

Why Confluence is More Powerful than Ample Sets in Probabilistic and Non-Probabilistic Branching Time

Henri Hansen^a, Mark Timmer^b

^a *Department of Software Systems, Tampere University of Technology
PO Box 553, FI-33101 Tampere, Finland
Email: henri.hansen@tut.fi*

^b *Formal Methods and Tools, Faculty of EEMCS
University of Twente, The Netherlands
Email: timmer@cs.utwente.nl*

Abstract

Confluence reduction and partial order reduction by means of ample sets are two different techniques for state space reduction in both traditional and probabilistic model checking. This paper provides an extensive comparison between these two methods, answering the long-standing question of how they relate. We prove that, while both preserve branching time properties, confluence reduction is strictly more powerful than partial order reduction: every reduction that can be obtained with partial order reduction can also be obtained with confluence reduction, but the converse is not true.

A core problem in the comparison is that confluence reduction was defined in an action-based setting, whereas partial order reduction was defined in a state-based setting. We therefore redefine confluence reduction in the state-based setting of Markov decision processes, and provide a nontrivial proof of its correctness. Additionally, we pinpoint precisely in what way confluence reduction is more general, and provide a restricted variant of confluence and relaxed variant of partial order reduction that exactly coincide. Clearly, the results we present also hold for non-probabilistic models, as they can just as well be applied in a context where all transitions are non-probabilistic.

To show the practical applicability of our results, we adapt a state space generation technique based on representative states, already known in combination with confluence reduction, so that it can also be applied with partial order reduction.

Keywords: Confluence reduction, Partial order reduction, Ample sets, Probabilistic branching time, Markov decision processes

1. Introduction

Probabilistic model checking has proved to be an effective way for improving the quality of communication protocols and encryption techniques, but also for studying biological systems or measuring the performance of networks. The omnipresent state space explosion poses a serious threat to the efficiency of model checking and similar methods; therefore, several reduction techniques have been introduced to deal with large systems.

Recently, two powerful reduction techniques from non-probabilistic model checking were generalised to the probabilistic setting: *partial order reduction* [1, 2, 3] and *confluence reduction* [4, 5]. Both use a notion of independence between transitions of a system, either explicitly or implicitly, and try to reduce the state space by eliminating redundant paths through the system (and therefore often also states). In the non-probabilistic setting, partial order reduction techniques have been defined for a large range of property classes, most notably variants that preserve $LTL_{\setminus X}$ and $CTL_{\setminus X}^*$ [6, 7, 8, 9]. Most work on confluence reduction has been designed such that the reduced system is branching bisimilar to the original system; thus, these techniques preserve virtually all branching properties (in particular, $CTL_{\setminus X}^*$). There is not as much work on weaker variants of confluence, though in [10] a variant is explored that makes no distinction between visible

and invisible actions and does not require acyclicity. The variant preserves deadlocks much in the same way as weaker versions of ample and stubborn sets [8].

Partial order reduction, in the form of *ample sets*, was the first of these methods to be applied in the probabilistic setting. In [11] and [12], the concept was lifted from labelled transition systems to Markov decision processes (MDPs), providing reductions that preserve quantitative $\text{LTL}_{\setminus X}$. These techniques were refined in [13] to also preserve probabilistic $\text{CTL}_{\setminus X}^*$, a branching logic. Later, a revision of partial order reduction for distributed schedulers was introduced and implemented in PRISM [14]. In [15], the use of fairness constraints in combination with ample sets for the quantitative analysis of MDPs was first introduced. Later, the so-called weak stubborn set method was also defined for a class of safety properties of MDPs under fairness constraints [16].

Recently, confluence reduction was lifted to the probabilistic realm as well. In [17, 18] a probabilistic variant was introduced that, just like the ample set reduction of [13], preserves branching properties. It was defined as a reduction technique for action-based probabilistic automata [19], but as we will show in this paper, it can also be used in the context of MDPs.

Ample sets and confluent transitions are defined and detected quite differently: ample sets are defined by first giving an independence relation for the action labels, whereas confluence is a property of a set of (invisible) transitions in the final state space. Even so, the underlying ideas are similar on the intuitive level. Therefore, an obvious question is: to what extent do they indeed coincide? This paper addresses that question by comparing the notion of probabilistic ample sets from [13] to the notion of strongly probabilistically confluent sets from [17]. We restrict to ample sets, because they are currently the most well-established notion of partial order reduction for MDPs.

Contributions. We first redefine confluence for MDPs. The task is nontrivial, because confluence is originally defined in a purely action-based formalism. We show that when preserving branching time behaviour, confluence reduction is strictly more powerful than ample set reduction, by proving that every nontrivial ample set can be mimicked by a confluent set, while also providing examples where confluent transitions do not qualify as ample sets. In such cases, confluence reduction is able to reduce more than ample set reduction. To continue, we pinpoint precisely in what way confluence is more general than ample sets, and restrict the definition of confluence as well as relax the definition of ample sets, to make them coincide.

While revealing exactly where the extra reduction with confluence comes from, the results we present support the idea that confluence reduction is a well-suited alternative to the thus far more often used partial order reduction methods. In particular, this is a major consideration in contexts where (1) detection of confluence using heuristics that make use of these more relaxed conditions is possible, or where (2) the conditions of confluence are just easier to check than their partial order reduction counterparts.

The first situation seems to occur in the context of statistical model checking. In [20], partial order reduction is used to remove spurious nondeterminism from models to allow them to be analysed statistically. As the reduction takes place on the full explicit model rather than a high-level specification, however, the more relaxed confluence conditions could also be applied. The authors of [20] indeed expressed the feeling that confluence reduction might be able to remove more nondeterminism than partial order reduction, thereby allowing more models to be analysed using their statistical model checking techniques. Additionally, they expect that the confluence conditions might be easier to check than the independence condition of partial order reduction. Hence, this is a promising direction for future work.

The second situation seems to arise when working with process-algebraic modelling languages. As demonstrated in [4] for the non-probabilistic and in [17] for the probabilistic setting, it is quite natural to detect confluence in such a context.

Alternatively, the relaxed definition of ample sets might be used in settings where the notion of partial order reduction is more natural. In addition to providing these practical opportunities, our precise comparison of confluence and partial order reduction fills a significant gap in the theoretical understanding of the two notions.

The theory is presented in such a way, that the results hold for non-probabilistic automata as well, as they form a special case of the theory where all probability distributions are deterministic. Hence, as a

side effect we also answered the long-standing question of how the non-probabilistic variants of partial order reduction and confluence reduction relate.

Our findings imply that results and techniques applicable to confluence can be used in conjunction with ample sets. As an example of such a technique, we show how a state space generation technique based on *representative states*, already known in the context of confluence reduction [4], can also be applied with partial order reduction. This is a very general technique for replacing a class of states by a single representative, and a quite similar method has also been used in conjunction with the so-called essential state abstraction in [21]. The technique replaces explicit checking of the cycle condition, in addition to further reducing the number of states and transitions. The latter is important, especially if the MDP is to be subjected to further analysis.

Overview of the paper. After recalling some basic preliminaries in Section 2, we present the notions of partial order reduction and confluence reduction in Section 3, also showing that confluence reduction for MDPs preserves $\text{PCTL}_{\setminus X}^*$ in the same way as ample sets. Then, in Section 4 we discuss how ample set reduction can be thought of as a special case of confluence reduction. We show what kind of restrictions and relaxations are needed to make them coincide, thereby pinpointing the exact differences of the methods. In Section 5 we consider the use of the so-called representation map in the context of ample set reduction. Section 6 concludes the paper and provides directions for future work.

2. Preliminaries

Definition 1 (Probability distributions). *A probability distribution over a countable set S is a function $\mu: S \rightarrow [0, 1]$ such that $\sum_{s \in S} \mu(s) = 1$. The support of a distribution is given by $\text{spt}(\mu) = \{s \in S \mid \mu(s) > 0\}$, and we write $\mathbb{1}_t$ for the deterministic distribution μ determined by $\mu(t) = 1$. We use $\text{Distr}(S)$ to denote the set that contains all probability distributions over S and the subdistribution \perp that assigns probability 0 to every $s \in S$.*

The model on which probabilistic partial order reduction is defined is the Markov decision process. It consists of states that are labelled by atomic propositions, an initial state, and a probabilistic action-labelled transition function. From each state s , a subset of the actions is enabled; for every such action a , a probability distribution $P(s, a)$ specifies for each other state s' the likelihood $P(s, a)(s')$ of ending up in s' after taking action a from s .

Definition 2 (MDPs). *A Markov decision process (MDP) is tuple $M = (S, \Sigma, P, s^0, \text{AP}, L)$, where*

- S is a finite set of states;
- Σ is a finite set of action labels;
- $P: (S \times \Sigma) \rightarrow \text{Distr}(S)$ is the probabilistic transition function;
- $s^0 \in S$ is the initial state;
- AP is the set of atomic propositions;
- $L: S \rightarrow 2^{\text{AP}}$ is the labelling function.

If $P(s, a) = \perp$, the action a is not enabled from s . Otherwise, $P(s, a)(s')$ is the probability of going to s' when executing a from s .

We use several notions when working with MDPs. The next definition introduces the set of transitions of the MDP, and introduces the notation (s, a, μ) to denote a transition from s , taking an action a and having a next-state distribution μ . Also, we introduce a notation for paths through an MDP.

Definition 3 (Notations for MDPs). Given an MDP $M = (S, \Sigma, P, s^0, \text{AP}, L)$, we denote the set of all possible transitions of M by

$$\Delta_M = \{(s, a, \mu) \in S \times \Sigma \times \text{Distr}(S) \mid P(s, a) = \mu \neq \perp\},$$

and write $s \xrightarrow{a} s'$ if there exists a distribution $\mu \in \text{Distr}(S)$ such that $(s, a, \mu) \in \Delta_M$ and $s' \in \text{spt}(\mu)$. Moreover, we write $s \xrightarrow{a} s'$ if $s \xrightarrow{a} s'$ for some $s' \in S$, and define $\text{en}(s) = \{a \in \Sigma \mid s \xrightarrow{a}\}$.

We write $s \xrightarrow{a_1 a_2 \dots a_n} s'$ if there exists a sequence of states $s_0 s_1 \dots s_n$ such that $s_0 = s$, $s_n = s'$ and $s_i \xrightarrow{a_{i+1}} s_{i+1}$ for every $0 \leq i < n$, and write $s \xrightarrow{a_1 a_2 \dots a_n} s'$ if $s \xrightarrow{a_1 a_2 \dots a_n} s'$ for some $s' \in S$.

For a given MDP, a wide class of reductions can be defined using the construct called a *reduction function*. Informally, such a function decides for each state which outgoing actions are enabled in the reduced MDP. The transition function of the reduced MDP then consists of all transitions that are still enabled after the reduction function is applied, and the set of states consists of all states that are still reachable using the reduced transition function.

Definition 4 (Reduction functions). Given an MDP $M = (S_M, \Sigma, P_M, s^0, \text{AP}, L_M)$, a reduction function is any function $R: S \rightarrow 2^\Sigma$ with $R(s) \subseteq \text{en}(s)$ for every $s \in S$. Given a reduction function R , the reduced MDP for M with respect to R is the minimal MDP $M_R = (S_R, \Sigma, P_R, s^0, \text{AP}, L_R)$ such that

- If $s \in S_R$ and $a \in R(s)$, then $P_R(s, a) = P_M(s, a)$ and $\text{spt}(P_M(s, a)) \subseteq S_R$;
- If $s \in S_R$ and $a \notin R(s)$, then $P_R(s, a) = \perp$;
- $L_R(s) = L_M(s)$ for every $s \in S_R$,

where minimal should be interpreted as having the smallest set of states.

Given a reduction function $R: S \rightarrow 2^\Sigma$, we define $\bar{R}: S \rightarrow 2^\Sigma$ by

$$\bar{R}(s) = \begin{cases} \emptyset & \text{if } R(s) = \text{en}(s) \\ R(s) & \text{otherwise} \end{cases}$$

We say that a reduction function is acyclic if there are no cyclic paths when only transitions from $\bar{R}(s)$ are considered.

Note that \bar{R} assigns to each state s the subset of actions that are enabled by R in case a real reduction is made for s . Otherwise, it assigns no actions to s . The transitions associated with \bar{R} are called the *nontrivial transitions* of the reduction.

Example 5. Figure 1 visualises an MDP M , consisting of 14 states. This MDP will be used throughout the paper as a running example. It represents a flow chart, specifying the way in which six tasks can be performed. The tasks occur in pairs: first $task_1$ and $task_2$ need to be executed, then $task_3$ and $task_4$, and finally we need to do $task_5$ and $task_6$. Each pair of tasks can be executed in either order. Furthermore, the execution of $task_2$ fails with probability $\frac{1}{10}$, in which case it can be attempted again. Moreover, after finishing the first two tasks and before starting the last two, we can quit or choose to continue. Finally, after all tasks have been completed, it is allowed to repeat either $task_5$ or $task_6$. We assume that the effect of the even-numbered tasks is visible to the environment (indicated by a change of atomic proposition due to such a transition).

Note that for this MDP we have $S = \{s_i \mid 0 \leq i \leq 13\}$ and $\Sigma = \{task_i \mid 1 \leq i \leq 6\} \cup \{quit, continue\}$. The probabilistic transition function is visualised by arrows. For instance, $P(s_0, task_2) = \mu$ such that $\mu(s_0) = \frac{1}{10}$ and $\mu(s_2) = \frac{9}{10}$, and $P(s_0, quit) = \perp$. Furthermore, we have $s^0 = s_0$ and $\text{AP} = \{p, q, r, s, t, u\}$. The labelling is indicated for each state, e.g., $L(s_2) = \{q\}$. We have $s_1 \xrightarrow{task_2} s_3$, for instance, and $\text{en}(s_3) = \{quit, task_3, task_4\}$ as well as $s_5 \xrightarrow{task_3 \text{ continue } task_5 \text{ } task_6} s_5$.

