On Reduction Criteria for Probabilistic Reward Models

Marcus Größer[a], Gethin Norman[b], Christel Baier[a],
Frank Ciesinski[a], Marta Kwiatkowska[b], David Parker[b]

[a] *Universität Bonn, Institut für Informatik I, Germany*
{baier | groesser | ciesinsk}@cs.uni-bonn.de
[b] *University of Birmingham, School of Computer Science, Edgbaston, United Kingdom*
{kwiatkowska | norman | parker}@cs.bham.ac.uk⋆

Abstract. In recent papers, the partial order reduction approach has been adapted to reason about the probabilities for temporal properties in concurrent systems with probabilistic behaviours. This paper extends these results by presenting reduction criteria for a probabilistic branching time logic that allows specification of constraints on quantitative measures given by a reward or cost function for the actions of the system.

1 Introduction

Partial order reduction [13, 25, 32] is one of the most prominent techniques for tackling the state explosion problem for concurrent software systems. It has been implemented in many tools and successfully applied to a large number of case studies, see e.g. [17]. Recently, the ample-set method [24] has been extended for concurrent probabilistic systems, both in the setting of quantitative linear time [5, 7] and branching time [4] properties. The underlying models used in this work are Markov decision processes (MDPs), an extension of transition systems where nondeterminism can be used e.g. to model the interleaving of concurrent activities, to represent the interface with an unknown system environment or for abstraction purposes, and where probability serves e.g. to model coin tossing actions or to specify the frequency of exceptional (faulty) behaviour (such as losing messages from a buffer). Thus, MDPs arise as natural operational models for randomized distributed algorithms and communication or security protocols and are widely used in model checking. Equipped with reward or cost functions MDPs are also standard models in many other areas, such as operations research, reinforcement learning and robot path planning. In those fields a lot work has already been done on reducing the state (and/or actions) space via aggregating states (and/or actions) [2, 29, 11]. Opposed to many results in the field of machine learning that yield only approximations to optimal solutions, the results in the field of model checking offer some work on exact process equivalences, like (weak) bisimulation. Contrary to those approaches that rely on partition refinement and need global knowledge of the state space, the approach with partial order reduction can be implemented with local conditions and therefore be intertwined with the state space search on-the-fly, provided an appropriate high-level representation of the system is given.

⋆ Supported in part by EPSRC grants GR/S11107 and GR/S46727

The contribution of this paper is reduction criteria which are shown to be sound for an extension of probabilistic computation tree logic (PCTL) [6] that serves to reason about rewards or costs. Our logic, called PCTL$_r$, essentially agrees with the logic suggested by de Alfaro [9, 8]. (PCTL$_r$ is also similar to the logic PRCTL [1, 23] which relies on a Markov chain semantics, while PCTL$_r$-formulae are interpreted over MDPs.) PCTL$_r$ allows specifications regarding e.g. the packet loss characteristics of a queueing system, the energy consumption, or the average number of unsuccessful attempts to find a leader in a distributed system. We first explain how the ample-set conditions suggested in [4] for PCTL can be modified to treat reward-based properties specified in PCTL$_r$ and then identify a fragment of PCTL$_r$ (which still contains a wide range of non-trivial reward properties) where the weaker criteria of [4] are sufficient. We also present results on a new logic PCTL$_c$, that treats the rewards with a discounting semantics. As in the case of previous publications on partial order reduction for probabilistic systems, the major difficulty was to provide the proof of correctness. The general proof technique follows the line of [12, 4] by establishing a bisimulation between the full and the reduced system. However, we depart here from these approaches by introducing a new variant of bisimulation equivalence for MDPs which borrows ideas from [21, 31] and relies on the concept of norm functions [22, 14]. This new type of bisimulation equivalence preserves PCTL$_r$-properties and might be useful also for other purposes.

Organization of the paper. Section 2 summarizes the basic definitions concerning Markov decision processes, reward structures and PCTL$_r$. Section 2 also recalls the partial order reduction approach for MDPs without reward structure and PCTL of [4] which we then extend to reason about rewards in Section 3. Section 4 identifies a class of reward-based properties that are preserved when using the weaker conditions of [4]. In Section 5 we discuss our approach in the setting of discounted rewards and Section 6 concludes the paper.

2 Preliminaries

Markov decision processes (MDPs), see e.g. [27]. An MDP is a tuple $\mathcal{M} = (S, Act, \mathbf{P}, s_{\text{init}}, \mathsf{AP}, L, \mathsf{rew})$ where S is a finite state space, $s_{\text{init}} \in S$ is the initial state, Act a finite set of actions, AP a set of atomic propositions, $L : S \to 2^{\mathsf{AP}}$ a labelling function, $\mathbf{P} : S \times Act \times S \to [0, 1]$ the three-dimensional transition probability matrix such that $\sum_{u \in S} \mathbf{P}(s, \alpha, u) \in \{0, 1\}$ for all states s and actions α, and a function rew that assigns to each action $\alpha \in Act$ a reward $\mathsf{rew}(\alpha) \in \mathbb{R}$.

Action α is called *enabled* in state s if $\sum_{u \in S} \mathbf{P}(s, \alpha, u) = 1$. We write $Act(s)$ for the set of actions that are enabled in s. The states t with $\mathbf{P}(s, \alpha, t) > 0$ are called α-successors of s. For technical reasons, we require that $Act(s) \neq \emptyset$ for all states s. Action α is called a *stutter action* iff for all $s \in S$ where α is enabled in s, $L(s) = L(u)$ for all α-successors u of s. That is, stutter actions do not change the state labelling. Action α is called *non-probabilistic* iff for all states s, there is at most one α-successor. That is, if α is enabled in s then there is a state s_α with $\mathbf{P}(s, \alpha, s_\alpha) = 1$, while $\mathbf{P}(s, \alpha, u) = 0$ for all other states u. In particular, if $\alpha \in Act(s)$ is a non-probabilistic stutter action then $L(s) = L(s_\alpha)$.

