CONTROLLER SYNTHESIS FOR PROBABILISTIC SYSTEMS (EXTENDED ABSTRACT)

Christel Baier[1*], Marcus Größer[1**], Martin Leucker[2***],
Benedikt Bollig[3], Frank Ciesinski[1*]

[1] *Institut für Informatik I, University of Bonn,* [baier|groesser|ciesinsk]@cs.uni-bonn.de
[2] *IT Department, Uppsala University,* leucker@it.uu.se
[3] *Lehrstuhl für Informatik II, RWTH Aachen,* bollig@cs.rwth-aachen.de

[*] Supported by the DFG-NWO-Project "VOSS".

[**] Supported by the DFG-Project "VERIAM" and the DFG-NWO-Project "VOSS".

[***] Supported by the European Research Training Network "Games".

Abstract Controller synthesis addresses the question of how to limit the internal behavior of a given implementation to meet its specification, regardless of the behavior enforced by the environment. In this paper, we consider a model with probabilism and nondeterminism where the nondeterministic choices in some states are assumed to be controllable, while the others are under the control of an unpredictable environment. We first consider probabilistic computation tree logic as specification formalism, discuss the role of strategy-types for the controller and show the NP-hardness of the controller synthesis problem. The second part of the paper presents a controller synthesis algorithm for automata-specifications which relies on a reduction to the synthesis problem for PCTL with fairness.

1. Introduction

In system design, the general goal is to develop systems that satisfy user requirement specifications. To simplify this development process, it should be automated as far as possible. One goal is to *synthesize* a system based on the requirements. Another, practically important task is to synthesize only a *controller* that limits or controls the behavior of an existing system, usually called *plant*, to meet the given specification.

In such a framework, the plant acts usually in an *environment*. The goal is to find a *schedule* for the controllable events that guarantees the specification to be satisfied considering all possible environmental behaviors. One can also understand the controller and environment as two *players*. The plant constitutes to the game board and controller synthesis becomes the problem of finding a *strategy* for the controller that satisfies the specification whatever move the environment does, or in other words, under any *adversary*.

The requirement specification can either be given *internally* or *externally*. Internal specifications impose restrictions for example on the number of visits of a state of the plant. Examples for external specifications are temporal logic formulas that are supposed to be satisfied by the controlled plant.

The controller synthesis problem has attracted a lot of attention in recent years. For discrete systems, the problem is meanwhile well understood [Thomas, 2003]. Recently, the problem was studied for timed systems [Bouyer et al., 2003; de Alfaro et al., 2003]. Here, the plant is modeled as a timed transition system and requirement specifications are given in timed temporal logic or as ω-regular winning conditions on the system.

We study the problem in a probabilistic setting. Our underlying model for the plant are Markov Decision Processes (MDPs), in which we, however, distinguish states that are under control of the plant from those that are under the control of the environment. This model is also known as *turn-based stochastic* $2\frac{1}{2}$*-player games* [Condon, 1992; Condon, 1993; Filar and Vrieze, 1997; de Alfaro et al., 1998; de Alfaro and Henzinger, 2000; Chatterjee et al., 2003], and it is a popular model in planning, AI, and control problems. Several solutions have been suggested for ω-regular winning objectives (e.g. reachability, Büchi and coBüchi, Rabin chain, parity condition) with *qualitative* winning criteria (sure, almost sure, limit sure) in the turn-based and concurrent case [Condon, 1993; de Alfaro et al., 1998; de Alfaro and Henzinger, 2000; Jurdzinski et al., 2003; Chatterjee et al., 2003]. We are interested here in *quantitative winning criteria* stating that the probability to win the game meets a given lower (or upper) probability bound as studied in [de Alfaro and Majumdar, 2001] for concurrent games and in the recent paper [Chatterjee et al., 2004] for $1\frac{1}{2}$- and $2\frac{1}{2}$-player stochastic games.

Translating the players to *system* and *environment*, one can construct a lot of examples of similar spirit, for example in domain of security analysis. The environment acts as an intruder and random moves are used to model different nuances [Mitchell, 2001].

In our setting, we study the problem to find a strategy for the plant such that a given external specification formalized as a *probabilistic temporal logic formula* is fulfilled, no matter how the opponent (environment) behaves. In the first part of the paper, we consider the synthesis problem where the specification is provided by means of a formula of probabilistic computation tree logic PCTL [Hansson and Jonsson, 1994; Bianco and De Alfaro, 1995]. As for strategies, we discuss several choices: The system or the opponent has to choose deterministically (D) or can choose randomly (R). Furthermore, he or she might choose according to the current state (M), also called stationary or Markovian, or, is allowed to look at the history of the game played so far (H). From a practical point of view, it would be desirable to be able to synthesize controllers that do not require extra memory to keep track of a history and

do not depend on random number generators. However, we show that this is not always possible. For security analysis, this implies that adversaries that act according to the information obtained so far are stronger than those not using this information. For the synthesis algorithms, it means that any of the strategy-classes HD, HR, MD and MR requires its own synthesis algorithm. We then show the NP-completeness of the synthesis problem for PCTL and MD-strategies and the NP-hardness of the synthesis problem for PCTL and the strategy-classes HD, HR and MR. Moreover, we show that these results already hold in the setting of $1\frac{1}{2}$-player games (where all states are assumed to be controllable) and for the sublogics $\text{PCTL}_{\setminus\bigcirc}$ and $\text{PCTL}_{\setminus\mathcal{U}}$ that do not use the next step and until operator, respectively. This result stands in contrast to the PCTL model checking problem which is solvable in polynomial-time and for which the strategy-class is irrelevant [Bianco and De Alfaro, 1995].
The second part of the paper addresses the synthesis problem for linear time specifications formalized by LTL-formulas. We show that an optimal HD-strategy for $\mathcal{M}$ and LTL-formula φ can be derived from an optimal MD-strategy for the product-MDP $\mathcal{M} \times \mathcal{A}$, built from the original MDP $\mathcal{M}$ and a deterministic Rabin automaton $\mathcal{A}$ for φ, that maximizes the probability to reach a so-called *winning component* under certain *fairness* assumptions for the adversary. We thus obtain a triple-exponential solution for the HD-controller synthesis problem for MDPs and LTL-specifications that relies on a reduction to the HD-controller synthesis problem for MDPs and Rabin automaton specifications. The latter is solvable via a reduction to the MD-synthesis problem for PCTL with fairness [Baier and Kwiatkowska, 1998]. The recent paper [Chatterjee et al., 2004] establishes the same complexity result for quantitative stochastic $2\frac{1}{2}$-player parity games. In fact, the latter is equivalent to the HD-controller synthesis problem for MDPs and Rabin automaton specifications because both the parity and Rabin-chain condition have the expressiveness of ω-regular winning conditions [Thomas, 1990; Emerson and Jutla, 1991; de Alfaro and Henzinger, 2000]. Thus, our algorithm, which relies on applying a model checker for PCTL with fairness to the MDPs induced by the MD-strategies, can be seen as an alternative to the algorithm suggested in [Chatterjee et al., 2004], which applies an algorithm to solve the quantitative $1\frac{1}{2}$-player parity game to the MDPs induced by the MD-strategies. For a full version of this paper see *http://web.informatik.uni-bonn.de/I/baier/publikationen.html*.

