Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 153 (2006) 97–116
www.elsevier.com/locate/entcs

Partial Order Reduction for Probabilistic Branching Time

Christel Baier [a,1] Pedro D'Argenio [b,2] Marcus Groesser [a,3]

[a] *Institut für Informatik I*
Universität Bonn
53117 Bonn, Germany

[b] *CONICET – FaMAF, Universidad Nacional de Córdoba*
Ciudad Universitaria
5000 Córdoba, Argentina

Abstract

In the past, partial order reduction has been used successfully to combat the state explosion problem in the context of model checking for non-probabilistic systems. For both linear time and branching time specifications, methods have been developed to apply partial order reduction in the context of model checking. Only recently, results were published that give criteria on applying partial order reduction for verifying quantitative linear time properties for probabilistic systems. This paper presents partial order reduction criteria for Markov decision processes and branching time properties, such as formulas of probabilistic computation tree logic. Moreover, we provide a comparison of the results established so far about reduction conditions for Markov decision processes.

Keywords: partial order reduction, Markov decision process, PCTL, model checking, probabilistic visible bisimulation, ample set

[1] Email: baier@cs.uni-bonn.de, supported by DFG-Project "VERIAM" and DFG-NWO-Project "VOSS II"

[2] Email: dargenio@famaf.unc.edu.ar, supported by the EC Project IST-2001-35304, "AMETIST", and the ANPCyT PICT 11-11738

[3] Email: groesser@cs.uni-bonn.de, supported by DFG-Project "VERIAM" and DFG-NWO-Project "VOSS II"

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2005.10.034

1 Introduction

Model checking is a technique that allows for the fully automatic verification of a property (often specified in a temporal logic) against a system that is modelled as a network of finite-state automata. It allows for the analysis of qualitative properties such as "every request is eventually answered". Following this example, there are systems whose nature may lead to some occasional unanswered request. Consider, for instance, a protocol that attempts to access a lossy medium a bounded number of times after which it aborts. A property like "access is eventually granted" is obviously false. Instead, to ensure quality of service, one would like that access is granted "often enough". For this purpose, model checking has been extended to deal with *quantitative* properties such as "access is eventually granted with at least 99% probability" [12,3]. In this case, systems are modelled as networks of Markov decision processes (MDP for short) [20].

To reason about non-probabilistic systems, a diversity of methods have been devised to tackle the state-explosion problem that arises when the network of automata is composed. A particular approach is partial order reduction [23,17,10,19, etc.] which is based on the observation that the execution order of concurrent operations does not usually change the validity of a property. Therefore, fixing one particular order of interleaving operations (without generating the others) helps to reduce the number of states and transitions that need to be explored while preserving the properties of interest.

Recently, Baier, Größer and Ciesinki [1] and D'Argenio and Niebert [5] developed independently from each other partial order reduction criteria for MDPs that preserve linear time properties, formalized as quantitative $\mathrm{LTL}_{\setminus X}$ properties. Both approaches rely on modifications of Peled's ample set methods [17,13,18].

The main contribution of this paper is the presentation of partial order reduction criteria for verifying *branching time* properties formalized by means of formulas of probabilistic computation tree logic [3]. Our criteria applied to ordinary transition systems reduce to the criteria suggested by Gerth et al. [10] for non-probabilistic branching time properties. Further on, we discuss the connections between the reduction criteria of [5,1] and those presented here and process equivalences (trace distribution equivalence, suitable notions of simulation and bisimulation).

Although the partial order reduction criteria for verifying branching time properties are rather strong and often might lead to a minor savings of states, our contribution has some impact under both practical and theoretical aspects. First, even a reduction that cannot shrink the state space of an MDP but only the transitions can increase the efficiency of the probabilistic model checking procedure. The latter relies on solving linear programs where the number of linear (in)equalities for any state s is given by the number of outgoing transitions from s. Thus, removing certain transitions via efficient reduction algorithms that operate on syntactic descriptions of the processes *simplifies the linear program* to be solved, and thus, can yield a speed-up of the analysis. Second, our reduction criteria provide the justification for modifying the given probabilistic program to be analyzed "by hand", e.g., using atomic regions for certain program fragments. [4] Third, in the context of the wide range of research results that discuss the possibility to adapt formal techniques to reason about non-probabilistic systems for the probabilistic setting, our results are of theoretical interest as they prove the existence of a *conservative probabilistic extension* of the partial order reduction criteria to preserve branching time properties. In fact, although research on model checking algorithms for probabilistic systems started about 20 years ago, the question whether partial order reduction for probabilistic systems is possible at all was open for a long time.

Organization of the paper. Section 2 briefly summarizes the preliminaries concerning our model (Markov decision processes). Section 3 recalls the criteria of the ample-method for linear time properties as suggested in [5,1]. The main result is presented in Section 4 where we provide the criteria to preserve probabilistic branching time properties. In Section 5, we explain the connections between the several reduction criteria and process equivalences. The paper ends with a brief conclusion in Section 6.

2 Preliminaries

In an MDP, any state s might have several outgoing action-labeled transitions, each of them is associated with a probability distribution which yields the probabilities for the successor states. As in [20,16,6] we assume here that for any state s, the outgoing transitions of s have different action labels. (This corresponds to the so-called reactive model in the classification of [25].) In addition, we assume here a labelling function that attaches to any state s a set of atomic propositions that are assumed to be fulfilled in state s. The atomic propositions will serve as atoms to formulate the desired properties in a temporal logical framework.

[4] The concept of atomic regions is known to be helpful to avoid the interleaving of concurrent activities, and thus, can be seen as a user-driven reduction technique to reduce the state space.

Definition 2.1 (Markov decision process (MDP), e.g. [20]) An MDP is a tuple $\mathcal{M} = (S, \mathsf{Act}, \mathsf{P}, s_{init}, \mathsf{AP}, \mathsf{L})$, where S is a finite set of states, Act a finite set of actions, $\mathsf{P} : (S \times \mathsf{Act} \times S) \to [0, 1]$ is the (three-dimensional) probability matrix, $s_{init} \in S$ the initial state, AP a finite set of atomic propositions, and $\mathsf{L} : S \to 2^{\mathsf{AP}}$ is a labeling function. $\mathsf{Act}(s)$ denotes the set of actions that are enabled in state s, i.e. the set of actions $\alpha \in \mathsf{Act}$ such that $\mathsf{P}(s, \alpha, t) > 0$ for some state $t \in S$. For any state $s \in S$, we require that $\mathsf{Act}(s) \neq \emptyset$ and $\sum_{s' \in S} \mathsf{P}(s, \alpha, s') = 1$ for any action $\alpha \in \mathsf{Act}(s)$. (In particular, we assume that $\mathcal{M}$ does not have terminal states.) □

The intuitive operational behavior of an MDP is as follows. If s is the current state then first one of the actions $\alpha \in \mathsf{Act}(s)$ is chosen non-deterministically. Afterwards, action α is executed leading to state t with probability $\mathsf{P}(s, \alpha, t)$.

We refer to t as an α-successor of s if $\mathsf{P}(s, \alpha, t) > 0$. Action α is called a *probabilistic action* if it has a random effect, i.e., if there is at least one state s where α is enabled and that has two or more α-successors. Otherwise, α is called non-probabilistic. In particular, if all actions in Act are non-probabilistic then our notion of an MDP reduces to an ordinary transition system with at most one outgoing α-transition per state and action α. When modelling realistic systems, most actions α will be non-probabilistic in the sense that they yield unique successor states.

