Probabilistic CTL*: The Deductive Way

Rayna Dimitrova[1], Luis María Ferrer Fioriti[2],
Holger Hermanns[2], and Rupak Majumdar[1]

[1]MPI-SWS, Germany [2]Saarland University, Germany

Abstract. Complex probabilistic temporal behaviours need to be guaranteed in robotics and various other control domains, as well as in the context of families of randomized protocols. At its core, this entails checking infinite-state probabilistic systems with respect to quantitative properties specified in probabilistic temporal logics. Model checking methods are not directly applicable to infinite-state systems, and techniques for infinite-state probabilistic systems are limited in terms of the specifications they can handle.
This paper presents a deductive approach to the verification of countable-state systems against properties specified in probabilistic CTL*, on models featuring both nondeterministic and probabilistic choices. The deductive proof system we propose lifts the classical proof system by Kesten and Pnueli to the probabilistic setting. However, the soundness arguments are completely distinct and go via the theory of martingales. Completeness results for the finite-state case and an infinite-state example illustrate the effectiveness of our approach.

1 Introduction

Temporal reasoning in the presence of choice and stochastic uncertainty is a fundamental problem in many domains. In the context of finite-state systems, such reasoning can be automated and a long line of research in probabilistic model checking has culminated in efficient tools that implement automatic model checking algorithms for Markov decision processes with specifications given in probabilistic temporal logics such as PCTL and PCTL* [26, 7, 9, 8, 2, 20]. When it comes to infinite-state systems, though, reasoning about probabilistic systems, barring a few special classes of properties such as safety or almost-sure termination, is mostly ad hoc. This is unfortunate, since many probabilistic systems are a priori infinite-state. For example, randomized distributed algorithms are often designed to work no matter how many agents participate in the system. Discrete time stochastic dynamical systems arising in control assume continuous and unbounded state spaces. More recently, probabilistic programming languages augment "normal" programming languages (with unbounded variables) with the ability to sample from probability distributions and to condition behaviors on observations. We would like to formally reason about the temporal behavior of these systems, but the current literature provides little direction.

In this paper, we extend the deductive approach to temporal logic verification to systems that combine non-determinism and probabilistic choice with

the (quantitative) probabilistic temporal logic PCTL*. Our central contribution is a novel set of proof rules enabling deductive proofs for PCTL and PCTL* properties on nondeterministic probabilistic programs with possibly infinite state space. We consider both qualitative and quantitative properties, and use martingale theory as our main mathematical tool. Conceptually, the rules we present for PCTL and PCTL* can be considered as a probabilistic enhancements of those developed by Kesten and Pnueli for CTL and CTL* [19]. At its core, the enhancement echoes the apparent analogy between classical *termination* proofs and proofs for *almost sure termination* of probabilistic programs. The latter was first studied in the pioneering work of Hart, Sharir, and Pnueli [16] as a particular liveness property. Their 0-1 law is the foundation of several semi-automatic approaches (e.g. [17, 21, 12]) for proving termination of finite and parametric systems. Pnueli [22] showed that the almost sure satisfaction of liveness properties on probabilistic systems can be reduced to the non-probabilistic case adding suitable fairness constraints. Pnueli and Zuck [23, 1] later extended this approach to a sound and complete characterization for finite state spaces. Almost sure properties do not depend on the actual probability values, but instead on the underlying graph structure. In contrast to this, the deductive rules developed in this paper do not rely on the graph structure. They instead reason about and deduce the "average" behaviour of the program. This makes it possible to analyse a considerably wider range of probabilistic programs and properties. We make use of Lyapunov ranking functions, a widely used technique for proving recurrence in Markov Chains. They were recently adapted to prove almost sure termination of term rewriting systems [5] and infinite-state (non)deterministic programs [6, 13]. We extend these techniques to full quantitative PCTL*.

When stretching the deductive approach of Kesten and Pnueli beyond PCTL, we must account for path formulas that describe ω-regular languages. In the non-probabilistic setting, Kesten and Pnueli reduce the reasoning about ω-regular properties to reasoning about safety or reachability under a *justice* assumption (justice is a form of fairness [15]). In the probabilistic setting, however, this reduction is unsound: a probabilistic program may not have any fair scheduler, thus the quantification over all fair schedulers is trivially satisfied, regardless of the original formula being invalid. The root cause of the problem is that a scheduler in the probabilistic setting generates a set of paths, opposed to just a single path in the non-probabilistic case. So, if a non-null set of paths is not fair, then the scheduler is not fair. To overcome this, we instead harvest and extend the martingale approach to checking qualitative termination [6, 13] with the power to directly handle general ω-regular conditions. This is achieved by a proof rule for Streett conditions which is complete in the finite-state case. The key step to prove soundness uses Levy's 0-1 law [11] to go to the limit behavior.

For finite-state systems, the proof rules we present are complete, but they are in general not complete for infinite-state systems. Technically, incompleteness is inherited from the fact that Lyapunov ranking functions are not complete for proving almost sure termination [13], in contrast to ranking functions wrt. ordinary termination. If they were complete, we would instantly obtain a com-

pleteness result, just as Kesten and Pnueli. However, even an incomplete set of proof rules can turn out to be very useful still, provided it can be effectively applied to interesting cases. For example, we can verify several parameterized randomized distributed algorithms, such as the choice coordination protocol by Rabin [24] using our proof system [10].

2 Probabilistic Systems and Logics

2.1 Probabilistic Systems

Preliminaries. A *probability space* [11] is a triple $(\Omega, \mathcal{F}, \mu)$ where Ω is a *sample space*, $\mathcal{F} \subseteq 2^{\Omega}$ is a σ-algebra, and $\mu : \mathcal{F} \to [0,1]$ is a probability measure. A *random variable* $X : \mathcal{F} \to \mathbb{R}$ on a probability space $(\Omega, \mathcal{F}, \mu)$ is a Borel-measurable function; it is *discrete* if there exists a countable set A such that $\mu(X^{-1}(A)) = 1$. A *random predicate* is a discrete random variable with co-domain $\{0,1\}$.

Given a probability space $(\Omega, \mathcal{F}, \mu)$, random predicates $P_1, \ldots, P_{n+1}$, real numbers $q_1, \ldots, q_n$, and binary relations $\bowtie_1, \ldots, \bowtie_n \in \{\leq, <, \geq, >, =\}$, the predicate $P_1 \otimes_{\bowtie_1 q_1} \ldots \otimes_{\bowtie_n q_n} P_{n+1}$ is valid iff there exist disjoint measurable sets $A_1, \ldots A_{n+1}$ with $\mu(A_1 \cup \ldots \cup A_{n+1}) = 1$ such that for all $k \in \{1, \ldots, n\}$, we have $A_k \models P_k$ and $\mu(A_k) \bowtie_k q_k$, and for $n+1$ we have $A_{n+1} \models P_{n+1}$.

In case of a countable sample space Ω, the powerset $\mathcal{P}(\Omega)$ is a σ-algebra; $\mathit{Distr}(\Omega)$ is the set of probability measures over $\mathcal{P}(\Omega)$; and for all $\mu \in \mathit{Distr}(\Omega)$ $\mathit{Supp}(\mu)$ denotes the set $\{\omega \in \Omega \mid \mu(\omega) > 0\}$.

Probabilistic guarded commands. We model probabilistic systems as programs in a probabilistic guarded-command language. A *probabilistic program* is a tuple $P = (\mathbf{x}, C)$, where $\mathbf{x}$ is a finite set of variables with countable domains and C is a finite set of guarded commands. A *deterministic guarded command* is of the form $g(\mathbf{x}) \mapsto \mathbf{x}' = \mathbf{e}(\mathbf{x})$, and a *probabilistic guarded command* has the form $g(\mathbf{x}) \mapsto \mathbf{x}' = \mathbf{e}_1(\mathbf{x}) \otimes_{=p_1} \ldots \otimes_{=p_k} \mathbf{x}' = \mathbf{e}_{k+1}(\mathbf{x})$, where $p_i \in [0,1]$ for each $1 \leq i \leq k$. The guard g is a predicate over the variables $\mathbf{x}$, and $\mathbf{e}$ and all $\mathbf{e}_i$ are expressions over $\mathbf{x}$. Intuitively, a probabilistic guarded command assigns to $\mathbf{x}$ the values of the expressions $\mathbf{e}_i$ with probability p_i, where $p_{k+1} = 1 - \sum_{j=1}^{k} p_j$.