An infinite *path* in an MDP is a sequence $\varsigma = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \cdots$ such that $\alpha_i \in Act(s_{i-1})$ and $\mathbf{P}(s_{i-1}, \alpha_i, s_i) > 0$ for all $i \geq 1$. We denote by *first*$(\varsigma) = s_0$ the starting state of ς and write $state(\varsigma,i)$ for the $(i+1)$th state in ς and $\rho(\varsigma,i)$ for the cumulative reward obtained through the first i actions. That is, if ς is as above then $state(\varsigma,i) = s_i$ and $\rho(\varsigma,i) = \mathsf{rew}(\alpha_1 \dots \alpha_i)$ where $\mathsf{rew}(\alpha_1 \dots \alpha_i) = \mathsf{rew}(\alpha_1) + \cdots + \mathsf{rew}(\alpha_i)$. If $T \subseteq S$ is a set of states then $\mathsf{Rew}(\varsigma,T)$ denotes the reward that is earned until a T-state is visited the first time. Formally, if $state(\varsigma,i) \in T$ and $state(\varsigma,j) \notin T$ for all $j < i$ then $\mathsf{Rew}(\varsigma,T) = \rho(\varsigma,i)$. If $state(\varsigma,i) \notin T$ for all $i \geq 0$ we set $\mathsf{Rew}(\varsigma,T) = \infty$. Finite paths (denoted by σ) are finite prefixes of infinite paths that end in a state. We use the notations $first(\sigma)$, $state(\sigma,i)$ and $\rho(\sigma,i)$ as for infinite paths and $|\sigma|$ for the length (number of actions). $Paths_{fin}(s)$ (resp. $Paths_{\omega}(s)$) denotes the set of all finite (resp. infinite) paths of $\mathcal{M}$ with $first(\cdot) = s$. Given a path $\varsigma = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \cdots$ we denote by $trace(\varsigma) = L(s_0), L(s_1), L(s_2), \dots$ the word over the alphabet 2^{AP} obtained by the projection of ς to the state labels. Two infinite paths ς_1 and ς_2 in an MDP are called *stutter equivalent* iff there is an infinite word $\ell_1, \ell_2, \dots$ over the alphabet 2^{AP} such that $trace(\varsigma_1) = \ell_1^{k_1}, \ell_2^{k_2}, \dots$ and $trace(\varsigma_2) = \ell_1^{n_1}, \ell_2^{n_2}, \dots$ where k_i, $n_i \geq 1$.

A *scheduler*, also often called policy, strategy or adversary, denotes an instance that resolves the nondeterminism in the states, and thus yields a Markov chain and a probability measure on the paths. We shall use here *history-dependent randomized schedulers* in the classification of [27]. They are defined as functions A that take as input a finite path σ and return a distribution over the actions $\alpha \in Act(last(\sigma))$.[1] A scheduler A is called deterministic if it chooses a unique action (with probability 1) for all finite paths. An A-path denotes an infinite or finite path σ that can be generated by A. Given a state s and a scheduler A, the behaviour of $\mathcal{M}$ under A can be formalised by a (possibly infinite-state) Markov chain. $\Pr^{A,s}$ denotes the standard probability measure on the Borel field of the infinite A-paths ς with $first(\varsigma) = s$. If $T \subseteq S$ then $\mathbb{E}^{A,s}(\Diamond T)$ denotes the expected value under A with starting state s for the random function $\varsigma \mapsto \mathsf{Rew}(\varsigma,T)$. Recall that $\mathsf{Rew}(\varsigma,T)$ denotes the reward that is earned by the prefix of ς that leads from the starting state s to a state in T and that $\mathsf{Rew}(\varsigma,T)$ equals ∞ if ς does not reach T. Thus, if there is a positive probability of not reaching T under scheduler A (from state s), then $\mathbb{E}^{A,s}(\Diamond T) = \infty$. If $s = s_{\text{init}}$ we simply write $\Pr^A$ and $\mathbb{E}^A$.

Probabilistic computation tree logic. PCTL is a probabilistic variant of CTL which has been introduced first for Markov chains [15] and then for Markovian models with nondeterminism [6, 31]. We follow here the approach of de Alfaro [9, 8] and extend PCTL with an operator $\mathcal{R}$ to reason about expected rewards. As partial order reduction relies on identifying stutter equivalent paths which might be distinguishable by the next step operator, we do not include the next step operator in the logic. PCTL$_r$-state formulae are therefore given by the grammar:

$$\Phi ::= \mathsf{true} \;\big|\; \mathsf{a} \;\big|\; \Phi \wedge \Phi \;\big|\; \neg\Phi \;\big|\; \mathcal{P}_J(\Phi_1 U_I \Phi_2) \;\big|\; \mathcal{R}_I(\Phi)$$

Here, $\mathsf{a} \in \mathsf{AP}$ is an atomic proposition, $J \subseteq [0,1]$ is a probability interval and $I \subseteq \mathbb{R} \cup \{-\infty, \infty\}$ a reward interval. We refer to the terms $\Phi_1 U_I \Phi_2$ as PCTL$_r$-path formulae. U_I denotes the standard until operator with a reward bound. The meaning of the

[1] By a distribution on a finite set X we mean a function $\nu : X \to [0,1]$ such that $\sum_{x \in X} \nu(x) = 1$ and refer to $\nu(x)$ as the probability for x.

path formula $\varphi = \Phi_1 U_I \Phi_2$ is that a Φ_2-state will be reached via a finite path σ where the cumulative reward is in I, while all states in σ, possibly except the last one, fulfil Φ_1. The state formula $\mathcal{P}_J(\varphi)$ holds for state s if for each scheduler A the probability measure of all infinite paths starting in s and fulfilling the path formula φ meets the probability bound given by J. On the other hand, $\mathcal{R}_I(\Phi)$ asserts that for any scheduler A the expected reward that is earned until a Φ-state has been reached meets the reward bound given by I. For instance, $\mathcal{R}_{[0,17]}(\mathsf{goal})$ asserts that independent of the scheduling policy the average costs to reach a goal state do not exceed 17. The formula $\mathcal{P}_{(0.9,1]}(\mathsf{true}\ U_{[0,4]}\ \mathsf{delivered})$ requires that the probability of a message being delivered with at most 4 retransmissions is greater than 0.9.