Paths. An infinite path in an MDP is a sequence $\varsigma = s_0, \alpha_1, s_1, \alpha_2, \ldots \in (S \times \mathsf{Act})^\omega$ such that $\alpha_i \in \mathsf{Act}(s_{i-1})$ and $\mathsf{P}(s_{i-1}, \alpha_i, s_i) > 0$ for any $i \geq 1$. We write paths in the form

$$\varsigma \;=\; s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$$

Then $\mathsf{first}(\varsigma) = s_0$ denotes the starting state of ς and $\mathsf{trace}(\varsigma) = \mathsf{L}(s_0), \mathsf{L}(s_1), \mathsf{L}(s_2), \ldots$ the word over the alphabet 2^{AP} obtained by the projection of ς to the state labels. Finite paths (denoted by the greek letter σ) are finite prefixes of infinite paths that end in a state. We use the notations $\mathsf{first}(\sigma)$ and $\mathsf{trace}(\sigma)$ as for infinite paths, $\mathsf{last}(\sigma)$ for the last state of σ and $|\sigma|$ for the length (number of actions).

Schedulers. A scheduler denotes an instance that resolves the nondeterminism in the states, and thus, yields a Markov chain and a probability measure on the paths. We consider here history dependent, randomized schedulers (briefly called schedulers) which are given by a function D that assigns to any finite path σ a probability distribution over $\mathsf{Act}(\mathsf{last}(\sigma))$. Schedulers are essential for the semantics of PCTL. For a formal definition see [20,1]. Intuitively, a scheduler takes as input the "history" of a computation (formalized by a finite path σ) and chooses the next action α randomly, according to the probabilities specified by the distribution $D(\sigma)$. Given a state s and a scheduler D, the behavior of $\mathcal{M}$ under D can be formalized by a (possibly infinite-state) Markov chain.

Probabilistic computation tree logic (PCTL). PCTL is a probabilistic variant of CTL [4] which has been introduced first for Markov chains [12] and then for Markovian models with non-determinism [11,3,22]. We follow here the state-labeled approach of Bianco and de Alfaro [3] and consider the full logic PCTL*, but without the next step operator. We define PCTL$^*_{\setminus X}$-state formulas (denoted by the capital greek letter Φ) and PCTL$^*_{\setminus X}$-path formulas (denoted by φ) by the following grammar:

$$\Phi ::= \mathit{true} \;\big|\; a \;\big|\; \Phi \wedge \Phi \;\big|\; \neg\Phi \;\big|\; \mathcal{P}_{\bowtie p}(\varphi)$$

$$\varphi ::= \mathit{true} \;\big|\; \Phi \;\big|\; \varphi \wedge \varphi \;\big|\; \neg\varphi \;\big|\; \varphi \mathsf{U} \varphi$$

Here, $a \in \mathsf{AP}$ is an atomic proposition. U denotes the standard until operator. The intuitive meaning of the path formula $\varphi_1 \mathsf{U} \varphi_2$ is that φ_2 will eventually hold while before continuously φ_1 is satisfied. In the state formula $\mathcal{P}_{\bowtie p}(\varphi)$, the subscript $\bowtie p$ describes a probability bound, say $\geq p$, $\leq p$, $> p$ or $< p$ where p is a real number in the interval $[0, 1]$. Thus, $\mathcal{P}_{\bowtie p}(\varphi)$ holds for state s if for each scheduler D the probability measure of all infinite paths starting in s and fulfilling the path formula φ meets the probability bound $\bowtie p$. We skip the formal definition of the semantics of PCTL$^*_{\setminus X}$ which can be found e.g. in [3].

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. State formula Φ is said to hold for an MDP $\mathcal{M}$ if $\mathcal{M}$'s initial state satisfies Φ, i.e., if $s_{\mathit{init}} \models \Phi$.

Other boolean connectives, such as disjunction $\vee$, implication $\rightarrow$, can be derived as usual. The temporal operators eventually $\Diamond$ and always $\Box$ are obtained in the standard way by $\Diamond\varphi = \mathit{true}\mathsf{U}\varphi$ and $\Box\varphi = \neg\Diamond\neg\varphi$.

PCTL$_{\setminus X}$ denotes the state formula fragment of PCTL$^*_{\setminus X}$ where the path subformulas in $\mathcal{P}_{\bowtie p}(\varphi)$ are of the form $\Phi_1 \mathsf{U} \Phi_2$. LTL$_{\setminus X}$ arises as the path formula fragment of PCTL$^*_{\setminus X}$ where all state subformulas are propositional formulas, i.e., do not contain the probabilistic operator $\mathcal{P}_{\bowtie p}(\cdot)$. By a quantitative LTL$_{\setminus X}$ property, we mean a PCTL$^*_{\setminus X}$ state formula of the form $\mathcal{P}_{\bowtie p}(\varphi)$ where φ is a LTL$_{\setminus X}$ formula.

Stutter actions. The correctness of partial order reduction criteria and temporal properties is typically formulated by means of an equivalence that identifies those states/paths whose traces agree up to stuttering. In this context, stuttering refers to the repetition of the same state-labels. For the partial order reduction we shall need the concept of *stutter actions*, i.e., actions that have no effect on the state-labels, no matter in which state they are taken. Formally, action α of an MDP $\mathcal{M}$ is called a stutter action iff for all states s, $t \in S$ we have: $\mathsf{P}(s, \alpha, t)$ ¿ 0 implies $\mathsf{L}(s) = \mathsf{L}(t)$.

We refer to $s \xrightarrow{\beta} t$ as a non-probabilistic stutter step if $\beta \in \mathsf{Act}(s)$ is a non-probabilistic stutter action and t the unique β-successor of s.

3 The ample set method for linear time properties

In this section, we summarize the main results of [5] and [1] and recall the argument why these techniques fail for branching time properties. [5] The starting point is an MDP $\mathcal{M} = (S, \mathsf{Act}, \mathsf{P}, s_{init}, \mathsf{AP}, \mathsf{L})$ to be verified against a linear time property. The rough idea is to assign to any reachable state s an action-set $\mathsf{ample}(s) \subseteq \mathsf{Act}(s)$ and to construct a reduced MDP $\hat{\mathcal{M}}$ that results by using the action-sets $\mathsf{ample}(s)$ instead of $\mathsf{Act}(s)$. Formally, given a function $\mathsf{ample} : S \to 2^{\mathsf{Act}}$ with $\mathsf{ample}(s) \subseteq \mathsf{Act}(s)$ for all states s, the state space of the reduced MDP $\hat{\mathcal{M}} = (\hat{S}, \mathsf{Act}, \hat{\mathsf{P}}, s_{init}, \mathsf{AP}, \hat{\mathsf{L}})$ induced by ample is the smallest set $\hat{S} \subseteq S$ that contains s_{init} and any state t where $\mathsf{P}(s, \alpha, t)$ ¿ 0 for some $s \in \hat{S}$ and $\alpha \in \mathsf{ample}(s)$. The labeling function $\hat{\mathsf{L}} : \hat{S} \to 2^{\mathsf{AP}}$ is the restriction of the original labeling function L to the state-set $\hat{S}$. [6] The transition probability matrix of $\hat{\mathcal{M}}$ is given by: $\hat{\mathsf{P}}(s, \alpha, t) = \mathsf{P}(s, \alpha, t)$ if $\alpha \in \mathsf{ample}(s)$ and 0 otherwise. State s is called fully expanded if $\mathsf{ample}(s) = \mathsf{Act}(s)$.