Example 1. As a running example, we consider the probabilistic model of a robot moving on a discrete plane, starting at an arbitrary position. At each step the robot performs a diagonal jump, and its goal is to visit the origin of the grid (the point with coordinates $(0,0)$) infinitely many times. A random force repels the robot, making the visits hard. Every time the robot performs a step, there is in each dimension a certain probability for it to go backwards a certain number of steps. The probability of going back and the number of steps is a function of the robot's position; this probability is higher when the robot is close to the origin.

The program has variables $l \in \{0,1,2\}, x \in \mathbb{Z}, y \in \mathbb{Z}$ and guarded commands:

$$
\begin{aligned}
c_{NE} &: l = 0 \mapsto x' = x + 1 \wedge y' = y + 1 \wedge l' = 1 \\
c_{SE} &: l = 0 \mapsto x' = x + 1 \wedge y' = y - 1 \wedge l' = 1 \\
c_{NW} &: l = 0 \mapsto x' = x - 1 \wedge y' = y + 1 \wedge l' = 1 \\
c_{SW} &: l = 0 \mapsto x' = x - 1 \wedge y' = y - 1 \wedge l' = 1
\end{aligned}
$$

$$c_x : l = 1 \mapsto (x' = x + 9 \cdot \mathrm{sign}(x) \otimes_{=\frac{1}{|x|+1}} x' = x) \wedge y' = y \wedge l' = 2$$
$$c_y : l = 2 \mapsto (y' = y + 9 \cdot \mathrm{sign}(y) \otimes_{=\frac{1}{|y|+1}} y' = y) \wedge x' = x \wedge l' = 0$$

The first four commands, enabled in location $l = 0$, correspond to the different jump directions of the robot (which controls the non-deterministic choices) can select. Locations $l = 1$ and $l = 2$ model the effect of the random repelling forces along the x and y co-ordinates, respectively. We assume that the force in the x-axis is independent from the one in the y-axis. Despite its simplicity, this problem cannot be solved using probabilistic model checking (the state space is infinite), nor using current deductive proof systems based on fairness (the probability values do matter). The proof system described in this paper, on the other hand, allows us to provide a simple and *modular* correctness argument. □

Semantics of probabilistic programs. The semantics of a probabilistic program $P = (\mathbf{x}, C)$ is a Markov decision process (MDP) $M = (S, \rho)$ [14]. The countable set of states S consists of the valuations of the variables $\mathbf{x}$ and $\rho : S \to \mathcal{P}(\mathit{Distr}(S))$ is the transition relation defined by the guarded commands in C. For a state $s \in S$ we have $\mu \in \rho(s)$ iff either (1) there exists a deterministic guarded command $c : g \mapsto \mathbf{x}' = \mathbf{e}$ in C such that $s \models g$, and for every $s' \in S$ it holds that $\mu(s') = 1$ if $s' = \mathbf{e}(s)$, and $\mu(s') = 0$ otherwise, where $\mathbf{e}(s)$ denotes the value of the expression $\mathbf{e}$ when the variables $\mathbf{x}$ are evaluated according to s, or (2) there exists a probabilistic guarded command $c : g \mapsto \mathbf{x}' = \mathbf{e}_1 \otimes_{=p_1} \ldots \otimes_{=p_k} \mathbf{x}' = \mathbf{e}_{k+1}$ in C such that $s \models g$, and for every $s' \in S$ it holds that $\mu(s') = \sum_{s' = \mathbf{e}_i(s)} p_i$. We assume w. l. o. g. that all programs are deadlock-free, i. e. $\rho(s) \neq \emptyset$. Note that with each state s and each command $c \in C$ with $s \models g_c$, where with g_c we denote the guard of c, the transition relation ρ associates a unique distribution $\mu_{s,c}$.

A *path* in M is a finite or infinite sequence $s_0, s_1, \ldots$ of states in S such that for each i there exists $\mu \in \rho(s_i)$, such that $\mu(s_{i+1}) > 0$. Given a state $s \in S$, we denote with $\mathsf{Paths}(M, s)$ the set of paths in M originating in the state s.

Schedulers. A *scheduler* is a function $\alpha : S^+ \to \mathit{Distr}(C)$ such that $\alpha(\tau \cdot s)(c) > 0$ implies $\mu_{s,c} \in \rho(s)$. We call α *memoryless* if $\alpha(\tau_1 \cdot s) = \alpha(\tau_2 \cdot s)$ for all $\tau_1, \tau_2 \in S^*$ and $s \in S$. A scheduler α is *deterministic* if $|\mathit{Supp}(\alpha(\tau))| = 1$ for all $\tau \in S^+$.

Given a probabilistic program $P = (\mathbf{x}, C)$ with a corresponding MDP $M = (S, \rho)$, a scheduler α defines a discrete time Markov chain (DTMC) $M_\alpha = (S^\alpha, \rho^\alpha)$, where $S^\alpha = S^* \times S$ is the state space and $\rho^\alpha : S^\alpha \to \mathit{Distr}(S^\alpha)$ is the Markov kernel defined as $\rho^\alpha((\tau, s), (\tau', s')) = \sum \rho(s, c, s') \cdot (\alpha(\tau \cdot s)(c))$ if $\tau' = \tau \cdot s$ and $\rho^\alpha((\tau, s), (\tau', s')) = 0$ otherwise. From any initial state $s \in S$ we can define a *unique* probability measure $\mathit{Prob}_{s,\alpha}$ over the set of infinite measurable paths that start at s and obey the probability laws of ρ^α [11].

Example 2. One possible strategy for the robot is to always choose in location $l = 0$ to decrease (when not accounting for the repelling force) the distance to the origin: if $x < 0$ and $y < 0$ then choose c_{NE}, if $x < 0$ and $y \geq 0$ then choose c_{SE}, if $x \geq 0$ and $y < 0$ then choose c_{NW}, and if $x \geq 0$ and $y \geq 0$ then choose c_{SW}. □

2.2 The Logics PCTL and PCTL*

We work with a simple variant of probabilistic computation tree logic (PCTL) in positive normal form [2]. Fix a set AP of *assertions* from an underlying assertion language closed under Boolean operations. The set of PCTL formulas over AP consists of two types of formulas: *state formulas* and *path formulas*.

State formulas are generated by the grammar $\Phi ::= a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \mathbb{P}^{\forall}_{\bowtie p}(\varphi) \mid \mathbb{P}^{\exists}_{\bowtie p}(\varphi)$, where $a \in AP$, Φ_1 and Φ_2 are state formulas, $\bowtie \in \{\leq, <, \geq, >\}$, $p \in \mathbb{R}_{\geq 0}$, and φ is a path formula. Path formulas are generated by the grammar $\varphi ::= \bigcirc \Phi \mid \Phi_1 \,\mathcal{U}\, \Phi_2 \mid \Phi_1 \,\mathcal{R}\, \Phi_2$, where Φ, Φ_1, Φ_2 are state formulas. $\mathcal{U}$ and $\mathcal{R}$ are the until and release operators of linear temporal logic (LTL), respectively. Recall that $\mathcal{R}$ is the dual of $\mathcal{U}$, that is, $\varphi \,\mathcal{R}\, \psi$ is equivalent to $\neg(\neg\varphi \,\mathcal{U}\, \neg\psi)$. As usual, we define the derived operators $\Diamond \varphi = \mathsf{tt} \,\mathcal{U}\, \varphi$ and $\Box \varphi = \neg \Diamond \neg \varphi = \mathsf{ff} \,\mathcal{R}\, \varphi$.

The logic PCTL* generalizes PCTL by allowing ω-regular languages over state formulas as path formulas. Let Φ be a PCTL* state formula. We call Φ a *basic state formula* if it is of the form or $\mathbb{P}^{\mathfrak{O}}_{\bowtie p}(\varphi)$ where $\mathfrak{O} \in \{\exists, \forall\}$ and φ is a PCTL* path formula which contains no probabilistic quantifiers (i.e. φ is an LTL formula). In the case when Φ is a PCTL formula, φ contains exactly one temporal operator, at the top level. We consider a presentation of PCTL* in which LTL formulas are given as *deterministic Streett automata* whose alphabet consists of sets of state formulas.[1] Recall that the set of accepting paths of a Streett automaton is measurable [26, 7].

The *qualitative* versions of PCTL and PCTL* restrict the constants p in $\mathbb{P}^{\mathfrak{O}}_{\bowtie p}(\varphi)$ to the set $\{0, 1\}$.

Semantics. Let $P = (\mathbf{x}, C)$ be a probabilistic program and $M = (S, \rho)$ be the corresponding MDP. Let AP consist of assertions over the variables $\mathbf{x}$.