A possible reduction function R for this MDP is given by $R(s_0) = \{task_1\}$, $R(s_1) = \{task_2\}$, $R(s_3) = \emptyset$ and $R(s_i) = \text{en}(s_i)$ for every other state. The reduced MDP with respect to R consists of solely the states s_0 , s_1 and s_3 and the two transitions connecting them. We have $\bar{R}(s_0) = \{task_1\}$ and $\bar{R}(s_1) = \emptyset$, and find that R is acyclic (which is immediate, as there is only one nontrivial transition). \square

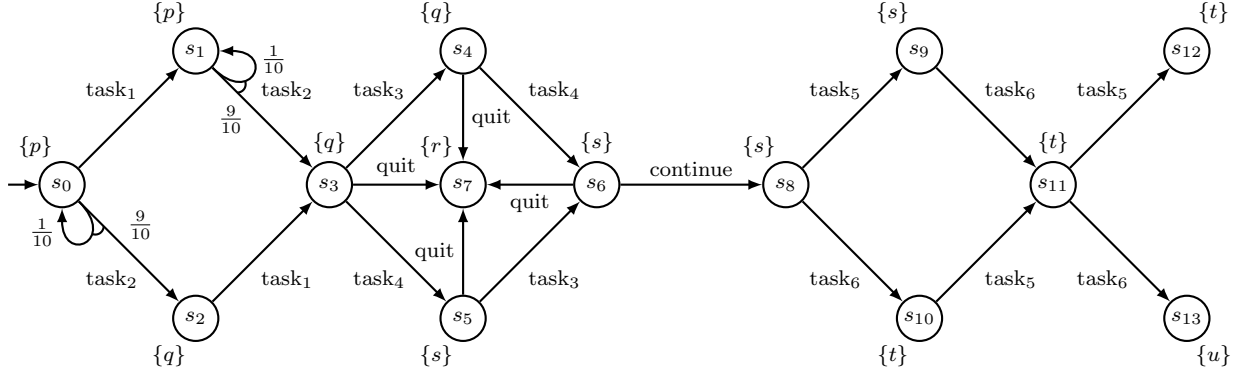


Figure 1: An MDP M representing a flow chart.

When reducing MDPs, we clearly want to retain some behaviour in order to still be able to verify certain properties. For the two reductions we deal with in this paper, $\text{PCTL}_{\setminus X}^*$ is preserved (a probabilistic variant of $\text{CTL}_{\setminus X}^*$; see for instance [22]).

3. Ample Sets and Confluence for MDPs

This section presents the theory of the ample set reduction and confluence reduction techniques. First, we introduce the concepts of deterministic, stuttering and invisible transitions, actions and paths. Also, we need the concepts of weight functions and probabilistic visible (bi)simulation [23], as they will be used to prove that our redefined variant of confluence for MDPs also preserves $\text{PCTL}_{\setminus X}^*$.

Definition 6 (Determinism, Stuttering and Invisibility). *Given an MDP $M = (S, \Sigma, P, s^0, \text{AP}, L)$,*

- A transition $(s, a, \mu) \in \Delta_M$ is deterministic if μ is deterministic, and an action $a \in \Sigma$ is a deterministic action if all a -labelled transitions are deterministic. We denote the set of all deterministic actions by $\Sigma_{\text{det}} \subseteq \Sigma$. Given a deterministic transition $(s, a, \mathbb{1}_t)$, we write $\text{target}(s, a) = t$;
- A transition $(s, a, \mu) \in \Delta_M$ is stuttering if $L(s') = L(s)$ for each $s' \in \text{spt}(\mu)$, and an action $a \in \Sigma$ is a stuttering action if all a -labelled transitions are stuttering. We denote the set of all stuttering actions by $\Sigma_{\text{st}} \subseteq \Sigma$;
- A transition $(s, a, \mu) \in \Delta_M$ is invisible if it is both deterministic and stuttering, and an action $a \in \Sigma$ is an invisible action if all a -labelled transitions are invisible. We denote the set of all invisible actions by $\Sigma_{\text{inv}} = \Sigma_{\text{st}} \cap \Sigma_{\text{det}}$;
- A finite path $s \xrightarrow{a_1 a_2 \dots a_n} s'$ or infinite path $s \xrightarrow{a_1 a_2 \dots}$ is invisible if every action a_i is invisible.

Note that (s, a, μ) may be an invisible transition even if a is not an invisible action, but not vice versa. Also note that given a sequence of invisible (and thus deterministic) actions $a_1 a_2 \dots a_n$, talking about “the path” of this sequence from some state s makes sense, because the states that are visited are unique. We do so for the rest of this paper.

Example 7. In the MDP M given in Figure 1, all transitions except for the two task_2 transitions are deterministic. Hence, all actions except for task_2 are deterministic. All transitions labelled by odd tasks are stuttering, as well as the continue transition, since the atomic propositions in their source and target states all correspond. Hence, all odd-labelled task actions and the continue action are stuttering. Combining this, we obtain $\Sigma_{\text{inv}} = \{\text{task}_i \mid i \in \{1, 3, 5\}\} \cup \{\text{continue}\}$. \square

Definition 8 (Weight functions). Let $\mathcal{R} \subseteq S_1 \times S_2$ be a binary relation and let $\mu \in \text{Distr}(S_1)$ and $\nu \in \text{Distr}(S_2)$ be probability distributions. We write $\mu \sqsubseteq_{\mathcal{R}} \nu$ if there exists a weight function $w: S_1 \times S_2 \rightarrow [0, 1]$ such that for all $s_1 \in S_1$ and $s_2 \in S_2$,

- $w(s_1, s_2) > 0$ implies $(s_1, s_2) \in \mathcal{R}$;
- $\sum_{s \in S_2} w(s_1, s) = \mu(s_1)$ and $\sum_{s \in S_1} w(s, s_2) = \nu(s_2)$.

Definition 9 (Probabilistic visible bisimulation). Let $M_1 = (S_1, \Sigma, P_1, s_1^0, \text{AP}, L_1)$ and $M_2 = (S_2, \Sigma, P_2, s_2^0, \text{AP}, L_2)$ be MDPs, and let $\mathcal{R} \subseteq S_1 \times S_2$ be a binary relation. Then, \mathcal{R} is a probabilistic visible simulation for (M_1, M_2) if $(s_1^0, s_2^0) \in \mathcal{R}$ and, for every $(s, s') \in \mathcal{R}$,

1. $L_1(s) = L_2(s')$;
2. If $a \in \text{en}(s)$, then either
 - (a) $a \in \Sigma_{\text{inv}}$ and $(\text{target}(s, a), s') \in \mathcal{R}$, or
 - (b) there is an invisible path $s' \xrightarrow{b_1 \dots b_n} s''$ in M_2 such that $(s, s'_i) \in \mathcal{R}$ for every state s'_i on this path, $a \in \text{en}(s'')$ and $P_1(s, a) \sqsubseteq_{\mathcal{R}} P_2(s'', a)$;
3. If there is an infinite invisible path $s \xrightarrow{b_1 b_2 \dots}$ in M_1 such that $(s_i, s'_i) \in \mathcal{R}$ for every s_i on this path, then there is a finite invisible path $s' \xrightarrow{a_1 \dots a_n} s'_n$ in M_2 , $n \geq 1$, such that $(s, s'_i) \in \mathcal{R}$ for every s'_i on this path (possibly excluding s'_n), and $(s_k, s'_n) \in \mathcal{R}$ for at least one s_k on the path $s \xrightarrow{b_1 b_2 \dots}$.

A binary relation \mathcal{R} is a probabilistic visible bisimulation for (M_1, M_2) if it is a probabilistic visible simulation for (M_1, M_2) and \mathcal{R}^{-1} is a probabilistic visible simulation for (M_2, M_1) .

We say that two MDPs M_1, M_2 are probabilistically visibly bisimilar, denoted by $M_1 \equiv_{\text{pvb}} M_2$, if there is a probabilistic visible bisimulation that relates them.

3.1. Ample sets

Although there are many techniques that are called “partial order reduction”, we focus on the ample set method as presented in [13], as it is the most well known and the only one we are aware of that has been defined so as to preserve probabilistic branching time properties. To present the definition, we first need to introduce the notion of independence. Intuitively, two actions a, b are independent if they don’t disable each other, and if the probability of ending up at any state by first taking a and then taking b is the same as when the actions are taken the other way around.

Definition 10 (Independence). Given an MDP $M = (S, \Sigma, P, s^0, \text{AP}, L)$, two actions $a, b \in \Sigma$ are independent if $a \neq b$ and for every state $s \in S$ such that $\{a, b\} \subseteq \text{en}(s)$ the following conditions hold:

- If $s' \in \text{spt}(P(s, a))$, then $b \in \text{en}(s')$ (and symmetrically);
- $\sum_{s' \in S} P(s, a)(s') \cdot P(s', b)(t) = \sum_{s' \in S} P(s, b)(s') \cdot P(s', a)(t)$, for every $t \in S$.

If a and b are not independent, we say that they are dependent. An action a is dependent on a set B if there exists at least one $b \in B$ on which a depends.

Based on this notion of dependence, the ample set constraints can be defined. We refer to [23] for an extended explanation of these conditions.

Definition 11 (Ample set reduction). Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP without any terminal states. Then, a reduction function $A: S \rightarrow 2^{\Sigma}$ for M is an ample set reduction function if it satisfies the following conditions in every state $s \in S$:

A0 $\emptyset \neq A(s) \subseteq \text{en}(s)$;

A1 If $A(s) \neq \text{en}(s)$, then $A(s) \subseteq \Sigma_{\text{st}}$;

A2 For every path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n \xrightarrow{b} t$ in M such that $b \notin A(s)$ and b depends on $A(s)$, there exists an $1 \leq i \leq n$ such that $a_i \in A(s)$;

A3 For every path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n$ in M_A with $s_n = s$, $A(s_i) = en(s_i)$ for at least one $1 \leq i \leq n$;

A4 If $A(s) \neq en(s)$, then $|A(s)| = 1$ and $A(s) \subseteq \Sigma_{\text{det}}$.

The sets $A(s)$ are called ample sets.

Note that we could also choose to allow MDPs *with* terminal states. In that case A0 should be changed to allow $A(s) = \emptyset$ if $en(s) = \emptyset$. Note also that conditions A1 and A4 can be combined by saying that either $A(s) = en(s)$ or $A(s)$ contains exactly one invisible action.

Example 12. In the MDP M given in Figure 1, the actions $task_1$ and $task_2$ are independent. After all, there is only one state in which both are enabled: s_0 . From there, indeed, these two actions do not disable each other. Moreover, when first executing $task_1$ and then executing $task_2$, the probability of ending up in s_1 is $\frac{1}{10}$ and the probability of ending up in s_3 is $\frac{9}{10}$. When executing the tasks the other way around, we obtain the same probabilities.

Similarly, it can be shown that $task_3$ and $task_4$ are independent. Note that $task_5$ and $task_6$ are not independent, as they are both enabled in s_{11} and from there can disable each other.

A valid ample set reduction function A for M is given by $A(s_0) = \{task_1\}$ and $A(s_i) = en(s_i)$ for all other states. Note that all ample set conditions vacuously hold for all fully-expanded states, so we only need to investigate s_0 . The conditions A0, A1 and A4 are trivial to verify. Also A3 is easy, since the only possible cycle in M_A is an infinite loop through s_1 (although this has probability 0): indeed $A(s_1) = en(s_1)$. Finally, to see why A2 holds, note that every path from s_0 either immediately traverses $task_1$ (which is indeed in $A(s_0)$) or starts with $task_1 task_2$; for all traces of the second kind, $task_1$ is independent of $A(s_0)$ and $task_2$ is in $A(s_0)$, satisfying the condition.

This reduction function only gets rid of state s_2 . Note that no additional reduction is possible. In s_3 , s_4 , s_5 and s_6 , no subset of the enabled actions can be chosen as an ample set, since none of the actions is independent of the *quit* action (as *quit* disables all other actions). Also, in s_8 no reduction is possible, since $task_5$ and $task_6$ are not independent. After all, in state s_{11} they can disable each other. \square

The following result from [13] indicates why ample sets are sound for MDP reduction.

Theorem 13. *If A is an ample set reduction function for M , then $M \equiv_{\text{pvb}} M_A$, and consequently M and M_A satisfy the same PCTL_X^* -formulae.*

3.2. Confluence

Confluence for action-based probabilistic automata was introduced in [17], and here we reformulate the theory in terms of MDPs in order to compare it to the ample set method. First, we need to introduce the notion of equivalence up to \mathcal{T} -steps: a way of saying that two probability distributions are basically the same, except possibly for some intermediate transitions from a set \mathcal{T} .

Definition 14 (Equivalence up to \mathcal{T} -steps). *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, $\mathcal{T} \subseteq \Delta_M$ a set of deterministic transitions of M , and $\mu, \nu \in \text{Distr}(S)$ two probability distributions. Then, we say that μ is equivalent up to \mathcal{T} -steps to ν , denoted by $\mu \rightsquigarrow_{\mathcal{T}} \nu$, if $\mu, \nu \neq \perp$ and there exists a partitioning $\text{spt}(\mu) = \bigsqcup_{i=1}^n S_i$ of the support of μ and an ordering $\text{spt}(\nu) = \{s_1, \dots, s_n\}$ of the support of ν , such that*

$$\forall 1 \leq i \leq n . \mu(S_i) = \nu(s_i) \wedge (S_i = \{s_i\} \vee (\forall s \in S_i . \exists a \in \Sigma . (s, a, \mathbb{1}_{s_i}) \in \mathcal{T})).$$

With respect to the notion of equivalence up to τ_c -steps of [17] this definition is slightly more general, as we allow states in the support of μ to directly correspond to states in the support of ν , without requiring a \mathcal{T} -step in between. As a result, the probabilistic variant of strong confluence that we define later on precisely corresponds to the nonprobabilistic variant if all transitions are deterministic.