2. Preliminaries

A *distribution* on a countable set X denotes a function $\mu : X \to [0,1]$ with $\sum_{x \in X} \mu(x) = 1$. $\mathsf{Distr}(X)$ denotes the set of all distributions on X. A MDP is a tuple $\mathcal{M} = (S, \mathsf{Act}, \mathsf{P}, s_{\mathit{init}}, \mathsf{AP}, L)$ where S is a countable set of *states*, Act a finite set of actions, $\mathsf{P} : S \times \mathsf{Act} \times S \to [0,1]$ is a three-dimensional

transition probability matrix such that $\sum_{t \in S} \mathsf{P}(s,\alpha,t) \in \{0,1\}$ for all states $s \in S$ and actions $\alpha \in \mathsf{Act}$, and $s_{init} \in S$ is the initial state. AP denotes a finite set of atomic propositions, and $L : S \to 2^{\mathsf{AP}}$ a labelling function which assigns to each state $s \in S$ the set $L(s)$ of atomic propositions that are (assumed to be) valid in s. For technical reasons, we require that none of the states is terminal, i.e., for each state s there exists an action α and a state s' with $\mathsf{P}(s,\alpha,s') > 0$. $\mathcal{M}$ is called finite if the state space S is finite. If $T \subseteq S$, then $\mathsf{P}(s,\alpha,T) = \sum_{t \in T} \mathsf{P}(s,\alpha,t)$ denotes the probability for s to move to a T-state, provided that action α has been selected in state s. We write $\mathsf{Act}_{\mathcal{M}}(s)$ or briefly $\mathsf{Act}(s)$ for the action-set $\{\alpha \in \mathsf{Act} \mid \mathsf{P}(s,\alpha,S) = 1\}$.

A *path* in $\mathcal{M}$ is a finite or infinite alternating sequence of states and actions $\sigma = s_1, \alpha_1, \ldots, \alpha_{n-1}, s_n$ or $\varsigma = s_1, \alpha_1, s_2, \alpha_2, \ldots$ such that $\mathsf{P}(s_i, \alpha_i, s_{i+1}) > 0$. $\sigma[i]$ denotes the i-th state of σ, $\mathsf{first}(\sigma) = \sigma[0]$, and $\mathsf{pref}(\sigma, i)$ denotes the i-th prefix of σ (ending in $\sigma[i]$). For finite paths, $\mathsf{last}(\sigma)$ denotes the last state of σ, while $\mathsf{length}(\sigma)$ stands for the number of transitions in σ. For infinite paths, $\mathsf{trace}(\varsigma)$ denotes the infinite word over the alphabet 2^{AP} which arises from ς by the projection of the induced state-sequence to the sequence of the labelings. If ς is as above then $\mathsf{trace}(\varsigma) = L(s_1), L(s_2), L(s_3), \ldots \in (2^{\mathsf{AP}})^{\omega}$. $\mathsf{Lim}(\varsigma)$ denotes the pair (T, A) where T is the set of states that occur infinitely often in ς and where $A : T \to \mathsf{Act}$ assigns to state $s \in T$ the set of actions $\alpha \in \mathsf{Act}(s)$ with $s = s_i$ and $\alpha = \alpha_i$ for infinitely many indices i. $\mathsf{Path}_{\mathcal{M}}(s)$ (briefly $\mathsf{Path}(s)$) stands for the set of infinite paths in $\mathcal{M}$ which start in state s.

In the sequel, we assume that $\mathcal{M}$ is a finite MDP and S_0 a nonempty subset of S consisting of the states which are under the control of the system, i.e., where the system may decide which of the possible actions is executed. The states in $S \setminus S_0$ are controlled by the environment. By a *strategy* for $(\mathcal{M}, S_0)$, we mean any instance D that resolves the nondeterminism in the S_0-states. We distinguish four types of strategies for $(\mathcal{M}, S_0)$, where M stands for Markovian, H for history-dependent, D for deterministic and R for randomized.

- A MD-strategy is a function $D : S_0 \to \mathsf{Act}$ such that $D(s) \in \mathsf{Act}(s)$.
- A MR-strategy is a function $D : S_0 \to \mathsf{Distr}(\mathsf{Act})$ with $D(s) \in \mathsf{Distr}(\mathsf{Act}(s))$
- A HD-strategy is a function D that assigns to any finite path σ in $\mathcal{M}$ with $\mathsf{last}(\sigma) = s \in S_0$ an action $D(\sigma) \in \mathsf{Act}(s)$.
- A HR-strategy is a function D that assigns to any finite path σ with $\mathsf{last}(\sigma) = s \in S_0$ a distribution $D(\sigma) \in \mathsf{Distr}(\mathsf{Act}(s))$.

MD-strategies are often called *simple* or *purely memoryless*. A D-path denotes a path that can be generated by D. E.g., if D is a HD-strategy and σ as above then σ is a D-path iff for all indices $i \in \{1, \ldots, n-1\}$ where $s_i \in S_0$ the chosen action α_i in σ agrees with $D(\mathsf{pref}(\sigma, i))$. We refer to the strategies for the environment as *adversaries*. Formally, for $\mathrm{X} \in \{\mathrm{MD}, \mathrm{MR}, \mathrm{HD}, \mathrm{HR}\}$, a X-adversary for $(\mathcal{M}, S_0)$ denotes a X-strategy for $(\mathcal{M}, S \setminus S_0)$. The notion of *policy* will be used to denote a decision rule that resolves both the internal

nondeterministic choices and the nondeterministic choices to be resolved by the environment. Thus, by a X-policy for $\mathcal{M}$ we mean a X-strategy for $(\mathcal{M}, S)$. We will use the letter D for strategies, E for adversaries and C for policies. Policies will often be written in the form $C = (D, E)$.

It is clear that the four strategy-types form a hierarchy. Each simple strategy can be viewed as a MR-strategy (which chooses for any state $s \in S_0$ a fixed action $\alpha \in \mathsf{Act}(s)$ with probability 1) and as a HD-strategy (which only looks for the last state of a path). Similarly, any HD-strategy can be viewed as a HR-strategy. Hence, the class of HR-strategies subsumes the other three strategy-classes MD, HD and MR.