PCTL* state formulas are interpreted over states of M, while path formulas are interpreted over paths. The satisfaction relations $\models$ are defined as usual for assertions, boolean and temporal operators [2]. Formulas containing the operators $\mathbb{P}^{\forall}$ and $\mathbb{P}^{\exists}$ are interpreted using a probability measure over sets of paths. More specifically, the satisfaction of $\mathbb{P}^{\forall}_{\bowtie p}(\varphi)$ (resp., $\mathbb{P}^{\exists}_{\bowtie p}(\varphi)$) in a state s is determined by the probability measures of the sets of paths $\{\tau \in \mathsf{Paths}(M_\alpha, s) \mid M_\alpha, \tau \models \varphi\}$ where α ranges over all (resp., some) possible schedulers, each of which defines a DTMC in which these sets are measurable. Formally,

$$
\begin{array}{lll}
P, s \models \mathbb{P}^{\forall}_{\bowtie p}(\varphi) & \text{iff} & \mathit{Prob}_{s,\alpha}(\{\tau \in \mathsf{Paths}(M_\alpha, s) \mid M_\alpha, \tau \models \varphi\}) \bowtie p \\
 & & \text{for every scheduler } \alpha \text{ inducing a DTMC } M_\alpha, \\
P, s \models \mathbb{P}^{\exists}_{\bowtie p}(\varphi) & \text{iff} & \mathit{Prob}_{s,\alpha}(\{\tau \in \mathsf{Paths}(M_\alpha, s) \mid M_\alpha, \tau \models \varphi\}) \bowtie p \\
 & & \text{for some scheduler } \alpha \text{ inducing a DTMC } M_\alpha.
\end{array}
$$

For convenience we use $P \models \Phi$ as an abbreviation for $P, s \models \Phi$ for all s. Finally, we note that both PCTL and PCTL* are effectively closed under negation.

[1] Usually, path formulas in PCTL* are defined using linear temporal logic (LTL) [2]. Since the analysis of PCTL* proceeds by first converting LTL to a deterministic automaton, we omit the intermediate step of converting LTL to automata and assume the path formulas are given as deterministic Streett automata.

$$\frac{\text{assertion } \pi \quad P \vdash \Phi[\Psi/\pi] \quad P \vdash \pi \to \Psi}{P \vdash \Phi}\text{BASIC-STATE} \qquad \frac{\theta \text{ is a valid assertion}}{P \vdash \theta}\text{GEN} \qquad \frac{P \vdash \pi \to \mathbb{P}^{\forall}_{=1}(\varphi) \quad P \vdash \pi \to \mathbb{P}^{\mathcal{O}}_{=1}(\varphi \to \psi)}{P \vdash \pi \to \mathbb{P}^{\mathcal{O}}_{=1}(\psi)}\text{MP}$$

$$\frac{P \vdash \pi \to \mathbb{P}^{\forall}_{=1}(\varphi_1) \quad P \vdash \pi \to \mathbb{P}^{\forall}_{=1}(\varphi_2)}{P \vdash \pi \to \mathbb{P}^{\forall}_{=1}(\varphi_1 \wedge \varphi_2)}\text{AND} \qquad \frac{P \vdash \pi \to \mathbb{P}^{\exists}_{>0}(\varphi_1) \quad P \vdash \pi \to \mathbb{P}^{\exists}_{>0}(\varphi_2)}{P \vdash \pi \to \mathbb{P}^{\exists}_{>0}(\varphi_1 \vee \varphi_2)}\text{OR}$$

Fig. 1. Preliminary rules for $\mathcal{O} \in \{\exists, \forall\}$, state formula Φ and path formulas $\varphi_1, \varphi_2, \varphi, \psi$.

3 A Deductive Proof System for PCTL

We now develop a deductive proof system for PCTL. We do this in three steps. First, we introduce some basic rules. Then, we show how to reason about qualitative formulas. Finally, we introduce rules for the full logic. For a probabilistic program P and a PCTL state formula Φ, we write the judgement $P \vdash \Phi$ to state that the proof system derives that program P satisfies Φ from every state.

We assume that we can establish validities in the underlying assertion language (first order logic, or a fragment of it) plus probabilities.

3.1 Preliminary Rules

Figure 1 shows the preliminary rules of our proof systems for PCTL and PCTL*.

The rule BASIC-STATE allows us to reduce the verification of Φ to the verification of formulas of the form $\pi \to \Psi$, where π is an assertion and Ψ is a basic state formula. A basic state formula Ψ occurring one or more times in Φ can be replaced by an assertion π which underapproximates the set of states satisfying the state formula Ψ. The rule's soundness is shown by induction. By successively applying the rule BASIC-STATE, in a bottom up manner, a proof obligation $P \vdash \Phi$ reduces to a set of proof obligations that are of the form $P \vdash \pi \to \Psi$, where Ψ is a basic state formula. We assume this form in subsequent rules.

The other rules lift proof rules of propositional logic to the probabilistic setting. The rule GEN concludes that a valid assertion (a tautology) holds in every state of a program P. The rules AND (resp. OR) formalize the distributivity of conjunction w. r. t. universal almost sure satisfaction (resp. the distributivity of disjunction w. r. t. existential satisfaction with positive probability).

Remark 1. For the rule MP in the existential case we must ensure that the scheduler from the second premise satisfies φ with probability 1. With an existential quantifier in the first premise we cannot guarantee that both schedulers are the same. This problem is also present in other proof rules. For simplicity of presentation, we impose a stronger condition that requires that φ is satisfied regardless of the resolution of the nondeterminism. Alternately, we could have a monolithic proof rule that combines the proof rules for the premises. The price would be more complex proof rules and lack of modularity.

$$\frac{\begin{array}{l}\text{assertion } \theta\\ \text{Lyapunov ranking function } \delta\\ P \vdash \pi \wedge \neg\psi \to \theta\\ P \vdash \theta \wedge \neg\psi \to \varphi\\ P \vdash \theta \wedge \neg\psi \to\\ \quad (\mathfrak{O}\, c \in C : g_c : \theta' \wedge \delta \succ \mathbb{E}(\delta' \mid s))\end{array}}{P \vdash \pi \to \mathbb{P}^{\mathfrak{O}}_{=1}(\varphi\, \mathcal{U}\, \psi)}\,\textsc{until}^{\mathfrak{O}}_{=1} \qquad \frac{\begin{array}{l}\text{assertion } \theta\\ P \vdash \pi \to \theta\\ P \vdash \theta \to \varphi\\ P \vdash \theta \wedge \neg\psi \to (\mathfrak{O}\, c \in C : g_c : \theta')\end{array}}{P \vdash \pi \to \mathbb{P}^{\mathfrak{O}}_{=1}(\psi\, \mathcal{R}\, \varphi)}\,\textsc{inv}^{\mathfrak{O}}_{=1}$$

$$\frac{\begin{array}{l}\text{assertion } \theta\\ \text{ranking function } \delta\\ P \vdash \pi \wedge \neg\psi \to \theta\\ P \vdash \theta \wedge \neg\psi \to \varphi\\ P \vdash \theta \wedge \neg\psi \to\\ \quad (\mathfrak{O}\, c \in C : g_c : (\theta' \wedge \delta \succ \delta') \otimes_{>0} \mathsf{tt})\end{array}}{P \vdash \pi \to \mathbb{P}^{\mathfrak{O}}_{>0}(\varphi\, \mathcal{U}\, \psi)}\,\textsc{until}^{\mathfrak{O}}_{>0} \qquad \frac{\begin{array}{l}\text{assertion } \theta\\ P \vdash \theta \to \varphi\\ P \vdash \pi \to \mathbb{P}^{\mathfrak{O}}_{>0}(\varphi\, \mathcal{U}\, \theta)\\ P \vdash \theta \to \mathbb{P}^{\mathfrak{O}}_{=1}(\psi\, \mathcal{R}\, \theta)\end{array}}{P \vdash \pi \to \mathbb{P}^{\mathfrak{O}}_{>0}(\psi\, \mathcal{R}\, \varphi)}\,\textsc{inv}^{\mathfrak{O}}_{>0}$$

$$\frac{P \vdash \pi \to (\mathfrak{O}\, c \in C : g_c : \varphi')}{P \vdash \pi \to \mathbb{P}^{\mathfrak{O}}_{=1}(\bigcirc \varphi)}\,\textsc{next}^{\mathfrak{O}}_{=1} \qquad \frac{P \vdash \pi \to (\mathfrak{O}\, c \in C : g_c : \varphi' \otimes_{>0} \mathsf{tt})}{P \vdash \pi \to \mathbb{P}^{\mathfrak{O}}_{>0}(\bigcirc \varphi)}\,\textsc{next}^{\mathfrak{O}}_{>0}$$

Fig. 2. Proof rules for qualitative properties, where $\mathfrak{O} \in \{\exists, \forall\}$. The quantification $(\mathfrak{O}\, c \in C : g_c : \chi(\mathbf{x}))$ stands for $\bigwedge_{c \in C}(g_c(\mathbf{x})) \to \chi(\mathbf{x}))$ if $\mathfrak{O} = \forall$ and for $\bigvee_{c \in C}(g_c(\mathbf{x}) \wedge \chi(\mathbf{x}))$ if $\mathfrak{O} = \exists$. The primed versions of assertions and expressions are obtained by replacing primed variables by the values assigned by the respective guarded command.

