I(\mathit{last}(\omega))| \leq |I(w)|$ and either $\mathit{last}(\omega)$ satisfies $\mathrm{Acc}^{>0}(\mathit{LS}(r,\omega))$ in $\mathcal{T}^n$, or $\mathit{last}(\omega)$ is a leaf of $\mathcal{T}^n$. In the former case we are done. Let us assume that $u' = \mathit{last}(\omega)$ is a leaf of $\mathcal{T}^n$. There is $m \geq n$ such that $\mathcal{T}^m = \mathcal{T}^{m-1} \odot_{u'} (\mathcal{T}_{(\mathit{last}(u'),q')} \odot_{v'} \mathcal{E}_{(\mathit{last}(v'),p')})$ for some $q'$, $p'$, and $v'$. Now we may resort to the previous case and conclude that $u'$ satisfies $\mathrm{Acc}^{>0}(\mathit{LS}(r,\omega))$ in $\mathcal{T}$, which implies that $w$ satisfies $\mathrm{Acc}^{>0}(r)$ in $\mathcal{T}$.

Now assume that $r \notin I(w)$. We proceed by induction on $|I(w)|$. If there is a finite path $\omega$ initiated in $w$ such that $\mathrm{Comp}(r,\omega) \neq \perp$ and $\mathit{LS}(r,\omega) \in I(\mathit{last}(\omega))$, then we may resort to the previous case and conclude that $w$ satisfies $\mathrm{Acc}^{>0}(r)$ in $\mathcal{T}$. We distinguish two cases:

1b. $w = u$ : There is $t$ such that $wt$ is a successor of $w$ and $\delta(r, \mathit{last}(w)) \in L_{>0}(wt)$. If $|I(w)| = 1$, then $\delta(r, \mathit{last}(w)) \in I(wt)$ which contradicts our assumptions. Hence, $|I(w)| \geq 2$ and by definition $|I(wt)| < |I(w)|$. We apply induction hypothesis and conclude that $wt$ satisfies $\mathrm{Acc}^{>0}(\delta(r, \mathit{last}(w)))$, and thus that $w$ satisfies $\mathrm{Acc}^{>0}(r)$.

2b. $w \neq u$ : It follows from our assumptions that for every finite path $\omega$ initiated in $w$ such that $\mathrm{Comp}(r, \omega) \neq \bot$ we have that $|I(\omega(i))| \leq |I(w)|$. Hence, we can use precisely the same argument as in 2a.

□

Given $s \in X$ and $q \in L_{>0}(s)$ we denote $\mu_{(s,q)}$ the probability of reaching a leaf of $\mathcal{T}_{(s,q)}$ along the path $wp(s, q)$. Let us denote $\mu = \min\{\mu_{(s,q)} \mid s \in X \text{ and } q \in L_{>0}(s)\}$. For every $n \geq 0$ and $v \in T^n$ we denote $p^v_n$ the probability of reaching a leaf of $T^n$ from $v$.

**Lemma B.8.** *Let $v \in \mathcal{T}^n$. For every $k \geq n$ there is $m \geq k$ such that*

$$p^v_m \leq \frac{1}{2}p^v_k(1 - \mu\frac{1}{3}) + \frac{1}{2}p^v_k \leq p^v_k(1 - \frac{\mu}{3})$$

*Hence,* $\lim_{k\to\infty} p^v_k = 0$.

*Proof.* Given $j \geq 0$, we denote $p^v_{k,\leq j}$ and $p^v_{k,>j}$ the probabilities of reaching a leaf of $\mathcal{T}^k$ from $v$ in at most $j$ steps and in at least $j + 1$ steps, respectively. There is $j \geq 0$ such that $p^v_{k,\leq j} \geq \frac{1}{2}p^v_k$ and hence $p^v_{k,>j} = p^v_k - p^v_{k,\leq j} \leq p^v_k - \frac{1}{2}p^v_k = \frac{1}{2}p^v_k$. By definition, there is $m \geq k$ such that each leaf reachable in $\mathcal{T}^k$ in at most $j$ steps is not a leaf in $\mathcal{T}^m$. Let $\mathcal{U}$ be the set of all leaves of $\mathcal{T}^k$ reachable from $v$ in at most $j$ steps. By Lemma B.4, the probability that a leaf of $\mathcal{T}^m$ is not reached from a fixed state of $\mathcal{U}$ is at least $\mu\frac{1}{3}$. Let $\nu$ be the maximal probability of reaching a leaf of $\mathcal{T}^m$ from a state of $\mathcal{U}$. Clearly $\nu \leq 1 - \mu\frac{1}{3}$.