If $\mathcal{M}$ is an MDP and s a state in $\mathcal{M}$ then we write $s \models \Phi$ to denote that state-formula Φ holds in state s, and similarly, $\varsigma \models \varphi$ to denote that path formula φ holds for the infinite path ς. The formal semantics of the propositional logic fragment is standard and the semantics of the $\mathcal{P}$- and $\mathcal{R}$-operator is formalised by :

$$\begin{array}{lll} s \models \mathcal{P}_J(\Phi_1 U_I \Phi_2) & \Leftrightarrow & \text{for all schedulers } A\text{:}\,\Pr^{A,s}\left\{\varsigma \in Paths_\omega(s) : \varsigma \models \Phi_1 U_I \Phi_2\right\} \in J \\ s \models \mathcal{R}_I(\Phi) & \Leftrightarrow & \text{for all schedulers } A\text{:}\,\mathbb{E}^{A,s}(\Diamond Sat(\Phi)) \in I \end{array}$$

If $\varsigma = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \cdots$ then $\varsigma \models \Phi_1 U_I \Phi_2$ iff $\exists i \geq 0$ s.t. $s_i \models \Phi_2 \ \wedge\ \rho(\varsigma, i) \in I \wedge \forall j < i.\ s_j \models \Phi_1$. The satisfaction set of Φ in $\mathcal{M}$ is $Sat(\Phi) = \left\{s \in S : s \models \Phi\right\}$. State formula Φ is said to hold for an MDP if the initial state satisfies Φ.

Note that one could also give the $\mathcal{R}_I$ operator a different semantics as follows. $s \models \mathcal{R}_I(\Phi)$ if and only if for all schedulers A, such that the probability to reach $Sat(\Phi)$ from s equals 1, it holds that $\mathbb{E}^{A,s}(\Diamond Sat(\Phi)) \in I$. But this is irrelevant for our purposes.

Derived operators. Other Boolean connectives, such as disjunction $\vee$, implication $\rightarrow$, can be derived as usual. The temporal operator eventually $\Diamond$ is obtained in the standard way by $\Diamond_I \Phi = \mathsf{true}\ U_I \Phi$. The always-operator can be derived as in PCTL by the duality of lower and upper probability bounds. For the trivial reward-interval $I = (-\infty, \infty)$, we obtain the standard eventually, always and until operator. We simply write U, $\Diamond$ and $\Box$ rather than $U_{(-\infty,\infty)}$, $\Diamond_{(-\infty,\infty)}$ and $\Box_{(-\infty,\infty)}$, respectively.

PCTL denotes the sublogic of PCTL$_r$ that does not use the $\mathcal{R}$-operator and where the path-formulae have the trivial reward interval. Since the reward structure is irrelevant for PCTL-formulae, they can be interpreted over MDPs without reward structure.

The ample set method for PCTL [4]. Before presenting the partial order reduction citeria for PCTL$_r$ in Section 3, we briefly summarize the results of [4] for applying the ample-set method to PCTL model checking. The starting point is an MDP $\mathcal{M} = (S, Act, \mathbf{P}, s_{\text{init}}, \mathsf{AP}, L)$, without reward structure, to be verified against a PCTL-formula. Following Peled's ample-set method [24], the idea is to assign to any reachable state s a nonempty action-set $ample(s) \subseteq Act(s)$ and to construct a reduced MDP $\hat{\mathcal{M}}$ by using the action-sets $ample(s)$ instead of $Act(s)$. Formally, given a function $ample : S \rightarrow 2^{Act}$ with $\emptyset \neq ample(s) \subseteq Act(s)$ for all states s, the state space of the reduced MDP $\hat{\mathcal{M}} = (\hat{S}, Act, \hat{\mathbf{P}}, s_{\text{init}}, \mathsf{AP}, \hat{L})$ induced by $ample$ is the smallest set $\hat{S} \subseteq S$ that contains s_{init} and any state u where $\mathbf{P}(s, \alpha, u) > 0$ for some $s \in \hat{S}$ and $\alpha \in ample(s)$. The labelling function $\hat{L} : \hat{S} \rightarrow 2^{\mathsf{AP}}$ is the restriction of the original labelling function L to the state-set $\hat{S}$. The

A1 **(Stutter-condition)** If $ample(s) \neq Act(s)$ then all actions $\alpha \in ample(s)$ are stutter actions.

A2 **(Dependence-condition)** For each path $\sigma = s \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} s_n \xrightarrow{\gamma} \cdots$ in $\mathcal{M}$ where γ is dependent on $ample(s)$ there exists an index $i \in \{1, \ldots, n\}$ such that $\alpha_i \in ample(s)$.

A3 **(Cycle-condition)** On each cycle $s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_n} s_n = s$ in $\hat{\mathcal{M}}$ there exists a state s_i which is fully expanded, i.e., $ample(s_i) = Act(s_i)$.

A4 **(Branching condition)** If $ample(s) \neq Act(s)$ then $ample(s)$ is a singleton consisting of a non-probabilistic action.

Fig. 1. Conditions for the ample-set method for PCTL [4]

transition probability matrix of $\hat{\mathcal{M}}$ is given by $\hat{\mathbf{P}}(s,\alpha,t) = \mathbf{P}(s,\alpha,t)$ if $\alpha \in ample(s)$ and 0 otherwise. State s is called fully expanded if $ample(s) = Act(s)$.

The main ingredient of any partial order reduction technique in the non-probabilistic or probabilistic setting is an adequate notion for the independence of actions. The definition for the independence of actions α and β in the composed transition system (which captures the semantics of the parallel composition of all processes that run in parallel) relies on recovering the interleaving 'diamonds'. Formally, two distinct actions α and β are called *independent* (in $\mathcal{M}$) iff for all states $s \in S$ with $\{\alpha, \beta\} \subseteq Act(s)$, (I1) $\alpha \in Act(u)$ for each β-successor u of s, (I2) $\beta \in Act(u)$ for each α-successor u of s, and (I3) $\mathbf{P}(s,\alpha\beta,w) = \mathbf{P}(s,\beta\alpha,w)$ for all $w \in S$ where $\mathbf{P}(s,\gamma\delta,w) = \sum_{u \in S} \mathbf{P}(s,\gamma,u) \cdot \mathbf{P}(u,\delta,w)$ for $\gamma, \delta \in Act$. Two different actions α and β are called *dependent* iff α and β are not independent. If $D \subseteq Act$ and $\alpha \in Act \setminus D$ then α is called independent of D iff for all actions $\beta \in D$, α and β are independent. Otherwise, α is called dependent on D.

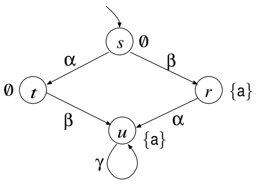
To preserve PCTL properties, [4] use the four conditions in Fig. 1. These rely on a slight modification of the conditions by Gerth et al [12] for preserving CTL-properties and can be implemented in an on-the-fly state space exploration [25, 3].