3.2 Proof Rules for Qualitative PCTL

Figure 2 shows rules for the qualitative fragment. Since we consider basic state formulas, the formulas φ and ψ in these rules are assertions. Using the duality between $\mathbb{P}^{\forall}$ and $\mathbb{P}^{\exists}$, and the closure of PCTL and PCTL* under negation, it is sufficient to restrict attention to the operators $\mathbb{P}^{\forall}_{=1}$, $\mathbb{P}^{\forall}_{>0}$, $\mathbb{P}^{\exists}_{=1}$, and $\mathbb{P}^{\exists}_{>0}$.

The rules use (Lyapunov) ranking functions. For a DTMC (S^α, ρ^α) and a well-founded set $(A, \succ)$, a function $\delta : S^\alpha \to A$ is a *ranking function* if δ decreases on each step, i. e., for each path s, s', we have $\delta(s) \succ \delta(s')$. A function $\delta : S^\alpha \to \mathbb{R}_{\geq 0}$ is a *Lyapunov ranking function* if δ decreases in expectation on each step, i. e., $\delta(s) \succ \mathbb{E}(\delta' \mid s) = \sum_{s' \in S^\alpha} \delta(s')\rho^\alpha(s, s')$ for all states $s \in S^\alpha$. We extend (Lyapunov) ranking functions to MDPs by quantifying over the set of enabled commands.

The rule $\textsc{until}^{\mathfrak{O}}_{=1}$ establishes almost sure liveness properties for states in some set of initial states described by π. The rule is standard: the premises require an assertion θ that defines an inductive invariant and a Lyapunov ranking function that decreases in expectation when taking transitions from θ-states that do not satisfy the target assertion ψ. The rule $\textsc{inv}^{\mathfrak{O}}_{=1}$ establishes almost sure invariance properties. In the case of universal quantification the rule corresponds to the respective rule for CTL, while the existence of a scheduler is equivalent to the existence of a winning strategy in a (non-probabilistic) safety game.

The proof rule $\textsc{until}^{\circlearrowleft}_{>0}$ allows us to establish liveness properties with positive probability. Here, the rule for the existential case corresponds to the one for CTL, while in the universal case the verification question is equivalent to the question about the existence of a strategy in a (non-probabilistic) reachability game. The proof rule $\textsc{inv}^{\circlearrowleft}_{>0}$ establishes invariance properties. The premises of this rule are rather strong: they require reaching with positive probability a set of states in which the temporal property holds almost surely. In Section 5.1 we give a weaker rule, for the (more general) case of satisfaction with probability at least p.

The rules $\textsc{next}^{\circlearrowleft}_{=1}$ and $\textsc{next}^{\circlearrowleft}_{>0}$ handle the next operator in the obvious way.

Consider a proof obligation $P \vdash \pi \to \Psi$, where π is an assertion (which can be tt) and Ψ is a basic state formula. By applying a rule corresponding to the temporal operator in Ψ we can reduce the proof obligation to a set of state validities $P \vdash \theta$ where θ is an assertion. Such proof obligations can be discharged by applying the rule GEN using a solver for the respective logical theory.

The proof system $\mathcal{P}_{\mathsf{qualitative}}$ consists of the proof rules GEN, BASIC-STATE, $\textsc{until}^{\circlearrowleft}_{=1}$, $\textsc{inv}^{\circlearrowleft}_{=1}$, $\textsc{next}^{\circlearrowleft}_{=1}$, $\textsc{until}^{\circlearrowleft}_{>0}$, $\textsc{inv}^{\circlearrowleft}_{>0}$ and $\textsc{next}^{\circlearrowleft}_{>0}$. The soundness of the proof system is proven by relatively standard reasoning. We defer the discussion about (in)completeness to Section 5.2.

Proposition 1. *$\mathcal{P}_{\mathsf{qualitative}}$ is sound: if $P \vdash \varphi$ in $\mathcal{P}_{\mathsf{qualitative}}$, then $P \models \varphi$.*

Example 3. Consider the probabilistic system P from Example 1. We want to prove $P \models \mathsf{tt} \to \mathbb{P}^{\exists}_{=1}(\Diamond \varphi_{close})$, where $\varphi_{close} \equiv |x| + |y| \leq 100$. Take the strategy which at location $l = 0$ selects the only command satisfying $x' = x - \mathrm{sign}(x) \wedge y' = y - \mathrm{sign}(y)$. Using rule $\textsc{until}^{\exists}_{=1}$, we have to find a Lyapunov ranking function δ that decreases in expectation whenever φ_{close} is not satisfied and we execute a command from the chosen strategy. Take the following function

$$\delta(l, x, y) = \begin{cases} x^2 + y^2 & \text{if } l = 0, \\ x^2 + y^2 + 120 & \text{if } l = 1, \\ x^2 + y^2 + 60 & \text{if } l = 2. \end{cases}$$

We analyse the behaviour of $\mathbb{E}(\delta' \mid x, y)$. At $l = 0$ we have $\mathbb{E}(x'^2 + y'^2 \mid x, y) = x^2 + y^2 - 2 \cdot (|x| + |y|) + 2 \leq x^2 + y^2 - 198$. For the unique command at $l = 1$ we have $\mathbb{E}(x'^2 + y'^2 \mid x, y) = x^2 + y^2 + \frac{18|x| + 9^2}{|x| + 1} \leq x^2 + y^2 + 59$. The case $l = 2$ is similar. □

3.3 Full PCTL

Figure 3 introduces proof rules for quantitative probabilities. The rule $\textsc{inv}^{\circlearrowleft}_{\bowtie p}$ for quantitative invariance is defined analogously to the respective rule for satisfaction with positive probability. The rule $\textsc{next}^{\circlearrowleft}_{\bowtie p}$ for the next operator can be defined in the obvious way, thus it is omitted here.

The rule $\textsc{until}^{\circlearrowleft}_{\geq p}$ establishes quantitative liveness properties. Its premises require two auxiliary assertions θ and $\theta_{\geq p}$ such that from each $\theta_{\geq p}$-state the set θ is almost surely reachable, and every time a θ-state is reached a Bernoulli trial

$$\frac{\begin{array}{l}\text{assertions } \theta_{\geq p}, \theta, \qquad \text{real number } \varepsilon > 0 \\ P \vdash \pi \wedge \neg\psi \to \theta_{\geq p} \qquad P \vdash \theta_{\geq p} \to \neg\psi \\ P \vdash \theta_{\geq p} \to \mathbb{P}^{\eth}_{=1}(\varphi \,\mathcal{U}\, \theta) \qquad P \vdash \theta \to \varphi \wedge \theta_{\geq p} \\ P \vdash \theta \to (\eth\, c \in C : g_c : \\ \quad (\exists q : q \geq \varepsilon : \psi' \otimes_{\geq pq} \neg\theta'_{\geq p} \otimes_{=q} \mathsf{tt}))\end{array}}{P \vdash \pi \to \mathbb{P}^{\eth}_{\geq p}(\varphi \,\mathcal{U}\, \psi)}\,\text{UNTIL}^{\eth}_{\geq p} \qquad \frac{\begin{array}{l}P \vdash \pi \to \mathbb{P}^{\eth}_{\bowtie p}(\varphi \,\mathcal{U}\, \theta) \\ P \vdash \theta \to \mathbb{P}^{\eth}_{=1}(\psi \,\mathcal{R}\, \varphi)\end{array}}{P \vdash \pi \to \mathbb{P}^{\eth}_{\bowtie p}(\psi \,\mathcal{R}\, \varphi)}\,\text{INV}^{\eth}_{\bowtie p}$$