Now a leaf of $\mathcal{T}^m$ can be reached from $v$ in two ways: Either follow a path of length less than or equal to $j$ to a state of $\mathcal{U}$ and then reach a leaf of $\mathcal{T}^m$, or avoid states of $\mathcal{U}$ in the first $j$ steps and then reach a leaf of $\mathcal{T}^m$. The first case has the probability at most $p^v_{k,\leq j} \cdot \nu \leq p^v_{k,\leq j} \cdot (1 - \mu\frac{1}{3})$. The second case has the probability at most $p^v_{k,>j}$. Hence, $p^v_m \leq p^v_{k,\leq j} \cdot (1 - \mu\frac{1}{3}) + p^v_{k,>j} \leq \frac{1}{2}p^v_k(1 - \mu\frac{1}{3}) + \frac{1}{2}p^v_k$.

Let us denote $\rho = 1 - \frac{\mu}{3} < 1$. We show that for every $j$ there is $k \geq n$ such that $p^v_k \leq \rho^j$ (this implies that $\lim_{k\to\infty} p^v_k = 0$ because $p^v_k \geq p^v_{k+1}$ for every $k \geq n$). For $j = 0$ we have $p^v_k \leq \rho^j = 1$. Assume that $p^v_k \leq \rho^j$. There is $m \geq k \geq n$ such that $p^v_m \leq p^v_k\rho \leq \rho^{j+1}$ and we are done. □

**Lemma B.9.** *If $w$ is a state of $\mathcal{T}$ and $r \in L_{=1}(w)$, then $w$ satisfies* $\mathrm{Acc}^{=1}(r)$.

*Proof.* Let us assume that $w \in \mathcal{T}^n$. For $m \geq n$ we denote $A_m \subseteq \mathit{IPath}[\mathcal{T}^m](w)$ and denote $B_m$ the set of all $\omega \in A_m$ such that $\mathsf{Comp}(r, \omega)$ is accepting. It follows from Lemma B.6 that $\mathcal{P}(B_m \mid A_m) = 1$. Also $B_m \subseteq B_{m+1} \subseteq \cdots$ and by Lemma B.8 we have that $\mathcal{P}(\bigcup_{k=m}^{\infty} B_m) = \lim_{m\to\infty} \mathcal{P}(B_m) = \lim_{m\to\infty} \mathcal{P}(A_m) = \lim_{m\to\infty} 1 - p_m^w = 1$. Thus almost all $\omega \in \mathit{IPath}[\mathcal{T}](w)$ satisfy that $\mathsf{Comp}(r, \omega)$ is accepting, and hence $w$ satisfies $\mathsf{Acc}^{=1}(r)$. □

### B.2.2 Complexity of Strategy Synthesis

Let us assume that for every $(s, q) \in \mathsf{EP}(X)$ there is a finite-memory strategy $\xi_{(s,q)}$ witnessing that $(s, q) \in \mathsf{EP}(X)$. In section B.2.3 we show that the size of each strategy $\xi_{(s,q)}$ is in $(|G||\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|mn)}$. Let us assume without the loss of generality that the size of each tree $\mathcal{T}_{(s,q)}$ is polynomial. Observe that each tree $\mathcal{E}_{(s,q)}$ is induced by the strategy $\xi_{(s,q)}$ which stops whenever the counter is 0. The strategy inducing $\mathcal{T}$ should behave like $\xi_{(s,q)}$ in $\mathcal{E}_{(s,q)}$ part of $\mathcal{T}$. Once a leaf of $\mathcal{E}_{(s,q)}$ is reached, then the strategy switches to the mode in which traverses the trees $\mathcal{T}_{(s,q)}$ until it reaches the root of another tree $\mathcal{E}_{(s',q')}$. The order in which the trees $\mathcal{T}_{(s,q)}$ are traversed is prescribed by the labeling I.