If the ample sets are "small" then we might expect that the linear program to be solved for $\hat{\mathcal{M}}$ is simpler than the one for $\mathcal{M}$. This is firstly, because the number of variables for $\hat{\mathcal{M}}$ is smaller than for $\mathcal{M}$, since there is one variable per state. Secondly the linear programs to be solved for $\hat{\mathcal{M}}$ contain less inequalities for any reachable state s that is not fully expanded (i.e., $\mathsf{ample}(s) \neq \mathsf{Act}(s)$).

Independence of actions. 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 rough idea is a formalization of actions belonging to different processes that are executed in parallel and do not affect each other, e.g. as they only refer to local variables and do not require any kind of synchronization. The formal 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)

[5] We adapt here the notations and conditions used in [5] and [1] for the purposes of this paper.

[6] Atomic propositions that do not occur in the given formula are assumed also not to occur in AP. This simple step can identify more paths as stutter equivalent, hence improving the reduction.

relies on recovering the interleaving diamonds. In non-probabilistic systems, independence of two actions α and β means that for any state s where both α and β are enabled the execution of α does not affect the enabledness of β (i.e., the α-successor of s has an outgoing β-transition), and vice versa, and in addition the action sequences $\alpha\beta$ and $\beta\alpha$ lead to the same state. In the probabilistic setting, the additional requirement that $\alpha\beta$ and $\beta\alpha$ have the same probabilistic effect is made:

Definition 3.1 (Independence of actions, cf. [5,1] Two actions α, β with $\alpha \neq \beta$ are called independent (in $\mathcal{M}$) iff for all states $s \in S$ with $\{\alpha, \beta\} \subseteq \mathsf{Act}(s)$:

(1) $\mathsf{P}(s, \alpha, t)$ ¿ 0 implies $\beta \in \mathsf{Act}(t)$,

(2) $\mathsf{P}(s, \beta, u)$ ¿ 0 implies $\alpha \in \mathsf{Act}(u)$

(3) for all states $w \in S$: $\sum_{t \in S} \mathsf{P}(s, \alpha, t) \cdot \mathsf{P}(t, \beta, w) = \sum_{u \in S} \mathsf{P}(s, \beta, u) \cdot \mathsf{P}(u, \alpha, w)$

Two different actions α and β are called dependent iff α and β are not independent. If $\mathsf{A} \subseteq \mathsf{Act}$ and $\alpha \in \mathsf{Act} \setminus \mathsf{A}$ then α is called independent from A iff for all actions $\beta \in \mathsf{A}$, α and β are independent. Otherwise, α is called dependent on A. □

Applying the above definition to non-probabilistic actions α and β (i.e., where $\mathsf{P}(s, \alpha, t)$, $\mathsf{P}(s, \beta, t) \in \{0, 1\}$ for all states s, t) yields the standard definition of independence of actions in ordinary transition systems.

Example 3.2 Fig. 1 shows a fragment of an MDP $\mathcal{M}_1$ representing the parallel execution of independent actions α and β. For example, α might stand for the outcome of the experiment of tossing a "one" with a dice, while β stands for tossing a fair coin. In general, whenever α and β stand for stochastic experiments that are independent in the classical sense then α and β viewed as actions of an MDP are independent. However, there are also other situations where two actions can be independent that do not have a fixed probabilistic branching pattern. E.g., actions α and β in the MDP $\mathcal{M}_2$ in Fig. 1 are independent. First notice that only in state s both α and β are enabled. The α-successors t, s of s have a β-transition to state u, while the β-successor u has a α-transition to itself. The effect under the action sequences $\alpha\beta$ and $\beta\alpha$ is the same as in either case state u is reached with probability 1. □

Criteria for linear time properties. To preserve linear time properties, both approaches [5] and [1] use a series of conditions (see Figure 2) that rely

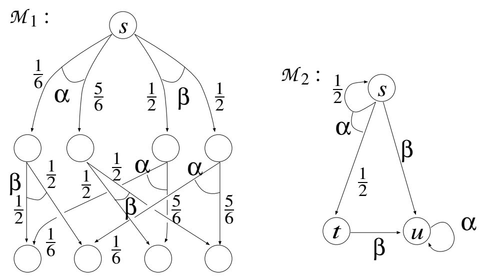

Fig. 1. **Examples for independent actions**

A0 **(Nonemptiness-condition)** For all states $s \in S$, $\emptyset \neq \mathsf{ample}(s) \subseteq \mathsf{Act}(s)$.

A1 **(Stutter-condition)** If $s \in \hat{S}$ and $\mathsf{ample}(s) \neq \mathsf{Act}(s)$ then all actions $\alpha \in \mathsf{ample}(s)$ are stutter actions.

A2 **(Dependence-condition)** For each path $\sigma = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \xrightarrow{\gamma} \ldots$ in $\mathcal{M}$ where $s \in \hat{S}$ and γ is dependent on $\mathsf{ample}(s)$ there exists an index $i \in \{1, \ldots, n\}$ such that $\alpha_i \in \mathsf{ample}(s)$.

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

A4 **(Branching condition)** ...

Fig. 2. **Conditions for the ample-sets**

on modifications of Peled's conditions for preserving $\mathrm{LTL}_{\backslash X}$-properties. The nonemptiness-condition (A0) ensures that the reduced system is a sub-MDP of the original one. [7] The stutter-condition (A1), dependence-condition (A2) and the cycle-condition (A3) agree exactly with Peled's conditions [17,13] for non-probabilistic systems and linear time properties. Instead of (A3), [1] suggests a weaker condition that uses the concept of de Alfaro's *end components* [6,7] in the style of the following condition:

A3' (End component condition) In each end component (T, A) in $\hat{\mathcal{M}}$ there is a fully expanded state, i.e., $\mathsf{ample}(t) = \mathsf{Act}(t)$ for some $t \in T$.

End components can be viewed as the MDP-counterpart to terminal strongly connected components in Markov chains. They consist of a state-set T and a nonempty action-set $A(t) \subseteq \mathsf{Act}(t)$ for each $t \in T$ such that for all $t \in T$ and actions $\alpha \in A(t)$ any α-successor of t belongs to T and the underlying directed graph of (T, A) is strongly connected. While in MDPs, infinite traversal of a cycle with at least one probabilistic action occurs with probability 0, almost

[7] Recall that in this paper we focus our attention to MDPs *without* terminal states. This explains why we require here as in [1], but unlike [5], the ample-sets to be non-empty.

all paths "end" in an end component (T, A), that is, once T is entered and only actions in $A(t)$ are scheduled, T will not be left and any state of T is visited infinitely often almost surely.

While conditions (A0)-(A3) ensure the equivalence of $\mathcal{M}$ and the reduced system $\hat{\mathcal{M}}$ in the non-probabilistic setting, they are not sufficient in the probabilistic setting, not even for reachability problems. Informally, the problem is that (A0)-(A3) allow for a reduction where the ample-set of a state s consists of e.g. two actions, say β, γ, while a certain probabilistic action $\alpha \in \mathsf{Act}(s)$ is not contained in the ample set of s. But then, a scheduler D for $\mathcal{M}$ might first schedule α and then – depending on the probabilistic outcome of α – decide to choose one of the ample actions β or γ (or to choose β and γ with appropriate probabilities). On the other hand, any scheduler for $\hat{\mathcal{M}}$ is forced to assign fixed probabilities to the actions β and γ *before* the outcome of the probabilistic experiment according to α is known. This explains why, with (A0)-(A3), $\mathcal{M}$ might have better strategies for reachability or other linear time properties than $\hat{\mathcal{M}}$. To remedy the situation a further condition is needed:

A4.1 (branching condition à la [5])
$|\mathsf{ample}(s)| = 1$ or $\mathsf{ample}(s) = \mathsf{Act}(s)$ for any state $s \in \hat{S}$.

