

A Comparison of Confluence and Ample Sets in Probabilistic and Non-Probabilistic Branching Time

Henri Hansen^a, Mark Timmer^b

^a *Tampere University of Technology
Institute of Mathematics
Email: henri.hansen@gmail.com*

^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, and answers the question how they relate in terms of reduction power when preserving branching time properties. We prove that, while both preserve the same 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.

The main challenge for the comparison is that confluence reduction was defined in an action-based setting, whereas ample set reduction is often 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 conditions under which the two notions coincide. 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 discuss 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 to ample sets.

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, for studying biological systems, and 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.

While reduction techniques preferably reduce as much as allowed by a relevant notion of bisimulation, in practice this is often infeasible. The computation may be complex and often requires the complete state space, while it is much more desirable to reduce on-the-fly, i.e., *prior to* the generation of the original state space. Therefore, reduction techniques often exchange reduction power for efficiency. Recently, two powerful techniques of this kind were generalised from non-probabilistic model checking 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

to guarantee that the reduced system is branching bisimilar to the original system; thus, these techniques preserve virtually all branching properties (in particular, CTL^*_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 CTL^*_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. Since both techniques are in general not able to achieve optimal reductions as compared to the bisimulation-minimal quotient, we are interested to see if there are scenarios that can be handled by one technique but not by the other, or whether their reduction capacities are equally powerful. Therefore, an obvious question is: to what extent do ample sets and confluent transition coincide? This paper addresses that question by comparing the notion of probabilistic ample sets from [13] to a state-based reformulation of 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 was originally defined in a purely action-based formalism. Also, the original definitions are insensitive to divergences, which in state-based approaches correspond to infinite stuttering. Unlike finite stuttering, infinite stuttering must be preserved in order to preserve PCTL^*_X . 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 show how the definitions need to be adjusted to make them coincide.

While revealing exactly where the extra reduction potential 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 and simulation. In this context, [20] used partial order reduction to remove spurious nondeterminism from models to allow them to be analysed statistically. As the reduction is applied directly to explicit models rather than high-level specifications, the more relaxed confluence conditions may come in handy. Indeed, [21] shows that confluence reduction is able to remove nondeterminism that partial order reduction could not, thereby allowing more models to be analysed using statistical model checking techniques. Our results provide theoretical support for this intuition. Since [20] applied a more powerful variant of partial order reduction, which only preserves linear time properties, there are also cases where confluence is able to reduce less [21]. Therefore, it seems beneficial to combine partial order reduction and confluence reduction for statistical model checking, applying both techniques if one of them fails.

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, our results (in particular Theorem 38) allow for the use of more relaxed definitions—incorporating a notion of *local* independence—if partial order reduction is used. 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 answer the question of how the non-probabilistic variants of ample set 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 [22]. 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 ample set 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 confluence and 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 $s \in S$. Given an equivalence relation $R \subseteq S \times S$ and two probability distributions $\mu, \nu \in \text{Distr}(S)$, we write $\mu \equiv_R \nu$ if $\mu(C) = \nu(C)$ for every equivalence class $C \in S/R$.*

The model on which probabilistic ample set 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 enabled action a , a probability distribution $P(s, a)$ specifies for every other state s' the likelihood $P(s, a)(s')$ of ending up in s' after taking 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;

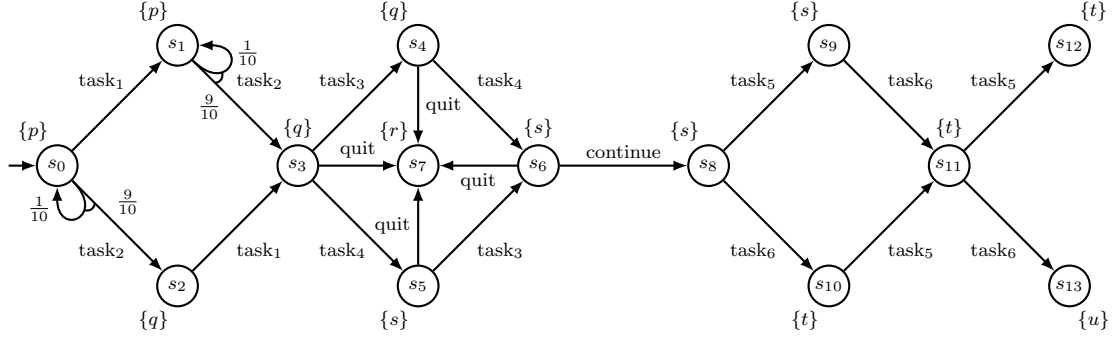


Figure 1: An MDP M representing a flow chart.