The MDP induced by a strategy. Any strategy D for $(\mathcal{M}, S_0)$ induces a MDP $\mathcal{M}_D$ which arises through unfolding $\mathcal{M}$ into a tree-like structure where the nondeterministic choices in the S_0-states are resolved according to D. E.g., if D is a HD-strategy for $(\mathcal{M}, S_0)$ then the states in $\mathcal{M}_D$ are the finite D-paths. The initial state of $\mathcal{M}_D$ is s_{init}, viewed as a path of length 0. If σ is a finite D-path and $\mathsf{last}(\sigma) \in S_0$ then $\mathsf{Act}_{\mathcal{M}_D}(\sigma) = \{D(\sigma)\}$ and $\mathsf{P}_D(\sigma, D(\sigma), \sigma') = \mathsf{P}(\mathsf{last}(\sigma), D(\sigma), s)$ if $\sigma' = \sigma, \alpha, s$ where $\alpha = D(\sigma)$, and $\mathsf{P}_D(\sigma, \alpha, \sigma') = 0$ in all other cases. If $\mathsf{last}(\sigma) \notin S_0$ then $\mathsf{Act}_{\mathcal{M}_D}(\sigma) = \mathsf{Act}_{\mathcal{M}}(\mathsf{last}(\sigma))$ and $\mathsf{P}_D(\sigma, \alpha, \langle \sigma, \alpha, s \rangle) = \mathsf{P}(\mathsf{last}(\sigma), \alpha, s)$ for all $\alpha \in \mathsf{Act}$ and $s \in S$. The MDP $\mathcal{M}_D$ for HR-strategies is defined in the same way except that $\mathsf{P}_D(\sigma, \alpha, \langle \sigma, \alpha, s \rangle) = D(\sigma)(\alpha) \cdot \mathsf{P}(\mathsf{last}(\sigma), \alpha, s)$ if $\mathsf{last}(\sigma) \in S_0$.

Markov chains and probability measure for policies. If $S_0 = S$ and $D = C$ is a policy for $\mathcal{M}$ then all nondeterministic choices are resolved in $\mathcal{M}_C$. Hence, for any HR-policy C, the MDP $\mathcal{M}_C$ is an infinite-state discrete-time Markov chain. If C is a stationary Markovian policy then all finite C-paths σ, σ' (viewed as states of $\mathcal{M}_C$) with $\mathsf{last}(\sigma) = \mathsf{last}(\sigma')$ can be identified. Hence, $\mathcal{M}_C$ can be viewed as a (discrete-time) Markov chain with state space S. If C is a policy for $\mathcal{M}$, then we write $\Pr^C_{\mathcal{M}}$ or briefly $\Pr^C$ to denote the (standard) probability measure on $\mathcal{M}_C$.

Probabilistic Computation Tree Logic (PCTL). PCTL (and its extension PCTL*) [Hansson and Jonsson, 1994; Bianco and De Alfaro, 1995] is a branching-time temporal logic à la CTL/CTL* where state-formulas are interpreted over states of a MDP and path-formulas over its paths. It incorporates an operator to refer to the probability of the occurrence of particular paths (rather than quantification over paths as in CTL). In the sequel, we assume a fixed set AP of atomic propositions and use the letter a to denote an atomic proposition (i.e., $a \in \mathsf{AP}$). The letter p stands for a probability bound (i.e., $p \in [0, 1]$). The symbol $\bowtie$ is one of the comparison operators $\leqslant$ or $\geqslant$. The syntax of PCTL*-state formulas (denoted by Φ, Ψ) and path formulas (denoted by φ) is as follows:

$$\begin{array}{lcl} \Phi & ::= & \mathrm{tt} \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathcal{P}_{\bowtie p}(\varphi) \\ \varphi & ::= & \Phi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \bigcirc \varphi \mid \varphi \,\mathcal{U}\, \varphi \end{array}$$

Intuitively, $\mathcal{P}_{\bowtie p}(\varphi)$ asserts that the probability measure of the paths satisfying φ meets the bound given by $\bowtie p$. The path modalities $\bigcirc$ (next step) and $\mathcal{U}$ (Until) have the same meaning as in CTL*. Other boolean connectives (e.g. $\vee$) and the temporal operators $\Diamond$ (eventually) and $\Box$ (always) can be derived as in CTL* by $\Diamond\varphi = \mathrm{tt}\,\mathcal{U}\,\varphi$ and $\Box\varphi = \neg\Diamond\neg\varphi$. PCTL denotes the sublogic where only path formulas of the form $\bigcirc\Phi$ and $\Phi\,\mathcal{U}\,\Psi$ are allowed. The always-operator can be derived in PCTL using the duality of lower and upper probability bounds, e.g. $\mathcal{P}_{\geqslant p}(\Box\Phi) = \mathcal{P}_{\leqslant 1-p}(\Diamond\neg\Phi)$. PCTL$_{\setminus\bigcirc}$ (PCTL$_{\setminus\mathcal{U}}$) denotes the fragment of PCTL that does not use the next step (until) operator. LTL (linear time logic) denotes the path-formula fragment of PCTL* where atoms are atomic propositions (rather than arbitrary state formulas).

Given a MDP $\mathcal{M}$ as before, the formal definition of the satisfaction relation $\models$ for PCTL*-path formulas and propositional PCTL*-state formulas is exactly as for CTL* and omitted here. For the probabilistic operator, the semantics is defined by$s \models \mathcal{P}_{\bowtie p}(\varphi)$ iff for all policies C : $\Pr^C(s,\varphi) \bowtie p$ where $\Pr^C(s,\varphi) = \Pr^C\{\varsigma \in \mathsf{Path}(s) \mid \varsigma \models \varphi\}$. We shall use $\Pr^C(\varphi)$ as an abbreviation for $\Pr^C(s_{\mathit{init}},\varphi)$. To distinguish the satisfaction relation for different MDPs, we sometimes write $(\mathcal{M},s) \models \Phi$ instead of $s \models \Phi$. We write $\mathcal{M} \models \Phi$ iff Φ holds in the initial state of $\mathcal{M}$.

The satisfaction relation for PCTL does not depend on the chosen policy-type because maximal and minimal probabilities for PCTL-path formulas under all HR-policies are reached with simple policies [Bianco and De Alfaro, 1995].