$$\frac{\begin{array}{l}\text{assertion } \theta, \quad \text{ranking function } \delta \\ P \vdash \pi \wedge \neg\psi \to \theta \\ P \vdash \pi \wedge \neg\psi \to \delta \leq m \\ P \vdash \theta \wedge \neg\psi \to \varphi \\ P \vdash \theta \wedge \neg\psi \to (\eth\, c \in C : g_c : \\ \quad (\delta' = \delta - 1 \wedge \theta') \otimes_{\geq p} \mathsf{tt})\end{array}}{P \vdash \pi \to \mathbb{P}^{\eth}_{\geq p^m}(\varphi \,\mathcal{U}\, \psi)}\,\text{UNTIL}^{\eth}_{\geq p^m} \qquad \frac{\begin{array}{l}\text{assertions } \theta, \overline{\theta}, \quad \text{r.f. } \delta \\ P \vdash \pi \to \theta \\ P \vdash \pi \to \delta \geq m \\ P \vdash \overline{\theta} \to \mathbb{P}^{\eth}_{=1}(\neg\varphi \,\mathcal{R}\, \neg\psi) \\ P \vdash \theta \wedge \neg\overline{\theta} \wedge \delta > 0 \to \varphi \wedge \neg\psi \\ P \vdash \theta \wedge \neg\overline{\theta} \to (\eth\, c \in C : g_c : \\ \quad (\theta' \wedge \delta \leq \delta') \vee \\ \quad ((\theta' \wedge \delta' = \delta - 1) \otimes_{\leq p} \overline{\theta}'))\end{array}}{P \vdash \pi \to \mathbb{P}^{\eth}_{\leq p^m}(\varphi \,\mathcal{U}\, \psi)}\,\text{UNTIL}^{\eth}_{\leq p^m}$$

$$\frac{\begin{array}{l}\text{assertion } \theta \\ P \vdash \pi \to \mathbb{P}^{\forall}_{\leq p}(\Diamond\, \theta) \\ P \vdash \pi \to \mathbb{P}^{\forall}_{=1}(\varphi \,\mathcal{U}(\theta \vee \psi))\end{array}}{P \vdash \pi \to \mathbb{P}^{\forall}_{\geq 1-p}(\varphi \,\mathcal{U}\, \psi)}\,\text{UNTIL}^{\forall}_{\geq 1-p} \qquad \frac{\begin{array}{l}\text{assertion } \theta \\ P \vdash \pi \to \mathbb{P}^{\eth}_{\geq p_1}(\varphi \,\mathcal{U}\, \theta) \\ P \vdash \theta \to \mathbb{P}^{\eth}_{\geq p_2}(\varphi \,\mathcal{U}\, \psi)\end{array}}{P \vdash \pi \to \mathbb{P}^{\eth}_{\geq p_1 \cdot p_2}(\varphi \,\mathcal{U}\, \psi)}\,\text{UNTIL}^{\eth}_{\geq p_1 \cdot p_2}$$

Fig. 3. Proof rules for $\mathbb{P}^{\eth}_{\bowtie p}$ for $p > 0$ and $\eth \in \{\exists, \forall\}$.

is executed. By adapting the premises to use bound $p + \Delta$ for some $\Delta > 0$, we can easily obtain a rule for strict inequalities.

The proof rules $\text{UNTIL}^{\eth}_{\geq p^m}$ and $\text{UNTIL}^{\eth}_{\leq p^m}$ are slightly more complex. They allow us to prove properties of the form $\mathbb{P}^{\eth}_{\bowtie q}(\varphi \,\mathcal{U}\, \psi)$ provided the bound q has a specific form. The rule $\text{UNTIL}^{\eth}_{\geq p^m}$ requires a ranking function which is initially bounded from above by m and which decreases at each step with probability at least p, thus guaranteeing that the target set of states is reached with probability at least p^m. The rule $\text{UNTIL}^{\eth}_{\leq p^m}$ establishes that an until formula is satisfied with probability at most p^m, by requiring a ranking function that is initially bounded from below by m and is such that in order to reach 0 there should be at least m occurrences of a command that has probability of at least $1 - p$ of going to a set of states from which the formula cannot be satisfied. Rule $\text{UNTIL}^{\forall}_{\geq 1-p}$ combines $\text{UNTIL}^{\eth}_{\leq p^m}$ and $\text{UNTIL}^{\eth}_{=1}$. Rule $\text{UNTIL}^{\eth}_{\geq p_1 \cdot p_2}$ lets us "chain" reachability proofs.

The proof system $\mathcal{P}_{\mathsf{quantitative}}$ consists of the rules in the proof system $\mathcal{P}_{\mathsf{qualitative}}$ together with the rules in Figure 3 and the rule $\text{NEXT}^{\eth}_{\bowtie p}$ (omitted here).

Proposition 2. *$\mathcal{P}_{\mathsf{quantitative}}$ is sound: if $P \vdash \varphi$ in $\mathcal{P}_{\mathsf{quantitative}}$, then $P \models \varphi$.*

Example 4. Consider the probabilistic system from Example 1. Here we show that $P \models \varphi_{close} \to \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$, i. e., we want to find a lower bound

on the probability of reaching the origin from any state in φ_{close}. Using rule $\text{UNTIL}^{\exists}_{\geq p^m}$ it is enough to find a ranking function that is bounded in φ_{close} and such that the probability of decreasing by one has a uniform lower bound in φ_{close}. For brevity, we consider a variant where the decision of the robot and the repelling disturbances occur at once, not sequentially. Then, the ranking function

$$\delta(l,x,y) = \begin{cases} \max(|x|,|y|) & \text{if } x \equiv y \mod 2 \\ \max(|x|,|y|)+5 & \text{if } x \not\equiv y \mod 2. \end{cases}$$

fullfils the requirements. When both coordinates have the same parity and one of them is not 0, it is always possible to decrease δ by selecting a proper command and assuming that the robot is not repelled in any direction. In case that they have different parity we have to consider the case when the robot is repelled in the coordinate with the largest absolute value. The lower bound is then $p = \frac{1}{101^2}$ as we have $|x|, |y| \leq 100$ for the states satisfying φ_{close}. □

4 Proof System for PCTL*

The proof rules presented in Section 3 are applicable to the PCTL fragment of PCTL*. We now extend the proof system $\mathcal{P}_{\mathsf{quantitative}}$ to reason about PCTL*.

The scope of the rules in Figure 1, and in particular BASIC-STATE, is not limited to PCTL. Thus, the rule BASIC-STATE can be applied to a PCTL* formula to arrive at a PCTL* formula $\mathbb{P}^{\circlearrowleft}_{\bowtie p}(\varphi)$, where the formula φ is a Streett automaton representing an ω-regular language over the alphabet of sets of assertions.

Streett Automata, Product Construction. Let AP be a finite set of assertions over $\mathbf{x}$. A *deterministic Streett automaton* is a tuple $\mathcal{A} = (Q, \Sigma, \rho, q_0, \{(E_i, F_i)\}_{i=1}^k)$, where Q is a finite set of states, $\Sigma \subseteq 2^{AP}$ is a finite input alphabet, $\rho \subseteq Q \times \Sigma \times Q$ is a transition relation, such that if $(q, \sigma_1, q_1) \in \rho$ and $(q, \sigma_2, q_2) \in \rho$ and $q_1 \neq q_2$ then $\varphi_{\sigma_1} \wedge \varphi_{\sigma_2}$ is unsatisfiable, where $\varphi_\sigma = (\bigwedge_{\theta \in \sigma} \theta) \wedge (\bigwedge_{\theta \in AP \setminus \sigma} \neg\theta)$ for $\sigma \in \Sigma$, $q_0 \in Q$ is the initial state, and for all $i = 1, \ldots, k$, $E_i \subseteq Q$ and $F_i \subseteq Q$.

A run of $\mathcal{A}$ on an infinite sequence of states (valuations of the variables $\mathbf{x}$) $\tau \in S^\omega$ is a sequence $\eta \in Q^\omega$ of automaton states such that $\eta[0] = q_0$ and for every $i \geq 0$ there exists $\sigma \in \Sigma$ such that $(\eta[i], \sigma, \eta[i+1]) \in \rho$ and $\tau[i] \models \varphi_\sigma$. A run η on τ is accepting if for every $i = 1, \ldots, k$ it holds that if $\mathsf{Inf}(\eta) \cap E_k \neq \emptyset$, then also $\mathsf{Inf}(\eta) \cap F_k \neq \emptyset$, where $\mathsf{Inf}(\eta) \subseteq Q$ is the set of states that occur infinitely often in η. A path τ is accepted by $\mathcal{A}$ iff there exists an accepting run of $\mathcal{A}$ on τ. We write $L(\mathcal{A})$ for the set of paths accepted by $\mathcal{A}$.

Consider a probabilistic program $P = (\mathbf{x}, C)$ and a deterministic Street automaton $\mathcal{A}$ with alphabet Σ which consists of sets of assertions over $\mathbf{x}$.