Example 15. Consider the MDP in Figure 2, and let $\mathcal{T} = \{(s_0, a, s_1), (s_2, a, s_6), (s_3, a, s_5), (s_4, a, s_5)\}$. Moreover, let $\mu = P(s_0, b)$ and $\nu = P(s_1, b)$. It now follows that $\mu \rightsquigarrow_{\mathcal{T}} \nu$, by taking the partitioning $\text{spt}(\mu) = S_1 \cup S_2$ with $S_1 = \{s_2\}$ and $S_2 = \{s_3, s_4\}$ and the ordering $\text{spt}(\nu) = \{s_6, s_5\}$. Now, indeed $\mu(S_1) = \frac{1}{3} = \nu(s_6)$ and $\mu(S_2) = \frac{2}{3} = \nu(s_5)$. Also, there is a transition in \mathcal{T} connecting s_2 to s_6 , and there are transitions in \mathcal{T} connecting s_3 and s_4 to s_5 .

Note that it also would have been fine if, for instance, s_0 directly went to s_6 instead of s_2 with probability $\frac{1}{3}$ as part of the b -transition. \square

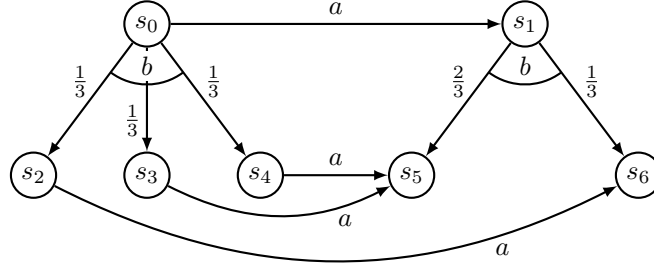


Figure 2: An MDP to demonstrate $\rightsquigarrow_{\mathcal{T}}$.

The next lemma states that, given a deterministic transition $(s, a, \mathbf{1}_{s'})$, the distribution from s associated with an action b independent of a is equivalent up to $\{(t, a, t')\}$ -steps to the distribution associated with the same action from s' .

Lemma 16. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, and let $a, b \in \Sigma$ two independent actions such that $a \in \Sigma_{\text{det}}$. Let $s \in S$ such that $\{a, b\} \subseteq \text{en}(s)$, and assume that $s \xrightarrow{a} s'$. If \mathcal{T} contains all outgoing a -transitions from states in the support of $P(s, b)$, i.e., $\mathcal{T} \supseteq \{(t, a, \mu) \in \Delta_M \mid t \in \text{spt}(P(s, b))\}$, then $P(s, b) \rightsquigarrow_{\mathcal{T}} P(s', b)$.*

Proof. For any $t \in \text{spt}(P(s', b))$, let $R_t = \{r \in \text{spt}(P(s, b)) \mid r \xrightarrow{a} t\}$ be the set of states that might be reached after the action b from s and can reach t by an a -action. As a and b are independent, R_t is not empty, and when taking into account the assumption that a is deterministic it follows that $\{R_t \mid t \in \text{spt}(P(s', b))\}$ is a partitioning of $\text{spt}(P(s, b))$. Now, indeed

$$\begin{aligned} P(s', b)(t) &= \sum_{s'' \in S} P(s, a)(s'') \cdot P(s'', b)(t) = \sum_{s'' \in S} P(s, b)(s'') \cdot P(s'', a)(t) \\ &= \sum_{s'' \in R_t} P(s, b)(s'') \cdot P(s'', a)(t) = P(s, b)(R_t). \end{aligned}$$

The first equality follows from the fact that a is deterministic, the second from the independence of a and b , the third from the definition of R_t and the fourth from the fact that a is deterministic.

Also, by definition of R_t and \mathcal{T} and the fact that $a \in \Sigma_{\text{det}}$, we have $\forall s \in R_t. \exists a \in \Sigma. (s, a, \mathbf{1}_t) \in \mathcal{T}$. \square

Using the \rightsquigarrow -relation introduced above, strong probabilistic confluence for MDPs can be defined easily. We define strong probabilistic confluence for sets of transitions \mathcal{T} , and require every transition in such a set to be invisible. Moreover, visible actions that were enabled before a confluent transition should still be enabled after that transition. Also, if a transition (s, b, μ) is to be mimicked by a transition (s', b, ν) , then there should be confluent transitions connecting μ and ν as defined by the relation $\rightsquigarrow_{\mathcal{T}}$. As an exception, transitions with invisible actions and having the same source and target state as a confluent transition do not have to be mimicked.

Definition 17 (Strong probabilistic confluence). *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP. A set $\mathcal{T} \subseteq \Delta_M$ of transitions of M is strongly probabilistically confluent if for every $(s, a, \mu) \in \mathcal{T}$ it holds that $a \in \Sigma_{\text{inv}}$ and*

- For every $b \in \text{en}(s)$, either $P(s, b) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s, a), b)$, or $b \in \Sigma_{\text{inv}}$ and $P(s, b) = \mu$.

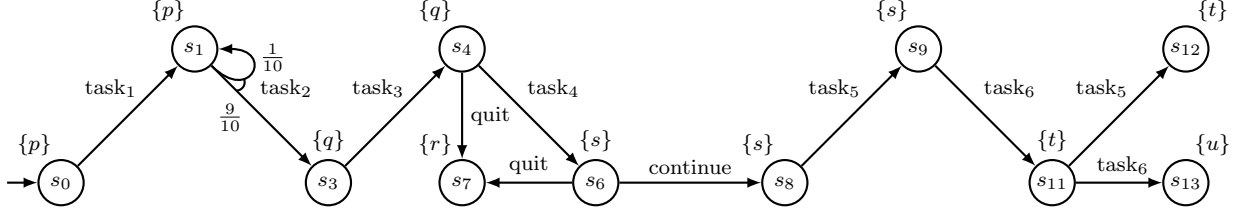


Figure 3: A reduced MDP.

A transition $(s, a, \mu) \in \Delta_M$ is said to be *strongly probabilistically confluent* if there exists a strongly probabilistically confluent set \mathcal{T} such that $(s, a, \mu) \in \mathcal{T}$.

Note that, although $P(s, b) = \mu$ implies that the b -transition from s is invisible, we do still need to require that $b \in \Sigma_{\text{inv}}$, since invisible transitions do not always have invisible actions (and this invisibility of the actions is later on required to prove probabilistic visible bisimulation).

From now on, we will use *confluent* as an abbreviation of strongly probabilistically confluent.

Definition 18 (Confluence reduction). *Given an MDP $M = (S, \Sigma, P, s^0, \text{AP}, L)$, a reduction function $T: S \rightarrow 2^\Sigma$ is a confluence reduction function for M if there exists some confluent set $\mathcal{T} \subseteq \Delta_M$ such that, for every $s \in S$,*

- if $T(s) \neq \text{en}(s)$, then $T(s) = \{a\}$ for some $a \in \Sigma_{\text{inv}}$ such that $(s, a, \mathbf{1}_{\text{target}(s,a)}) \in \mathcal{T}$.

In such a case, we also say that T is a confluence reduction function under \mathcal{T} .

Note that, in every state, a confluence reduction function either fully explores all outgoing transitions or explores only one of them (which is then required to be confluent). This way, the possibility exists that confluent transitions are taken indefinitely, ignoring the presence of other actions. This problem is well known in the theory of partial order reduction as the *ignoring problem* [24], and is dealt with by the cycle condition A3 of the ample set method. We can just as easily deal with it in the context of confluence reduction by requiring the reduction function to be acyclic. In Section 5 we will look at an alternative approach.

Example 19. Consider again the MDP M given in Figure 1. We define $\mathcal{T} = \{(s_0, \text{task}_1, s_1), (s_2, \text{task}_1, s_3), (s_3, \text{task}_3, s_4), (s_5, \text{task}_3, s_6), (s_8, \text{task}_5, s_9), (s_{10}, \text{task}_5, s_{11})\}$. Note that, indeed, all of these transitions are invisible. Moreover, it is easy to verify that, for instance, $P(s_0, \text{task}_2) \rightsquigarrow_{\mathcal{T}} P(s_1, \text{task}_2)$. This is the only proof obligation for the transition $(s_0, \text{task}_1, s_1)$ in \mathcal{T} . For $(s_2, \text{task}_1, s_3)$ there is nothing we have to prove, since there are no other transitions from s_2 .

Note that $(s_5, \text{task}_3, s_6)$ is a valid element of \mathcal{T} , since $P(s_5, \text{quit}) \rightsquigarrow_{\mathcal{T}} P(s_6, \text{quit})$. After all, both of these probability distributions assign probability 1 to s_7 and hence equivalence up to \mathcal{T} -steps is trivial. The validity of the other transitions is shown similarly.

Based on \mathcal{T} , we can define the reduction function T given by $T(s_0) = \text{task}_1$, $T(s_3) = \text{task}_3$, $T(s_8) = \text{task}_5$ and $T(s) = \text{en}(s)$ for all other states s . The reduced MDP obtained in this way is shown in Figure 3. Note that, compared to the maximal ample set reduction that could be obtained for this MDP, we reduced on two more occasions in the MDP. \square

Correctness. Before proving that acyclic confluence reductions preserve probabilistic visible bisimulation, we provide a lemma connecting the relations \rightsquigarrow and \sqsubseteq . Note that the relation \mathcal{R} that is used in this lemma relates states that are either identical or connected by a confluent transition.

Lemma 20. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, and $\mathcal{T} \subseteq \Delta_M$ a confluent set of transitions of M . Let \mathcal{R} be the reflexive closure of the set $\{(s, s') \in S \times S \mid \exists a \in L. (s, a, \mathbf{1}_{s'}) \in \mathcal{T}\}$. Finally, let $\mu, \nu \in \text{Distr}(S)$. Then, $\mu \rightsquigarrow_{\mathcal{T}} \nu$ implies $\mu \sqsubseteq_{\mathcal{R}} \nu$.*

Proof. Let $\mu \rightsquigarrow_{\mathcal{T}} \nu$, so there exists a partitioning $\text{spt}(\mu) = \bigsqcup_{i=1}^n S_i$ of the support of μ and an ordering $\text{spt}(\nu) = \{s_1, \dots, s_n\}$ of the support of ν , such that

$$\forall 1 \leq i \leq n . \mu(S_i) = \nu(s_i) \wedge (S_i = \{s_i\} \vee (\forall s \in S_i . \exists a \in \Sigma . (s, a, \mathbb{1}_{s_i}) \in \mathcal{T})).$$

Now, we define the function $w: S \times S \rightarrow [0, 1]$ as follows. For every $1 \leq i \leq n$, for every $s \in S_i$, let $w(s, s_i) = \mu(s)$. Let $w(s', s'') = 0$ for every other pair of states. We show that w is a weight function for μ and ν under \mathcal{R} , thereby proving that $\mu \sqsubseteq_{\mathcal{R}} \nu$.

To see that $w(s_1, s_2) > 0$ implies $(s_1, s_2) \in \mathcal{R}$, observe that by definition of \rightsquigarrow it follows from $w(s_1, s_2) > 0$ that either $s_1 = s_2$ or $\exists a \in \Sigma . (s_1, a, \mathbb{1}_{s_2}) \in \mathcal{T}$. In the first case $(s_1, s_2) \in \mathcal{R}$ due to the reflexivity of \mathcal{R} , and in the second case $(s_1, s_2) \in \mathcal{R}$ due to the definition of the relation underlying \mathcal{R} (which relates states that are connected by confluent transitions).

Furthermore, given a state $s \in \text{spt}(\mu)$, contained in some S_i , we have $\sum_{s' \in S} w(s, s') = w(s, s_i) = \mu(s)$ by definition of w and the fact that every state $s \in \text{spt}(\mu)$ is contained in exactly one class S_i . For any state $s \notin \text{spt}(\mu)$ it directly follows that $\sum_{s' \in S} w(s, s') = 0 = \mu(s)$.

Finally, given a state $s_i \in \text{spt}(\nu)$, we have $\sum_{s' \in S} w(s', s_i) = \sum_{s' \in S_i} w(s', s_i) = \mu(S_i) = \nu(s_i)$ by definition of w and the definition of \rightsquigarrow . For any state $s \notin \text{spt}(\nu)$ it directly follows that $\sum_{s' \in S} w(s', s) = 0 = \nu(s)$. \square

The next lemma will also be of use in the proofs to follow. Note that the relation \mathcal{R} that is used in this lemma relates states that are connected by ‘undirected’ paths of confluent transitions; i.e., for every $(s, s') \in \mathcal{R}$ there is a path from s to s' consisting only of confluent transitions, where we may ignore the direction of the arrows. The lemma shows that pairs of states that are related in this way always have a joining state that they can both reach via only confluent transitions.

Lemma 21. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, and $\mathcal{T} \subseteq \Delta_M$ an acyclic confluent set of transitions. Let \mathcal{R} be the reflexive, symmetric and transitive closure of the set $\{(s, s') \in S \times S \mid \exists a \in L . (s, a, \mathbb{1}_{s'}) \in \mathcal{T}\}$. Let T be a confluence reduction under \mathcal{T} . Then, if $(s, s') \in \mathcal{R}$, there exists a state s'' such that there exist two paths*

$$s_0 \xrightarrow{b_1} s_1 \xrightarrow{b_2} \dots \xrightarrow{b_{n-1}} s_{n-1} \xrightarrow{b_n} s_n \qquad s'_0 \xrightarrow{c_1} s'_1 \xrightarrow{c_2} \dots \xrightarrow{c_{m-1}} s'_{m-1} \xrightarrow{c_m} s'_m$$

with $s_0 = s$, $s'_0 = s'$ and $s_n = s'_m = s''$, such that every transition is in \mathcal{T} and, if s' is in S_T , the second path is also in M_T .

Proof. In this proof, we will use $s \xrightarrow{\tau} s'$ to denote an anonymous confluent transition in \mathcal{T} . Also, we will write ‘ \mathcal{T} -confluent path’ to refer to a path consisting only of confluent transitions that are in the set \mathcal{T} .