A4.2 (branching condition à la [1]) If $\sigma = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \xrightarrow{\gamma} \ldots$ is a path in $\mathcal{M}$ where $s \in \hat{S}$, $\alpha_1, \ldots, \alpha_n, \gamma \notin \mathsf{ample}(s)$ and γ is probabilistic then $|\mathsf{ample}(s)| = 1$.

Clearly, (A4.2) is weaker than (A4.1). Moreover, (A4.2) is irrelevant for ordinary transition systems viewed as MDPs where all actions are non-probabilistic. [5] shows that under conditions (A0)-(A3) and (A4.1), $\mathcal{M}$ and $\hat{\mathcal{M}}$ are simulation equivalent with regard to a probabilistic variant of forward simulation, and thus, $\mathcal{M}$ and $\hat{\mathcal{M}}$ are equivalent for all properties that are preserved by the simulation relation. [1] provides a direct proof for the preservation of measurable, stutter insensitive linear time properties under conditions (A0)-(A3) and (A4.2).

[5] and [1] observed that although (A4.1) agrees with the condition made by Gerth et al. [10,18] which in combination with (A0)-(A3) ensures the preservation of branching time properties for non-probabilistic systems, (A0)-(A3) and (A4.1) may fail for verifying probabilistic branching time properties specified in $\mathrm{PCTL}_{\backslash X}$. The counterexample, given in Fig. 3, that illustrates this observation is a probabilistic variant of the example presented in [10,18] to demonstrate that (A0)-(A3) cannot guarantee that a non-probabilistic system and the reduced system are $\mathrm{CTL}_{\backslash X}$-equivalent. In the MDP $\mathcal{M}$ in Fig. 3, a and b are atomic propositions. [8] Dark states are labelled with $\{a, b\}$, grey

[8] In Fig. 3 we have grouped together the states of the system $\mathcal{M}$ that are probabilistic

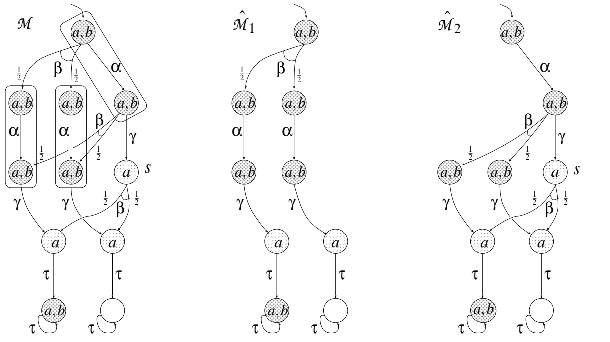

Fig. 3. **(A0)-(A3) and (A4.1) are not sufficient for PCTL$_{\setminus X}$**

states with $\{a\}$ and white states with $\emptyset$. α and β are independent stutter actions. Moreover, β and γ are independent. Thus, conditions (A0)-(A3) and (A4.1) are fulfilled when choosing the singleton ample-set $\{\beta\}$ in the initial state which leads to the reduced MDP $\hat{\mathcal{M}}_1$ in Fig. 3. But then, the PCTL$_{\setminus X}$-formula

$$\mathcal{P}_{=1}(\ \Box(\ (a \wedge \neg b) \rightarrow (\mathcal{P}_{=1}[\Diamond b] \vee \mathcal{P}_{=1}[\Diamond \neg a])\ )\ )$$

holds for $\hat{\mathcal{M}}_1$, but not for $\mathcal{M}$. An intuitive explanation for this phenomenon is the fact that (A4.1) still allows for probabilistic branches in non-fully expanded states leading to states that are not PCTL$_{\setminus X}$-equivalent.

4 Preserving branching time properties

As a consequence of the previous example, requirements (A4.1) and (A4.2) need to be strengthened. Therefore, we adopt the following stronger condition.

A4 (branching condition for branching time properties) If $\mathsf{ample}(s) \neq \mathsf{Act}(s)$ then $\mathsf{ample}(s)$ is a singleton consisting of a non-probabilistic action.

Notice that condition (A4) collapses to (A4.1) for ordinary transition systems, i.e., MDPs where all actions are non-probabilistic. Thus, the five conditions (A0)-(A4) that we suggest for a reduction that preserves probabilistic branching time properties yield a conservative adaption of the conditions (A0)-(A3), (A4.1) suggested by Gerth et al. [10,18] for non-probabilistic systems and CTL$^*_{\setminus X}$-properties.

To handle branching time properties, the cycle-condition (A3) could also be replaced with the weaker end component condition (A3'). However, in combination with (A4), conditions (A3') and (A3) are equivalent. This follows from the fact that for any end component in $\hat{\mathcal{M}}$ where none of its states is

visible bisimilar, which will be explained later in definition 4.5.

fully expanded, the ample-sets of all its states are singletons consisting of a non-probabilistic action. Thus, the end component under consideration is a cycle.

Example 4.1 In contrast to $\hat{\mathcal{M}}_1$, the reduced system $\hat{\mathcal{M}}_2$ in Fig. 3 which is obtained from $\mathcal{M}$ by choosing the ample-set of the initial state to be $\{\alpha\}$ fulfills (A0)-(A4). Thus, as will be shown, $\mathcal{M}$ and $\hat{\mathcal{M}}_2$ satisfy the same branching time properties. □

The remainder of this section is concerned with the proof of the correctness of our approach which is stated in the following theorem.

Theorem 4.2 (Correctness of (A0)-(A4)) *If (A0)-(A3) and (A4) are fulfilled then $\mathcal{M}$ and $\hat{\mathcal{M}}$ satisfy the same* $\mathrm{PCTL}^*_{\setminus X}$ *state formulas.*

We use a proof technique similar to those of [10,18] where (A0)-(A3) and (A4.1) are shown to be sufficient for $\mathrm{CTL}^*_{\setminus X}$ properties and non-probabilistic transition systems. However, we have here the additional difficulty to reason about probabilistic behaviors.

Definition 4.3 (Weight function, cf. [12]) Let S, S' be finite sets and $\mathcal{R} \subseteq S \times S'$. If μ and μ' are distributions on S and S' respectively [9] then a weight function for (μ, μ') with respect to $\mathcal{R}$ denotes a function $w : S \times S' \to [0, 1]$ such that

- $w(s, s') > 0$ implies $(s, s') \in \mathcal{R}$,
- $\sum_{s' \in S'} w(s, s') = \mu(s)$ for all $s \in S$ and $\sum_{s \in S} w(s, s') = \mu'(s')$ for all $s' \in S'$.

We write $\mu \cong_{\mathcal{R}} \mu'$ to denote the existence of a weight function for (μ, μ') w.r.t. $\mathcal{R}$ and refer to $\cong_{\mathcal{R}}$ as the lifting of $\mathcal{R}$ to distributions. □

In the sequel, we will use the following observation which is e.g. shown in [2,8]:

Proposition 4.4 (Transitivity of $\cong_{\mathcal{R}}$) *If $\mathcal{R}$ is a binary, transitive relation on a set S and μ, μ', μ'' are distributions on S such that $\mu \cong_{\mathcal{R}} \mu'$ and $\mu' \cong_{\mathcal{R}} \mu''$ then $\mu \cong_{\mathcal{R}} \mu''$.*

The following definition can be viewed as a probabilistic variant of the so-called *visible bisimulation* that has been introduced in [10].