- $L: S \rightarrow 2^{\text{AP}}$ 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 (Supporting 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 $s' \in S$. Given a set $\mathcal{T} \subseteq \Delta_M$ we write $s \rightarrow_{\mathcal{T}} s'$ (reachability) if there is a path $s \xrightarrow{a_1 a_2 \dots a_n} s'$ and all the transitions of this path are in \mathcal{T} . In addition, we write $s \rightarrow_{\leftarrow \mathcal{T}} s'$ (joinability) if there is a state t such that $s \rightarrow_{\mathcal{T}} t$ and $s' \rightarrow_{\mathcal{T}} t$.

A subset of transitions of an MDP is acyclic if there does not exist a cycle in the subgraph of the MDP when only considering the transitions in this set.

Example 4. 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), while the odd-numbered tasks are invisible.

Note that for this MDP we have $S = \{s_i \mid 0 \leq i \leq 13\}$ and $\Sigma = \{\text{task}_i \mid 1 \leq i \leq 6\} \cup \{\text{quit}, \text{continue}\}$. The probabilistic transition function is visualised by arrows. For instance, $P(s_0, \text{task}_2) = \mu$ such that $\mu(s_0) = \frac{1}{10}$ and $\mu(s_2) = \frac{9}{10}$, and $P(s_0, \text{quit}) = \perp$. 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 $s_1 \xrightarrow{\text{task}_2} s_3$, for instance, and $\text{en}(s_3) = \{\text{quit}, \text{task}_3, \text{task}_4\}$ as well as $s_5 \xrightarrow{\text{task}_3 \text{ continue } \text{task}_5 \text{ task}_6} s_8$. \square

Definition 5 (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 on it is invisible.

We sometimes abuse the notation a little by writing (s, a, s') instead of $(s, a, \mathbb{1}_{s'})$ for deterministic transitions.

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 6. 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

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 7 (Reduction functions). Given an MDP $M = (S_M, \Sigma, P_M, s^0, \text{AP}, L_M)$, a reduction function is any function $R: S_M \rightarrow 2^\Sigma$ with $R(s) \subseteq \text{en}(s)$ for every $s \in S_M$. 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

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

The transitions in \bar{R} are called the nontrivial transitions of the reduction. We say that a reduction function R is acyclic if the original MDP restricted to the transitions in \bar{R} is acyclic.

In other words, \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 . Note that the reduction function is acyclic if there is no cycle of nontrivial transitions in the MDP.

Example 8. A possible reduction function R for the MDP in Figure 1 is given by $R(s_0) = \{task_1\}$, $R(s_1) = \{task_2\}$, $R(s_3) = \emptyset$ and $R(s_i) = en(s_i)$ for every other state s_i . 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 and this transition is no self-loop). \square

When reducing MDPs, we clearly want to retain some behaviour to still be able to verify certain properties. The reductions we deal with preserve $PCTL^*_X$ (a probabilistic variant of CTL^*_X ; see for instance [23]).

3. Ample Sets and Confluence for MDPs

This section presents the theory of the ample set reduction and confluence reduction techniques. While the ample set technique is just taken from literature, our definitions and correctness proofs for confluence reduction for MDPs are novel—although inspired by confluence reduction for PAs [17].

First, we need the concepts of weight functions and probabilistic visible (bi)simulation [24], as they will be used to prove that our redefined variant of confluence for MDPs also preserves $PCTL^*_X$.