Let $(s, s') \in \mathcal{R}$, so there must be path such as $s \xrightarrow{\tau} s_0 \xrightarrow{\tau} s_1 \xleftarrow{\tau} s_2 \xleftarrow{\tau} s_3 \xrightarrow{\tau} s_4 \xleftarrow{\tau} s'$. Potentially, as is the case here, the path contains a fragment of the form $s_i \xleftarrow{\tau} s_{i+1} \xrightarrow{\tau} s_{i+2}$, which we call a ‘bad’ fragment. Clearly this violates the conditions for the path to show that $s \xrightarrow{\tau} \dots \xrightarrow{\tau} s'' \xleftarrow{\tau} \dots \xleftarrow{\tau} s'$, as the arrows point in the wrong direction. Also, note that a path without such a fragment is what we need to prove this lemma. Therefore, we will show that in any path that can be used to show $(s, s') \in \mathcal{R}$, we can eliminate these kind of fragments, obtaining another path in the model that does show $s \xrightarrow{\tau} \dots \xrightarrow{\tau} s'' \xleftarrow{\tau} \dots \xleftarrow{\tau} s'$.

If $s_i \xleftarrow{\tau} s_{i+1} \xrightarrow{\tau} s_{i+2}$, then by definition of confluence either (1) $s_i = s_{i+2}$, or (2) $P(s_{i+1}, \tau) \rightsquigarrow_{\mathcal{T}} P(s_{i+2}, \tau)$ (where the τ in $P(s_{i+1}, \tau)$ is meant to denote the same action as the one in $s_i \xleftarrow{\tau} s_{i+1}$). In the first case, the whole fragment can just be reduced to the state s_i , indeed eliminating the bad fragment. In the second case, by definition of $\rightsquigarrow_{\mathcal{T}}$ we have either $s_i = \text{target}(s_{i+2}, \tau)$ or $s_i \xrightarrow{\tau} \text{target}(s_{i+2}, \tau)$. So, the fragment $s_i \xleftarrow{\tau} s_{i+1} \xrightarrow{\tau} s_{i+2}$ can be replaced by either $s_i \xleftarrow{\tau} s_{i+2}$ or $s_i \xrightarrow{\tau} \text{target}(s_{i+2}, \tau) \xleftarrow{\tau} s_{i+2}$, respectively. Repeating this for all bad fragments, a path of the form $s \xrightarrow{\tau} \dots \xrightarrow{\tau} s'' \xleftarrow{\tau} \dots \xleftarrow{\tau} s'$ appears.

Even though the removal of bad fragments can introduce new bad fragments, this approach indeed converges. To show this, we first define the ‘badness’ of a path to be the number of pairs of transitions that are pointing away from each other. More formally, given a path such as $\pi = s_0 \xrightarrow{\tau_1} s_1 \xleftarrow{\tau_2} s_2 \xleftarrow{\tau_3} s_3 \xrightarrow{\tau_4} s_4$, the badness of π equals the size of the set $\{(i, j) \mid i < j \wedge (s_i \xleftarrow{\tau_i} s_{i+1} \text{ on } \pi) \wedge (s_{j-1} \xrightarrow{\tau_j} s_j \text{ on } \pi)\}$. For this example, that would be the set $\{(1, 4), (2, 4)\}$, so the badness of π is 2. Note that this path has only one bad fragment, and that after the first bad fragment removal, it could for instance become

$\pi' = s_0 \xrightarrow{\tau_1} s_1 \xleftarrow{\tau_2} s_2 \xrightarrow{\tau_3} \text{target}(s_4, \tau) \xleftarrow{\tau_4} s_4$. Although this path still has one bad fragment, its badness has decreased from 2 to 1. It is easy to see that each of the above three bad fragment eliminations reduces the badness by at least one. Since a path with badness 0 has no bad fragments anymore, this approach indeed terminates.

Now, let $M_T = (S_T, \Sigma, P_T, s^0, \text{AP}, L_T)$ be the reduced MDP with respect to T . Based on the path $s \xrightarrow{\tau} \dots \xrightarrow{\tau} s'' \xleftarrow{\tau} \dots \xleftarrow{\tau} s'$ derived above (which may but does not have to be in M_T), we show that the two paths promised by the lemma exist in such a way that $s'_0 \xrightarrow{c_1} s'_1 \xrightarrow{c_2} \dots \xrightarrow{c_{m-1}} s'_{m-1} \xrightarrow{c_m} s'_m$ is guaranteed to be in M_T .

If in every state s'_i on the path $s'' \xleftarrow{\tau} \dots \xleftarrow{\tau} s'$ there is only one outgoing transition that is in \mathcal{T} , the situation is easy. After all, in that case M_T has to preserve this path as in each state it can only choose that one confluent transition (or enable all transitions). Therefore, the path $s \xrightarrow{\tau} \dots \xrightarrow{\tau} s'' \xleftarrow{\tau} \dots \xleftarrow{\tau} s'$ provides the two paths we were looking for.

If, however, the set of confluent transitions \mathcal{T} underlying the confluence reduction T contains multiple confluent transitions from some state s'_i , it might be the case that the transition $s'_i \xrightarrow{c_{i+1}} s'_{i+1}$ of the above path $s' \xrightarrow{\tau} \dots \xrightarrow{\tau} s''$ is not the one chosen by T . Let $s'_i \xrightarrow{d_{i+1}} s^*$ be the confluent transition from s'_i that is chosen by T . Now, since $(s, s') \in \mathcal{R}$ and s' has a \mathcal{T} -confluent path to s^* , also $(s, s^*) \in \mathcal{R}$. Therefore, by the same arguments used in the first half of this proof, there is a state s^{**} such that both s and s^* have a \mathcal{T} -confluent path to s^{**} . Hence, there also is a \mathcal{T} -confluent path from s' to s^{**} that goes via s^* . Note that the length of the part of the path from s' to the joining state s^{**} that is in M_T increased by at least one, compared to the part of the path from s' to the previous joining state s'' . This procedure can be repeated, and it must terminate due to acyclicity and the fact that the state space is assumed to be finite. Hence, we can find \mathcal{T} -confluent paths from s and s' that join at some state in such a way that the path from s' only contains transitions that are in M_T . These two paths satisfy this lemma. \square

Theorem 22. *If T is an acyclic confluence reduction for an MDP M , then $M \equiv_{\text{pvb}} M_T$.*

Proof. Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP and T an acyclic confluence reduction for M , and let $M_T = (S_T, \Sigma, P_T, s^0, \text{AP}, L_T)$ be the reduced MDP with respect to T , according to Definition 4. To prove that $M \equiv_{\text{pvb}} M_T$, we provide a relation $\mathcal{R}^* \subseteq S \times S_T$, relating states that are connected by confluent transitions. We first prove that \mathcal{R}^* is a probabilistic visible bisimulation for (M, M_T) , and then that $(\mathcal{R}^*)^{-1}$ is one for (M_T, M) .

Let \mathcal{T} be the set of confluent transitions associated with T (by definition of confluence reductions). In this proof, whenever we write that a transition is ‘confluent’, we mean that it is in this set \mathcal{T} . Note that this is actually a stronger statement: some transitions can be confluent, but might nonetheless not be present in \mathcal{T} . Similarly, a ‘confluent path’ in this proof is a path consisting only of transitions from \mathcal{T} .

Let \mathcal{R} be the reflexive, symmetric and transitive closure of $\{(s, s') \in S \times S \mid \exists a \in L. (s, a, \mathbb{1}_{s'}) \in \mathcal{T}\}$. Finally, let $\mathcal{R}^* = \mathcal{R} \cap (S \times S_T)$. Note that \mathcal{R}^* is still transitive, and that it is reflexive for the states of S_T . Basically, \mathcal{R}^* relates all states that are connected by an ‘undirected’ confluent path; i.e., for every $(s, s') \in \mathcal{R}^*$ it holds that there is a path in M from s to s' consisting only of confluent transitions, if we ignore the direction of the arrows.

To see that \mathcal{R}^* is a probabilistic visible simulation for (M, M_T) , first note that $(s^0, s^0) \in \mathcal{R}^*$. After all, \mathcal{R}^* is reflexive for the states in S_T and indeed $s^0 \in S_T$ since reduction functions preserve initial states. Moreover, for every $(s, s') \in \mathcal{R}^*$, the conditions presented in Definition 9 hold:

1. Since all confluent transitions are invisible by definition, they are stuttering and hence $L(s) = L_T(s')$.
2. Let $a \in \text{en}(s)$. By Lemma 21 we know that there exists a state $t \in S$ such that there are two paths

$$s_0 \xrightarrow{b_1} s_1 \xrightarrow{b_2} \dots \xrightarrow{b_{n-1}} s_{n-1} \xrightarrow{b_n} s_n \qquad s'_0 \xrightarrow{c_1} s'_1 \xrightarrow{c_2} \dots \xrightarrow{c_{m-1}} s'_{m-1} \xrightarrow{c_m} s'_m$$

with $s_0 = s$, $s'_0 = s'$ and $s_n = s'_m = t$, such that every transition is in \mathcal{T} , and, as s' is in M_T , the second path is also in M_T .

Without loss of generality, we can assume that $T(t) = \text{en}(t)$, i.e., t is fully expanded. Such a t exists, because of the acyclicity of the reduction. As t is fully expanded, note that $P_T(t, a) = P(t, a)$. Now, we make a case distinction between (i) $s = t$ and (ii) $s \neq t$.

- (i) If $s = t$, then condition 2(b) of Definition 9 holds. After all, we have seen that there is a confluent path from s' to t in M_T , so due to $s = t$ there is a confluent path from s' to s in M_T . By definition of confluence this path is invisible, and by definition of \mathcal{R}^* all the states on this path are related to s . Moreover, as this path ends in s and s is fully expanded, $a \in \text{en}(s)$ in M_T and $P(s, a) \sqsubseteq_{\mathcal{R}^*} P_T(t, a)$ since $P_T(t, a) = P(t, a) = P(s, a)$ and $\sqsubseteq_{\mathcal{R}^*}$ is reflexive.
- (ii) If $s \neq t$, then as mentioned above there is a confluent path $s_0 \xrightarrow{b_1} s_1 \xrightarrow{b_2} \dots \xrightarrow{b_{n-1}} s_{n-1} \xrightarrow{b_n} s_n$ with $s_0 = s$ and $s_n = t$. We prove by induction on the length of the subpaths of this path that either we have $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}^*$, or $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_i, a)$ for every $1 \leq i \leq n$. The first part of this disjunction coincides with condition 2(a) of Definition 9, the second implies condition 2(b) by instantiating it with $i = n$ and taking the confluent path from s' to t (which we already showed to be in M_T).

Note that we prove $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_i, a)$ using \mathcal{R} instead of \mathcal{R}^* , since not every state s_i has to be in S_T . However, as $s_n = t$ is fully explored, every state in the support of $P(s_n, a)$ is in S_T , and therefore $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_n, a)$ implies the required $P(s, a) \sqsubseteq_{\mathcal{R}^*} P(s_i, a)$.

Base case. To prove the base case, we make a case distinction based on whether or not $a = b_1$.

- = If a and b_1 coincide, then $a \in \Sigma_{\text{inv}}$ by definition of confluence. Moreover, $\text{target}(s, a) = s_1$ and $(s_1, s') \in \mathcal{R}^*$, as there is a confluent path from s_1 to t and one from s' to t (and therefore an undirected confluent path from s_1 to s'). Hence, $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}^*$.
- \neq If a and b_1 do not coincide, observe that either $P(s, a) \rightsquigarrow_{\mathcal{T}} P(s_1, a)$ or a is an invisible action and $P(s, a) = \mathbb{1}_{s_1}$ (since the b_1 -transition is confluent). In the former case, we use Lemma 20 — observing that the relation \mathcal{R} used in the current proof is a superset of the relation \mathcal{R} used in that lemma and applying Proposition 5.2.1.5 of [25] (which states that $\mu \sqsubseteq_{R_1} \mu'$ implies $\mu \sqsubseteq_{R_2} \mu'$ if $R_2 \supseteq R_1$) — to obtain $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_1, a)$. In the latter case, $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}^*$ holds again by the same reasoning as above.

Inductive case. Let either $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}^*$, or $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_i, a)$ for every $1 \leq i \leq k$ for some $k < n$. In case the first part of this disjunction is true we are done. So, assuming that $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_i, a)$ for every $1 \leq i \leq k < n$, we prove the statement for $k + 1$. To do so, we make a case distinction based on whether or not $a = b_{k+1}$.

- = If a and b_{k+1} coincide, then $a \in \Sigma_{\text{inv}}$ by definition of confluence. Moreover, from the assumption that $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_k, a)$ and the definition of \mathcal{R} it follows that $\text{target}(s, a)$ has an undirected confluent path to $\text{target}(s_k, a)$. As $\text{target}(s_k, a) = s_{k+1}$ and this state has a confluent path to t (and therefore an undirected confluent path to s'), also $\text{target}(s, a)$ has an undirected confluent path to s' . Hence, $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}^*$.
- \neq If a and b_{k+1} do not coincide, either $P(s_k, a) \rightsquigarrow_{\mathcal{T}} P(s_{k+1}, a)$ or a is an invisible action and $P(s_k, a) = \mathbb{1}_{s_{k+1}}$ (because the b_{k+1} -transition is confluent). In the latter case, again $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}^*$ holds by the same reasoning as above.

In the former case, we obtain $P(s_k, a) \sqsubseteq_{\mathcal{R}} P(s_{k+1}, a)$ in the same way as for the base case. Now, using Proposition 5.2.1.2 from [25] and the fact that $R \circ R = R$ for reflexive and transitive relations R , we observe that $\sqsubseteq_{\mathcal{R}}$ is transitive. Therefore, $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_k, a)$ and $P(s_k, a) \sqsubseteq_{\mathcal{R}} P(s_{k+1}, a)$ yield $P(s, a) \sqsubseteq_{\mathcal{R}} P(s_{k+1}, a)$.

3. Let $s \xrightarrow{b_1 b_2 \dots}$ be an infinite invisible path of M , such that $(s_i, s') \in \mathcal{R}^*$ for every state s_i on this path. By Lemma 21 and the fact that $(s, s') \in \mathcal{R}^*$, we know that there exists a state $t \in S$ such that there are two paths

$$s'_0 \xrightarrow{c_1} s'_1 \xrightarrow{c_2} \dots \xrightarrow{c_{n-1}} s'_{n-1} \xrightarrow{c_n} s'_n \qquad s''_0 \xrightarrow{d_1} s''_1 \xrightarrow{d_2} \dots \xrightarrow{d_{m-1}} s''_{m-1} \xrightarrow{d_m} s''_m$$

with $s'_0 = s$, $s''_0 = s'$ and $s'_n = s''_m = t$, such that every transition is in \mathcal{T} and the second path is also in M_T . Without loss of generality, we can assume that $T(t) = \text{en}(t)$, i.e., t is fully expanded. Such a t exists, because of the acyclicity of the reduction.