Definition 4.5 (Probabilistic visible bisimulation (pvb))
Let $\mathcal{M} = (S, \mathsf{Act}, \mathsf{P}, s_{init}, \mathsf{AP}, \mathsf{L})$ and $\mathcal{M}' = (S', \mathsf{Act}', \mathsf{P}', s'_{init}, \mathsf{AP}, \mathsf{L}')$ be two MDPs with the same set of atomic propositions and let $\mathcal{R} \subseteq S \times S'$ be a binary

[9] By a distribution on a finite set S we mean a function $\mu : S \to [0, 1]$ such that $\sum_{s \in S} \mu(s) = 1$.

relation. Then, $\mathcal{R}$ is called a probabilistic visible simulation if $(s_{init}, s'_{init}) \in \mathcal{R}$ and for any pair (s, s') on $\mathcal{R}$ the following three conditions are fulfilled.

(1) $\mathsf{L}(s) = \mathsf{L}'(s')$

(2) For any action $\alpha \in \mathsf{Act}(s)$ at least one of the following two conditions holds:

(2.1) α is a non-probabilistic stutter action such that $(t, s') \in \mathcal{R}$ for the unique α-successor t of s,

(2.2) There is a finite path σ' of the form $s' = s'_0 \xrightarrow{\beta_0} s'_1 \ldots \xrightarrow{\beta_{n-1}} s'_n$ in $\mathcal{M}'$ s.t. [10]

- $\beta_0, \ldots, \beta_{n-1}$ are non-probabilistic stutter actions,
- $(s, s'_i) \in \mathcal{R}$ for $1 \le i \le n$,
- $\alpha \in \mathsf{Act}'(s'_n)$ and $\mathsf{P}(s, \alpha, \cdot) \cong_{\mathcal{R}} \mathsf{P}'(s'_n, \alpha, \cdot)$.

(3) If there is an infinite path ς of the form $s = t_0 \xrightarrow{\beta_0} t_1 \xrightarrow{\beta_1} t_2 \xrightarrow{\beta_2} t_3 \xrightarrow{\beta_3} \ldots$ in $\mathcal{M}$ consisting of non-probabilistic stutter actions $\beta_0, \beta_1, \beta_2, \ldots$ and such that $(t_i, s') \in \mathcal{R}$, $i = 0, 1, 2, \ldots$ then there is a finite path σ' of the form $s' = t'_0 \xrightarrow{\gamma_0} t'_1 \xrightarrow{\gamma_1} \ldots \xrightarrow{\gamma_{j-1}} t'_j \xrightarrow{\gamma_j} t'_{j+1}$ in $\mathcal{M}'$ such that $(s, t'_i) \in \mathcal{R}$, $i = 0, 1, \ldots, j$, $(t_1, t'_{j+1}) \in \mathcal{R}$, and $\gamma_0, \gamma_1, \ldots, \gamma_{j-1}, \gamma_j$ are non-probabilistic stutter actions.

$\mathcal{R}$ is called a probabilistic visible bisimulation for $(\mathcal{M}, \mathcal{M}')$ if $\mathcal{R}$ is a probabilistic visible simulation for $(\mathcal{M}, \mathcal{M}')$ and $\mathcal{R}^{-1}$ is a probabilistic visible simulation for $(\mathcal{M}', \mathcal{M})$. We write $\mathcal{M} \approx_{\text{pvb}} \mathcal{M}'$ iff there exists a probabilistic visible bisimulation for $(\mathcal{M}, \mathcal{M}')$.

Our goal is to show that $\mathcal{M} \approx_{\text{pvb}} \hat{\mathcal{M}}$ where $\mathcal{M}$ denotes the "full" MDP and $\hat{\mathcal{M}}$ the reduced MDP that results from ample-sets satisfying (A0)-(A4). The following proposition completes then our argumentation.

Proposition 4.6 (Soundness of pvb for $PCTL^*$ without next step) *Let $\mathcal{M}$ and $\mathcal{M}'$ be two MDPs as in Definition 4.5 such that $\mathcal{M} \approx_{pvb} \mathcal{M}'$. Then, $\mathcal{M}$ and $\mathcal{M}'$ satisfy the same* $\mathrm{PCTL}^*_{\setminus X}$ *state formulas.*

Proof. *(Sketch).* One proof obligation relies on proving that the coarsest probabilistic visible bisimulation $\mathcal{R}$ is a divergence-sensitive probabilistic branching bisimulation and the latter is sound for $\mathrm{PCTL}^*_{\setminus X}$ [22,21]. [11]

[10] The case $n = 0$, i.e., $\sigma' = s'$, is allowed.

[11] There are some minor differences between our approach and those in [22,21], e.g. they use an action-labelled setting and prove the preservation result under the assumption of probabilistic convergence (rather than considering a divergence-sensitive variant of probabilistic branching bisimulation). However, the main argumentation for the preservation result for a notion of divergence-sensitive probabilistic branching bisimulation will be the same.

Another different proof obligation is to provide a direct proof for the claim and to show by structural induction on the syntax of $\mathrm{PCTL}^*_{\setminus X}$ state/path formulas that whenever $\mathcal{R}$ is a probabilistic visible bisimulation then

- for all $\mathrm{PCTL}^*_{\setminus X}$ state formulas Φ and $(s, s') \in \mathcal{R}$: $s \models \Phi$ iff $s' \models \Phi$
- for all $\mathrm{PCTL}^*_{\setminus X}$ path formulas φ and $(\varsigma, \varsigma') \in \mathcal{R}_{\mathrm{path}}$: $\varsigma \models \varphi$ iff $\varsigma' \models \varphi$

Here, $\mathcal{R}_{\mathrm{path}}$ denotes the "lifting" of $\mathcal{R}$ to paths (which has to be defined in an appropriate way). We skip the details of this proof obligation too as it relies on standard arguments provided e.g. in [22,21] (and also [9] for an MDP-like model where probabilistic and non-deterministic states alternate). □

In the sequel, we assume that conditions (A0)-(A4) hold. Our goal is now to establish a probabilistic visible bisimulation that relates $\mathcal{M}$ and $\hat{\mathcal{M}}$.

Definition 4.7 (Forming path, relation $\rightsquigarrow$) Let $\mathcal{M}$ be an MDP as before and let $s, s' \in S$. A forming path from s to s' is a finite path σ of the form

$$s = s_0 \xrightarrow{\beta_0} s_1 \xrightarrow{\beta_1} \ldots \xrightarrow{\beta_{n-1}} s_n = s' \qquad (*)$$

where $\beta_0, \ldots, \beta_{n-1}$ are non-probabilistic stutter actions and, for $i = 0, 1, \ldots, n-1$, the singleton action-set $\{\beta_i\}$ fulfills the dependence condition (A2) for state s_i.[12] We write $s \rightsquigarrow s'$ iff there exists a forming path from s to s'. $\cong$ denotes the lifting of $\rightsquigarrow$ to distributions on S via weight functions as in definition 4.3 (i.e., $\cong\, = \,\cong_{\rightsquigarrow}$). □

As the formal definition of forming paths only refers to non-probabilistic actions and agrees exactly with the definition of forming paths in the non-probabilistic setting [10,18], the following properties that were established for non-probabilistic systems also hold for MDPs. First, we observe that the