Theorem 1 **([4]).** *If (A1)-(A4) hold then $\mathcal{M}$ and $\hat{\mathcal{M}}$ fulfil the same* PCTL*-formulae.*

3 Reduction Criteria for Rewards

In the sequel, we assume that we are given an MDP $\mathcal{M}$ and discuss the partial order reduction approach for properties specified in PCTL_r. We first show that (A1)-(A4) are not sufficient to preserve PCTL_r properties with nontrivial reward bounds. To treat full PCTL_r, we shall need a modification of the branching condition (A4).

Example 1. We begin with a simple example illustrating that (A1)-(A4) cannot ensure that all PCTL_r-formulae are preserved. Consider the following MDP with the actions α, β, γ that are all non-probabilistic and where $\mathsf{rew}(\alpha) = \mathsf{rew}(\beta) = \mathsf{rew}(\gamma) = 1$.



Since α and β are independent and α is a stutter action, (A1)-(A4) allow for a reduction obtained through *ample*$(s)=\{\alpha\}$. Thus, $\hat{S}=\{s,t,u\}$. Consider the PCTL$_r$ formula $\Phi=\mathcal{R}_{[2,\infty)}(\mathsf{a})$. Then, the reduced system $\hat{\mathcal{M}}$ satisfies Φ, while the original system $\mathcal{M}$ does not, because $\mathcal{M}$ might choose action β in s which yields the expected reward 1 to reach an a-state. □

We now discuss how to strengthen conditions (A1)-(A4) such that reward-based properties are preserved. We start with some simple observations. First, as $\hat{\mathcal{M}}$ is a sub-MDP of the original system $\mathcal{M}$, any scheduler A for $\hat{\mathcal{M}}$ is also a scheduler for $\mathcal{M}$. Thus:

Lemma 1. *Let Φ_1,Φ_2 be PCTL$_r$-formulae with $Sat_{\mathcal{M}}(\Phi_i)\cap\hat{S}=Sat_{\hat{\mathcal{M}}}(\Phi_i)$, $i=1,2$.*
(i) $\mathcal{M}\models\mathcal{R}_I(\Phi_1) \Rightarrow \hat{\mathcal{M}}\models\mathcal{R}_I(\Phi_1)$,
(ii) $\mathcal{M}\models\mathcal{P}_J(\Phi_1 U_I \Phi_2) \Rightarrow \hat{\mathcal{M}}\models\mathcal{P}_J(\Phi_1 U_I \Phi_2)$.

The converse directions in Lemma 1 do not hold in general as $\mathcal{M}$ might have "more" schedulers than $\hat{\mathcal{M}}$. To get a feeling of how to modify the reduction criteria for PCTL$_r$, let us first give some informal explanations. In [4], the soundness proof of (A1)-(A4) for PCTL establishes a kind of bisimulation between the full MDP $\mathcal{M}$ and the reduced MDP $\hat{\mathcal{M}}$ which allows one to transform any scheduler A for $\mathcal{M}$ into a scheduler B for $\hat{\mathcal{M}}$ such that A and B yield the same probabilities for PCTL-path formulae. As in the case of the ample-set method for verifying linear time properties (where (A1)-(A3) and a weaker form of (A4) are sufficient [5, 7]) this scheduler-transformation yields a transformation of the A-paths into "corresponding" B-paths. Let us look at this path-transformation "path ς in $\mathcal{M} \rightsquigarrow$ path $\hat{\varsigma}$ in $\hat{\mathcal{M}}$" which, in fact, is already known from the non-probabilistic case [24]. The path $\hat{\varsigma}$ in $\hat{\mathcal{M}}$ is obtained through a sequence of paths $\varsigma_0,\varsigma_1,\varsigma_2,\ldots$ in $\mathcal{M}$ such that the first i-steps in ς_i and ς_{i+1} agree and are composed of transitions in $\hat{\mathcal{M}}$. The switch from ς_i to ς_{i+1} is performed as follows.

Let $\pi=s_1\xrightarrow{\alpha_1}s_2\xrightarrow{\alpha_2}\cdots$ be the suffix of ς_i starting with the $(i{+}1)$th step (by the above, s_1 is a state in $\hat{\mathcal{M}}$). Our goal is to construct a stutter equivalent path $\hat{\pi}$ from s_1 that starts with an action in *ample*(s_1). We then may compose the prefix of ς_i from *first*(ς_i) to s_1 with $\hat{\pi}$ to obtain the path ς_{i+1}. If $\alpha_1\in$ *ample*(s_1) then we may put $\pi=\hat{\pi}$. Let us now assume that $\alpha_1\notin$ *ample*(s_1). Then, by (A4), *ample*(s_1) consists of a single non-probabilistic action.

(T1) If there is some index $j\geq 2$ such that $\alpha_j\in$ *ample*(s_1) then choose the smallest such index j and replace the action sequence $\alpha_1\ldots\alpha_{j-1}\alpha_j\alpha_{j+1}\ldots$ with $\alpha_j\alpha_1\ldots\alpha_{j-1}\,\alpha_{j+1}\ldots$. This is possible since by (A2) the actions $\alpha_1,\ldots,\alpha_{j-1}$ are independent of α_j. The resulting path $\hat{\pi}$ is stutter-equivalent to π by condition (A1).
(T2) If $\alpha_j\notin$ *ample*(s_1) for all $j\geq 1$ and *ample*$(s_1)=\{\beta\}$ then replace the action sequence $\alpha_1\alpha_2\ldots$ with $\beta\alpha_1\alpha_2\ldots$. Again, (A2) ensures that each α_j is independent of β. (A1) yields the stutter-equivalence of π and the resulting path $\hat{\pi}$.