We make a case distinction based on whether (1) $s' \neq t$ or (2) $s' = t$.

- (i) If $s' \neq t$, then we can take $s' \xrightarrow{d_1} s'_1$ as the required finite path. By definition of confluence d_1 is indeed invisible, $(s, s') \in \mathcal{R}^*$ by assumption and $(s, s'_1) \in \mathcal{R}^*$ by transitivity of \mathcal{R}^* .
- (ii) Let $s' = t$. In this case, one of the states on the confluent path $s \xrightarrow{c_1 \dots c_n} s'$ is the last one to appear in the infinite path $s \xrightarrow{b_1 b_2 \dots}$, let us say this is s'_i . Note that the index i here refers to the index of this state on the confluent path from s to s' , and that the index of this state on $s \xrightarrow{b_1 b_2 \dots}$ may be different. We denote the states on that infinite path without prime, so let's say that $s'_i = s_k$.

Now, we prove by induction on the length of the path from s'_i to s' (also denoted by s'_n) that every state s'_j on that path (including s') has an infinite path of invisible transitions, i.e., $s'_j \xrightarrow{\tau \tau \dots}$ (where again we use τ to denote an anonymous invisible action), such that every state on that path is reachable by a directed confluent path from at least one state s_l of the path $s \xrightarrow{b_1 b_2 \dots}$.

Base case. Since s'_i is on the path $s \xrightarrow{b_1 b_2 \dots}$, it clearly has an infinite invisible path, just continuing $s_k \xrightarrow{b_{k+1}} s_{k+1} \xrightarrow{b_{k+1}} \dots$. Also, each state on this path is reachable by a directed confluent path from some state on $s \xrightarrow{b_1 b_2 \dots}$, as they even are on $s \xrightarrow{b_1 b_2 \dots}$ and therefore empty paths suffice.

Inductive case. Let s'_j (with $s'_j \neq s'$) be a state on the confluent path from s'_i to s' such that $s'_j \xrightarrow{\tau \tau \dots}$ and every state on $s'_j \xrightarrow{\tau \tau \dots}$ is reachable by a directed confluent path from at least one state of the path $s \xrightarrow{b_1 b_2 \dots}$. We show that s'_{j+1} also has such an infinite invisible path. If s'_{j+1} lies on $s'_j \xrightarrow{\tau \tau \dots}$, this is obviously true.

So, from now on assume that the infinite invisible path from s'_j does not involve s'_{j+1} . Note that s'_{j+1} is reachable by a directed confluent path from some state on $s \xrightarrow{b_1 b_2 \dots}$, namely the state s_k mentioned above, as there is a directed confluent path from $s_k = s'_i$ to s'_{j+1} (this is after all a part of the confluent path from s to s').

Now, let s^* be s'_j 's successor on its infinite invisible path $s'_j \xrightarrow{\tau \tau \dots}$, so let's say $s'_j \xrightarrow{b} s^*$ (here the b -transition is invisible but not necessarily confluent). As s'_j also has a confluent transition to s'_{j+1} , by definition of confluence either $P(s'_j, b) \rightsquigarrow_{\tau} P(s'_{j+1}, b)$ or $\text{target}(s'_j, b) = s'_{j+1}$. Note that the second option is not possible, since $\text{target}(s'_j, b) = s^*$ is on $s'_j \xrightarrow{\tau \tau \dots}$ and we assumed that s'_{j+1} is not. The first option translates to either (a) $s^* = \text{target}(s'_{j+1}, b)$ or (b) there is a confluent transition from s^* to $\text{target}(s'_{j+1}, b)$.

- (a) In this case, clearly s'_{j+1} also has an infinite invisible path, first taking it's b -transition and then continuing on the infinite invisible path from s^* . All states on this path are reachable by a directed confluent path from at least one of the states of the path $s \xrightarrow{b_1 b_2 \dots}$ due to the induction hypothesis and the earlier observation that this holds for s'_{j+1} .
- (b) In this case, there is a state u such that $s'_{j+1} \xrightarrow{b} u$ and s^* has a confluent transition to u .
If u is on $s'_j \xrightarrow{\tau \tau \dots}$, then s'_{j+1} has an infinite invisible path, and the directed confluent paths exist for the same reason as in case (a).

If u is not on $s'_j \xrightarrow{\tau \tau \dots}$, then again u is reachable by a directed confluent path from some state on $s \xrightarrow{b_1 b_2 \dots}$, since s^* is and there is a confluent transition from s^* to u . Moreover, from u the exact same situation that we started with appears again. So, we can repeat the argument until case (a) occurs, or if that doesn't happen (b) occurs infinitely often and s'_{j+1} has an infinite invisible path as well.

So, s' has an infinite invisible path such that every state on this path is reachable by a directed confluent path from at least one of the states of the path $s \xrightarrow{b_1 b_2 \dots}$. Let $s' \xrightarrow{b} s^*$ be the first transition of this path from s' , then the path $s' \xrightarrow{b} s^*$ satisfies condition 3 of Definition 9. After all, this path is in M_T since t was assumed to be fully expanded and $s' = t$. Moreover, there indeed is some state v on $s \xrightarrow{b_1 b_2 \dots}$ with a directed confluent path to s^* , so $(v, s^*) \in \mathcal{R}^*$.

To see that $(\mathcal{R}^*)^{-1}$ is a probabilistic visible simulation for (M_T, M) , we can use the same or much simpler arguments than above:

1. As above.

2. Every state $s \in S_T$ will have either exactly one outgoing confluent transition, or exactly the outgoing transitions that are in M . In the first case 2(a) holds, and in the second, 2(b), trivially.
3. The same reasoning applies as before, with the simplification that each infinite execution of M_T is at the same time an infinite execution of M . \square

Proposition 3.4.10 from [23], gives the following corollary.

Corollary 23. *If T is an acyclic confluence reduction function for M , then M and M_T satisfy the same $\text{PCTL}_{\setminus X}^*$ -formulae.*

4. A Comparison of Ample Sets and Confluence

The relationship between ample sets and confluence is not straightforward. In this section, we will first see that confluence is strictly more general. In addition to this, we cover exactly the aspects that differentiate ample sets from confluence. Based on these observations we restrict confluence and relax ample sets, so as to make them coincide. We provide a justification for each of these restrictions and relaxations, that reflects on how the methods are used in practice.

4.1. Why confluence is strictly more powerful

The starting point of our investigation is given by the following theorem. It shows that, if the ample set method allows a state to explore only one of its outgoing transitions, the confluence method also allows this. Therefore, any reduction that can be achieved by the use of ample sets can also be achieved by using confluence.

Recall that $\bar{A}(s)$ contains the actions that are enabled from s by a reduction function A , in case s is not fully explored; otherwise, $\bar{A}(s)$ is the empty set.

Theorem 24. *Let A be an ample set reduction function for an MDP $M = (S, \Sigma, P, s^0, \text{AP}, L)$. Then, the set $\mathcal{T}_A = \{(s, a, \mu) \in \Delta_M \mid a \in \bar{A}(s)\}$ is acyclic, and consists of confluent transitions.*

Proof. Firstly, the fact that \mathcal{T}_A is acyclic follows from the ample set condition A3: a cycle of nontrivial transitions would violate the condition. Secondly, to show that all the transitions in \mathcal{T}_A are confluent, we need to find a confluent set of transitions $\mathcal{T}_A^* \supseteq \mathcal{T}_A$ in which they are contained. Let \mathcal{T}_A^* be defined as the minimal set that satisfies the following:

- $\mathcal{T}_A^* \supseteq \mathcal{T}_A$;
- If $(s, a, \mathbf{1}_t) \in \mathcal{T}_A^*$ and $b \in \text{en}(s)$ ($b \neq a$), then $\{(s_0, a, \mu) \in \Delta_M \mid s_0 \in \text{spt}(P(s, b))\} \subseteq \mathcal{T}_A^*$.

To prove that \mathcal{T}_A^* is confluent, first note that by conditions A1 and A4 of the definition of ample sets and by construction of \mathcal{T}_A^* , only transitions with invisible actions are ever added to the set. Second, let $(s, a, \mathbf{1}_t) \in \mathcal{T}_A^*$ and let (s, b, μ) be a transition of M . If b equals a , then the condition for confluence is trivially fulfilled, so assume that $b \neq a$. If we can prove that a and b are independent, confluence follows from Lemma 16. Note that this lemma is indeed applicable, since by construction \mathcal{T}_A^* contains all a -transitions from the support of $P(s, b)$.

By definition of \mathcal{T}_A^* , there must be some state s^* and a (possibly empty) path $s^* \xrightarrow{b_1 \dots b_n} s$ such that $b_i \neq a$ for each i , and $a \in \bar{A}(s^*)$. Then, $\bar{A}(s^*) = \{a\}$, by condition A4 of ample sets. Condition A2 guarantees that if b depends on a , we would have at least one $b_i \in \bar{A}(s^*)$, contradicting A4. Thus, a and b are independent. \square

Note that this implies that the other notions of confluence from [17] (probabilistic confluence and weak probabilistic confluence), which are even more powerful than strong probabilistic confluence (though less practically applicable), are also strictly more powerful than partial order reduction.

On the other hand, it is not the case that every confluent transition can be chosen to be in a nontrivial ample set. Confluence reduction turns out to be more liberal on several aspects, as illustrated by the following examples.

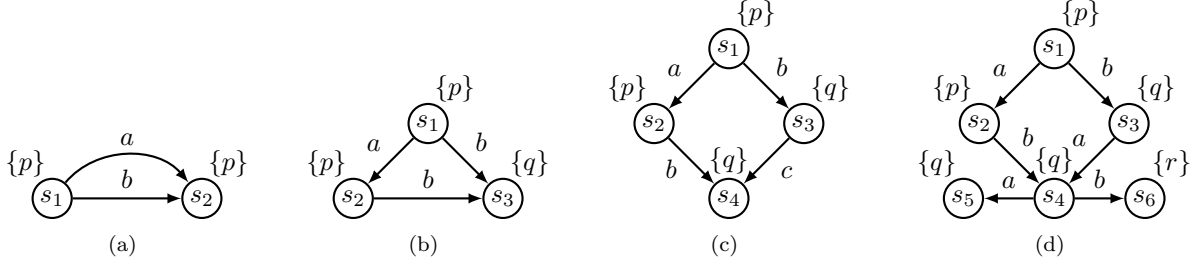


Figure 4: Confluence triumphs over ample sets.

Example 25. Consider the MDPs in Figure 4 (with the atomic propositions per state indicated in brackets). For these MDPs, all transitions are deterministic. Note also that all a -transitions are stuttering and therefore invisible. Even more, they are constructed in such a way that the outgoing a -transitions from every state s_1 are confluent. Hence, confluence reduction is allowed to omit their outgoing b -transitions, removing six transitions and two states.

In Figure 4(a), also the b -transition is invisible. Due to the part $b \in \Sigma_{\text{inv}}$ and $P(s, b) = \mu$ of the disjunction in Definition 17, this transition does not prohibit the a -transition from being confluent. After all, this part basically allows confluent transitions to disable other invisible transitions having the same source and target state as the confluent transition, as illustrated here. Therefore, confluence reduction is allowed to choose either one of these two transitions and could for instance reduce based on $\mathcal{T} = \{(s_1, a, \mathbb{1}_{s_2})\}$. The ample set conditions do not allow this; they require complete independence between a and b for $\{a\}$ to be a valid ample set for s_1 . Hence, the only valid ample set for s_1 is $\{a, b\}$.

In Figure 4(b), the b -transition is not invisible anymore. Also, a and b are again dependent since b disables a . However, the a -transition from s_1 can still be considered confluent, taking $\mathcal{T} = \{(s_1, a, \mathbb{1}_{s_2})\}$ as the underlying confluent set for confluence reduction, due to the part $S_i = \{s_i\}$ of the disjunction in Definition 14 (so the reduction is enabled by the weakening of this definition with respect to [17]). This part of the definition makes sure that although visible actions must still be enabled after a confluent transition, the confluent action does not need to still be enabled after the visible action. Again, however, partial order reduction would not work since a and b are not independent.

Although it might seem that allowing reduction in case of triangle constructions such as Figure 4(b) only removes some transitions, it can in theory make a significant difference in the number of states. Imagine for instance a system in which every state has a transition *quit* to a single deadlock state (as is partially the case in Figure 1). Then, not one action is independent of *quit*, and partial order reduction would not be able to provide any reduction. However, such transitions would not interfere with confluence. Every confluence reduction that would be possible without the *quit* transitions is still possible with the *quit* transitions.

In Figure 4(c), the a -transition can be considered confluent since the diamond shape is closed perfectly (taking $\mathcal{T} = \{(s_1, a, \mathbb{1}_{s_2}), (s_3, c, \mathbb{1}_{s_4})\}$). Even though b disables a , there is a transition from s_3 to s_4 that can easily be seen confluent. The ample set conditions strictly require invisible transitions to be mimicked by equally-named invisible transitions, not allowing any reduction for this model.

In Figure 4(d), the outgoing a -transition from s_1 is confluent since the diamond shape of independence is present (taking $\mathcal{T} = \{(s_1, a, \mathbb{1}_{s_2}), (s_3, a, \mathbb{1}_{s_4})\}$). The fact that a can disable b later on in the system does not matter for confluence. The ample set conditions, however, do require a and b to be globally independent for $\{a\}$ to be a valid ample set for s_1 . As this is not the case, no reductions can be achieved with partial order reduction. \square

One large contributor to why confluence provides more reduction stems from the fact that it is defined based on the actual low-level transitions at a given state of the model, whereas the independence notion of ample set reduction works on higher-level actions and is considered to be global. That is, the dependency relation is assumed to be the same for every state. In practice, however, heuristics for detecting confluent transitions symbolically often also take this action-based point of view, diminishing the differences [4, 17].