Rabin automata. A *deterministic Rabin automaton* is a structure $\mathcal{A} = \big(Q,\Pi,\delta,q_0,\mathsf{Acc}\big)$ where Q is a finite state space, Π the alphabet, $q_0 \in Q$ the starting state, and $\delta : Q \times \Pi \to Q$ the transition function. (To encode an LTL-formula by a Rabin automaton the alphabet $\Pi = 2^{\mathsf{AP}}$ is used.) The acceptance condition Acc is a set of tuples (H_i,K_i) consisting of subsets H_i and K_i of Q. The run for an infinite word $\pi = \pi[0],\pi[1],\ldots \in \Pi^\omega$ in $\mathcal{A}$ means the infinite sequence $q_0,q_1,q_2,\ldots$ of automata-states where $q_{i+1} = \delta(q_i,\pi[i])$. Acceptance of π under the Rabin condition $\mathsf{Acc} = \{(H_i,K_i) : i = 1,\ldots,m\}$ can be described by the LTL-formula $\mathsf{acc}(\mathcal{A}) = \bigvee_{1\leq i\leq m} \Diamond\Box(H_i \wedge \Diamond K_i)$. That is, a run $\rho = q_0,q_1,q_2,\ldots$ in $\mathcal{A}$ is accepting if there is at least one pair (H_i,K_i) in the acceptance condition Acc of $\mathcal{A}$ such that $\lim(\rho) \subseteq H_i$ and $\lim(\rho) \cap K_i \neq \varnothing$ where $\lim(\rho)$ denotes the set of all states in Q which appear infinitely often in ρ. $\mathcal{L}(\mathcal{A})$ denotes the accepted language of $\mathcal{A}$, i.e., the set of infinite words $\rho \in \Pi^\omega$ whose run in $\mathcal{A}$ is accepting. Given a MDP $\mathcal{M} = (S,\mathsf{Act},\mathsf{P},s_{\mathit{init}},\mathsf{AP},L)$, policy C for $\mathcal{M}$ and Rabin automaton $\mathcal{A} = (Q,2^{\mathsf{AP}},\delta,Q_0,\mathsf{Acc})$, we write $\Pr^C(s,\mathcal{A})$ for the probability measure of all C-paths that start in state s and that generate a trace which is accepted by $\mathcal{A}$, i.e., we put $\Pr^C(s,\mathcal{A}) = \Pr^C\{\varsigma \in \mathsf{Path}(s) : \mathsf{trace}(\varsigma) \in \mathcal{L}(\mathcal{A})\}$. $\Pr^C(\mathcal{A})$ stands short for $\Pr^C(s_{\mathit{init}},\mathcal{A})$.

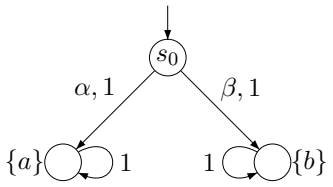


Fig. 1(a) randomization helps

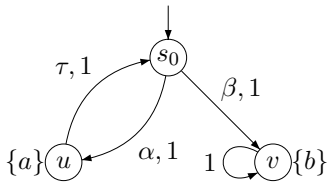


Fig. 1(b) history helps

3. The controller synthesis problem for PCTL

The controller synthesis problems discussed in this paper are formalized by triples $(\mathcal{M}, S_0, \mathit{Spec})$ where $\mathcal{M}$ is a finite MDP, S_0 a set of controllable states in $\mathcal{M}$ and Spec a temporal-logical or ω-regular specification. The question is to find a strategy D for $(\mathcal{M}, S_0)$ such that Spec holds for the MDP $\mathcal{M}_D$, no matter how the environment (adversary) behaves.

This section addresses the case where Spec is a PCTL-state formula and first discusses the role of the strategy-type. Let $\mathrm{X} \in \{\mathrm{MD}, \mathrm{MR}, \mathrm{HD}, \mathrm{HR}\}$ be a strategy class. The X-controller synthesis problem for PCTL is as follows:

Given: a finite MDP $\mathcal{M}$, a subset S_0 of states and a PCTL-state formula Φ.

Wanted: a X-strategy D for $(\mathcal{M}, S_0)$ such that $\mathcal{M}_D \models \Phi$ (if one exists).

Clearly, any solution of the MD-strategy controller synthesis problem (i.e., any simple strategy D with $\mathcal{M}_D \models \Phi$) is at the same time a solution for the controller synthesis problem for any other strategy-class which subsumes the simple strategies (in particular, for the strategy-classes MR, HD and HR). With the same argument, if the HD- or MR-controller synthesis problem is solvable then so is the HR-controller synthesis problem.

The question arises whether (as for the PCTL satisfaction relation) e.g. simple strategies are as powerful as HD-strategies to solve the controller synthesis problem and the same question for other strategy-classes X_1 (instead of MD) and X_2 (instead of HD) with $\mathrm{X}_2 \not\subseteq \mathrm{X}_1$. The answer is *no* in either case (more precisely, for the strategy-classes MD, MR, HD and HR discussed here), even for the sublogics $\mathrm{PCTL}_{\setminus\bigcirc}$ and $\mathrm{PCTL}_{\setminus\mathcal{U}}$.

For the MDP $\mathcal{M}$ in Fig. 1(a) with $s_0 \in S_0$ and the $\mathrm{PCTL}_{\setminus\mathcal{U}}$-formula $\Phi = \mathcal{P}_{\geqslant 0.5}(\bigcirc a) \wedge \mathcal{P}_{\geqslant 0.5}(\bigcirc b)$ the HD-controller synthesis problem is not solvable for $\mathcal{M}$, S_0 and Φ. On the other hand, $(\mathcal{M}_D, s_0) \models \Phi$ for the MR-strategy D which assigns probability 1/2 to actions α and β in state s_0. Hence, the MR-controller synthesis problem is solvable for $\mathcal{M}$, S_0 and Φ. The same argument applies to the $\mathrm{PCTL}_{\setminus\bigcirc}$-formula $\mathcal{P}_{\geqslant 0.5}(\Diamond a) \wedge \mathcal{P}_{\geqslant 0.5}(\Diamond b)$. Thus, randomized strategies (MR, HR) can be more powerful than deterministic (MD, HD) strategies to solve the controller synthesis problem for $\mathrm{PCTL}_{\setminus\mathcal{U}}$ or $\mathrm{PCTL}_{\setminus\bigcirc}$.

The following shows that there are instances $(\mathcal{M}, S_0, \Phi)$ for which the controller synthesis problem for the strategy-class HD is solvable but not for the MR-strategies. For the MDP shown in Figure 1(b), with $s_0 \in S_0$, and $\Phi =$

$\mathcal{P}_{\geqslant 1}(\bigcirc a) \wedge \mathcal{P}_{\geqslant 1}(\bigcirc \Psi)$, with $\Psi = \mathcal{P}_{\geqslant 1}(\bigcirc \mathcal{P}_{\geqslant 1}(\bigcirc b))$ there is a HD-strategy D with $(\mathcal{M}_D, s_0) \models \Phi$. On the other hand, the only MR-strategy D which guarantees for s_0 that with probability 1 the next state is an a-state is given by $D(s_0)(\alpha) = 1$, and $D(s_{init})(\beta) = 0$. For this MR-strategy D, we have $(\mathcal{M}_D, s_0) \not\models \mathcal{P}_{\geqslant 1}(\bigcirc \Psi)$, and hence, $(\mathcal{M}_D, s_0) \not\models \Phi$. The same argument applies to the PCTL$_{\setminus\bigcirc}$-formula $\mathcal{P}_{\geqslant 1}(\Diamond a) \wedge \mathcal{P}_{\geqslant 1}(\Diamond b)$ where the only chance for a MR-strategy to reach the a-state u with probability 1 is to select action α in state s_0 with probability 1.

The previous remarks show that the role of strategy-types for controller synthesis is completely different from the situation in PCTL model checking. While a single algorithm suffices for PCTL model checking, for controller synthesis, any strategy type requires its own synthesis algorithm!