The information needed by a strategy $\tau$ which generates $\mathcal{T}$ is following:

1. values of the labelings I, J, C in the current state;
2. the vertex $t$ that occurred directly after the last change of the counter C, the number of occurrences of $t$ after the last change of the counter C (up to $\lambda$), and the information whether all of these occurrences of $t$ were followed by the same vertex;
3. the number of transitions after the last change of the counter C (up to the numer $\mathit{size}$)
4. the position in a tree $\mathcal{T}_{(s,q)}$ if the play is currently in the $\mathcal{T}_{(s,q)}$ part of $\mathcal{T}$;
5. the current state of the strategy $\xi_{(s,q)}$ if the play is currently in the $\mathcal{E}_{(s,q)}$ part of $\mathcal{T}$.

It is easy to see that the size of this information is in $(\lambda \cdot |G||\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|mn)}$. Moreover, this information can be stored in states of a finite-state automaton and updated while reading vertices of the MDP G. Thus, it is a mere technicality to define a one-counter strategy $\tau$ of the size $(\lambda \cdot |G||\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|mn)}$ which generates $\mathcal{T}$.

### B.2.3 Proof of Proposition 3.3: Left to Right

We define $X$ to be the set of all vertices reachable from $s_{in}$ in $G_\sigma$. Let $s_0 \in X$ and let $q_0 \in L_{>0}(s_0)$. Also, let us fix $v_0$ a state reachable from $s_{in}$ in $G_\sigma$ such that $\mathtt{last}(v_0) = s_0$.

We inductively define a sequence of functions $I_0, I_1 ... I_{|Q|} : V^+ \times Q \to \{\bot, \bullet\} \cup \mathcal{A}$, states $u_0, u_1, \dots, u_{|Q|} \in V^+$ and sets of runs $R_0, R_1, \dots R_{|Q|} \subseteq \mathit{Run}[G_\sigma](s_{in})$ as follows.

**Base step** There is a state $u_0$ reachable from $v_0$, a set $A \in \mathcal{A}$ and a set of runs $R_0$ of nonzero measure such that the following holds: Every $\omega \in R_0$ contains $u_0$, and we have $\mathtt{occ}(\mathtt{Comp}(\bar{q}, \omega')) = \mathtt{inf}(\mathtt{Comp}(\bar{q}, \omega')) = A$ where $\bar{q} = \mathtt{LS}(q_0, v_0 \dots u_0)$ and $\omega'$ is the prefix of $\omega$ initiated in $u_0$.

We put $I_0(v', q') = A$ for all tuples $(v', q')$ such that $q' = \mathtt{LS}(\bar{q}, u_0 \dots v')$ and $u_0 \dots v'$ is a finite path contained in some run in $R_0$. Also, for all tuples $(v', q')$ such that there are $v''$ and $q''$ satisfying $I_0(v'', q'') = A$ and $\mathtt{LS}(q', v' \dots v'') = q''$ we define $I_0(v', q') = \bullet$. Otherwise, we set $I_0(v', q') = \bot$.

**Induction step** Now suppose we have fixed $I_i$, $u_i$ and $R_i$. First, suppose there is $u \in V^+$ and $q \in L(u)$ such that $I_i(u, q) = \bot$ and there is a nonzero measure of runs $\omega \in R_i$ such that $\mathtt{Comp}(q, \omega')$ (where $\omega'$ is the suffix of $\omega$ initiated in $v$) is accepting. Then there is $u_{i+1} \in V^+$, $q_{i+1} \in L(v_{i+1})$, $A \in \mathcal{A}$ and $R_{i+1} \subseteq R_i$ such that $\mathtt{inf}(\mathtt{Comp}(q_{i+1}, \omega)) = \mathtt{occ}(\mathtt{Comp}(q_{i+1}, \omega)) = A$ for all $\omega \in \mathtt{Run}[G_\sigma](u_{i+1})$ that are a suffix of some run from $R_{i+1}$. For all states $v'$ and $q' \in L(v')$ we define $I_{i+1}(v', q')$ as follows:

- If $I_i(v', q') \neq \bot$, then $I_{i+1}(v', q') = I_i(v', q')$
- We put $I_{i+1}(v', q') = A$ if $q' = \mathtt{LS}(q_{i+1}, u_{i+1} \dots v')$ for $u_{i+1} \dots v'$a finite path contained in some run from $R_{i+1}$.
- For all tuples $(v'', q'')$ such that $I_0(v', q') = A$ and $\mathtt{LS}(q'', v'' \dots v') = q'$ and $v'' \dots v'$ is a finite path contained in some run from $R_{i+1}$, we define $I_0(v'', q'') = \bullet$.
- Otherwise, we set $I_0(v', q') = \bot$.