Definition 9 (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 $\mu, \nu \neq \perp$ and 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 10 (Probabilistic visible bisimulation). *Let $M_1 = (S_1, \Sigma, P_1, s_1^0, AP, L_1)$ and $M_2 = (S_2, \Sigma, P_2, s_2^0, AP, L_2)$ 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 $a \in en(s)$, 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, $a \in en(s'')$ 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 (with $k > 0$) 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 11 (Independence). *Given an MDP $M = (S, \Sigma, P, s^0, AP, L)$, two actions $a, b \in \Sigma$ are independent if $a \neq b$ and for every state $s \in S$ such that $\{a, b\} \subseteq en(s)$ 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 [24] for an extended explanation of these conditions.

Definition 12 (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 $s_n = s$, $A(s_i) = \text{en}(s_i)$ for at least one $1 \leq i \leq n$;
- A4** If $A(s) \neq \text{en}(s)$, 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 $\text{en}(s) = \emptyset$. Note also that conditions A1 and A4 can be combined by saying that either $A(s) = \text{en}(s)$ or $A(s)$ contains exactly one invisible action.

Example 13. 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 $A(s_0) = \{\text{task}_1\}$ and $A(s_i) = \text{en}(s_i)$ 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 $A(s_1) = \text{en}(s_1)$. 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 a number of times task_2 and then task_1 ; for all traces of the second kind, task_2 is independent of $A(s_0)$ and task_1 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 14. *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.*

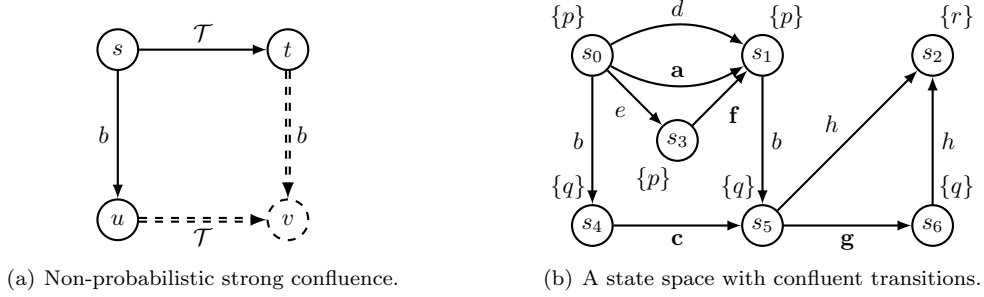


Figure 2: Non-probabilistic motivation.

3.2. Confluence

Confluence for action-based probabilistic automata was introduced in [17]. Here, we reformulate the theory in terms of MDPs in order to compare it to the ample set method. In [17], three variants of probabilistic confluence were introduced, differing in their ease of detection and their reduction power. Traditionally, notions of confluence (as of bisimulation) that are able to *distinguish* many systems are called *strong*, while the more *relaxed* notions are called *weak*. Hence, reduction power is inversely related to the strength of a notion's distinctive character.

In this work, we redefine strong probabilistic confluence. The weaker variants are more difficult to use in practice and would therefore not provide a fair comparison to ample sets. We weaken the notion of strong probabilistic confluence slightly from [17], to make it a true probabilistic generalisation of the earlier notion of non-probabilistic strong confluence from [5], except for the ability to preserve divergences. This way, our results also hold for strong confluence in a non-probabilistic setting, as presented in previous work. We first discuss non-probabilistic strong confluence as presented in [5], to provide motivation and intuition as a foundation for the probabilistic variant.

3.2.1. Non-probabilistic strong confluence

Strong confluence is based on a set \mathcal{T} of invisible transitions, that are all called *strongly confluent* if the set satisfies a certain confluence property. Basically, it should be the case that the transitions from \mathcal{T} can never interfere with observable behaviour. That is, an action that is enabled before a confluent transition should still be enabled after the transition (so that it can be *mimicked*). Moreover, the system should end up in the same state, regardless of whether the other transition is taken before or after the confluent transition. Because of this, confluent transitions can basically be given priority, omitting all other transitions from the states in which they are enabled.