The naïve idea to solve the MD-controller synthesis problem for $\mathcal{M}$, S_0 and PCTL-formula Φ is to consider all simple strategies D for $(\mathcal{M}, S_0)$ and to apply a standard PCTL model checking algorithm to $\mathcal{M}_D$ and Φ. The time complexity is linear in the length of Φ and exponential in $\mathsf{size}(\mathcal{M})$, but we should not expect an algorithm which is efficient for all MDPs because of the following theorem which shows that the decision variant of the controller synthesis problem is NP-complete. The decision variant asks for the *existence* of a simple strategy D such that $\mathcal{M}_D \models \Phi$ but not for such a strategy. To prove membership in NP we need the existence of a polynomial-time algorithm that calculates the *precise* maximal or minimal probabilities for PCTL-path formulas under simple policies (rather than approximation algorithms). For instance, this is possible if all probabilities in the given MDP $\mathcal{M}$ and all probability bounds in the given PCTL formula Φ are rational. In this case, we may apply the PCTL model checking procedure à la Bianco and de Alfaro [Bianco and De Alfaro, 1995] using precise methods to solve linear programs.

THEOREM 1 *Under the above conditions, the decision variant of the MD-controller synthesis problem for PCTL and its sublogics* PCTL$_{\setminus\mathcal{U}}$ *and* PCTL$_{\setminus\bigcirc}$ *is NP-complete, even when we require all states in the MDP to be controllable.*

THEOREM 2 *The decision variant of the MR/HD/HR-controller synthesis for PCTL and its sublogics* PCTL$_{\setminus\bigcirc}$ *and* PCTL$_{\setminus\mathcal{U}}$ *is NP-hard, even when all states are required to be controllable.*

PCTL with fairness. In Section 4, we shall need a variant of the controller synthesis problem for PCTL where fairness assumptions about the adversaries are made. The X-controller synthesis problem for PCTL with fairness assumes a finite MDP $\mathcal{M} = (S, \mathsf{Act}, \mathsf{P}, s_{init}, \mathsf{AP}, L)$, a subset S_0 of S, a PCTL-formula Φ and, in addition, a fairness condition for the adversaries. It asks for a X-strategy D such that $\mathcal{M}_D \models_{fair} \Phi$ where the satisfaction relation $\models_{fair}$ is defined as the standard satisfaction relation $\models$, except for the probabilistic op-

erator: $s \models_{\mathit{fair}} \mathcal{P}_{\bowtie p}(\varphi)$ iff for all fair policies C: $\Pr^C(s,\varphi) \bowtie p$. Several fairness notions for MDPs have been suggested [Vardi, 1985; Pnueli and Zuck, 1986; Baier and Kwiatkowska, 1998]. In Section 4 we shall use the notion of a fair adversary (for a given strategy D) to denote an adversary F such that almost all (D,F)-paths are fair.

The NP-completeness established in Theorem 1 for PCTL without fairness carries over to PCTL with fairness. To solve the MD-controller synthesis problem for PCTL with fairness conditions, we may apply the model checking algorithm suggested in [Baier and Kwiatkowska, 1998] to each MDP $\mathcal{M}_D$ induced by a simple strategy D. For other strategy types (MR, HD or HR), the complexity or even the decidability of the controller synthesis problem for PCTL (without or with fairness) is an open problem.

4. HD-controller synthesis for automata-specifications

We now address the controller synthesis problem where the specification is provided by means of an ω-automaton and a probability bound "$\bowtie p$". Using an automata-representation for a given LTL-formula, the techniques suggested here also solve the controller synthesis problem for LTL.

In the rest of this section, $\mathcal{M} = (S, \mathsf{Act}, \mathsf{P}, s_{\mathit{init}}, \mathsf{AP}, L)$ is a finite MDP, $S_0 \subseteq S$, and $\mathcal{A} = (Q, 2^{\mathsf{AP}}, \delta, q_0, \mathsf{Acc})$ a deterministic Rabin automaton as in Sect. 2. The X-controller synthesis problem for $\mathcal{M}$, S_0, $\mathcal{A}$, "$\bowtie p$" asks whether there is a X-strategy D such that $\Pr^{(D,E)}(\mathcal{A}) \bowtie p$ for all HD-adversaries E.

To see the difference between the controller synthesis problems for the strategy-classes HD and MD resp. MR, consider the following. Let $\mathcal{M}$ be as in figure 1(b), $s_0 \in S_0$ and $\varphi = (\Diamond a \wedge \Diamond b)$ and $\varphi' = (\bigcirc a \wedge \bigcirc \bigcirc \bigcirc b)$ be $\mathrm{LTL}_{\setminus\bigcirc}$ and $\mathrm{LTL}_{\setminus\mathcal{U}}$ formulas respectively. The controller synthesis problem for $\mathcal{M}$, S_0, φ (resp. φ') and probability bound "$\geqslant 1$" is solvable for HD, but not for MD or MR strategies. The controller synthesis problem for $\mathcal{M}$, S_0, φ (resp. φ') and probability bound "$\geqslant p$" is solvable for MR, but not for MD strategies for $0 < p < 1$ (resp. $0 < p \leqslant \frac{1}{4}$). So any of the strategy types MD, MR, HD requires its own synthesis algorithm.

On the other hand, the two history-dependent strategy types HD and HR are equivalent for the controller synthesis problem for automata-specifications as HR-strategies can be viewed as convex combinations of (possibly infinitely many) HD-strategies, see e.g. [Derman, 1970; Puterman, 1994].

In the following, we present a solution for the HD-controller synthesis problem for $\mathcal{M}$, S_0, $\mathcal{A}$ and *lower* probability bounds "$\geqslant p$". Thus, our goal is the construction of a HD-strategy D such that $\Pr^{(D,E)}(s_{\mathit{init}}, \mathcal{A}) \geqslant p$ for all HD-adversaries E. Upper probability bounds can be treated in a similar way.

DEFINITION 3 (PRODUCT-MDP [DE ALFARO, 1997]) The MDP $\mathcal{M} \times \mathcal{A} = (S \times Q, \mathsf{Act}, \mathsf{P}, t_{\mathit{init}}, \mathsf{AP}', L')$ is defined as follows: The initial state t_{init} is

$\langle s_{init}, q_{init}\rangle$ where $q_{init} = \delta(q_0, L(s_{init}))$. The values of the transition probability matrix are given by $\mathsf{P}(\langle s,q\rangle, \alpha, \langle s', \delta(q, L(s'))\rangle) = \mathsf{P}(s,\alpha,s')$ and $\mathsf{P}(\cdot) = 0$ in all other cases. The set AP' is $\mathsf{AP} \cup (S \times Q) \cup Q$ where AP, $S \times Q$ and Q are supposed to be pairwise disjoint. The labeling function L' is given by $L'(\langle s,q\rangle) = L(s) \cup \{\langle s,q\rangle, q\}$. The "liftings" of the sets $H_i, K_i \subseteq Q$ in the acceptance condition of $\mathcal{A}$ are defined by $\bar{H}_i = S \times H_i$, and $\bar{K}_i = S \times K_i$. If $P \subseteq (S \times Q) \cup Q$ then we write P for the propositional formula $\bigvee_{q \in P} q$. ■