If there is no such $u$, we put $I_{i+1} = I_i$, $u_{i+1} = u_i$ and $R_{i+1} = R_i$. Finally, we denote $v = u_{|Q|}$, $R = R_{|Q|}$ and $I = I_{|Q|}$.

In the rest of this section, we prove the following lemma.

**Lemma B.10.** *Let* $q = \mathtt{Comp}(\bar{q}, v_0 \dots v)$*. The tuple* $(\mathtt{last}(v), q)$ *is an entry point for* $X$*.*

To see this, we define a HD strategy $\xi$ that witnesses that $(\mathtt{last}(v), q)$ is an entry point for $X$.

For each state $u$ reachable from $v$ in $G_\sigma$ and for all $r \in L(u)$ we fix a sequence $\alpha(u, r) = u_0u_1 \dots u_n$ of states such that all of the following holds:

- $u_0 = u$
- for all $i \geq 0$ and $r' \in L(u_i)$ we have that if $\delta(r', u_i) \in L(u_{i+1})$, then $I(u_i, r') \leq I(u_{i+1}, \delta(r', u_i))$ (here, we impose the ordering in which $\perp < \bullet$ and $\bullet < A$ for all $A \in \mathcal{A}$)
- $\sigma$ assigns a nonzero probability to go to $\mathtt{last}(u_{i+1})$ from $u_i$ if $u_i \in V^*V_\Box$.
- If $I(u_0, r) = A$ for some $A \in \mathcal{A}$, then $\mathtt{Comp}(r, u_0 \dots u_n)$ contains all states from $A$.
- If $I(u_0, r) = \bullet$, then $I(u_n, r') = A$ for some $A \in \mathcal{A}$ where $r' = \mathrm{LS}(r, u_0 \dots u_n)$.
- If $I(u_0, r) = \perp$, then one of the following holds:
  - For $r' = \mathrm{LS}(q, u_0 \dots u_n)$ we have $I(u_n, r') = \bullet$ or $I(u_n, r') = A$ for some $A \in \mathcal{A}$.
  - No vertex from $V_\bigcirc$ is reachable from $u$ in $G_\sigma$, $\mathtt{last}(u_{n-1}) \in V_R$ and for the successor $v'$ of $\mathtt{last}(u_{n-1})$ different from $\mathtt{last}(u_n)$, we have that $\mathtt{Comp}(r, u_0 \dots u_{n-1}u')$ is defined.

In the following, we define the strategy $\xi$ inductivelly. Together with $\xi$, we build a function $J : V^+ \times Q \to \{\perp, \bullet\} \cup \mathcal{A}$ and a function $K$ that to each state $u \in V^+$ assigns a linear order on $L(u)$.

Let $\alpha(v, q) = u_0 \dots u_n$ (where $q \in L(v)$ is arbitrary state) and let us denote $s_i = \mathtt{last}(u_i)$ for all $0 \leq i \leq n$. For each $0 \leq i < n$, whenever $s_i \in V_\Box$, we define $\xi(s_0, \dots, s_i) = s_{i+1}$. For each $0 \leq i \leq n$ and $r \in L(s_i)$ we define $J(s_0 \dots s_i, r) = I(u_i, r)$ and for all states $v'$ that are of the form $s' = s_0 \dots s_i s'_{i+1} \dots s'_m$ where $s'_j \in V_\bigcirc$ for $i < j \leq m$, and for all $r \in L(v')$ we define $J(v', r) = I(u_i s'_{i+1} \dots s'_m, r)$.

For all states $v'$ that are of the form $s_0 \dots s_i s'_{i+1} \dots s'_m$ where $0 < i \leq n$ and $m \leq i$, we define $K(v')$ so that it satisfies the following (here, $us$ and $uss'$ are two successive states):

- If for some $r', r'' \in L(s)$ we have $r' < r''$ in the order $K(us)$, then $\delta(r', s) < \delta(r'', s)$ in the order $K(uss')$.
- All states $r' \in L(s')$ for which there is no $r'' \in L(s)$ such that $\delta(r', s) = r''$ are placed at the end of the order $K(uss')$.