4.2. Making confluence and ample sets coincide

As a first step towards making confluence coincide with ample sets, we precisely prohibit all the liberal aspects of confluence that make the reductions in Figure 4(a), 4(b) and 4(c) work. As a second step, we loosen the independence concept of ample sets so that it better corresponds to the more local approach of confluence, allowing partial order reduction to optimise Figure 4(d). Note that we do this safely, i.e., Theorem 22 is never compromised in the process, as all these notions will still be confluent in the sense used in the theorem.

Strengthening confluence. First of all, equivalence up to \mathcal{T} -steps should always occur in the diamond structure of independence. Therefore, the part $S_i = \{s_i\}$ of the disjunction has to be removed. This results in confluence not being able to reduce Figure 4(b) anymore.

Definition 26 (Equivalence up to \mathcal{T} -steps (strengthened)). *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, $\mathcal{T} \subseteq \Delta_M$ a set of deterministic transitions of M , and $\mu, \nu \in \text{Distr}(S)$ two probability distributions. Then, we say that μ is equivalent up to \mathcal{T} -steps to ν , denoted by $\mu \rightsquigarrow_{\mathcal{T}} \nu$, if $\mu, \nu \neq \perp$ and there exists a partitioning $\text{spt}(\mu) = \bigsqcup_{i=1}^n S_i$ of the support of μ and an ordering $\text{spt}(\nu) = \{s_1, \dots, s_n\}$ of the support of ν , such that*

$$\forall 1 \leq i \leq n . \mu(S_i) = \nu(s_i) \wedge \forall s \in S_i . \exists a \in \Sigma . (s, a, \mathbb{1}_{s_i}) \in \mathcal{T}.$$

When symbolic analysis is carried out for ample sets and similar methods, the relations that are extracted are usually assumed symmetric. This is much due to the way algorithms for generating them work. This stronger version of up-to-equivalence is mostly symmetric: the only asymmetric feature that remains is that the \mathcal{T} -transitions are required to be deterministic. The practical interpretation of this restriction is that it matches the symmetric nature of relations that ample sets and related methods often (though not always, see for instance [16]) require.

In addition to strengthening equivalence up to \mathcal{T} -steps, also strong probabilistic confluence is restricted to no longer allow an action b from a state s with a confluent transition $(s, a, \mathbb{1}_t)$ to immediately go to t and not be mimicked there; the practical interpretation is similar to the one mentioned above. After this change, no reduction is possible anymore in the model of Figure 4(a).

Definition 27 (Strong probabilistic confluence (strengthened)). *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP. A set $\mathcal{T} \subseteq \Delta_M$ of transitions of M is strongly probabilistically confluent if for every $(s, a, \mu) \in \mathcal{T}$ it holds that $a \in \Sigma_{\text{inv}}$ and*

- For every $b \in \text{en}(s)$ ($b \neq a$), it holds that $P(s, b) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s, a), b)$.

Here, $\rightsquigarrow_{\mathcal{T}}$ is according to Definition 26. We call a reduction function with an underlying confluent set satisfying these strengthened requirements a strengthened confluence reduction function.

Note that we had to add the restriction $b \neq a$; otherwise, confluent transitions would not commute with themselves anymore.

Finally, we saw in Figure 4(d) that for confluence it can happen that invisible transitions are mimicked by actions with different names. To get closer to the notions coinciding, we need to make sure that actions are not allowed to rely on other actions to ‘close their diamonds’. From the point of view of symbolic analysis, this restriction matches the practical methods of analysis used in conjunction with partial order reduction: this way only pairwise analysis of actions is required, and the algorithms for generating ample sets or similar notions mostly rely on these sort of binary relations. For this purpose we introduce the concept of *action-separability*, requiring that each subset of \mathcal{T} that can be obtained by only keeping one specific action is confluent. That way, confluence reduction functions such as the one in Figure 4(d) are not allowed anymore.

Definition 28 (Action-separable confluence). *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, then a confluent set $\mathcal{T} \subseteq \Delta_M$ of transitions of M is action-separable if for every action $a \in \Sigma$ the subset $\mathcal{T}_a = \{(s, a, \mu) \in S \times \{a\} \times \text{Distr}(S) \mid (s, a, \mu) \in \mathcal{T}\}$ of a -labelled confluent transitions is either empty or confluent.*

A confluence reduction function $T: S \rightarrow 2^{\Sigma}$ is action-separable if its underlying confluent set \mathcal{T} is.

Relaxing ample sets. Independence is judged by the ample set constraints in a global manner, whereas confluence deals with the notion of “equivalence up to”, which is much more local.

To make the methods coincide, independence should also be judged locally, i.e., given a state, dependency of a and b makes a difference only in parts of the MDP that can be reached without executing the ample action first. This corresponds to the fact that confluence only puts restrictions on commutation of actions before a confluent transition.

The practical side of this lies in dynamic analysis. We can, for instance, initially consider that a and b are dependent due to symbolic analysis. However, after finishing exploring some part of the possible states following a state s , we might come to the conclusion that the dependency never manifests anywhere where a has not been executed yet, and thus declare a independent of b locally in s . This idea originates from [26] and [27], and also exactly corresponds to the way the stubborn set definitions (see, e.g., [8]) deal with dependency in the non-probabilistic case: only executions starting from the current state, that do not include any stubborn actions, are relevant from the point of view of commutativity.

To define local independence, let $R_a(s) \subseteq S$ be the set of states s' such that $s \xrightarrow{c_1 \dots c_n} s'$ for some sequence where there is no i such that $c_i = a$.

Definition 29 (Local independence). *Given an MDP $M = (S, \Sigma, P, s^0, \text{AP}, L)$, a state $s \in S$, and two actions $a, b \in \Sigma$, we say that a is independent of b at s if $a \neq b$ and for every state $s' \in R_a(s)$ such that $\{a, b\} \subseteq \text{en}(s')$ the following conditions hold:*

- *If $s^* \in \text{spt}(P(s', a))$, then $b \in \text{en}(s^*)$ (and symmetrically);*
- $$\sum_{s^* \in S} P(s', a)(s^*) \cdot P(s^*, b)(t) = \sum_{s^* \in S} P(s', b)(s^*) \cdot P(s^*, a)(t), \text{ for every } t \in S.$$

If a is not independent of b at s , we say that it is dependent of b at s .

Note that local (in)dependence is not a symmetric relation. For a to be independent of b at s we only look at the states in $R_a(s)$; this is in general a set different from $R_b(s)$.

Example 30. In Example 12 we noticed that the actions task_5 and task_6 in Figure 1 were not independent, since there is a state (s_{11}) in which they can disable each other. However, taking local independence, we see that $R_{\text{task}_5}(s_8) = \{s_8, s_{10}\}$ and $R_{\text{task}_6}(s_8) = \{s_8, s_9\}$, and we can verify that the independence conditions are satisfied by all of these states. Hence, task_5 is independent of task_6 at s_8 and also task_6 is independent of task_5 at s_8 . Therefore, if the ample set conditions would use local independence instead of global independence, it would be allowed to take either task_5 or task_6 as an ample set for s_8 . \square

Under these definitions, we have the following lemma.

Lemma 31. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, $a \in \Sigma_{\text{det}}$ a deterministic action, $s \in S$ a state, and $\mathcal{T} \supseteq \{(t, a, \mu) \in \Delta_M \mid t \in R_a(s)\}$ a set containing all a -labelled transitions enabled from some state that is reachable from s without doing any a -transitions. For any action $b \in \Sigma$ such that $b \neq a$, the implication $\{a, b\} \subseteq \text{en}(s') \Rightarrow P(s', b) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s', a), b)$ holds for every $s' \in R_a(s)$ if and only if a is independent of b at s .*

Proof. (\Rightarrow) To prove the “only if” part of this lemma, take an arbitrary action $b \neq a$ and consider any state $s' \in R_a(s)$ such that $\{a, b\} \subseteq \text{en}(s')$. According to the assumptions of the lemma the a -transition from s' has to be in \mathcal{T} , so let $(s', a, \mathbb{1}_t) \in \mathcal{T}$.

Due to the implication assumed by this lemma, $P(s', b) \rightsquigarrow_{\mathcal{T}} P(t, b)$. Now, from the fact the \mathcal{T} only contains a -transitions and the part $\forall s^* \in S_i. \exists a \in \Sigma. (s^*, a, \mathbb{1}_{s_i}) \in \mathcal{T}$ of the conjunction in the revised definition of $\rightsquigarrow_{\mathcal{T}}$, the first condition for independence is satisfied. For the second condition, observe that

$$\begin{aligned} \sum_{s^* \in S} P(s', a)(s^*) \cdot P(s^*, b)(u) &= P(t, b)(u) = \sum_{s^* \in S_u} P(s', b)(s^*) = \sum_{\substack{s^* \in S \\ s^* \xrightarrow{a} u}} P(s', b)(s^*) \\ &= \sum_{s^* \in S} P(s', b)(s^*) \cdot P(s^*, a)(u) \end{aligned}$$

where the first and last step follow from the fact that a is deterministic, the second and third from the definition of $\rightsquigarrow_{\mathcal{T}}$. We used S_u to denote the class in the partitioning according to $\rightsquigarrow_{\mathcal{T}}$, corresponding to state u .

(\Leftarrow) For the “if” part of this lemma, assume that a is independent of b at s , and let $s' \in R_a(s)$ be an arbitrary state such that $\{a, b\} \subseteq \text{en}(s')$. Carrying out exactly the same calculations as in Lemma 16 for s' (note that \mathcal{T} indeed contains all a -transitions from the support of $P(s', b)$ since all these states are also in $R_a(s)$), we see that $P(s', b) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s', a), b)$. \square

Under the local dependency condition, we can now relax the ample set conditions slightly.

Definition 32 (Relaxed ample sets). *A set $A(s)$ is a relaxed ample set if it meets the criteria of Definition 11, except that A2 is replaced by the following condition:*

A2 For every path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n \xrightarrow{b} t$ in M such that $b \notin A(s)$ and some $a \in A(s)$ is dependent on b at s , there exists an $1 \leq i \leq n$ such that $a_i \in A(s)$;*

Comparison. Our main theorem is now ready to be proven:

Theorem 33. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP. Then, $T: S \rightarrow 2^\Sigma$ is an acyclic action-separable strengthened confluence reduction function if and only if T is a relaxed ample set reduction function.*

Proof. (\Rightarrow) To prove the “only if” part of the theorem, let \mathcal{T} be the action-separable strengthened confluent set underlying T , and let $s \in S$ be an arbitrary state. In this proof, when we write that a transition is confluent we mean that it is confluent *and* that it is in \mathcal{T} . If $T(s) = \text{en}(s)$, then all ample set conditions hold vacuously, so assume that $T(s) \neq \text{en}(s)$. Thus, by definition of confluence reduction functions, $T(s) = \{a\}$ for some $a \in \Sigma_{\text{inv}}$.

Condition A0 is clearly satisfied. Moreover, A1 follows from fact that only transitions with invisible (and thus stuttering) actions can be confluent, A3 from the acyclicity of T and A4 by construction and from the fact that all confluent transitions are deterministic.

For condition A2*, we prove the contrapositive: given an arbitrary path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n \xrightarrow{b} t$ in M such that $b \notin T(s)$ and $a_i \notin T(s)$ for every i , we show that $T(s)$ is independent of b at s . Due to Lemma 31, it is enough to prove that $(s', a, \mathbb{1}_{\text{target}(s', a)}) \in \mathcal{T}$ for every $s' \in R_a(s)$ and additionally $P(s', b) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s', a), b)$ if $\{a, b\} \subseteq \text{en}(s')$.

Let $s' \in R_a(s)$, so there is a path $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_m} s_m$ such that $a_i \neq a$ for every i and $s_m = s'$. Since there is a confluent a -transition from s and also $a_1 \in \text{en}(s)$ and $a_1 \neq a$, by definition of confluence $P(s, a_1) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s, a), a_1)$. Now, by the strengthened definition of \rightsquigarrow and using action-separability, there has to be a confluent a -labelled transition from s_1 . Repeating this argument from s_1 we find that $P(s_1, a_2) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s_1, a), a_2)$ and that there is a confluence a -labelled transition from s_2 , and continuing this way that $P(s_{m-1}, a_m) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s_{m-1}, a), a_m)$ and that there is a confluent a -labelled transition from s_m . So, since $s_m = s'$, indeed $(s', a, \mathbb{1}_{\text{target}(s', a)}) \in \mathcal{T}$. Now, if $\{a, b\} \subseteq \text{en}(s')$, then the same argument can be applied once more from s' , obtaining $P(s', b) \rightsquigarrow_{\mathcal{T}} P(\text{target}(s', a), b)$. (Note that indeed $b \neq a$ since it was assumed that $b \notin T(s)$.)

(\Leftarrow) To prove the “if” part of the theorem, let \mathcal{T}_a be the set of nontrivial actions of the ample set reduction function that are labelled by a . Now, the construction and proof of confluence of a set $\mathcal{T}_a^* \supseteq \mathcal{T}_a$ works almost exactly as in Theorem 24: the construction never adds actions that have a label that is different from a to the set (so action-separability is guaranteed), and the proof of confluence does not rely in any way on the liberal parts that we removed from the definitions.

The only difference is that now, due to the relaxed condition A2*, a and b are not necessarily globally independent anymore. However, confluence can still be proven. To see this, let $(s, a, \mathbb{1}_t) \in \mathcal{T}_a^*$ and let (s, b, μ) be a transition of M . If b equals a , then again the condition for confluence is trivially fulfilled, so assume that $b \neq a$. Now, by definition of \mathcal{T}_a^* , there must be some state s^* and a (possibly empty) path $s^* \xrightarrow{b_1 \dots b_n} s$ such that $b_i \neq a$ for each i , and $a \in \overline{T}(s^*)$. Then, $\overline{T}(s^*) = \{a\}$, by condition A4 of ample sets. Condition A2* guarantees that if a depends on b at s^* , we would have at least one $b_i \in \overline{T}(s^*)$, contradicting A4. Thus, a is independent of b at s^* . As $s \in R_a(s^*)$, the conditions of local independence hold at s .