There is a one-to-one correspondence between the paths in $\mathcal{M}$ and $\mathcal{M} \times \mathcal{A}$. Given a (finite or infinite) path π in $\mathcal{M}$, we lift π to a path $\pi^\times$ in $\mathcal{M} \times \mathcal{A}$ by adding automata components which describe the run of π in $\mathcal{A}$. Vice versa, given a path π in $\mathcal{M} \times \mathcal{A}$, the projection $\pi|_{\mathcal{M}}$ of π to the state sequence in $\mathcal{M}$ is a path in $\mathcal{M}$ while the projection $\pi|_{\mathcal{A}}$ of π to the sequence of automata-states is the run for $\pi|_{\mathcal{M}}$ in $\mathcal{A}$. This observation yields a one-to-one correspondence between the HD-strategies for $(\mathcal{M}, S_0)$ and $(\mathcal{M} \times \mathcal{A}, S_0 \times Q)$ in the following sense. If D is a strategy for $(\mathcal{M}, S_0)$ then we may define a strategy $D^\times$ for $(\mathcal{M} \times \mathcal{A}, S_0 \times Q)$ by $D^\times(\sigma) = D(\sigma|_{\mathcal{M}})$. Vice versa, given a strategy D for $(\mathcal{M} \times \mathcal{A}, S_0 \times Q)$, we may define the "corresponding" strategy $D|_{\mathcal{M}}$ for $(\mathcal{M}, S_0)$ by $D|_{\mathcal{M}}(\sigma) = D(\sigma^\times)$. The described transformation $D \mapsto D^\times$ is type-preserving in the sense that if D is a X-strategy for $(\mathcal{M}, S_0)$ then $D^\times$ is a X-strategy for $(\mathcal{M} \times \mathcal{A}, S_0 \times Q)$, while the converse transformation $D \mapsto D|_{\mathcal{M}}$ may yield a HD-strategy $D|_{\mathcal{M}}$ for $(\mathcal{M}, S_0)$ if D is a simple strategy for $(\mathcal{M} \times \mathcal{A}, S_0 \times Q)$. If C is a HD-policy for $\mathcal{M}$ and $C^\times$ the induced HD-policy in $\mathcal{M} \times \mathcal{A}$ then $\Pr^C_{\mathcal{M}}(\mathcal{A}) = \Pr^{C^\times}_{\mathcal{M}\times\mathcal{A}}(\mathcal{A}) = \Pr^{C^\times}_{\mathcal{M}\times\mathcal{A}}(\mathsf{acc}(\mathcal{A}))$. By the one-to-one-relation for both the adversaries E and strategies D, we get:

LEMMA 4 $\sup_D \inf_E \Pr^{(D,E)}_{\mathcal{M}}(\mathcal{A}) = \sup_D \inf_E \Pr^{(D,E)}_{\mathcal{M}\times\mathcal{A}}(\mathcal{A})$

Lemma 4 allows us to focus on the product-MDP. From now on, if not stated otherwise, by a strategy (an adversary) we mean a strategy (an adversary) for $(\mathcal{M} \times \mathcal{A}, S_0 \times Q)$. [de Alfaro, 1997] defines end components of the product-MDP as the MDP-analogue of recurrent sets in discrete-time Markov chains. Intuitively, end components are sub-MDPs for which a policy can be defined such that almost all paths in the end component visit any state of the end component infinitely often. Formally, an *end component* [de Alfaro, 1997] for the MDP $\mathcal{M} \times \mathcal{A}$ denotes a pair (T, A) consisting of a nonempty subset T of $S \times Q$ and a function $A : T \to \mathsf{Act}$ such that (i) $\varnothing \neq A(t) \subseteq \mathsf{Act}(t)$ for all states $t \in T$, (ii) $\mathsf{P}(t, \alpha, T) = 1$ for all $t \in T$ and $\alpha \in A(t)$ and (iii) the induced digraph $(T, \longrightarrow_A)$ is strongly connected. (Here, $t \longrightarrow_A t'$ iff $\mathsf{P}(t, \alpha, t') > 0$ for some $\alpha \in A(t)$.) An *accepting end component* (AEC) is an end component (T, A) such that $T \subseteq \bar{H}_i$ and $T \cap \bar{K}_i \neq \varnothing$ for some index $i \in \{1, \ldots, m\}$.

[de Alfaro, 1997] shows that for each policy C, the probability measure for the infinite paths ς where $\mathsf{Lim}(\varsigma)$ is an end component is 1. Hence, we have $\Pr^{C}_{\mathcal{M}\times\mathcal{A}}(\mathcal{A}) = \Pr^{C}_{\mathcal{M}\times\mathcal{A}}\{\varsigma : \mathsf{Lim}(\varsigma) \text{ is an AEC}\}$.
For our purposes, we need a variant of accepting end components, called winning components. The idea is that for any state t of a winning component there is a strategy such that—independent on how the adversary resolves the nondeterminism—almost all paths starting in t will eventually reach an AEC and stay there forever.

DEFINITION 5 (WINNING COMPONENT) A winning component denotes a pair (T, A) consisting of a nonempty subset T of $S \times Q$ and a function $A : T \to \mathsf{Act}$ such that (1) $A(t) \subseteq \mathsf{Act}(t)$ and $|A(t)| = 1$ for all $t \in T \cap (S_0 \times Q)$, (2) $A(t) = \mathsf{Act}(t)$ for all $t \in T \cap ((S \setminus S_0) \times Q)$, (3) $\mathsf{P}(t, \alpha, T) = 1$ for all $t \in T$ and $\alpha \in A(t)$ and (4) for any simple adversary E and any bottom strongly connected component U of the digraph $(T, \longrightarrow_E)$ with $t \longrightarrow_E t'$ iff $\mathsf{P}(t, E(t), t') > 0$ there exists an index $i \in \{1, \ldots, m\}$ such that $U \subseteq \bar{H}_i$ and $U \cap \bar{K}_i \neq \varnothing$. WC denotes the set of all states $t \in S \times Q$ that are contained in some winning component. ■

Our goal is now to show that the best strategy to generate $\mathcal{A}$-paths can be derived from the best strategy to reach a winning component.

LEMMA 6 *For any state $t_0 \in (S \times Q) \setminus \mathsf{WC}$ and HD-strategy D, there exists a HD-adversary E such that $\Pr^{(D,E)}_{\mathcal{M}\times\mathcal{A}}(t_0, \mathcal{A}) < 1$.*

We now show that any strategy can be improved by forcing the system to stay in WC as soon as WC is reached.