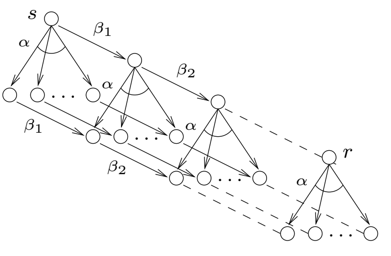

Fig. 4. **Illustration of Prop. 4.8**

relation $\rightsquigarrow$ is transitive and reflexive (even though, in general, non-symmetric). Second, if σ is a forming path from s to s' of length n as in (*) in definition 4.7

[12] That is, for any finite path $s_i \xrightarrow{\alpha_0} t_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{m-1}} t_m \xrightarrow{\gamma} \ldots$ where γ is dependent on β_i there exists $j \in \{0, 1, \ldots, m-1\}$ such that $\alpha_j = \beta_i$.

then $s_i \rightsquigarrow s_j$ for $0 \leq i \leq j \leq n$. In addition, forming paths enjoy the property that they can be replicated after an independent operation is performed. In a probabilistic setting this can be depicted as in Fig. 4 and is formally stated in the next proposition.

Proposition 4.8 (Properties of forming paths) *Let s, s' be two states in $\mathcal{M}$ such that $s \rightsquigarrow s'$ and let $\alpha \in \mathsf{Act}(s)$.*

(a) If there is a forming path from s to s' in which α does not occur then $\alpha \in \mathsf{Act}(s')$ and $\mathsf{P}(s, \alpha, \cdot) \cong \mathsf{P}(s', \alpha, \cdot)$.

(b) If α is a non-probabilistic stutter action with $s \xrightarrow{\alpha} t$ and $t \not\rightsquigarrow s'$ then $\alpha \in \mathsf{Act}(s')$ and $s' \xrightarrow{\alpha} t'$ where $t \rightsquigarrow t'$. (In particular, we also have $\mathsf{P}(s, \alpha, \cdot) \cong \mathsf{P}(s', \alpha, \cdot)$.)

Proof. The proof for (a) can be provided using induction on the length n of a forming path from s to s' where α does not occur. The basis of induction $n = 0$ is obvious as we then have $s = s'$. In the induction step $n - 1 \Longrightarrow n (n \geq 1)$ we assume that

$$s = s_0 \xrightarrow{\beta_0} s_1 \xrightarrow{\beta_1} \ldots \xrightarrow{\beta_{n-2}} s_{n-1} \xrightarrow{\beta_{n-1}} s_n = s'$$

is a forming path from s to s' such that $\alpha \notin \{\beta_0, \ldots, \beta_{n-1}\}$. By induction hypothesis we have $\alpha \in \mathsf{Act}(s_{n-1})$ and

$$\mathsf{P}(s, \alpha, \cdot) \cong \mathsf{P}(s_{n-1}, \alpha, \cdot) \tag{+}$$

As the dependence condition (A2) holds for state s_{n-1} and the singleton action-set $\{\beta_{n-1}\}$, actions α and β_{n-1} are independent. Since β_{n-1} is a non-probabilistic stutter action, $s_n = s'$ is the unique β_{n-1}-successor of s_{n-1}. Thus $\alpha \in \mathsf{Act}(s_n)$ (see Def. 3.1) and for any α-successor t of s_{n-1}, we have $\beta_{n-1} \in \mathsf{Act}(t)$. Moreover, condition (A2) also holds for any α-successor t of s_{n-1} and the singleton action-set $\{\beta_{n-1}\}$, since α and β_{n-1} are independent and (A2) holds for state s_{n-1} and the singleton action-set $\{\beta_{n-1}\}$. Let u_t be the unique β_{n-1}-successor of t. We then have $t \rightsquigarrow u_t$. As the probabilistic effect of the action sequences $\alpha\beta_{n-1}$ and $\beta_{n-1}\alpha$ in state s_{n-1} are the same we have:

$$\textstyle\sum_{\substack{t \in S \\ u_t = u}} \mathsf{P}(s_{n-1}, \alpha, t) = \mathsf{P}(s_n, \alpha, u)$$

for any state $u \in S$. Thus, we may deal with the weights $w(t, u_t) = \mathsf{P}(s_{n-1}, \alpha, t)$ and $w(\cdot) = 0$ in all remaining cases. Hence, $\mathsf{P}(s_{n-1}, \alpha, \cdot) \cong \mathsf{P}(s_n, \alpha, \cdot)$. Using (+) and the transitivity of $\cong$ (cf. Proposition 4.4) we get $\mathsf{P}(s, \alpha, \cdot) \cong \mathsf{P}(s_n, \alpha, \cdot)$. The proof for part (b) can be provided with similar arguments, also using induction of the length of a forming path from s to s'. $\square$

Note that part (a) of Proposition 4.8 applies to all actions $\beta \in \mathsf{Act}(s)$ which are probabilistic or which are non-stutter actions. But, in addition, there might be also non-probabilistic stutter actions β enabled in s that do not occur on at least one forming path from s to the given state s'.

Definition 4.9 (Relation $\mathcal{R}$) The relation $\mathcal{R}$ is given by $\mathcal{R} = \{(s, \hat{s}) \in S \times \hat{S} : s \rightsquigarrow \hat{s}\}$. (As before, S is the state space of $\mathcal{M}$ and $\hat{S}$ the state space of $\hat{\mathcal{M}}$.) □

In the sequel, a forming path in $\hat{\mathcal{M}}$ means a forming path $s_0 \xrightarrow{\beta_0} s_1 \xrightarrow{\beta_1} \ldots \xrightarrow{\beta_{n-1}} s_n$ as in definition 4.7 where $s_0, s_1, \ldots, s_n \in \hat{S}$ and $\beta_i \in \mathsf{ample}(s_i)$, $i = 0, 1, \ldots, n-1$.

Proposition 4.10 (Forming paths in the reduced MDP) *Let $\hat{s}$ be a state in $\hat{\mathcal{M}}$.*

(a) If $\hat{\sigma}$ is a forming path in $\hat{\mathcal{M}}$ starting in state $\hat{s}$ and $(s, \hat{s}) \in \mathcal{R}$ then $(s, \hat{u}) \in \mathcal{R}$ for all states $\hat{u}$ in $\hat{\sigma}$.

(b) There exists a forming path from $\hat{s}$ in $\hat{\mathcal{M}}$ to some fully expanded state.

(c) If $\alpha \in \mathsf{Act}(\hat{s})$ then there exists a forming path $\hat{\sigma}$ in $\hat{\mathcal{M}}$ from $\hat{s}$ to some state $\hat{u}$ such that $\alpha \in \mathsf{ample}(\hat{u})$ and $\mathsf{P}(\hat{s}, \alpha, \cdot) \cong \mathsf{P}(\hat{u}, \alpha, \cdot)$.

Proof. Part (a) is immediate by transitivity of $\rightsquigarrow$.

Part (b) follows from the fact that any finite path in $\hat{\mathcal{M}}$ where none of its states is fully expanded is a forming path (because of conditions (A1), (A2) and (A4)). As $\hat{\mathcal{M}}$ is finite-state, the non-emptiness condition (A0) and the cycle condition (A3) ensure the existence of a forming path from $\hat{s}$ to a fully expanded state.

Part (c) can be derived from (b) and Proposition 4.8 as follows.