Strong confluence can be defined diagrammatically, as in Figure 2(a). Here, we write \mathcal{T} above an arrow to indicate that the corresponding transition is in \mathcal{T} (and we do not care about the action name). The solid lines should be matched universally, while the dashed lines are meant to be existential. That is, for all states s, t, u, v in a system M such that there are transitions $(s, a, t) \in \mathcal{T}$ and $(s, b, u) \in \Delta_M$, there has to be a state v with transitions $(t, b, v) \in \Delta_M$ and $(u, c, v) \in \mathcal{T}$ for some c . Additionally, if the left b -transition is in \mathcal{T} , so should the right one be. Some states might coincide, so for instance a self-loop (s, b, s) can be seen as the transition (s, b, u) with $u = s$. The property should hold for all transitions in \mathcal{T} and all actions b .

To make things slightly more liberal, invisible actions do not necessarily have to be mimicked. Therefore, the double lines indicate that, in case $b \in \Sigma_{\text{inv}}$, it is also fine if t cannot do a b -transition, as long as $t = v$. Similarly, since all transitions in \mathcal{T} are invisible, it is also fine if there is no \mathcal{T} -transition from u and $u = v$.

Example 15. As an example, consider the MDP in Figure 2(b). Note that the actions a, c, d, e, f and g are invisible. We show that $\mathcal{T} = \{a, c, f, g\}$ is a valid strongly confluent set (indicated in bold in the figure).

First note that the confluence diagram always holds for a confluent transition with itself. Take for instance the match $s = s_4$ and $t = u = s_5$, with the c -transition being both of the outgoing transitions from

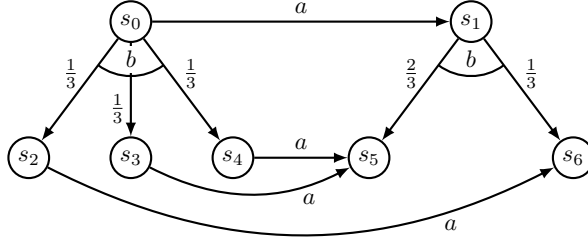


Figure 3: An MDP to demonstrate $\sim_{\mathcal{T}}$.

s . Since every \mathcal{T} -transition is invisible, we can take both $t = v$ and $u = v$ due to the double lines (which is valid since $t = u$), and the diagram holds vacuously. For c and f there is nothing more to check.

For a , there are three additional transitions for which the diagram has to hold. The b -transition can easily be seen to satisfy the diagram, since it can indeed be mimicked from s_1 and there is a \mathcal{T} -transition between the target states. For the d -transition, the diagram holds by taking $t = u = v$. Finally, for the e -transition (which is invisible itself), we can take $t = v$ and see that the diagram fits using $s = s_0$, $t = v = s_1$ and $u = s_3$. For g we should have u and v coincide, and take $s = s_5$, $t = s_6$ and $u = v = s_2$. \square

3.2.2. Probabilistic strong confluence

In a probabilistic setting, transitions do not have a single target state anymore; they go to a distribution over target states. Therefore, instead of requiring as above that u should have a \mathcal{T} -transition to v , we should require that the distribution corresponding to the b -transition from s is somehow related by \mathcal{T} -transitions to the distribution corresponding to the b -transition from t .

To define strong probabilistic confluence, we therefore introduce the notion of *equivalence up to \mathcal{T} -steps*: a way of saying that two probability distributions are basically the same, except for some intermediate transitions from a set \mathcal{T} .

Definition 16 (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 $\mu \sim_{\mathcal{T}} \nu$, 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 (by the $S_i = \{s_i\}$ clause). This corresponds to the fact that the lower \mathcal{T} -transition in Figure 2(a) is dashed, and is needed for our probabilistic notion of strong confluence to coincide with the existing non-probabilistic notion in a non-probabilistic context (except for divergences).

Example 17. Consider the MDP in Figure 3, 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 $\mu \sim_{\mathcal{T}} \nu$, 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

The next lemma states that, given a deterministic transition $(s, a, \mathbb{1}_{s'})$, the distribution from s associated with an action b independent of a is equivalent up to a -labeled-steps to the distribution associated with the same action from s' .