LEMMA 7 *There is a simple strategy D_{WC} such that $\Pr^{(D_{\mathsf{WC}},E)}_{\mathcal{M}\times\mathcal{A}}(t, \mathcal{A}) = 1$ for all HD-adversaries E and all states $t \in \mathsf{WC}$. and for any infinite D_{WC}-path ς, if $\varsigma[i] \in \mathsf{WC}$ then $\varsigma[j] \in \mathsf{WC}$ for all $j \geqslant i$.*

Lemma 6 and 7 yield:

COROLLARY 8 *For any state $t \in \mathcal{M} \times \mathcal{A}$: $t \in \mathsf{WC}$ iff there exists a HD-strategy D with $\Pr^{(D,E)}_{\mathcal{M}\times\mathcal{A}}(t, \mathcal{A}) = 1$ for all HD-adversaries E.*

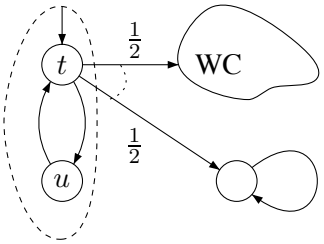
LEMMA 9 *For any HD-strategy D there is a HD-strategy $\widehat{D}$ such that for all HD-adversaries E:*

(1) *For any infinite $\widehat{D}$-path ς, if $\varsigma[i] \in \mathsf{WC}$ then $\varsigma[j] \in \mathsf{WC}$ for all $j \geqslant i$.*

(2) $\Pr^{(\widehat{D},E)}_{\mathcal{M}\times\mathcal{A}}(\Diamond\mathsf{WC}) = \Pr^{(\widehat{D},E)}_{\mathcal{M}\times\mathcal{A}}(\Diamond\mathsf{WC} \wedge \mathsf{acc}(\mathcal{A}))$

(3) $\Pr^{(\widehat{D},E)}_{\mathcal{M}\times\mathcal{A}}(\mathcal{A}) \geqslant \Pr^{(D,E)}_{\mathcal{M}\times\mathcal{A}}(\mathcal{A})$

Our rough goal is to show $\sup_D \inf_E \Pr^{(D,E)}(\mathcal{A}) = \sup_D \inf_E \Pr^{(D,E)}(\Diamond\mathsf{WC})$.



Lemma 9 yields "$\geqslant$". Unfortunately, "$\leqslant$" does not hold in general. For instance, in the MDP shown aside, we assume that states t and u build an AEC which is not contained in WC. If $t \notin S_0$ then the best adversary, $\widehat{E}$, chooses the transition that leaves the AEC $\{t, u\}$ and moves with probability 1/2 to WC. On the other hand, under the adversary E that forces the system to stay forever in the AEC $\{t, u\}$, WC is never reached, and thus, E minimizes the probability for $\Diamond\mathsf{WC}$. However, any adversary, which—as in the example aside—forces the system to stay in an AEC that does not intersect with WC, can be improved by leaving the AEC, even if WC is reached under the modified adversary.

LEMMA 10 *For any HD-strategy $\widehat{D}$ which fulfills condition (1) and (2) of Lemma 9 and any HD-adversary E there exists a HD-adversary $\widehat{E}$ such that*

(4) $\Pr_{\mathcal{M}\times\mathcal{A}}^{(\widehat{D},\widehat{E})}(\Gamma) = 0$ *where* Γ *denotes the set of infinite paths* ς *that start in* t_{init} *and where* $\mathsf{Lim}(\varsigma) = (T, A)$ *is an AEC with* $T \cap \mathsf{WC} = \varnothing$.

(5) $\Pr_{\mathcal{M}\times\mathcal{A}}^{(\widehat{D},E)}(\mathcal{A}) \;\geqslant\; \Pr_{\mathcal{M}\times\mathcal{A}}^{(\widehat{D},\widehat{E})}(\mathcal{A}) \;=\; \Pr_{\mathcal{M}\times\mathcal{A}}^{(\widehat{D},\widehat{E})}(\Diamond\mathsf{WC})$

For policies $(\widehat{D}, \widehat{E})$ where (1), (2) and (4) hold, the probability to reach WC agrees with the probability for the $\mathcal{A}$-paths. Thus, using Lemma 7, 9 and 10, we obtain: $\sup_D \inf_E \Pr_{\mathcal{M}\times\mathcal{A}}^{(D,E)}(\mathcal{A}) = \sup_{\widehat{D}} \inf_{\widehat{E}} \Pr_{\mathcal{M}\times\mathcal{A}}^{(D,\widehat{E})}(\Diamond\mathsf{WC})$ where D, E range over all HD-strategies/HD-adversaries and $\widehat{D}$, $\widehat{E}$ over all HD-strategies/HD-adversaries satisfying (1), (2) and (4). We now show that the adversaries where (4) holds are exactly the adversaries that are fair in the following sense:

DEFINITION 11 (AEC-FAIRNESS) For any state $t \in ((S\setminus S_0)\times Q)\cap(\mathsf{AEC}\setminus \mathsf{WC})$, let $\mathsf{FairAct}(t)$ be the set of actions $\alpha \in \mathsf{Act}(t)$ such that $\mathsf{P}(t, \alpha, S \setminus \mathsf{AEC}) > 0$. Here, AEC denotes the set of all states that are contained in some AEC. For any other state $t \in S \times Q$, we put $\mathsf{FairAct}(t) = \varnothing$. An infinite path $\varsigma = s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \ldots$ is called AEC-fair iff for all $t \in ((S\setminus S_0)\times Q)\cap(\mathsf{AEC}\setminus \mathsf{WC})$ where $\mathsf{FairAct}(t) \neq \varnothing$: If t occurs infinitely often in ς then there are infinitely many indices i with $s_i = t$ and $\alpha_i \in \mathsf{FairAct}(s_i)$. An HD-adversary F is called AEC-fair for strategy D if $\Pr_{\mathcal{M}\times\mathcal{A}}^{(D,F)}\{\varsigma : \varsigma \text{ is AEC-fair}\} = 1$. ■

Given a strategy $\widehat{D}$ satisfying (1) and (2), any adversary $\widehat{E}$ that fulfills condition (4) in Lemma 10 is AEC-fair for $\widehat{D}$. Vice versa, (4) holds for any adversary F that is AEC-fair for $\widehat{D}$. Thus:

$$\inf_E \Pr_{\mathcal{M}\times\mathcal{A}}^{(\widehat{D},E)}(\mathcal{A}) \;=\; \inf_{F \text{ fair}} \Pr_{\mathcal{M}\times\mathcal{A}}^{(\widehat{D},F)}(\mathcal{A}) \;=\; \inf_{F \text{ fair}} \Pr_{\mathcal{M}\times\mathcal{A}}^{(\widehat{D},F)}(\Diamond\mathsf{WC})$$

LEMMA 12 $\sup_D \inf_E \Pr^{(D,E)}_{\mathcal{M}\times\mathcal{A}}(\mathcal{A}) = \sup_D \inf_{F \text{ fair}} \Pr^{(D,F)}_{\mathcal{M}\times\mathcal{A}}(\Diamond \mathsf{WC})$
where D ranges over all HD-strategies, E over all HD-adversaries and F over all HD-adversaries that are AEC-fair for D. This follows from Lemma 9.

According to Lemma 12, the HD-controller synthesis problem for Rabin-automata specifications is reducible to the HD-controller synthesis problem for PCTL with fairness. Although the controller synthesis problem for PCTL depends on the chosen strategy-type, for probabilistic reachability properties such as $\mathcal{P}_{\geqslant p}(\Diamond \mathsf{WC})$ we may switch from HD-strategies to simple strategies.