Let $\hat{\sigma}$ be a forming path in $\hat{\mathcal{M}}$ from $\hat{s}$ where $\hat{v} = \mathsf{last}(\hat{\sigma})$ is fully expanded and let $\alpha \in \mathsf{Act}(\hat{s})$. If α does not occur in $\hat{\sigma}$ then Proposition 4.8(a) yields

$$\alpha \in \mathsf{Act}(\hat{v}) = \mathsf{ample}(\hat{v}) \text{ and } \mathsf{P}(\hat{s}, \alpha, \cdot) \cong \mathsf{P}(\hat{v}, \alpha, \cdot).$$

If α appears in $\hat{\sigma}$ then we consider the longest prefix $\hat{\pi}$ of $\hat{\sigma}$ where α does not occur. Let $\hat{u} = \mathsf{last}(\hat{\pi})$. Then, $\hat{\sigma}$ has the form

$$\underbrace{\hat{s} \to \ldots \to \hat{u}}_{=\hat{\pi}} \xrightarrow{\alpha} \ldots \to \hat{v}.$$

In particular, $\alpha \in \mathsf{ample}(\hat{u})$. Again, Proposition 4.8 (a) yields $\mathsf{P}(\hat{s}, \alpha, \cdot) \cong \mathsf{P}(\hat{u}, \alpha, \cdot)$. □

We are now ready to prove that $\mathcal{M}$ and $\hat{\mathcal{M}}$ are probabilistic visible bisimilar.

Theorem 4.11 *$\mathcal{R}$ is a probabilistic visible bisimulation for $(\mathcal{M}, \hat{\mathcal{M}})$.*

Proof. Clearly, $(s_{init}, s_{init}) \in \mathcal{R}$. We show that for any $(s, \hat{s}) \in \mathcal{R}$ conditions (1)-(3) in Def. 4.5 hold, and conversely, that (1)-(3) are fulfilled for the "inverse" pair $(\hat{s}, s) \in \mathcal{R}^{-1}$.

(1) It is obvious that $\mathsf{L}(s) = \mathsf{L}(\hat{s})$ as all actions on a forming path are stutter actions. Thus, all states on a forming path have the same labeling.

(2) We first show that condition (2) in Def. 4.5 holds for $(s, \hat{s}) \in \mathcal{R}$. Let $\alpha \in \mathsf{Act}(s)$.

If α is a non-probabilistic stutter action and $s \xrightarrow{\alpha} t$ where $t \rightsquigarrow \hat{s}$ (and thus, $(t, \hat{s}) \in \mathcal{R}$) then we are in the situation of condition (2.1) in Definition 4.5.

Let us now assume that α is probabilistic or a non-stutter action or $s \xrightarrow{\alpha} t$ is a non-probabilistic stutter step where $t \not\rightsquigarrow \hat{s}$. In either case, we may apply part (a) or (b) of Proposition 4.8 which yields

$$\alpha \in \mathsf{Act}(\hat{s}) \text{ and } \mathsf{P}(s, \alpha, \cdot) \cong \mathsf{P}(\hat{s}, \alpha, \cdot). \tag{+}$$

As $\hat{s}$ is a state in the reduced MDP $\hat{\mathcal{M}}$, part (c) of Proposition 4.10 yields the existence of a forming path from $\hat{s}$ in the reduced MDP $\hat{\mathcal{M}}$ to some state $\hat{u}$ where $\alpha \in \mathsf{ample}(\hat{u})$ and $\mathsf{P}(\hat{s}, \alpha, \cdot) \cong \mathsf{P}(\hat{u}, \alpha, \cdot)$. Hence, by (+) and the transitivity of $\cong$ (see Proposition 4.4) we obtain:

$$\alpha \in \mathsf{Act}(\hat{u}) \text{ and } \mathsf{P}(s, \alpha, \cdot) \cong \mathsf{P}(\hat{u}, \alpha, \cdot). \tag{++}$$

We may compose the forming paths in $\mathcal{M}$ from s to $\hat{s}$ (which exists as we have $s \rightsquigarrow \hat{s}$) and the forming path $\hat{\sigma}$ from $\hat{s}$ to $\hat{u}$ in $\hat{\mathcal{M}}$ and obtain $s \rightsquigarrow \hat{u}$. As $\hat{u} \in \hat{S}$, we get $(s, \hat{u}) \in \mathcal{R}$. By part (a) of Proposition 4.10 we get $(s, \hat{v}) \in \mathcal{R}$ for all states $\hat{v}$ in $\hat{\sigma}$. Thus, (++) yields that we are in the situation of condition (2.2) in Def. 4.5.

Let now be $(\hat{s}, s) \in \mathcal{R}^{-1}$ and an action $\alpha \in \mathsf{ample}(\hat{s})$. As $(\hat{s}, s) \in \mathcal{R}^{-1}$ we have $s \rightsquigarrow \hat{s}$. Thus, there is a forming path σ' from s to $\hat{s}$. Using the trivial fact that $\mathsf{P}(\hat{s}, \alpha, \cdot) \cong \mathsf{P}(\hat{s}, \alpha, \cdot)$ and part (a) of Proposition 4.10 we are in the case of condition (2.2) in Def. 4.5 (note, that $\mathcal{R}^{-1} \subseteq \hat{S} \times S$).

(3) As the divergence condition in Definition 4.5 only refers to non-probabilistic actions and agrees with the third condition of visible simulations in non-probabilistic systems, condition (3) can be established applying exactly the same arguments as in the non-probabilistic case [10,18]. $\square$

Theorem 4.11 together with Proposition 4.6 completes the proof of Theorem 4.2.

5 Partial order reduction versus process equivalences

In this section we give a brief overview of the connections between the partial order reduction criteria presented here and in the papers [5] and [1] on the one hand and probabilistic process equivalences on the other hand. With suitable notions of stutter equivalence, simulation and bisimulation equivalence (see below) we have:

(a) If conditions (A0)-(A3) and (A4.2) hold then $\mathcal{M}$ and $\hat{\mathcal{M}}$ are stutter equivalent (see [1]), but in general $\hat{\mathcal{M}}$ does not simulate $\mathcal{M}$.

(b) If conditions (A0)-(A3) and (A4.1) hold then $\mathcal{M}$ and $\hat{\mathcal{M}}$ are simulation equivalent (see [5]), but in general not bisimilar.

(c) If conditions (A0)-(A3) and (A4) hold then $\mathcal{M}$ and $\hat{\mathcal{M}}$ are bisimilar.

(c) is our Theorem 4.11, the underlying notion of bisimulation is probabilistic visible bisimulation (as defined in Def. 4.5) and could also be replaced with a divergence-sensitive, state-based variant of probabilistic branching bisimulation defined in the style of [22].

The underlying notion of *stutter equivalence* essentially agrees with *trace distribution equivalence* [21] (reformulated for our model and state labels rather than action labels). Stutter equivalence for paths identifies those paths ς_1 and ς_2 where $\mathsf{trace}(\varsigma_1)$ and $\mathsf{trace}(\varsigma_2)$ arise from the same sequence of labels by repetition of state-labels, that is $\mathsf{trace}(\varsigma_1) = \ell_1^{k_1}, \ell_2^{k_2}, \ldots$ and $\mathsf{trace}(\varsigma_2) = \ell_1^{n_1}, \ell_2^{n_2}, \ldots$ where $\ell_1\ell_2\ldots$ is an infinite word over 2^{AP} and k_i, $n_i \geq 1$. Two MDPs $\mathcal{M}$ and $\mathcal{M}'$ are said to be stutter equivalent if for any scheduler D for $\mathcal{M}$ there is a scheduler D' for $\mathcal{M}'$ such that their probability measures agree on all measurable unions of stutter equivalence classes (of paths starting in the initial state of $\mathcal{M}$ and $\mathcal{M}'$ respectively). In particular, stutter equivalent MDPs satisfy the same quantitative $\mathrm{LTL}_{\backslash X}$ properties.