Note, that the insertion of the additional action in transformation (T2) possibly changes the cumulative reward. Since we are interested in the cumulative reward that is gained until a certain state labelling is reached, the action permutation in transformation (T1)

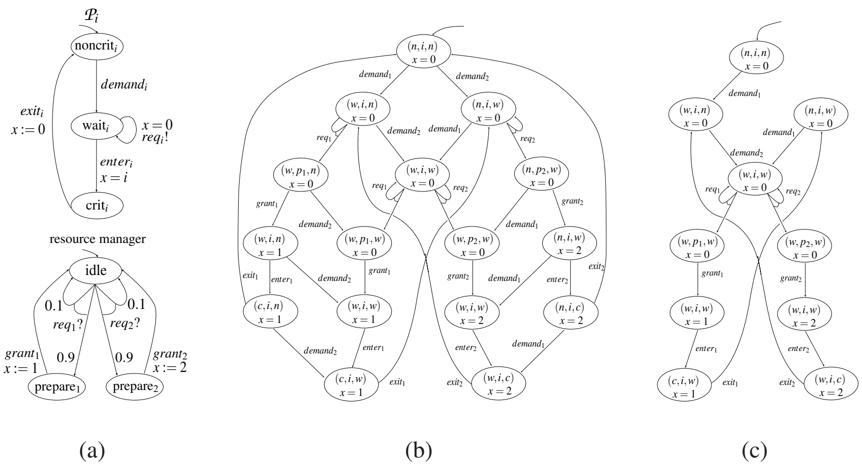


Fig. 2. Mutual exclusion example: (a) components, (b) full system and (c) reduced MDP

possibly changes this reward, as can be seen in Example 1 (note that a stutter action is permuted to the front of the action sequence).

To establish the equivalence of $\mathcal{M}$ and $\hat{\mathcal{M}}$ for PCTL$_r$ it seems to be sufficient to ensure that, in transformation (T2), the additional action β has zero reward, and in transformation (T1), the stutter action α_j, that is permuted to the front of the action sequence, has zero reward. This motivates the following stronger branching condition:

A4$'$ (New branching condition) If $ample(s) \neq Act(s)$ then $ample(s) = \{\beta\}$ for some non-probabilistic action with $\mathsf{rew}(\beta) = 0$.

Theorem 2. *If (A1)-(A3), (A4$'$) hold then $\mathcal{M}$ and $\hat{\mathcal{M}}$ satisfy the same PCTL$_r$ formulae.*

Example 2. To illustrate our approach we consider a simple mutual exclusion protocol in which the processes P_1 and P_2 attempt to access a common resource controlled by a resource manager. A shared variable x is used to guarantee mutual exclusion and we assume that the communication is unreliable (requests to the resource manager are corrupted/lost with probability 0.1). Fig. 2(a) presents the different components of the system. Associating a reward of 1 with the actions req_1 and req_2 and 0 with all other actions, using PCTL$_r$ one can, for example, specify:

- $R_{\leq 1.4}(\mathsf{crit}_1 \vee \mathsf{crit}_2)$: the expected number of requests before a process enters the critical section is at most 1.4;
- $\mathcal{P}_{>0.7}(\mathsf{true}\ U_{[0,6]}\ \mathsf{crit}_1 \vee \mathsf{crit}_2)$: the probability that a process enters its critical section after at most 6 requests have been issued is strictly greater than 0.7.

Fig. 2(b) gives the full MDP for the system and (assuming $\mathsf{AP} = \{\mathsf{crit}_1, \mathsf{crit}_2\}$) one can construct the reduced system given in Fig. 2(c) satisfying conditions (A1)-(A4$'$). $\square$

4 Preservation Result for (A1)-(A4) and Reward-Based Properties

We now turn to the question of which properties with nontrivial reward bounds are preserved by (A1)-(A3) and the original branching condition (A4) in Fig. 1. Let us again look at the path transformation described in (T1) and (T2) where, given a path π in $\mathcal{M}$ a path $\hat{\pi}$ is generated, where either the action sequence of $\hat{\pi}$ is a permutation of the action sequence of π (T1) or $\hat{\pi}$ starts with a non-probabilistic stutter action and then performs the same action sequence as the original path π (T2). As the rewards are in $\mathbb{R}$ we do not know, how the cumulative reward of $\hat{\pi}$ has changed compared to that of π. If we however require that the rewards of all actions are *non-negative*, along the modified path $\hat{\pi}$ a reward equal or greater will be earned than that along π. This yields an informal explanation why the additional power of $\mathcal{M}$ can lead to smaller minimal expected rewards, but the maximal expected rewards agree in $\mathcal{M}$ and $\hat{\mathcal{M}}$. Similarly, we might expect that the minimal probabilities for events of the form $\mathsf{a}_1 U_{[0,r]} \mathsf{a}_2$ agree under $\mathcal{M}$ and $\hat{\mathcal{M}}$. The same holds for maximal probabilities for events of the form $\Box_{[0,r]}\mathsf{a}$. This motivates the definition of the following sublogic of PCTL$_r$.

Let PCTL$_r^-$ be the sublogic of PCTL$_r$ which only uses the $\mathcal{R}$-operator with upper reward bounds, i.e., formulae of the form $\mathcal{R}_{[0,r]}(\Phi)$, and where the probabilistic operator is only used in combination with PCTL-path formulae $\Phi_1 U \Phi_2$ or with the until-operator in combination with upper reward and lower probability bounds or in combination with lower reward and upper probability bounds or with the always-operator in combination with upper reward and upper probability bounds or in combination with lower reward and lower probability bounds, e.g. $\mathcal{P}_{[0,p]}(\Box_{[0,r]}\Phi)$ or $\mathcal{P}_{(p,1]}(\Phi_1 U_{[0,r]} \Phi_2)$. Note that PCTL is contained in PCTL$_r^-$. (The result stated in Theorem 3 would still hold when dealing with a release- or weak until operator rather than the always-operator.)

Theorem 3. *If (A1)-(A4) hold and* $\mathsf{rew}(\alpha) \geq 0$ *for all* $\alpha \in Act$ *then* $\mathcal{M}$ *and* $\hat{\mathcal{M}}$ *satisfy the same* PCTL$_r^-$ *formulae.*

Proof. (*sketch*) As is the case for many other types of (bi)simulation relations for probabilistic systems, our notion of bisimulation equivalence will use the concept of *weight functions* [18, 19]. Let S, S' be finite sets and $R \subseteq S \times S'$. If ν and ν' are distributions on S and S' respectively then a weight function for (ν, ν') with respect to R denotes a function $w : S \times S' \to [0,1]$ such that $\{(s,s') : w(s,s') > 0\} \subseteq R$, $\sum_{u' \in S'} w(s,u') = \nu(s)$ and $\sum_{u \in S} w(u,s') = \nu'(s')$ for all $s \in S$, $s' \in S'$. We write $\nu \sqsubseteq_R \nu'$ iff there exists a weight function for (ν, ν') with respect to R and refer to $\sqsubseteq_R$ as the lifting of R to distributions.