Now, confluence follows from Lemma 16. (Note that, technically, this lemma is not applicable: although by construction \mathcal{T}_a^* contains all a -transitions from the support of $P(s, b)$, a and b are not globally independent. However, the fact that the independence equations hold at s is the only thing that is used in the proof of Lemma 16, so the result is still valid.)

Note that the union of these confluent sets \mathcal{T}_a is an action-separable confluent set, as the action-specific subsets are exactly the sets \mathcal{T}_a constructed above. Thus, we get the result by taking the union of every \mathcal{T}_a , as a ranges over all (invisible) actions: the resulting action-separable confluent set \mathcal{T} contains all nontrivial transitions of T and therefore proves that T is an acyclic action-separable strengthened confluence reduction function. \square

Note that action-separable confluence strengthened reduction is just a special case of the liberal definition of confluence, used in Theorem 22, so it too preserves probabilistic visible bisimulation. Since relaxed ample set reduction functions coincide with confluence now, we immediately have the result that they too still preserve probabilistic visible bisimulation.

As all of our propositions and theorems hold just as well in case there are no probabilistic transitions, and the probabilistic notions of ample set reduction and confluence reduction in that case reduce to their non-probabilistic variants, the following corollary is also immediate.

Corollary 34. *In the non-probabilistic setting, confluence reduction is able to reduce more than ample set reduction. With some adjustments (as in Definitions 26, 27, 28, 29 and 32), the two notions coincide.*

5. Practical Implications of the Theory

Instead of requiring acyclicity directly, the probabilistic confluence reduction technique of [17] uses the method of representative states, as introduced in [4]. A highly similar construction was used in [21] for representing sets of states for the so-called essential state abstraction. Basically, for this we perceive the system as being partitioned into sets of states that can reach a common *representative* through confluent transitions. As each state in such a set S_i can simulate all other states in S_i , we can just take one of them as a representative for that set and omit the other states. To make sure that all visible transition are enabled immediately from the representative (without the need of some confluent transitions first), we need to choose the representative from the terminal strongly connected component (TSCC) of the sub-MDP restricted by the states of S_i and the confluent transitions. A TSCC is a set of states that can all reach each other and cannot reach any state not in this set. The representative can easily be found using a slightly adapted variant of Tarjan's algorithm for strongly connected components, as explained in detail in [4, 5].

We now introduce the technicalities needed to work with representatives. First, we define a notation for paths containing only transitions from a given set. Second, we introduce a representation map to be a function assigning a representative state to every state of the MDP.

Definition 35 (Restricted paths). *Let $M = (S, \Sigma, P, s^0, AP, L)$ be an MDP, and $\mathcal{T} \subseteq \Delta_M$ a subset of its transitions. We write $s \twoheadrightarrow_{\mathcal{T}} s'$ if there is a path from s to s' that consists solely of transitions that are in \mathcal{T} .*

Definition 36 (Representation map). *Let M be an MDP, and $\mathcal{T} \subseteq \Delta_M$ a subset of its transitions. Then, a function $\phi_{\mathcal{T}}: S \rightarrow S$ is a representation map for M under \mathcal{T} , if*

- $\forall s, s' \in S . (s, a, \mathbb{1}_{s'}) \in \mathcal{T} \Rightarrow \phi_{\mathcal{T}}(s) = \phi_{\mathcal{T}}(s')$;
- $\forall s . s \twoheadrightarrow_{\mathcal{T}} \phi_{\mathcal{T}}(s)$.

The first condition makes sure that states that can reach each other via \mathcal{T} -transitions have the same representative, and the second ascertains that every representative is in a TSCC when restricting to \mathcal{T} -transitions.

Proposition 37. *Let $M = (S, \Sigma, P, s^0, AP, L)$ be an MDP, and \mathcal{T} the set of nontrivial transitions of either a confluence reduction function or an ample set reduction function augmented by all invisible transitions $s \xrightarrow{a} s'$ such that $|en(s)| = 1$. Then, there exists a representation map for M under \mathcal{T} .*

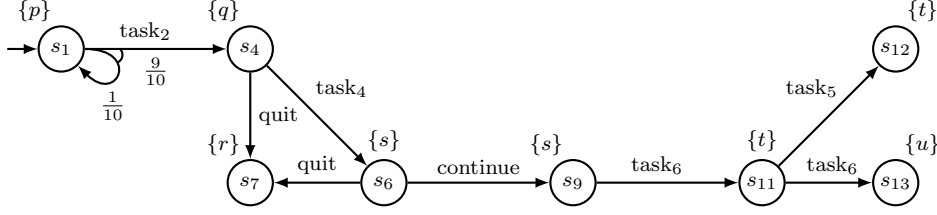


Figure 5: A quotient MDP under a representation map.

Proof. By definition of confluence reduction functions and ample set reduction functions, each state has at most one outgoing nontrivial transition. Also, the other transitions in \mathcal{T} have source states that have only one outgoing transition. Hence, for every state $s \in S$ it holds that there is at most one transition $(s, a, \mathbb{1}_{s'}) \in \mathcal{T}$. Hence, no branching into distinct TSCCs is possible inside a \mathcal{T} -connected subgraph.

This implies that the representative of a state s can easily be chosen by following the unique path of \mathcal{T} -transitions from s : either this path terminates at some point (which should then be taken as the representative for s), or it is in the form of a lasso and keeps cycling. In the second case, any state s' on the cycle can be chosen as the representative for s — but that same state should then be chosen as the representative for all states s'' such that $s'' \rightarrow_{\mathcal{T}} s'$. This way, each state can reach its representative by traversing only \mathcal{T} -transitions. Also, all \mathcal{T} -connected states get assigned the same representative. \square

The following definition states how, given an MDP and a representation map, the reduced MDP is constructed.

Definition 38 (Quotient MDP). *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be a finite MDP, and ϕ a representation map for M under a set \mathcal{T} . The quotient MDP of M under ϕ is defined as $M_\phi = (S_\phi, \Sigma, P_\phi, s_\phi^0, \text{AP}, L_\phi)$ where*

- $S_\phi = \{\phi(s) \mid s \in S\}$;
- $P_\phi(s, a) = \mu$ if and only if $\forall s' \in S_\phi \cdot \mu(s') = \sum_{s^* \in \phi^{-1}(s')} P(s, a)(s^*)$;
- $s_\phi^0 = \phi(s^0)$;
- $L_\phi(s) = L(s)$ for every $s \in S_\phi$.

The definition of the quotient is slightly different from the one given in [18]. This definition induces a self-loop to the states of the quotient in case there is an outgoing confluent transition from the representative; its justification is to handle infinite invisible paths correctly in probabilistic visible bisimulation.

Example 39. Consider again the MDP in Figure 1 and the confluence reduction function T provided in Example 19. The set \mathcal{T} of nontrivial transitions of T augmented by all invisible transitions $s \xrightarrow{a} s'$ such that $|en(s)| = 1$, is $\mathcal{T} = \{(s_0, \text{task}_1, \mathbb{1}_{s_1}), (s_3, \text{task}_3, \mathbb{1}_{s_4}), (s_8, \text{task}_5, \mathbb{1}_{s_9}), (s_2, \text{task}_1, \mathbb{1}_{s_3}), (s_{10}, \text{task}_5, \mathbb{1}_{s_{11}})\}$. A possible representation map under \mathcal{T} is given by

$$\phi_{\mathcal{T}}(s_0) = s_1 \quad \phi_{\mathcal{T}}(s_2) = s_4 \quad \phi_{\mathcal{T}}(s_3) = s_4 \quad \phi_{\mathcal{T}}(s_8) = s_9 \quad \phi_{\mathcal{T}}(s_{10}) = s_{11}$$

and $\phi_{\mathcal{T}}(s) = s$ for all other states s . The quotient MDP under this representation map is shown in Figure 5. \square

The following theorem states that representation maps can be used to safely reduce a state space.

Theorem 40. *Let M be a finite MDP and \mathcal{T} be the set of nontrivial transitions of a confluence reduction function (not necessarily acyclic) augmented by all invisible transitions $s \xrightarrow{a} s'$ such that $|en(s)| = 1$. If ϕ is a representation map for M under \mathcal{T} , then $M \equiv_{\text{pvb}} M_\phi$.*

Proof. First of all, we observe that all transitions in \mathcal{T} are confluent. The nontrivial transitions of a confluence reduction function are confluent by definition of confluence reduction functions, and each transition that is invisible and has a source state with no other outgoing transitions satisfies the confluence condition vacuously.

Let $\mathcal{R} \subseteq S \times S_\phi$ be the relation that contains exactly all pairs $(s, \phi(s))$. We prove that \mathcal{R} is a probabilistic visible bisimulation for (M, M_ϕ) , by first showing that it is a probabilistic visible simulation for (M, M_ϕ) and then that \mathcal{R}^{-1} is a probabilistic visible simulation for (M_ϕ, M) .

Note that, by definition of quotient MDPs, the initial states of M and M_ϕ are indeed related by \mathcal{R} . Now, let $(s, s') \in \mathcal{R}$, so $s' = \phi(s)$. By definition of representation maps, and taking into account that ϕ is based on a set of confluent transitions, there is a confluent path from s to s' . Let $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s'$ be such a path. Then, the conditions of probabilistic visible simulation hold as follows.

1. $L(s) = L_\phi(s')$ is obvious from the existence of a confluent path from s to s' and the fact that confluent transitions are invisible.
2. Let $a \in \text{en}(s)$. If $(s, a, \mu) \in \mathcal{T}$, then by definition of representation maps it must hold that $\phi(s) = \phi(\text{target}(s, a))$ and therefore $(\text{target}(s, a), \phi(s)) \in \mathcal{R}$. Hence, condition 2(a) of Definition 9 follows immediately. Even if the transition is not in \mathcal{T} , but it does hold that $a \in \Sigma_{\text{inv}}$ and $\phi(\text{target}(s, a)) = \phi(s)$, this is still true.

For the other cases, we show that condition 2(b) holds. The invisible path in the quotient that is allowed by this condition is always the empty path. So, we just need to prove that $a \in \text{en}(s')$ in M_ϕ and $P(s, a) \sqsubseteq_{\mathcal{R}} P_\phi(s', a)$. To see that indeed a is enabled from s' , note that, due to the existence of the confluent path, either (1) $P(s, a) \rightsquigarrow_{\mathcal{T}} P(s_1, a) \rightsquigarrow_{\mathcal{T}} \dots \rightsquigarrow_{\mathcal{T}} P(s', a)$ or (2) somewhere on the path the clause $(b \in \Sigma_{\text{inv}}$ and $P(s, b) = \mu)$ of the definition of confluence was applied. However, it is easy to see that the second option yields $a \in \Sigma_{\text{inv}}$ and $\phi(\text{target}(s, a)) = \phi(s)$, which was already covered above. So, from now on we assume that $P(s, a) \rightsquigarrow_{\mathcal{T}} P(s_1, a) \rightsquigarrow_{\mathcal{T}} \dots \rightsquigarrow_{\mathcal{T}} P(s', a)$, and hence $a \in \text{en}(s')$ in M . By definition of the quotient, this also implies that $a \in \text{en}(s')$ in M_ϕ .

To prove $P(s, a) \sqsubseteq_{\mathcal{R}} P_\phi(s', a)$, we define a function $w: S \times S_\phi \rightarrow [0, 1]$ and show that it is a weight function. Given any pair $(s_1, s_2) \in S \times S_\phi$, let w be given by

$$w(s_1, s_2) = \begin{cases} P(s, a)(s_1) & \text{if } s_2 = \phi(s_1) \\ 0 & \text{otherwise} \end{cases}$$

By definition, $w(s_1, s_2) > 0$ implies that $(s_1, s_2) \in \mathcal{R}$. Also, given any $s_1 \in S$, by definition $w(s_1, s^*)$ is only nonzero if $s^* = \phi(s_1)$. Moreover, since $w(s_1, \phi(s_1)) = P(s, a)(s_1)$, we indeed obtain that $P(s, a)(s_1) = \sum_{s^* \in S_\phi} w(s_1, s^*)$.

For w to be a weight function, we additionally need to show that $P_\phi(s', a)(s_2) = \sum_{s^* \in S} w(s^*, s_2)$ for every s_2 . Since $P(s, a) \rightsquigarrow_{\mathcal{T}} P(s_1, a) \rightsquigarrow_{\mathcal{T}} \dots \rightsquigarrow_{\mathcal{T}} P(s', a)$, there is a partitioning $\text{spt}(P(s, a)) = \bigsqcup_{i=1}^m S_i$, and an ordering $\{s'_1, \dots, s'_m\} = \text{spt}(P(s', a))$, such that $P(s, a)(S_i) = P(s', a)(s'_i)$ and there is a (possibly trivial) confluent path from all states of S_i to s'_i .

Let $s_2 \in \text{spt}(P_\phi(s', a))$ be an arbitrary state in the support of $P_\phi(s', a)$. Without loss of generality, assume that $\{s'_1, \dots, s'_k\} = \phi^{-1}(s_2) \cap \text{spt}(P(s', a))$ for some $k \leq m$, i.e., the first k states in the ordering of $\text{spt}(P(s', a))$ are the ones that map to s_2 . Then, we have

$$P_\phi(s', a)(s_2) = \sum_{s^* \in \phi^{-1}(s_2)} P(s', a)(s^*) = \sum_{i=1}^k P(s', a)(s'_i) = \sum_{i=1}^k P(s, a)(S_i) = \sum_{s_1 \in S_1 \cup \dots \cup S_k} P(s, a)(s_1)$$

where the first equality is due to the definition of the quotient, the second and third by the assumptions above, and the fourth by the fact that the S_i 's form a partitioning.

Because each state has exactly one representative, we have $\text{spt}(P(s, a)) \cap \phi^{-1}(s_2) = S_1 \cup \dots \cup S_k$, which means the above can be used to find

$$P_\phi(s', a)(s_2) = \sum_{s_1 \in S_1 \cup \dots \cup S_k} P(s, a)(s_1) = \sum_{s_1 \in \phi^{-1}(s_2)} P(s, a)(s_1) = \sum_{s_1 \in \phi^{-1}(s_2)} w(s_1, s_2) = \sum_{s^* \in S} w(s^*, s_2)$$

proving the claim.