Now let $u \in V^*V_\Box$ be a state such that $\xi$ is defined for all predecessors of $u$ but is not defined for $u$, and let $r$ be the first state in $K(u)$. Let us take a state $u'$ that occurs in some $\omega \in R$ such that $L(u) = L(u')$ and for all $r' \in L(u)$, $J(u, r') = I(u', r')$. Let $\alpha(u', r) = u_0, \dots u_n$ and let us denote $s_i = last(u_i)$ for all $0 \leq i \leq n$. For each $0 \leq i < n$, whenever $s_i \in V_\Box$, we define $\xi(us_1 \dots s_i) = s_{i+1}$. For all states $v'$ that are of the form $v' = us_0 \dots s_i s'_{i+1} \dots s'_m$ (where $m \geq i$ and $s_j \in V_\bigcirc$ for $i \leq j \leq m$), and for all $r' \in L(v')$, we define $J(v', r') = I(u_i s_0 \dots s_i s'_{i+1} \dots s'_m, r')$.

For all states $v'$ that are of the form $us_0 \dots s_i s'_{i+1} \dots s'_m$ where $0 < i \leq n$, $m \leq i$ and $s_j \in V_\bigcirc$ for $i \leq j \geq m$, we define $K(v')$ so that it satisfies the following (here, $v's$ and $v'ss'$ are two successive states):

- $\delta(q, v')$ is the last state in the order $K(v's_1)$
- If for some $r', r'' \in L(s)$ we have $r' < r''$ in the order $K(v's)$, then $\delta(r', s) < \delta(r'', s)$ in the order $K(v'ss')$. (with the exception of the previous item)
- All states $r' \in L(s')$ for which there is no $r'' \in L(s)$ such that $\delta(r', s) = r''$ are placed at the end of the order $K(v'ss')$.

In the remaining, we argue that the strategy $\xi$ is the strategy satisfying the conditions of Definition 3.2.

**Lemma B.11.** *For every state* $ut$ *of* $(G|X)$ *and every* $p \in L_{=1}(t)$ *we have* $ut \models_\xi Acc^{=1}(p)$

*Proof.* First, if $I(ut, p) = A$ for some $A \in \mathcal{A}$, then for all runs $\omega$ initiated in $ut$, $occ(Comp(p, \omega)) = inf(Comp(p, \omega)) = A$. Indeed, due to the definition of $\xi$ there can not exist $p' \notin A$ and $k$ such that $p' = LS(p, \omega(0) \dots \omega(k))$ (this is ensured by the definition of I). On the other hand, for almost all runs $\omega$ there is $k$ such that for $p' = LS(p, \omega(0) \dots \omega(k))$ we have that $p'$ is the first state in the ordering defined by $\omega(k)$. Then, there is a finite path $u_0, u_1, \dots u_n$ initiated in $\omega(k)$ such that $Comp(p', u_0, u_1 \dots u_n)$ contains all states of A. Because $(G|X)_\xi$ may be viewed as a finite Markov chain, one can easily see that $ut \models_\xi Acc^{=1}(p)$.

If $I(ut, p) = \bullet$, then for almost all runs $\omega$ initiated in $ut$ there is $k$ such that for $p' = LS(p, \omega(0) \dots \omega(k))$ we have $I(ut, p') = A$ for some $\mathcal{A}$. The remaining follows from the above paragraph.

The case $I(ut, p) = \perp$ may not occur due to the definition of $I_i$. ∎

To prove that $s \models_\xi Acc^{=1}(q)$, one uses arguments similar to the arguments from the first paragraph of Lemma B.11.

**Lemma B.12.** *For all states* $ut$ *of* $(G|X)_\xi$ *and all* $p \in L_{>0}(t)$ *we have the following: if there is no state of* $V^*V_\bigcirc$ *reachable from* $ut$ *in* $(G|X)_\xi$, *then either* $wt \models_\xi \mathtt{Acc}^{=1}(p)$, *or there is a finite path* $u_0t_0, \dots, u_kt_k$ *initiated in* $ut$ *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* $\xi(u_kt_k)$ *selects the edge* $(t_k, r_1)$ *and* $\delta(p, t_0 \cdots t_k) \in L_{>0}(r_2)$.

*Proof.* Suppose that there is no state of $V^*V_\bigcirc$ reachable from $ut$ in $(G|X)_\xi$ and let $\omega$ be the run initiated in $ut$. If $I(ut, p) \neq \perp$, one may apply same arguments as in the proof of Lemma B.11.