Definition 1 (Normed (bi)simulation). Let $\mathcal{M} = (S_{\mathcal{M}}, Act, \mathbf{P}_{\mathcal{M}}, s_{\text{init}}^{\mathcal{M}}, \mathsf{AP}, L_{\mathcal{M}}, \mathsf{rew})$ and $\mathcal{N} = (S_{\mathcal{N}}, Act, \mathbf{P}_{\mathcal{N}}, s_{\text{init}}^{\mathcal{N}}, \mathsf{AP}, L_{\mathcal{N}}, \mathsf{rew})$ be two MDPs with the same set of atomic propositions, the same action set Act and the same reward structure $\mathsf{rew} : Act \to \mathbb{R}_{\geq 0}$. A normed reward simulation for $(\mathcal{M}, \mathcal{N})$ with respect to rew is a triple (R, η_1, η_2) consisting of a binary relation $R \subseteq S_{\mathcal{M}} \times S_{\mathcal{N}}$ and functions $\eta_1, \eta_2 : R \to \mathbb{N}$ such that $(s_{\text{init}}^{\mathcal{M}}, s_{\text{init}}^{\mathcal{N}}) \in R$ and for each pair $(s,s') \in R$ the following conditions hold.

(N1) $L_{\mathcal{M}}(s) = L_{\mathcal{N}}(s')$
(N2) If $\alpha \in Act_{\mathcal{M}}(s)$ then at least one of the following conditions holds:
(N2.1) α is enabled in s' (i.e., $\alpha \in Act_{\mathcal{N}}(s')$) and $\mathbf{P}_{\mathcal{M}}(s, \alpha, \cdot) \sqsubseteq_R \mathbf{P}_{\mathcal{N}}(s', \alpha, \cdot)$,

(N2.2) α is a non-probabilistic stutter action s. th. $(s_\alpha, s') \in R$ and $\eta_1(s_\alpha, s') < \eta_1(s, s')$.

(N2.3) There is a non-probabilistic stutter action $\beta \in Act_{\mathcal{N}}(s')$ with $(s, s'_\beta) \in R$ and $\eta_2(s, s'_\beta) < \eta_2(s, s')$.

A normed bisimulation for $(\mathcal{M}, \mathcal{N})$ is a tuple $(R, \eta_1, \eta_2, \eta_1^-, \eta_2^-)$ such that (R, η_1, η_2) and $(R^{-1}, \eta_1^-, \eta_2^-)$ are normed simulations for $(\mathcal{M}, \mathcal{N})$, resp. $(\mathcal{N}, \mathcal{M})$. □

We write $\mathcal{M} \approx_{nb} \mathcal{N}$ iff there exists a normed bisimulation for $\mathcal{M}$ and $\mathcal{N}$.

A forming path from s to $\hat{s}$ means a path $s = s_0 \xrightarrow{\beta_0} s_1 \xrightarrow{\beta_1} \cdots \xrightarrow{\beta_{n-1}} s_n = \hat{s}$ where $\beta_0, \ldots, \beta_{n-1}$ are non-probabilistic stutter actions, and for $0 \le i < n$, the singleton action-set $\{\beta_i\}$ fulfils the dependence condition (A2) for state s_i. A shortest forming path from s to $\hat{s}$ means a forming path from s to $\hat{s}$ where the cumulative reward is minimal under all forming paths from s to $\hat{s}$ and where the length (number of actions) is minimal under all forming paths with minimal cumulative reward. We will write $\mu(s, \hat{s})$ for the cumulative reward of all/some shortest forming path from s to $\hat{s}$. $s \rightsquigarrow \hat{s}$ denotes the existence of a forming path from s to $\hat{s}$ and we put $R = \{(s, \hat{s}) \in S \times \hat{S} : s \rightsquigarrow \hat{s}\}$.

If $(s, \hat{s}) \in R$ then

$$\Pr^{A,s}(\Pi(s, r+\mu(s,\hat{s}), C_1, \ldots, C_n)) \ge \Pr^{B,\hat{s}}(\Pi(\hat{s}, r, C_1, \ldots, C_n)) \qquad (*)$$

and $\Pr^{A,s}(\Pi(s, C_1, \ldots, C_n)) = \Pr^{B,\hat{s}}(\Pi(\hat{s}, C_1, \ldots, C_n))$. Here, we used the following notation. Let $u \in S$, $C_1, C_2, \ldots, C_n$ be a sequence of $\approx_{nb}$-equivalence classes with $C_i \ne C_{i+1}$ for $1 \le i < n$ and $r \ge 0$. Then, $\Pi(u, r, C_1, \ldots, C_n)$ denotes the set of all infinite paths that have a finite prefix of the form $u_0 \to^*_{C_1} \tilde{u}_1 \xrightarrow{\gamma_1} u_2 \to^*_{C_2} \tilde{u}_2 \xrightarrow{\gamma_2} \cdots \xrightarrow{\gamma_{n-2}} u_{n-1} \to^*_{C_{n-1}} \tilde{u}_{n-1} \xrightarrow{\gamma_{n-1}} u_n$ where $u_0 = u$ and the total reward is $\le r$ and $u_n \in C_n$. The actions γ_i are arbitrary. In this context, $v \to^*_C \tilde{v}$ means a finite path built out of non-probabilistic stutter actions such that v, $\tilde{v}$ and all intermediate states of that path belong to C. $\Pi(u, C_1, \ldots, C_n)$ stands for the union of the path-sets $\Pi(u, r, C_1, \ldots, C_n)$ for arbitrary $r \ge 0$. For $s = s_{\text{init}} = \hat{s}$ we have $\mu(s, \hat{s}) = \mu(s_{\text{init}}, s_{\text{init}}) = 0$.