The underlying simulation relation has been formally defined in [5] and is a variant of *probabilistic forward simulation* as introduced by Segala [21]. This kind of simulation allows a state s to be simulated by a distribution over states (rather than a single state). For the example in Figure 3, $\mathcal{M}$ and $\hat{\mathcal{M}}_1$ are simulation equivalent. The intuitive argument why $\hat{\mathcal{M}}_1$ can simulate $\mathcal{M}$ is that state s is simulated by the distribution that assigns probability 1/2 to the two a-states in $\hat{\mathcal{M}}_1$. Thus, (A0)-(A3) and (A4.1) guarantee the equivalence of $\mathcal{M}$ and $\hat{\mathcal{M}}_1$ up to *trace distribution congruence* [21], and thus, they are suitable for compositional reasoning.

In fact, in Figure 3, $\mathcal{M}$ and $\hat{\mathcal{M}}_1$ are not bisimilar because there is *no state* in $\hat{\mathcal{M}}_1$ that corresponds to state s in $\mathcal{M}$. Thus, Figure 3 yields an example for a reduction satisfying (A0)-(A3) and (A4.1) where $\mathcal{M}$ and $\hat{\mathcal{M}}_1$ are not bisimilar (as stated in (b)). Figure 5 illustrates a reduction satisfying (A0)-(A3) and

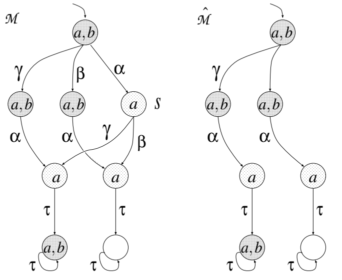

Fig. 5. **(A0)-(A3) and (A4.2) hold, but $\hat{\mathcal{M}}$ does not simulate $\mathcal{M}$**

(A4.2) where $\hat{\mathcal{M}}$ does not simulate $\mathcal{M}$ (as stated in (a)). Here, $\mathcal{M}$ and $\hat{\mathcal{M}}$ do not contain probabilistic actions, and hence, can be viewed as ordinary transition systems. The intuitive argument why $\hat{\mathcal{M}}$ does not simulate $\mathcal{M}$ is that there is no possibility to mimic the nondeterministic choice in state s via a probabilistic choice over the two a-states in $\hat{\mathcal{M}}$. Note that the schedulers for $\mathcal{M}$ in the upper a-state s might choose β and γ (and thus, combine the two lower a-states) with arbitrary probabilities while probabilistic forward simulation would require a *fixed* probabilistic distribution over the two lower a-states to mimic the possible behaviors of s (which is not possible).

In (a) and (b), the cycle condition (A3) can be replaced with the weaker end component condition (A3'), while the switch from (A3) to (A3') is irrelevant for (c) as explained in Section 4. Statement (a) with (A3') rather than (A3) is the original formulation in [1]. In (b), the end component condition (A3') requires a notion of probabilistic forward simulation that allows for (certain) infinite computations to simulate a single transition, while for the cycle condition (A3) a simpler version of simulation suffices where any transition of the simulated process has to be matched by a finite computation tree of the simulating process. The approach of [5] works with the cycle condition (A3) and a formalization of finite computation trees by means of SOS-rules. Yet, to deal with (A3') and possibly infinite computation trees a further rule that captures the semantics of infinite behaviors could be added.

6 Conclusion

In this paper, we presented partial order criteria to preserve probabilistic branching time properties. This is of theoretical relevance, since it represents the branching counterpart to the linear approach of [1,5] and is the natural probabilistic extension of the reduction techniques for CTL [10]. Although we cannot yet report on experimental results, we expect that our results have also some impact under practical aspects as explained in the introduction. Thus our results can be seen as an alternative to the symbolic BDD-based methods used e.g. in the PCTL model checker PRISM [15], but they can also be applied in combination with symbolic methods using a static variant of the partial order reduction criteria as in [14].

On the practical side we are currently implementing a model checker for quantitative LTL and PCTL and intend to integrate the partial order techniques presented here and in [1,5].

Further directions include a more detailed discussion about partial order reduction criteria and process equivalences. We will have a deeper look at the action-labeled case and study which variants of Valmari's conditions for various non-probabilistic process equivalences [23,24] can be adapted to the probabilistic case.

References

[1] C. Baier, M. Größer, and F. Ciesinski. Partial order reduction for probabilistic systems. In *Proc. 1st QEST*, pages 230–239, IEEE Computer Society Press, 2004.

[2] Christel Baier. *On the algorithmic verification of probabilistic systems*. Universität Mannheim, 1998. Habilitation Thesis.

[3] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *FST & TCS*, *LNCS* 1026, pages 499–513, 1995.

[4] E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, April 1986.

[5] P.R. D'Argenio and P. Niebert. Partial order reduction on concurrent probabilistic programs. In *Proc. 1st QEST*, pages 240–249, IEEE Computer Society Press, 2004.

[6] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.

[7] L. de Alfaro. Stochastic transition systems. In *Proc. 9th CONCUR*, *LNCS* 1466, pages 423–438, 1998.

[8] J. Desharnais. *Labelled Markov Processes*. PhD thesis, McGill University, 1999.

[9] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Weak bisimulation is sound and complete for PCTL*. In *Proc. CONCUR 2002*, *LNCS* 2421.

[10] 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 Computer Society Press, 1995.

[11] H. Hansson. *Time and Probability in Formal Design of Distributed Systems*. Series in Real-Time Safety Critical Systems. Elsevier, 1994.

[12] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[13] G. Holzmann and D. Peled. An improvement in formal verification. In *Proc. Formal Description Techniques, FORTE94*, pages 197–211, Berne, Switzerland, October 1994. Chapman & Hall.

[14] R. Kurshan, V. Levin, M. Minea, D. Peled, H. Yenign. Static partial order reduction. In *Proc. TACAS*, *LNCS* 1384:345–357, 1998.

[15] M. Kwiatkowska, G. Norman, D. Parker. Probabilistic symbolic model checking with PRISM: A hybrid approach. In *International Journal on Software Tools for Technology Transfer (STTT)*, volume 6, number 2, pages 128-142, 2004.

[16] K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

[17] D. Peled. All from one, one for all: On model checking using representatives. In *Proc. 5th CAV*, *LNCS* 697, pages 409–423, 1993.

[18] D. Peled. Partial order reduction: Linear and branching time logics and process algebras. In [19], pages 79–88, 1996.

[19] D. Peled, V. Pratt, and G. Holzmann, editors. *Partial Order Methods in Verification*, volume 29(10) of *DIMACS*. American Mathematical Society, 1997.

[20] M. L. Puterman. *Markov Decision Processes—Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, 1994.

[21] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, Massachusetts Institute of Technology, 1995.

[22] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.

[23] A. Valmari. A stubborn attack on state explosion. *Formal Methods in System Design*, 1:297–322, 1992.

[24] A. Valmari. Stubborn set methods for process algebras. In [19], pages 79–88, 1996.

[25] R. van Glabbeek, S. Smolka, B. Steffen, and C. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Proc. 5th LICS*, pages 130–141. IEEE Computer Society Press, 1990.