The *product of P and $\mathcal{A}$* is the probabilistic program $P_\mathcal{A} = (\mathbf{x}^\mathcal{A}, C^\mathcal{A})$, where $\mathbf{x}^\mathcal{A} = \mathbf{x} \dot\cup \{x_q\}$, for a fresh variable x_q with domain Q, and $C^\mathcal{A}$ is the set of guarded commands defined as follows. The set $C^\mathcal{A}$ contains one guarded command for each pair of transition $(q, \sigma, q') \in \rho$ and probabilistic guarded command $c : g \mapsto \mathbf{x}' = \mathbf{e}_1 \otimes_{=p_1} \ldots \otimes_{=p_k} \mathbf{x}' = \mathbf{e}_{k+1}$ in C, where (c, q, σ, q') is the label of

$$\frac{\begin{array}{l}\text{assertions } \theta, \overline{\theta} \\ \text{constant } p > 0 \\ P \vdash \pi \to \overline{\theta} \\ P \vdash \overline{\theta} \to \mathbb{P}^{\forall}_{=1}(\Diamond\, \theta) \\ P \vdash \theta \to \mathbb{P}^{\forall}_{=1}(\Box \theta) \\ P \vdash \theta \wedge \varphi \to \mathbb{P}^{\forall}_{\geq p}(\Diamond\, \psi)\end{array}}{P \vdash \pi \to \mathbb{P}^{\forall}_{=1}(\Box \Diamond \varphi \to \Box \Diamond \psi)}\,\text{REC}^{\forall}_{=1} \qquad \frac{\begin{array}{l}\text{assertions } \theta, \overline{\theta} \\ \text{constant } p > 0 \\ P \vdash \pi \to \overline{\theta} \\ P \vdash \overline{\theta} \to \mathbb{P}^{\exists}_{=1}(\Diamond\, \theta) \\ P \vdash \theta \to \mathbb{P}^{\forall}_{=1}(\Box \theta) \\ \text{for all } i = 1, \ldots, m: \\ P \vdash \theta \wedge \varphi^i \to \mathbb{P}^{\exists}_{\geq p}\left(\Diamond\, \psi^i\right)\end{array}}{P \vdash \pi \to \mathbb{P}^{\exists}_{=1}\left(\bigwedge_{i=1}^{m}(\Box \Diamond \varphi^i \to \Box \Diamond \psi^i)\right)}\,\text{REC}^{\exists}_{=1}$$

Fig. 5. Proof rules for almost sure repeated reachability, where $\mathfrak{H} \in \{\exists, \forall\}$.

the product guarded command, and the assertion $\varphi_\sigma(\mathbf{x})$ represents the letter σ: $(c, q, \sigma, q') : g_c \wedge x_q = q \wedge \varphi_\sigma(\mathbf{x}) \mapsto x'_q = q' \wedge (\mathbf{x}' = \mathbf{e}_1 \otimes_{=p_1} \ldots \otimes_{=p_k} \mathbf{x}' = \mathbf{e}_{k+1})$. Similarly, for deterministic guarded commands. For a given scheduler α and an initial state s, the set of paths of $P_{\mathcal{A}}$ on which $\mathcal{A}$ has an accepting run, denoted $Acc(P, \mathcal{A})_{\alpha,s}$, is measurable [26, 7].

Basic Path Rule. Given a Streett automaton $\mathcal{A}$, the rule shown in Figure 4 reduces the proof obligation $P \vdash \pi \to \mathbb{P}^{\mathfrak{H}}_{\bowtie p}(L(\mathcal{A}))$ to proving a statement of the form $P \vdash \pi' \to \mathbb{P}^{\mathfrak{H}}_{\bowtie p}(Acc(P, \mathcal{A}))$, where π' is an assertion.

Proposition 3. *If the premises of the proof rule* BASIC-PATH *are satisfied then it holds that* $P \models \pi \to \mathbb{P}^{\mathfrak{H}}_{\bowtie p}(L(\mathcal{A}))$.

Rules for Repeated Reachability. The Streett acceptance condition of $\mathcal{A}$ can be encoded as repeated reachability formulas of the form $\bigwedge_{i=1}^{k}\left(\Box \Diamond \varphi_i \to \Box \Diamond \psi_i\right)$, where φ_i and ψ_i are assertions over $\mathbf{x}^{\mathcal{A}}$ encoding the sets E_i and F_i for $i = 1, \ldots, k$. Figure 5 shows the corresponding rules for the almost sure case.

$$\frac{P_{\mathcal{A}} \vdash (\pi \wedge x_q = q_0) \to \mathbb{P}^{\mathfrak{H}}_{\bowtie p}(Acc(P, \mathcal{A}))}{P \vdash \pi \to \mathbb{P}^{\mathfrak{H}}_{\bowtie p}(L(\mathcal{A}))}$$

Fig. 4. Rule BASIC-PATH

Proposition 4 (Soundness of $\text{REC}^{\forall}_{=1}$). *Rules* $\text{REC}^{\forall}_{=1}$ *and* $\text{REC}^{\exists}_{=1}$ *are sound.*

Proof (Sketch). We prove soundness of $\text{REC}^{\forall}_{=1}$. Fix an arbitrary scheduler α and consider M_α. We can restrict the proof to the infinite paths that start in a θ-state since any infinite path of P eventually visits only states in θ. Let $S_0, S_1, \ldots$ be the random process such that S_k is the state visited after executing exactly k instructions, and $\mathcal{F}_k$ be the smallest σ-algebra that makes S_k measurable. Let $\Diamond^{\geq n}\, \psi$ denote the event $\{\exists m \geq n : S_m \in \psi\}$ and $[\mathcal{E}]$ denote the indicator function for the event $\mathcal{E}$. Notice that $\lim_n \Diamond^{\geq n}\, \psi = \Box \Diamond \psi$.

$$\begin{aligned}[\Diamond^{\geq m}\, \psi] &= \lim_n \mathbb{P}(\Diamond^{\geq m}\, \psi \mid \mathcal{F}_n) \geq \limsup_n \mathbb{P}(\Diamond^{\geq n}\, \psi \mid \mathcal{F}_n) \\ &\geq \liminf_n \mathbb{P}(\Diamond^{\geq n}\, \psi \mid \mathcal{F}_n) \geq \lim_n \mathbb{P}(\Box \Diamond \psi \mid \mathcal{F}_n) = [\Box \Diamond \psi]\end{aligned}$$

The equalities are a consequence of Levy's 0-1 law [11, Theorem 5.5.8] and the fact that $\Diamond^{\geq m} \psi$ and $\Box\Diamond\psi$ are measurable in $\sigma(\bigcup_n \mathcal{F}_n)$. If we let m go to infinity both extremes coincide and therefore $\lim_n \mathbb{P}(\Diamond^{\geq n} \psi \mid \mathcal{F}_n) = [\Box\Diamond\psi]$.

From the last premise of the rule we have $\mathbb{P}(\Diamond^{\geq n} \psi \mid \mathcal{F}_n) \geq p[S_n \in \varphi]$, i. e. the probability of reaching a ψ-state from a φ-state is at least p. Take ω an arbitrary point event that satisfies $\Box\Diamond\varphi$, then for infinitely many n we have $\mathbb{P}(\Diamond^{\geq n} \psi \mid \mathcal{F}_n)(\omega) \geq p > 0$, and therefore $[\Box\Diamond\psi](\omega) = 1$. We thus conclude that $P \models \pi \to \mathbb{P}^{\forall}_{=1}(\Box\Diamond\varphi \to \Box\Diamond\psi)$.

The soundness of the existential rule is proved in a similar way; additionally, one has to show how a witness scheduler can be constructed from the individual schedulers that guarantee reachability of each ψ^i for $i = 1, \ldots, m$. □

In the special case $\varphi := \mathsf{tt}$ in rule $\text{REC}^{\forall}_{=1}$, we obtain a proof rule for unconditional recurrence as the rule given by Hart and Sharir [16, Lemma 3.3].

The rule for $\text{REC}^{\exists}_{=1}$ in Figure 5 requires that the assertion θ is invariant under all possible schedulers instead of under some scheduler. The reason is the following: the fact that there exist a scheduler that ensures the invariance and schedulers that ensure reachability does not imply that these schedulers can be combined in a scheduler that achieves both properties. Instead of referring to the rule for proving $\mathbb{P}^{\exists}_{\geq p}\left(\Diamond\psi^i\right)$ we can alternatively include the respective premisses and incorporate the requirement that the scheduler should guarantee that θ is invariant. We omit this more complicated rule for simplicity of the presentation.