3. Let $s \xrightarrow{b_1 b_2 \dots}$ be an infinite invisible path of M such that $(s_i, s') \in \mathcal{R}$ for every s_i on this path, i.e., $\phi(s_i) = s'$ for every s_i on the path.

We show that there is an invisible self-loop at s' in M_ϕ , which then provides the finite invisible path $s' \rightarrow s'$ in M_ϕ that satisfies the condition (since indeed $(s', s') \in \mathcal{R}$). To show the existence of such a self-loop, it is sufficient to show that s' has an outgoing invisible transition in M to some state s^* such that $\phi(s^*) = s'$ (as the self-loop in M_ϕ then follows by definition of the quotient).

Note that in exactly the same way as we did in part 3 of the proof of Theorem 22, it can be shown that s' has an infinite invisible path such that every state on that path is reachable by a directed confluent path from at least one of the states of the path $s \xrightarrow{b_1 b_2 \dots}$. Now, let $s' \xrightarrow{b} s^*$ be the first transition of this path from s' , and let v be the state on $s \xrightarrow{b_1 b_2 \dots}$ with a directed confluent path to s^* . Because of this confluent path, $\phi(s^*) = \phi(v)$ by definition of representation maps, and therefore $\phi(s^*) = s'$ since we assumed that $\phi(v) = s'$, finishing the proof.

In the other direction, let $(s, s') \in \mathcal{R}^{-1}$ and note that $s = \phi(s')$.

1. $L_\phi(s) = L(s')$ is again obvious from the existence of a confluent path from s' to s and the fact that confluent transitions are invisible.
2. Let $a \in \text{en}(s)$. Then, by definition of the quotient there is some transition (s, a, μ) in M such that $P_\phi(s, a)(t) = \sum_{s^* \in \phi^{-1}(t)} \mu(s^*)$ for every $t \in S_\phi$.

Now, to show condition 2(b), take a confluent path from s' to s (which exists since $\phi(s') = s$). By definition of confluence this path is indeed invisible, and $(s, s'_i) \in \mathcal{R}^{-1}$ for every state s'_i on this path by definition of confluence. Finally, as explained above, also $a \in \text{en}(s)$ in M . It remains to show that $P_\phi(s, a) \sqsubseteq_{\mathcal{R}} P(s, a)$. For this, we define a function $w: S_\phi \times S \rightarrow [0, 1]$ and show that it is a weight function. Given any pair $(s_1, s_2) \in S_\phi \times S$, let w be given by

$$w(s_1, s_2) = \begin{cases} P(s, a)(s_2) & \text{if } s_1 = \phi(s_2) \\ 0 & \text{otherwise} \end{cases}$$

Note that this mirrors the function defined in the proof that \mathcal{R} is a probabilistic visible simulation.

By definition, $w(s_1, s_2) > 0$ implies that $(s_1, s_2) \in \mathcal{R}^{-1}$. Also, given any $s_2 \in S$, by definition $w(s^*, s_2)$ is only nonzero if $s^* = \phi(s_2)$. Moreover, since $w(\phi(s_2), s_2) = P(s, a)(s_2)$, we indeed obtain that $P(s, a)(s_2) = \sum_{s^* \in S_\phi} w(s^*, s_2)$.

Finally, it holds that

$$P_\phi(s, a)(s_1) = \sum_{s^* \in \phi^{-1}(s_1)} P(s, a)(s^*) = \sum_{s^* \in S} w(s_1, s^*)$$

where the first equality follows from the definition of the quotient and the second from the definition of w .

3. Let $s \xrightarrow{b_1 b_2 \dots}$ be an infinite invisible path of M such that $(s_i, s') \in \mathcal{R}^{-1}$ for every s_i on this path. By definition of representation maps, that implies that all states s_i coincide, so the infinite path is just a self-loop of s .

By definition, this invisible self-loop of s in the quotient corresponds to an invisible transition (s, a, s^*) in M such that $\phi(s^*) = s$. Since $s = \phi(s')$, there is a confluent path from s' to s . If this path is nonempty, it proves the conditions. After all, for every state s'_i on this path indeed $(s, s'_i) \in \mathcal{R}^{-1}$, by definition of representation maps. If the path is empty (so $s = s'$), then we can take the path $s' \xrightarrow{a} s^*$ to prove the condition, since $\phi(s^*) = s$ and hence $(s, s^*) \in \mathcal{R}^{-1}$. \square

Theorem 40 is useful, not only for confluence reduction, but also for ample set reduction. Using the representation map, the cycle condition A3 does not need to be checked explicitly. After all, from Theorem 24 we know that every ample set reduction is an acyclic confluence reduction. It is easy to see that every ample set reduction without A3 is still a confluence reduction, albeit possibly not acyclic anymore. However, as acyclicity is also not needed for Theorem 40, the representative approach can just as well be applied to

partial order reduction using ample sets without using the cycle condition explicitly. Then, the algorithm that is used for finding representatives makes sure that acyclicity is guaranteed.

Basically, using this result, state space generation of MDPs using either confluence or ample sets can be done as follows. Whenever a state is visited during the generation of the state space, nontrivial transitions of the reduction function are traversed until a TSCC is reached (see [5] for the details of a variant of Tarjan’s algorithm that does precisely this). Then, instead of the original state, a representative state from this TSCC is chosen as its replacement. That way, cyclicity of the reduction is avoided, and a smaller state space is obtained.

6. Conclusions and Future Work

We redefined probabilistic confluence reduction to an MDP-based setting, enabling a comparison to probabilistic partial order reduction based on ample sets. We proved that every nontrivial ample set can be mimicked by a confluent set, and that in some cases reductions are possible using confluence but not using ample sets. Therefore, at least in theory confluence reduction is able to reduce more than the ample set method. We also showed the exact way in which confluence has to be strengthened and ample sets have to be relaxed for the two notions to coincide. These results hold for the non-probabilistic variants of the two reduction techniques as well.

Our observation that probabilistic ample set reduction can be mimicked by probabilistic confluence reduction has additional implications, some of which are highly practical. One such implication is that the use of a representation map for reduced state space generation, already applied earlier in combination with confluence reduction, can also be applied for partial order reduction. Furthermore, as a nontrivial result, the cycle condition can be replaced by the representation map.

As both ample sets and confluence are detected symbolically on the language level, the quality of the heuristics applied there will decide which notion works best in practice. The results in this paper already strengthen our theoretical understanding of the two methods, and this is independent of the heuristics that are applied. Also, no matter how such heuristics might be improved, the results in this paper will remain valid. Even though a case study on probabilistic confluence reduction in [17] seemed to outperform similar reduction based on ample sets, future work could focus more on the relative merits of the two notions in practice and potentially on the improvement of the syntactical heuristics, if some “best of both worlds” approach is found.

A natural question is, whether there are similar results that could be proven for weaker semantics, like reductions that preserve $LTL_{\setminus X}$. For most part, the answer is obvious: confluence reduction preserves branching time properties, so it also preserves $LTL_{\setminus X}$. However, since confluence is designed to preserve branching properties, it has the inherent restriction that confluent transitions must lead to bisimilar states. This means that we must be able to take single confluent transitions, for if we couldn’t, we would lose some state that is not bisimilar to the current state. Ample sets, and similar methods, do not need such a restriction when dealing with weaker semantics.

One class of open and interesting questions remains, however. When aiming to prove Theorem 33, we worked mostly by *restricting* confluence. It is sensible to ask, if we could have proven the theorem by relaxing the ample set conditions more and restricting the confluence conditions less, while maintaining a practical method that can make use of the extra reduction. How would the less restrictive conditions of confluence (e.g., the original asymmetric up-to-equivalence), or the absence of action separability, be used in conjunction with ample sets or other partial order reduction methods? Could similar conditions be used when partial order reduction preserves weaker properties, like $LTL_{\setminus X}$? Future work might focus on answering these questions.

Acknowledgements. This research has been partially funded by NWO under grants 612.063.817 (SYRUP) and Dn 63-257 (ROCKS), and by the European Union under FP7-ICT-2007-1 grant 214755 (QUASIMODO), and the Finnish Foundation for Technology Promotion. We thank Stefan Blom, Mariëlle Stoelinga and the anonymous reviewers for their helpful suggestions.

References

- [1] P. Godefroid, Partial-order Methods for the Verification of Concurrent Systems: an Approach to the State-explosion Problem, volume 1032 of *Lecture Notes in Computer Science*, Springer, 1996.
- [2] D. Peled, All from one, one for all: on model checking using representatives, in: Proceedings of the 5th International Conference on Computer Aided Verification (CAV), volume 697 of *Lecture Notes in Computer Science*, Springer, 1993, pp. 409–423.
- [3] A. Valmari, Stubborn sets for reduced state space generation, in: Proceedings of the 10th International Conference on Application and Theory of Petri Nets, volume 483 of *Lecture Notes in Computer Science*, Springer, 1989, pp. 491–515.
- [4] S. C. C. Blom, J. C. van de Pol, State space reduction by proving confluence, in: Proceedings of the 14th International Conference on Computer Aided Verification (CAV), volume 2404 of *Lecture Notes in Computer Science*, Springer, 2002, pp. 596–609.
- [5] S. C. C. Blom, Partial τ -confluence for efficient state space generation, Technical Report SEN-R0123, CWI, 2001.
- [6] R. Gerth, R. Kuiper, D. Peled, W. Penczek, A partial order approach to branching time logic model checking, in: Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems (ISTCS), pp. 130–139.
- [7] B. Willems, P. Wolper, Partial-order methods for model checking: From linear time to branching time, in: Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS), IEEE Computer Society, 1996, pp. 294–303.
- [8] A. Valmari, Stubborn set methods for process algebras, in: Proceedings of the DIMACS workshop on Partial order methods in verification (POMIV), AMS Press, 1996, pp. 213–231.
- [9] D. Peled, Ten years of partial order reduction, in: Proceedings of the 10th International Conference on Computer Aided Verification (CAV), volume 1427 of *Lecture Notes in Computer Science*, Springer, 1998, pp. 17–28.
- [10] F. Lang, R. Mateescu, Partial order reductions using compositional confluence detection, in: Proceedings of the 2nd World Congress on Formal Methods (FM), volume 5850 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 157–172.
- [11] C. Baier, M. Größer, F. Ciesinski, Partial order reduction for probabilistic systems, in: Proceedings of the 1st International Conference on Quantitative Evaluation of Systems (QEST), IEEE Computer Society, 2004, pp. 230–239.
- [12] P. R. D’Argenio, P. Niebert, Partial order reduction on concurrent probabilistic programs, in: Proceedings of the 1st International Conference on Quantitative Evaluation of Systems (QEST), IEEE Computer Society, 2004, pp. 240–249.
- [13] C. Baier, P. R. D’Argenio, M. Größer, Partial order reduction for probabilistic branching time, in: Proceedings of the Third Workshop on Quantitative Aspects of Programming Languages (QAPL), volume 153(2) of *ENTCS*, Elsevier, 2006, pp. 97–116.
- [14] S. Giro, P. R. D’Argenio, L. M. F. Fioriti, Partial order reduction for probabilistic systems: A revision for distributed schedulers, in: Proceedings of the 20th International Conference on Concurrency Theory (CONCUR), volume 5710 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 338–353.
- [15] C. Baier, M. Größer, F. Ciesinski, Quantitative analysis under fairness constraints, in: Proceedings of the 7th International Symposium on Automated Technology for Verification and Analysis (ATVA), volume 5799 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 135–150.
- [16] H. Hansen, M. Kwiatkowska, H. Qu, Partial order reduction for model checking Markov decision processes under unconditional fairness, in: Proceedings of the 8th International Conference on Quantitative Evaluation of Systems (QEST), IEEE Computer Society, 2011, pp. 203–212.
- [17] M. Timmer, M. I. A. Stoelinga, J. C. van de Pol, Confluence reduction for probabilistic systems, in: Proceedings of the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), volume 6605 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 311–325.
- [18] M. Timmer, M. I. A. Stoelinga, J. C. van de Pol, Confluence Reduction for Probabilistic Systems (extended version), Technical Report 1011.2314, ArXiv e-prints, 2010.
- [19] R. Segala, Modeling and Verification of Randomized Distributed Real-Time Systems, Ph.D. thesis, Massachusetts Institute of Technology, 1995.
- [20] J. Bogdoll, L. M. F. Fioriti, A. Hartmanns, H. Hermanns, Partial order methods for statistical model checking and simulation, in: Proceedings of the Joint 13th IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS) and 31th IFIP International Conference on FORmal TEchniques for Networked and Distributed Systems (FORTE), volume 6722 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 59–74.
- [21] P. R. D’Argenio, B. Jeannet, H. E. Jensen, K. G. Larsen, Reduction and refinement strategies for probabilistic analysis, in: Proceedings of the 2nd Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification (PAPM-PROBMIV), volume 2399 of *Lecture Notes in Computer Science*, Springer, 2002, pp. 57–76.
- [22] C. Baier, J.-P. Katoen, Principles of Model Checking, MIT Press, 2008.
- [23] M. Größer, Reduction Methods for Probabilistic Model Checking, Ph.D. thesis, Technische Universität Dresden, 2008.
- [24] S. Evangelista, C. Pajault, Solving the ignoring problem for partial order reduction, International Journal on Software Tools for Technology Transfer 12 (2010) 155–170.
- [25] M. I. A. Stoelinga, Alea jacta est: verification of probabilistic, real-time and parametric systems, Ph.D. thesis, University of Nijmegen, 2002.
- [26] S. Katz, D. Peled, Defining conditional independence using collapses, Theoretical Computer Science 101 (1992) 337–359.
- [27] P. Godefroid, D. Pirotin, Refining dependencies improves partial-order verification methods, in: Proceedings of the 5th International Conference on Computer Aided Verification (CAV), volume 697 of *Lecture Notes in Computer Science*, Springer, 1993, pp. 438–449.