Otherwise, one of the following holds

- $\mathtt{Comp}(p, \omega) = \perp$. Then, let $\ell$ be the biggest number such that $\mathtt{Comp}(p, \omega(0) \dots \omega(\ell))$ is defined. Surely, we must have that $\mathtt{last}(\omega(k))$ is a state of $V_R$ satisfying the requirements of the lemma.
- $\mathtt{Comp}(p, \omega) \neq \perp$. Then, there is $\ell$ such that for $p' = LS(p, \omega(0) \dots \omega(\ell))$ we have that $p'$ is the first state of $K(\omega(k))$. Then there is a finite path $\bar{u}_1\bar{u}_2 \dots \bar{u}_n$ initiated in $\omega(k)$ such that $\mathtt{last}(\bar{u}_n) \in V_R$ is a state satisfying the conditions of the lemma.

□

Now let us analyze the size of $\xi$. Surely, $\xi$ is a finite memory strategy. Thus, one can perform the analysis using the results of [6]. First, we need to show that we can reduce the problem of the existence of the strategy satisfying the objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$ to the problem of the existence of a strategy satisfiyng a detPECTL* formula $\psi$. We construct the formula $\psi$ as follows. Let us order the set $Q$ into a sequence $q_1, q_2, \dots q_{|Q|}$. Remember that the winning condition $A$ of $\mathcal{M}$ is defined using two sets $\mathit{Rab} = \{(A_i, B_i) | 1 \leq i \leq n\}$ and $\mathit{Saf} = \{(C_i, D_i) | 1 \leq i \leq m\}$ where the first set specifies a Rabin winning condition and the latter specifies a Streett winning condition. For each tuple $(X, Y) \in \mathit{Rab}$ and state $q \in Q$ we construct an automaton $\mathcal{B}_{q,(X,Y)}$ as follows. The set of states of $\mathcal{B}_{(X,Y)}$ is equivallent to the set of states of $\mathcal{M}$, the accepting set is $X$ and the transitions are defined as follows:

- for all $q' \in Q \setminus Y$ and $q'' \in Q$ we have $q' \xrightarrow{T} q''$ if there is a transiton $q' \xrightarrow{s} q''$ in $\mathcal{M}$ and $T = \{i | q_i \in L(s)\}$.
- for all $q' \in Y$, $q' \xrightarrow{\emptyset} q'$.

For each tuple $(X, Y) \in \mathit{Saf}$ state $q \in Q$ we construct an automaton $\mathcal{B}_{q,(X,Y)}$ as follows. The states of $\mathcal{B}_{q,(X,Y)}$ are tuples $(q', x)$ where $q' \in Q$ and $x \in \{1, 2\}$, the initial state is $(q, 1)$, the accepting set is $\{(q, x)|x = 1 \vee q \in Y\}$ and the transitions are defined as follows:

- for all $q' \in Q \setminus X$ and $q'' \in Q$, we have $(q', 1) \xrightarrow{T} (q'', 1)$ if there is a transition $q' \xrightarrow{s} q''$ in $\mathcal{M}$ such that $T = \{i|q_i \in L(s)\}$.
- for all $q' \in X$ and $q'' \in Q$ we have $(q', 1) \xrightarrow{T} (q'', 2)$ if there is a transition $q' \xrightarrow{s} q''$ in $\mathcal{M}$ such that $T = \{i|q_i \in L(s)\}$.
- for all $q', q'' \in Q$ we have $(q', 2) \xrightarrow{T} (q'', 2)$ if there is a transition $q' \xrightarrow{s} q''$ in $\mathcal{M}$ such that $T = \{i|q_i \in L(s)\}$.

For each $\mathcal{B}_{q,(X,Y)}$ we create a formula $\psi_{q,(X,Y)} = \mathcal{B}_q^{=1}(q_1, \dots, q_{|Q|})$. Let us order all these formulae into an (arbitrary) sequence and let us denote $\psi(i)$ the i-th formula in the sequence.