We can give a proof rule $\text{REC}^{\circlearrowleft}_{>0}$ for repeated reachability with positive probability that is analogous to the rule $\text{INV}^{\circlearrowleft}_{>0}$: Its premises require that some set of states θ is reached with positive probability and in every state in that set the repeated reachability property is satisfied almost surely. Analogously, we can obtain a rule $\text{REC}^{\exists}_{\geq p}$ for the existential quantitative repeated reachability. The rule $\text{REC}^{\forall}_{\geq p}$ for the universal quantitative case is a straightforward adaptation of $\text{REC}^{\forall}_{=1}$: It requires that some set of states θ is invariant with probability at least p and from every state in θ that satisfies φ a ψ-state is reached with probability at least q for some $q > 0$. Strict inequalities are handled as in the PCTL case.

Example 5. We want to prove that there is a strategy for the robot in Example 1 that visits infinitely often the origin regardless of the initial state. This can be specified in PCTL* as $P \models \mathsf{tt} \to \mathbb{P}^{\exists}_{=1}(\Box\Diamond(x = 0 \wedge y = 0))$. From Example 3 we have $P \vdash \mathsf{tt} \to \mathbb{P}^{\exists}_{=1}(\Diamond\varphi_{close})$, and from Example 4 we have $P \vdash \varphi_{close} \to \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$. Then, we can conclude that $P \vdash \mathsf{tt} \to \mathbb{P}^{\exists}_{\geq p}(\Diamond(x = 0 \wedge y = 0))$. The desired property follows immediately from the rule $\text{REC}^{\exists}_{=1}$ as $P \vdash \mathsf{tt} \to \mathbb{P}^{\exists}_{=1}(\Box\Diamond\mathsf{tt})$ is a tautology. □

Unlike the deductive proof systems for CTL* [19] and ATL* [25] here we cannot encode the accepting condition of the automaton $\mathcal{A}$ as a fairness requirement in the product system. In [19] LTL formulas are translated to temporal testers with fairness conditions, and their synchronous product with the original system yields a fair discrete system. Justice (a specific form of fairness) is then handled by specialized proof rules. Similarly, in [25] an LTL formula

is transformed to a deterministic automaton, whose synchronous composition with the system yields an alternating discrete system with fairness conditions and the resulting proof condition then contains strategy quantifiers ranging over fair strategies. Subsequently, fair strategy quantifiers are transformed into unfair ones and the fairness conditions are made explicit in the resulting temporal formula, which is of a specific form and is treated by special proof rules.

The example below demonstrates that in the probabilistic case the encoding of the winning condition of the automaton as a fairness constraint is not equivalent to an explicit encoding in the temporal formula.

Example 6. Consider the probabilistic program P over variables $s \in \{0,1,2\}$ and $x, y \in \mathbb{B}$. The transition relation is described by the guarded commands:

$$c_0 : s = 0 \mapsto (s' = 1 \wedge x' = 0 \wedge y' = 0) \otimes_{=\frac{1}{2}} (s' = 2 \wedge x' = 1 \wedge y' = 1),$$
$$c_1 : s = 1 \mapsto s' = 1 \wedge x' = 0 \wedge y' = 0,$$
$$c_2 : s = 2 \mapsto s' = 2 \wedge x' = 1 \wedge y' = 1.$$

Initially we have $\iota \equiv s = 0 \wedge x = 0 \wedge y = 0$. A scheduler α is fair w.r.t. the fairness requirement $\varphi \equiv \Box\Diamond(x = 1)$ if in the resulting DTMC starting from any ι-state, $\Box\Diamond(x = 1)$ holds with probability 1. Thus, the set of schedulers that are fair w.r.t. φ is empty and hence if quantifiers are interpreted over the set of all fair schedulers we have that $P \models \iota \to \mathbb{P}^{\exists}_{\geq 1/2}(\Box\Diamond(y = 1))$ does not hold and $P \models \iota \to \mathbb{P}^{\forall}_{\geq 1/2}(\Box\Diamond(y = 1))$ holds trivially. On the other hand, when quantifiers range over all possible schedulers, we have that $P \models \iota \to \mathbb{P}^{\exists}_{\geq 1/2}(\Box\Diamond(x = 1) \to \Box\Diamond(y = 1))$ and $P \models \iota \to \mathbb{P}^{\forall}_{\geq 1/2}(\Box\Diamond(x = 1) \to \Box\Diamond(y = 1))$ are satisfied. □

The proof system $\mathcal{P}^*_{\mathsf{quantitative}}$ consists of the rules in $\mathcal{P}_{\mathsf{quantitative}}$ together with the rules MP, AND, OR, the rule BASIC-PATH and the rules for repeated reachability $\text{REC}^{\forall}_{=1}, \text{REC}^{\exists}_{=1}, \text{REC}^{\circlearrowleft}_{>0}, \text{REC}^{\forall}_{\bowtie p}$ and $\text{REC}^{\exists}_{\bowtie p}$.

Proposition 5. *$\mathcal{P}^*_{\mathsf{quantitative}}$ is sound: if $P \vdash \varphi$ in $\mathcal{P}^*_{\mathsf{quantitative}}$, then $P \models \varphi$.*

5 Discussion

We have presented the first deductive proof system for PCTL*. Our initial experience with the proof system has been positive: for example, we can prove the termination of Rabin's choice coordination problem with probability at least $1 - 2^{-\frac{M}{2}}$, for a parameter M denoting the size of the alphabet used in the protocol, for *any* number of processes. Like with any deductive proof system, one has to come up with invariants and Lyapunov ranking functions. While we currently do this manually, it will be interesting to combine our proof system with recent automated techniques [18]. We conclude with two technical discussions: relaxations of our proof rules and completeness.

5.1 Variants of the Deduction Rules

Our choice of deduction rules has been driven by the intention to keep the exposition simple. We now discuss some possible relaxations to our rules, motivated by the incompleteness of some of the original rules.

Invariant with positive probability. As a first example, consider the rule for $\textsc{inv}^{\forall}_{\geq p}$, which checks if a set of states that each satisfy the invariant with probability one can be reached with probability at least p.

Consider the probabilistic program P with a single variable x over $\mathbb{N}$ that describes a biased random walk. The initial state is $x = 1$ and the state $x = 0$ is absorbing. At each step x increases by 1 with probability $3/4$ and decreases by 1 with probability $1/4$. We have that $P \models (x = 1) \rightarrow \mathbb{P}^{\forall}_{\geq \frac{2}{3}}(\Box(x > 0))$ holds. However, from every state of P the state $x = 0$ is reached with positive probability. Thus, we cannot provide an assertion θ as required by the premisses of rule $\textsc{inv}^{\forall}_{\geq p}$, as no subset of the set of states where $x > 0$ holds is invariant.

The rule $\overline{\textsc{inv}}^{\forall}_{\geq p}$ in Figure 6 is a generalisation of $\textsc{inv}^{\forall}_{\geq p}$. The idea is to provide assertions, $\theta_1, \theta_2, \ldots$ such that from each θ_i-state there is high enough probability to eventually move to some θ_j where $j > i$, meaning that the infinite product of these probabilities converges to the desired probability p for the invariant.

We can apply the rule $\overline{\textsc{inv}}^{\forall}_{\geq p}$ in Figure 6 to this random walk example as follows. Let $\theta_k = (x = k)$ for each $k > 0$. Then, clearly, $P \vdash (x = 1) \rightarrow \bigvee_{k=1}^{\infty} \theta_k$ and for all $k > 0$ we have $P \vdash \theta_k \rightarrow \varphi$. The probability of reaching θ_{k+1} from a state in θ_k is $p_k = \frac{1-3^{-k}}{1-3^{-(k+1)}}$ and thus $P \vdash \theta_k \rightarrow \mathbb{P}^{\forall}_{\geq p_k}(\Diamond \bigvee_{j=k+1}^{\infty} \theta_j)$ for all $k > 0$. Finally, $\prod_{k=1}^{\infty} p_k = 2/3$ which completes the proof. Clearly, the expressivity comes at a price of more complex premises.

Repeated Reachability. As a second example, consider rule $\overline{\textsc{rec}}^{\forall}_{=1}$ in Figure 6, which takes a different approach from the one in Figure 5. Instead of ensuring that after visiting a state satisfying φ we reach with probability at least p a state satisfying ψ, we ensure that it is almost impossible to visit an infinite number of φ-states without visiting a single ψ-state. The latter is a more relaxed condition. Take any program that satisfies the former and add a self loop in a state satisfying $\neg\varphi \wedge \neg\psi$ that is reachable from a φ-state. The modified program does not satisfy the premise of the original rule, although the property still holds. The modified rule does not suffer from this.