The above yields that for each scheduler A for $\mathcal{M}$ there exists a scheduler B for $\hat{\mathcal{M}}$ such that $\Pr^A(\Pi(s_{\text{init}}, r, C_1, \ldots, C_n)) \ge \Pr^B(\Pi(s_{\text{init}}, r, C_1, \ldots, C_n))$ for all $r \ge 0$ and all $\approx_{nb}$-equivalence classes $C_1, \ldots, C_n$. From this we can derive that $\mathcal{M}$ and $\hat{\mathcal{M}}$ fulfil the same PCTL$_r^-$ formulae. □

Example 3. Let us return to Example 2 and redefine the rewards such that the only nonzero rewards are for actions *demand*$_1$ and *demand*$_2$ which have reward 1. Now, in this situation the reduced MDP in Fig. 2(c) can no longer be constructed using (A1)-(A4′). However, this construction is still possible under (A1)-(A4).

This is demonstrated by the fact that both the reduced and full MDP satisfy the PCTL$_r^-$ property $\mathcal{R}_{[0,2]}(\mathsf{crit}_1 \vee \mathsf{crit}_2)$ (the maximum expected number of processes that can attempt to enter the critical section before one of them does so is at most 2), while only the reduced model satisfies the PCTL$_r$ property $\mathcal{R}_{[2,\infty)}(\mathsf{crit}_1 \vee \mathsf{crit}_2)$ (the minimum expected number is at least 2). □

5 Reward Properties w.r.t Discounted Rewards

In many research areas (e.g. economics, operations research, control theory) rewards are treated with a different semantics, namely as so-called *discounted rewards* [27], where given a discount factor $0 < c < 1$, the reward of the i-th action of a path is multiplied with c^{i-1}. This interpretation of rewards reflects the fact that a reward (e.g. a payment) in the future is not worth quite as much as it is now (e.g. due to inflation). In this Section we investigate our partial order approach for discounted rewards.

Given a path $\varsigma = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \cdots$ and a discount factor $c \in (0,1)$, we denote by $\rho_c(\varsigma, i) = \mathsf{rew}_c(\alpha_1 \dots \alpha_i) = c^0 \cdot \mathsf{rew}(\alpha_1) + c^1 \cdot \mathsf{rew}(\alpha_2) + \cdots + c^{i-1} \cdot \mathsf{rew}(\alpha_i)$ the cumulative discounted reward obtained through the first i actions.

With this on hand we can define the logic PCTL$_c$, which is a variant of PCTL$_r$. In PCTL$_c$, we use the new operators U_I^c and $\mathcal{R}_I^c$ instead of U_I and $\mathcal{R}_I$, where instead of the cumulative reward $\rho(\varsigma, i)$ the cumulative discounted reward $\rho_c(\varsigma, i)$ is used in the semantics of those new operators. The semantics of the U_I^c operator is as follows. Given a path $\varsigma = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \cdots$, we say that $\varsigma \models \Phi_1 U_I^c \Phi_2$ iff $\exists i \geq 0$ s.t $s_i \models \Phi_2 \wedge \forall j < i : s_j \models \Phi_1 \wedge \rho_c(\varsigma, i) \in I$. Similarly, given a set of states $T \subseteq S$ we denote by $\mathsf{Rew}_c(\varsigma, T)$ the discounted reward that is earned until a T-state is visited the first time. Formally, if $state(\varsigma, i) \notin T$ for all $i \geq 0$ then $\mathsf{Rew}_c(\varsigma, T) = \infty$. If $state(\varsigma, i) \in T$ and $state(\varsigma, j) \notin T$ for all $j < i$ then $\mathsf{Rew}_c(\varsigma, T) = \rho_c(\varsigma, i)$. For $T \subseteq S$ and a scheduler A, $\mathbb{E}_c^{A,s}(\Diamond T)$ denotes the expected value under A with starting state s for the random function $\varsigma \mapsto \mathsf{Rew}_c(\varsigma, T)$. Then $s \models \mathcal{R}_I^c(\Phi)$ iff $\forall$ schedulers A: $\mathbb{E}_c^{A,s}(\Diamond Sat(\Phi)) \in I$.

A simple example shows that theorem 2 does not hold for PCTL$_c$ (even if all rewards are nonnegative). Consider the MDP $\mathcal{M}$ in example 1 on page 5. We assign the following rewards : $\mathsf{rew}(\alpha) = 0, \mathsf{rew}(\beta) = \mathsf{rew}(\gamma) = 1$. Choosing $ample(s) = \{\alpha\}$, conditions (A1)-(A3) and (A4') are satisfied. However, if we consider the formula $\Phi = \mathcal{R}^c_{[0,c]}(a)$, we gain that the reduced system $\hat{\mathcal{M}}$ satisfies Φ while the original system $\mathcal{M}$ does not, because $\mathcal{M}$ might choose action β in state s which yields the expected discounted reward to reach an a-state to be $c^0 \cdot \mathsf{rew}(\beta) = 1 > c$.

The reader should notice that due to the discounting, the transformations (T1) and (T2) described in Section 3 on page 6 change the reward of a given path, even under condition (A4') which requires the ample set of a non-fully expanded state to be a singleton consisting of a non-probabilistic action with zero reward. Nevertheless, the following holds: given an MDP M with only *non-negative* rewards, ample-sets that satisfy (A1)-(A3) and (A4') and a path ς in $\mathcal{M}$, let $\hat{\varsigma}$ be a path that emanates from ς by applying transfomation (T1) or (T2). Then $\rho_c(\hat{\varsigma}, i) \leq \rho_c(\varsigma, i)$. Similarly as in Section 4 this informally explains that the additional power of $\mathcal{M}$ can lead to greater maximal expected rewards, but the minimal expected rewards agree in $\mathcal{M}$ and $\hat{\mathcal{M}}$. Also, the maximal probabilities for events of the form $a_1 U^c_{[0,r]} a_2$ agree under $\mathcal{M}$ and $\hat{\mathcal{M}}$. This motivates the definition of the following sublogic of $PCTL_c$.

Let $PCTL_c^-$ be the sublogic of $PCTL_c$ which uses the $\mathcal{R}^c$ operator only with lower reward bounds (i.e $\mathcal{R}^c_{[r,\infty)}\Phi$) and where the probabilistic operator is only used in combination with PCTL-path formulae $\Phi_1 U \Phi_2$ or with the until-operator in combination with lower reward and lower probability bounds or in combination with upper reward and upper probability bounds or with the always-operator in combination with upper reward and lower probability bounds or in combination with lower reward and upper probability bounds, e.g. $\mathcal{P}_{[0,p]}(\Box_{[r,\infty)}\Phi)$ or $\mathcal{P}_{[0,p]}(\Phi_1 U_{[0,r]} \Phi_2)$. Note that PCTL is contained in PCTL_c^-.