Lemma 18. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, and $a, b \in \Sigma$ 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 $\text{spt}(P(s, b))$. We use this partitioning to show that $P(s, b) \rightsquigarrow_{\mathcal{T}} P(s', b)$. 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 $\forall s \in R_t. \exists a \in \Sigma. (s, a, \mathbb{1}_t) \in \mathcal{T}$. \square

We define strong probabilistic confluence for sets of transitions \mathcal{T} , and require every transition in such a set to have an invisible action. Actions that were enabled before a confluent transition should still be enabled after the transition, and if a transition (s, b, μ) is to be mimicked by a transition (t, 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, as an equivalent transition already exists by definition.

Definition 19 (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 all its transitions have invisible actions, and for every $(s, a, \mathbb{1}_t) \in \mathcal{T}$ and every $b \in \text{en}(s)$ either*

- $P(s, b) \rightsquigarrow_{\mathcal{T}} P(t, b)$ and, if $(s, b, P(s, b)) \in \mathcal{T}$, then also $(t, b, P(t, b)) \in \mathcal{T}$, or
- $b \in \Sigma_{\text{inv}}$ and $P(s, b) = \mathbb{1}_t$.

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 $(s, a, \mu) \in \mathcal{T}$.

To motivate this definition, consider again the diagram in Figure 2(a). The first clause of our definition corresponds to the case where the b -transition from s is indeed mimicked from t . In the non-probabilistic case, we then required that there either is a confluent transition from u to v , or that $u = v$. In the probabilistic case, this corresponds to requiring that $P(s, b) \rightsquigarrow_{\mathcal{T}} P(t, b)$. Also, just like in the non-probabilistic case, we require that if the b -transition from s is confluent, then so is the one from t .

If $b \in \Sigma_{\text{inv}}$, as stated by the second clause, then the states t and v in Figure 2(a) are allowed to coincide. If $P(s, b) = \mathbb{1}_t$, then apparently also t and u coincide, and the diagram holds vacuously. So, the second clause corresponds to the case that both the path from u to v and the path from t to v is empty.

Looking at the non-probabilistic diagram, there is one last possibility: if $b \in \Sigma_{\text{inv}}$, it is also allowed that it is not mimicked by t (so $t \neq v$), that $u \neq v$, and there is a \mathcal{T} -transition from u to v . A clause dealing with this case would be “ $b \in \Sigma_{\text{inv}}$ and $\exists c. (\text{target}(s, b), c, \mathbb{1}_t) \in \mathcal{T}$ ”. In the non-probabilistic action-based setting, the object is to preserve branching bisimilarity. Branching bisimulation as an equivalence does not require loops of invisible actions, i.e., divergences, to be preserved. In the current context we want to prove probabilistic visible bisimulation, which requires divergence to be preserved by confluent transitions (otherwise, minimal reachability probabilities might change). To demonstrate that the suggested third clause would not preserve enough behaviour, consider the MDP in Figure 4.

Here, $\mathcal{T} = \{b\}$ would be a valid strongly probabilistically confluent set if this additional third clause would be taken into account. After all, a is invisible and indeed there is a confluent transition from s_0 to s_1 .

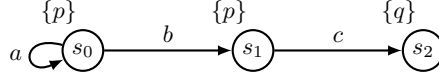


Figure 4: An MDP to demonstrate strong probabilistic confluence.

A reduction based on this set, keeping only the b -transition from s_0 and omitting the a -transition, would change the minimal reachability probability from s_0 of the atomic proposition q from 0 to 1. For this reason, we need a stronger condition than the non-probabilistic version of strong confluence and omit this clause. We are now ready to define confluence reduction functions.

Definition 20 (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* [3, 25], and is dealt with by the cycle condition A3 of the ample set method. With strong confluence this problem can be dealt with by requiring acyclicity, and in Section 5 we will look at an alternative approach.