More specifically, rule $\overline{\textsc{rec}}^{\forall}_{=1}$ in Figure 6 requires the existence of a Lyapunov ranking function that decreases in expectation in states where φ holds but ψ does not hold, and cannot increase in expectation in states that do not satisfy ψ. Thus, the rule can be successfully applied also in cases where a φ state is visited only finitely many times. Its completeness is discussed in Section 5.2.

5.2 Completeness for Finite State Systems

Our proof rules are in general incomplete for infinite-state probabilistic programs. For example, the rule $\textsc{until}^{\circlearrowleft}_{=1}$ relies on Lyapunov ranking functions that are known to be incomplete for almost sure termination [5, 13]. We focus the

$$\frac{\begin{array}{l}\text{assertions } \theta_1, \theta_2, \ldots \\ \text{constants } p_1, p_2, \ldots \\ \prod_{k=1}^{\infty} p_k \geq p \\ P \vdash \pi \to \bigvee_{k=1}^{\infty} \theta_k \\ \text{for all } k > 0 : \\ P \vdash \theta_k \to \varphi \\ P \vdash \theta_k \to \mathbb{P}^{\forall}_{\geq p_k}(\Diamond \bigvee_{j=k+1}^{\infty} \theta_j)\end{array}}{P \vdash \pi \to \mathbb{P}^{\forall}_{\geq p}(\Box \varphi)}\,\overline{\text{INV}}^{\forall}_{\geq p} \qquad \frac{\begin{array}{l}\text{assertion } \theta, \text{ Lyapunov r. f. } \delta \\ P \vdash \pi \to \theta \\ P \vdash \theta \to \mathbb{P}^{\forall}_{=1}(\Box \theta) \\ P \vdash \theta \wedge \neg\psi \wedge \varphi \to \\ \qquad \big(\forall c \in C : g_c : \theta' \wedge \delta \succ \mathbb{E}(\delta' \mid s)\big) \\ P \vdash \theta \wedge \neg\psi \quad \to \\ \qquad \big(\forall c \in C : g_c : \theta' \wedge \delta \geq \mathbb{E}(\delta' \mid s)\big)\end{array}}{P \vdash \pi \to \mathbb{P}^{\forall}_{=1}\left(\Box\Diamond \varphi \to \Box\Diamond \psi\right)}\,\overline{\text{REC}}^{\forall}_{=1}$$

Fig. 6. More advanced proof rules.

discussion to programs with *finite* state spaces, as most of our rules —or slight variations thereof— are complete for this class. The completeness of the rule $\text{INV}^{\circlearrowleft}_{=1}$ and the rules for positive probability follows from the non-probabilistic case [19] (even for countable state spaces).

Until. If a program P satisfies almost surely $\varphi \,\mathcal{U}\, \psi$ regardless of the scheduler, then given an initial state s the expected amount of steps before reaching a ψ-state is bounded. Moreover, there is an optimal memoryless scheduler that maximizes this quantity for all states [3]. Then, the mapping that assigns to each state the expected time of reaching a ψ-state using the optimal scheduler is a valid Lyapunov ranking function. For the completeness of $\text{UNTIL}^{\exists}_{=1}$ we have that there is a memoryless and deterministic scheduler that satisfies $\varphi \,\mathcal{U}\, \psi$ [4]. Then we have to take θ as the set of states visited by the scheduler, and build a Lyapunov ranking function for this sub-MDP in a similar way as above.

Streett Condition. The rule $\text{REC}^{\exists}_{=1}$ is not complete as the premise $P \vdash \theta \to \mathbb{P}^{\forall}_{=1}(\Box \theta)$ is too strong. The monolithic proof rule (see Remark 1) that guarantees that θ is invariant w. r. t. the schedulers of the last premise is complete. We have to choose θ as the states that the scheduler visits infinitely often with non-zero probability. The set θ is almost surely reached and each of its states belongs to at least one end component [8]. If a φ^i-state is visited infinitely often, then the end component that the scheduler reaches must have a ψ^i-state, otherwise the property will be violated. Then, the last premise is satisfied.

The rule $\overline{\text{REC}}^{\forall}_{=1}$ presented in Section 5.1 is complete. We need to analyze the maximal end components of the program. Consider the sub-MDP obtained from an end component E. From every state the maximum expected number of φ-states visited before reaching a ψ-state is finite, since the maximal probability of returning to a φ-state without visiting a ψ state is less than one. This quantity can be used to build a Lyapunov function that decreases every time that a $\varphi \wedge \neg\psi$-state is visited. Consider now the quotient MDP that is obtained by lumping every maximal end component into a single state and removing self-loops. It has no end component except for deadlock states. Then we can build a Lyapunov ranking function that ensures that a deadlock state is reached almost surely. We can combine all these local Lyapunov functions to build a global one that satisfies the conditions of the rule $\overline{\text{REC}}^{\forall}_{=1}$.

Acknowledgements This work is supported by the EU FP7 projects 295261 (MEALS) and 318490 (SENSATION), by the DFG Transregional Collaborative Research Centre SFB/TR 14 AVACS, and by the CDZ project 1023 (CAP).

References

1. Tamarah Arons, Amir Pnueli, and Lenore D. Zuck. Parameterized verification by probabilistic abstraction. In *FOSSACS*, pages 87–102, 2003.
2. Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.
3. Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.
4. Andrea Bianco and Luca de Alfaro. Model checking of probabalistic and nondeterministic systems. In *FSTTCS*, pages 499–513, 1995.
5. Olivier Bournez and Florent Garnier. Proving positive almost-sure termination. In *RTA*, pages 323–337, 2005.
6. Aleksandar Chakarov and Sriram Sankaranarayanan. Probabilistic program analysis with martingales. In *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, pages 511–526, 2013.
7. Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.
8. Luca de Alfaro. *Formal verification of probabilistic systems*. PhD thesis, Standford, 1997.
9. Luca de Alfaro, Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Roberto Segala. Symbolic model checking of probabilistic processes using mtbdds and the kronecker representation. In *TACAS 2000*, volume 1785 of *Lecture Notes in Computer Science*, pages 395–410. Springer, 2000.
10. Rayna Dimitrova, Luis María Ferrer Fioriti, Holger Hermanns, and Rupak Majumdar. PCTL*: The deductive way (extended version). Reports of SFB/TR 14 AVACS 114, 2016. Available at http://www.avacs.org.
11. Rick Durrett. *Probability: Theory and Examples*. Series in Statistical and Probabilistic Mathematics. Cambridge University Press, fourth edition, 2010.
12. Javier Esparza, Andreas Gaiser, and Stefan Kiefer. Proving termination of probabilistic programs using patterns. In P. Madhusudan and Sanjit A. Seshia, editors, *Computer Aided Verification - 24th International Conference, CAV*, volume 7358 of *LNCS*, pages 123–138. Springer, 2012.
13. Luis María Ferrer Fioriti and Holger Hermanns. Probabilistic termination: Soundness, completeness, and compositionality. In *POPL*, pages 489–501, 2015.
14. Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer, 1997.
15. Nissim Francez. *Fairness*. Texts and Monographs in Computer Science. Springer, 1986.
16. Sergiu Hart, Micha Sharir, and Amir Pnueli. Termination of probabilistic concurrent program. *ACM Trans. Program. Lang. Syst.*, 5(3):356–380, 1983.
17. Joe Hurd. *Formal verification of probabilistic algorithms*. PhD thesis, University of Cambridge, 2001.

18. Joost-Pieter Katoen, Annabelle McIver, Larissa Meinicke, and Carroll C. Morgan. Linear-invariant generation for probabilistic programs: - automated support for proof-based methods. In *Static Analysis (SAS 2010)*, volume 6337 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2010.
19. Yonit Kesten and Amir Pnueli. A compositional approach to CTL* verification. *Theor. Comput. Sci.*, 331(2-3):397–428, 2005.
20. Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM: probabilistic model checking for performance and reliability analysis. *SIGMETRICS Performance Evaluation Review*, 36(4):40–45, 2009.
21. Annabelle McIver and Carroll Morgan. *Abstraction, Refinement and Proof for Probabilistic Systems*. Monographs in Computer Science. Springer, 2005.
22. Amir Pnueli. On the extremely fair treatment of probabilistic algorithms. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pages 278–290, 1983.
23. Amir Pnueli and Lenore D. Zuck. Probabilistic verification. *Inf. Comput.*, 103(1):1–29, 1993.
24. Michael O. Rabin. The choice coordination problem. *Acta Informatica*, 17:121–134, 1982.
25. Matteo Slanina, Henny B. Sipma, and Zohar Manna. Deductive verification of alternating systems. *Form. Asp. Comput.*, 20(4-5):507–560, 2008.
26. Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, pages 327–338, 1985.