Theorem 4. *If (A1)-(A3) and (A4') hold and* $\mathsf{rew}(\alpha) \geq 0$ *for all* $\alpha \in Act$ *then* $\mathcal{M}$ *and* $\hat{\mathcal{M}}$ *satisfy the same* PCTL_c^- *formulae.*

6 Conclusion

The goal of this paper was to study the theoretical foundations of the ample-set approach for the logic PCTL_r, a variant of PCTL with reward-bounded temporal modalities and an expectation operator. The main results of this paper are that the ample-set conditions presented in [4] for PCTL preserve a class of non-trivial reward-based properties (Theorem 3) and that a slight modification of the conditions of [4] are sufficient to treat full PCTL_r (Theorem 2). The proofs of these results have been established by means of a new notion of weak bisimulation for MDPs which preserves PCTL_r and – since it is simpler than other notions of weak bisimulation equivalence for MDPs – might also be useful for other purposes. Moreover we investigated the logic PCTL_c, a variant of PCTL_r where the rewards are given a discounting semantics. We presented ample-set conditions that preserve a non-trivial subset of PCTL_c properties if all given rewards are non-negative (Theorem 4).

Besides being of theoretical interest, the results of this paper also have a practical impact. First experimental results on the ample set approach for MDPs (without reward structure) with the forthcoming model checker LiQuor [3] show that although the criteria needed for probabilistic systems are stronger than in the non-probabilistic case, good reductions can be obtained. Furthermore, the bottleneck in analysis of probabilistic systems modelled by MDPs are the required techniques for solving linear programs. Since the amount of time required for the construction of the reduced MDP is negligible compared to the running time of linear program solvers, even small reductions can increase the efficiency of the quantitative analysis.

In future work, we plan to integrate the partial order reduction techniques suggested here in the symbolic MTBDD-based model checker PRISM [16] by constructing a syntactic representation of the reduced MDP at compile time, in the style of static partial order reduction [20] which permits a combination of partial order reduction with symbolic BDD-based model checking.

References

1. S. Andova, H. Hermanns, and J.-P. Katoen. Discrete-time rewards model-checked. In *Proc. FORMATS*, volume 2791 of *Lecture Notes in Computer Science*, pages 88–104, 2003.
2. Mehran Asadi and Manfred Huber. Action dependent state space abstraction for hierarchical learning systems. In *Proc. IASTED*, Insbruck, Austria, 2005.
3. C. Baier, F. Ciesinski, and M. Groesser. Quantitative analysis of distributed randomized protocols. In *Proc.FMICS*, 2005.
4. C. Baier, P. D'Argenio, and M. Größer. Partial order reduction for probabilistic branching time. In *Proc. QAPL*, 2005.

5. C. Baier, M. Größer, and F. Ciesinski. Partial order reduction for probabilistic systems. In QEST 2004 [28], pages 230–239.
6. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. FST & TCS*, volume 1026 of *LNCS*, pages 499–513, 1995.
7. P.R. D'Argenio and P. Niebert. Partial order reduction on concurrent probabilistic programs. In QEST 2004 [28], pages 240–249.
8. L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.
9. L. de Alfaro. Temporal logics for the specification of performance and reliability. In *Proc. STACS*, volume 1200 of *Lecture Notes in Computer Science*, pages 165–179, 1997.
10. L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *Proc. 13th LICS*, IEEE Press, pages 454–465, 1998.
11. Thomas Dean, Robert Givan, and Sonia Leach. Model reduction techniques for computing approximately optimal solutions for markov decision processes. In *Proc. 13th UAI*, pages 124–131, San Francisco, California, 1997. Morgan Kaufmann Publishers.
12. R. Gerth, R. Kuiper, D. Peled, and W. Penczek. A partial order approach to branching time logic model checking. In *Proc. 3rd ISTCS'95*, pages 130–139. IEEE Press, 1995.
13. P. Godefroid. *Partial Order Methods for the Verification of Concurrent Systems: An Approach to the State Explosion Problem*, volume 1032 of *LNCS*. Springer-Verlag, 1996.
14. D. Griffioen and F. Vaandrager. Normed simulations. In *Proc. 10th International Computer Aided Verification Conference*, volume 1427 of *LNCS*, pages 332–344, 1998.
15. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
16. A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In *Proc. 12th TACAS*, 2006. To appear.
17. G. Holzmann. *The SPIN Model Checker, Primer and Reference Manual*. Addison Wesley, 2003.
18. C. Jones. *Probabilistic Non-Determinism*. PhD thesis, University of Edinburgh, 1990.
19. B. Jonsson and K. Larsen. Specification and refinement of probabilistic processes. In *Proc. LICS*, pages 266–277. IEEE CS Press, 1991.
20. R. Kurshan, V. Levin, M. Minea, D. Peled, and H. Yenign. Static partial order reduction. In *Proc. TACAS*, volume 1384 of *LNCS*, pages 345–357, 1998.
21. K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.
22. K. Namjoshi. A simple characterization of stuttering bisimulation. In *Proc. FSTTCS*, volume 1346 of *LNCS*, pages 284–296, 1997.
23. N. Pekergin and Sana Younes. Stochastic model checking with stochastic comparison. In *Proc. EPEW/WS-FM*, volume 3670 of *LNCS*, pages 109–123, 2005.
24. D. Peled. All from one, one for all: On model checking using representatives. In *Proc. 5th CAV*, volume 697 of *LNCS*, pages 409–423, 1993.
25. D. Peled. Partial order reduction: Linear and branching time logics and process algebras. In [26], pages 79–88, 1996.
26. D. Peled, V. Pratt, and G. Holzmann, editors. *Partial Order Methods in Verification*, volume 29(10) of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1997.
27. M. L. Puterman. *Markov Decision Processes—Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, 1994.
28. *Proceedings of the 1st International Conference on Quantitative Evaluation of SysTems (QEST 2004).* Enschede, the Netherlands. IEEE Computer Society Press, 2004.
29. Balaraman Ravindran and Andrew G. Barto. Model minimization in hierarchical reinforcement learning. In *Proc. SARA*, pages 196–211. LNCS, 2002.

30. R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology, 1995.
31. R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.
32. A. Valmari. Stubborn set methods for process algebras. In [26], pages 79–88, 1996.