LEMMA 13 $\sup_D \inf_{F \text{ fair}} \Pr^{(D,F)}_{\mathcal{M}\times\mathcal{A}}(\Diamond \mathsf{WC}) = \max_{\widetilde{D} \text{ simple}} \inf_{F \text{ fair}} \Pr^{(\widetilde{D},F)}_{\mathcal{M}\times\mathcal{A}}(\Diamond \mathsf{WC})$
where D ranges over all HD-strategies, $\widetilde{D}$ over all simple strategies and F over all HD-adversaries that are AEC-fair for D resp. $\widetilde{D}$. And finally we get

THEOREM 14 *There is a HD-strategy D for $(\mathcal{M}, S_0)$ which solves the controller synthesis problem for $\mathcal{M}$, S_0, $\mathcal{A}$ and probability bound "$\geqslant p$" iff there is a simple strategy $\widetilde{D}$ for the MD-controller synthesis problem for $\mathcal{M} \times \mathcal{A}$, $S_0 \times Q$, the PCTL-formula $\mathcal{P}_{\geqslant p}(\Diamond \mathsf{WC})$ and AEC-fairness (Def. 11).*

In summary, the HD-controller synthesis problem for automata specifications and lower probability bounds can be solved by performing the following steps: (i) Built the product-MDP $\mathcal{M} \times \mathcal{A}$, (ii) calculate WC, (iii) check whether there is a simple strategy $\widetilde{D}$ for $(\mathcal{M} \times \mathcal{A}, S_0 \times Q)$ such that $(\mathcal{M} \times \mathcal{A})_{\widetilde{D}} \models_{fair} \mathcal{P}_{\geqslant p}(\Diamond \mathsf{WC})$ and (iv) if no such simple strategy $\widetilde{D}$ exists then return "No." Otherwise return the HD-scheduler D as in the proof of Theorem 14. In step (ii), we may make use of Corollary 8 which yields that WC is the set of states that have a winning strategy for the Rabin-chain winning objective (formalized by the LTL-formula $\mathsf{acc}(\mathcal{A})$) and the almost-sure winning criterion. Reformulating $\mathsf{acc}(\mathcal{A})$ as a parity winning condition, we may apply the reduction technique suggested in [Chatterjee et al., 2003] from qualitative stochastic $2\frac{1}{2}$-player parity games to (non-stochastic) 2-player parity games to calculate WC with known methods [Emerson et al., 1993; Jurdzinski, 2000; Vöge and Jurdzinski, 2000]. In step (iii), the naïve method that applies a model checking algorithm for PCTL with fairness [Baier and Kwiatkowska, 1998] to any of the MDPs $(\mathcal{M} \times \mathcal{A})_{\widetilde{D}}$, the space complexity is bounded by $\mathcal{O}(\mathsf{size}(\mathcal{M}) \cdot \mathsf{size}(\mathcal{A}))$, but the worst-case running time is exponential in $\mathsf{size}(\mathcal{M})$ and $\mathsf{size}(\mathcal{A})$. (Note that the number of simple strategies is $\prod_{s \in S_0} |\mathsf{Act}(s)|^{|Q|} \geqslant 2^{|S_0| \cdot |Q|}$ if $|\mathsf{Act}(s)| \geqslant 2$ for all states $s \in S_0$.)

References

Baier, C. and Kwiatkowska, M. (1998). Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155.

Bianco, A. and De Alfaro, L. (1995). Model checking of probabilistic and non-deterministic systems. In Proc. FST & TCS, LNCS 1026, pages 499–513.

Bouyer, P., D'Souza, D., Madhusudan, P., and Petit, A. (2003). Timed control with partial observability. In Proc. CAV LNCS 2725, pages 180–192.

Chatterjee, K., Jurdzinski, M., and Henzinger, T. (2003). Simple stochastic parity games. In Proc. CSL LNCS 2803, pages 100–113.

Chatterjee, K., Jurdzinski, M., and Henzinger, T. (2004). Quantitative simple stochastic parity games. In *Proceedings of the Annual Symposium on Discrete Algorithms (SODA)*. SIAM.

Condon, A. (1992). The complexity of stochastic games. *Inf. and Comp.*, 96:203–224.

Condon, A. (1993). On algorithms for simple stochastic games. DIMACS, 13:51–71.

de Alfaro, L. (1997). *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University. Technical report STAN-CS-TR-98-1601.

de Alfaro, L., Faella, M., Henzinger, T., Majumdar, R., and Stoelinga, M. (2003). The element of surprise in timed games. In *Proc. CONCUR*, LNCS 2761, pages 144–158.

de Alfaro, L. and Henzinger, T. (2000). Concurrent omega-regular games. In Proc. LICS, pages 141–154. IEEE Computer Society Press.

de Alfaro, L., Henzinger, T., and Kupferman, O. (1998). Concurrent reachability games. In Proc. FOCS, pages 564–575. IEEE Computer Society Press.

de Alfaro, L. and Majumdar, R. (2001). Quantitative solution of omega-regular games. In Proc. STOC'01, pages 675–683. ACM Press.

Derman, C. (1970). *Finite-State Markovian Decision Processes*. Academic Press.

Emerson, E. and Jutla, C. (1991). Tree automata, mu-calculus and determinacy. In Proc.FOCS, pages 368–377. IEEE Computer Society Press.

Emerson, E. A., Jutla, C. S., and Sistla, A. P. (1993). On model-checking for fragments of mu-calculus. In Courcoubetis, C., editor, Proc. CAV, LNCS 697, pages 385–396.

Filar, J. and Vrieze, K. (1997). *Competitive Markov Decision Processes*. Springer.

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

Jurdzinski, M. (2000). Small progress for solving parity games. In *Proc. STACS*, volume 1770 of *LNCS*, pages 290–301.

Jurdzinski, M., Kupferman, O., and Henzinger, T. (2003). Trading probability for fairness. In Proc.CSL, LNCS 2471, pages 292–305.

Mitchell, J. C. (2001). Probabilistic polynomial-time process calculus and security protocol analysis. In Proc. ESOP LNCS 2028, pages 23–29.

Pnueli, A. and Zuck, L. (1986). Verification of multiprocess probabilistic protocols. *Distributed Computing*, 1:53–72.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY.

Thomas, W. (1990). Automata on infinite objects. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science*, volume B, chapter 4, pages 133–191. Elsevier Science Publishers

Thomas, W. (2003). Infinite games and verification In Poc. CAV LNCS 2725, pages 58–64.

Vardi, M. Y. (1985). Automatic verification of probabilistic concurrent finite-state programs. In Proc. FOCS, pages 327–338, Portland, Oregon. IEEE.

Vöge, J. and Jurdzinski, M. (2000). A discrete strategy improvement algorithm for solving parity games. In Proc. CAV, LNCS 1855, pages 202–215.