Now for all $q \in Q$ we define an automaton $\mathcal{B}_q$. The automaton has states $Q \cup \{accept\}$, where $accept$ is a distinguished state. The state $accept$ is the only accepting state of $\mathcal{B}_q$, $q$ is the initial state and the transition function of $\mathcal{B}_q$ is defined as follows:

- for all $q', q'' \in Q$ we have $q' \xrightarrow{T} q''$ if there is a transiton $q' \xrightarrow{s} q''$ in $\mathcal{M}$ and $T = \{i|q_i \in L(s)\}$.
- for all $q' \in Q$ and $(X, Y) \in Rab$ there is a transiton $q' \xrightarrow{\{j\}} q''$ where $\psi(j - |Q|) = \psi_{q',(X,Y)}$.
- for all $q' \in Q$ there is a transiton $q' \xrightarrow{\{j_1,\dots,j_m\}} q''$ where $\psi(j_i - |Q|) = \psi_{q',(C_i,D_i)}$.
- $accept \xrightarrow{\emptyset} accept$

Now for $1 \leq i \leq |Q|$ we define $\phi(i) = \mathcal{B}_{q_i}^{\sim r}(q_1, \dots q_{|Q|}, \psi(1), \dots, \psi(|Q| * (m + n)))$ where $\sim r$ is either $> 0$ or $= 1$, depending on whether $q_i \in L_{>0}$, or $q_i \in L_{=1}$

Now we construct an automaton $\mathcal{B}$. $\mathcal{B}$ has the set of states $V$, initial state $s_{in}$ and transitions $s \xrightarrow{T} t$ where $T = \{i|q_i \in L(s)\}$. All states of $\mathcal{B}$ are accepting. We put $\psi = \mathcal{B}^{=1}(\phi(1), \dots, \phi(|Q|))$. It is easy to verify that the resulting formula is a detPECTL* formula and that the construction described above can be performed in polynomial time.

**Lemma B.13.** *For all FD strategies $\sigma$ the following holds: The consistency objective $(\mathcal{M}, (Q_{>0}, Q_{=1}, L)$ is achievable by $\sigma$ iff the formula $\psi$ is satisfied in $G_\sigma$ with valuation $L$.*

Now we can use the results of [6] and deduce that iff the consistency objective is achievable, then there is a strategy of size $(|G||\delta|^{\mathcal{O}(1)} 2^{|Q|mn}$. Moreover, such strategy can be synthesized in time $(|G||\delta|^{\mathcal{O}(1)} 2^{|Q|mn}$.

## B.3 Step (b)

**Corollary B.14.** *The problem of existence of a consistent HC strategy is decidable in time* $(|G| \cdot |\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|nm)}$. *Moreover, existence of a consistent HC strategy implies existence of a one-counter consistent HC strategy computable in time* $(|G| \cdot |\delta|)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(|Q|nm)}$.

*Proof.* There is a game $G' = (V', E', (V'_\Box, V_\bigcirc), \mathrm{Prob}')$, an automaton $\mathcal{M}' = (Q', V', \delta', \mathcal{A})$, a set $Q'_{>0} \subseteq Q'$, a set $V'_R \subseteq V'_\Box$, a set $V_D = V_\Box \setminus V_R$ and a labeling $L' : V' \to 2^Q$ such that there is a consistent HC strategy in $G$ iff there is a srHC strategy in $G'$ that achieves a consistency objective $(\mathcal{M}, (Q_{>0}, Q_{=1}), L)$. Applying Proposition 3.3, we obtain an algorithm that decides the existence of a consistent HC strategy in $G$.

We only sketch the construction of $G'$ here. For each $s \in V_R$ with successors $t_1$ and $t_2$ we add vertices $s_1$, $s_2$ and $s_{1,2}$ to $V_\Box$, remove transitions $(s, t_1)$, $(s, t_2)$ from $E$ and add transitions $(s_1, t_1)$, $(s_{1,2}, t_1)$, $(s_{1,2}, t_2)$ and $(s_2, t_2)$ to $E$. We put $L(s_1) = L(s_2) = L(s_{1,2}) = \{q' | \delta(q, s) = q' \text{ for some } q \in L(s)\}$ and $\delta(s_1, q) = \delta(s_2, q) = \delta(s_{1,2}, q) = q$ for all $q$. The set $V'_R$ is created from $V_R$ and contains the vertex $s_{1,2}$ instead of $s$.

The constructed game need not have the form required in Proposition 3.3 (e.g. it may not have binary branching), but it can be transformed using constructions given earlier in the Appendix. □