Example 21. 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 have invisible actions. 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 due to the clause $S_i = \{s_i\}$ in its definition. 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 5. 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

3.2.3. Correctness

Our main result here is Theorem 25, which establishes the correctness of acyclic confluence reduction functions. The following two lemmas and corollary first give us some tools. For starters, we provide a lemma stating that the joinability relation for confluent transitions (i.e., $\rightarrow \leftarrow \mathcal{T}$) is an equivalence relation.

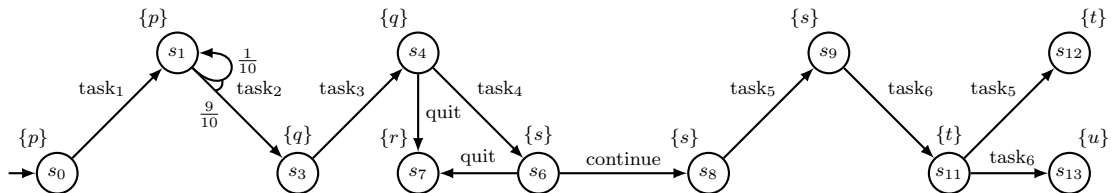


Figure 5: A reduced MDP.

Lemma 22. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, and $\mathcal{T} \subseteq \Delta_M$ a strongly probabilistically confluent set of transitions of M . Then, the set $R = \{(s, s') \mid s \twoheadrightarrow_{\leftarrow \mathcal{T}} s'\}$ is an equivalence relation.*

Proof. Clearly, R is reflexive and symmetric by construction. So, we only need to prove transitivity.

Let $s' \twoheadrightarrow_{\leftarrow \mathcal{T}} s \twoheadrightarrow_{\leftarrow \mathcal{T}} s''$. We show that $s' \twoheadrightarrow_{\leftarrow \mathcal{T}} s''$. Let t' be a state such that $s \twoheadrightarrow_{\mathcal{T}} t'$ and $s' \twoheadrightarrow_{\mathcal{T}} t'$, and likewise, let t'' be a similar state for s and s'' . If we can show that there is some state t such that $t' \twoheadrightarrow_{\mathcal{T}} t$ and $t'' \twoheadrightarrow_{\mathcal{T}} t$, we have the result. Let a minimal confluent path from s to t' be given by $s_0 \xrightarrow{a_1 a_2 \dots a_n} s_n$, with $s_0 = s$ and $s_n = t'$. By induction on the length of this path, we show that for each state s_i on it, there is some state t such that $s_i \twoheadrightarrow_{\mathcal{T}} t$ and $t'' \twoheadrightarrow_{\mathcal{T}} t$. Since t' is also on the path, this completes the argument.

Base case. There clearly is a state t such that $s_0 \twoheadrightarrow_{\mathcal{T}} t$ and $t'' \twoheadrightarrow_{\mathcal{T}} t$, namely t'' itself. After all, $s_0 = s$ and $s \twoheadrightarrow_{\mathcal{T}} t''$, and $\twoheadrightarrow_{\mathcal{T}}$ is reflexive.

Inductive case. Let there be a state t_k such that $s_k \twoheadrightarrow_{\mathcal{T}} t_k$ and $t'' \twoheadrightarrow_{\mathcal{T}} t_k$. We show that there exists a state t_{k+1} such that $s_{k+1} \twoheadrightarrow_{\mathcal{T}} t_{k+1}$ and $t'' \twoheadrightarrow_{\mathcal{T}} t_{k+1}$. Let $(s_k, a, u) \in \mathcal{T}$ be the first transition on the \mathcal{T} -path from s_k to t_k . Let $(s_k, a_{k+1}, s_{k+1}) \in \mathcal{T}$ be the \mathcal{T} -transition between s_k and s_{k+1} . By Definition 19, either (1) $s_{k+1} = u$, or (2) $\mathbb{1}_{s_{k+1}} \rightsquigarrow_{\mathcal{T}} P(u, a_{k+1})$ and $(u, a_{k+1}, P(u, a_{k+1})) \in \mathcal{T}$.

In case (1), we directly find $s_{k+1} \twoheadrightarrow_{\mathcal{T}} t_k$. Hence, we can just take $t_{k+1} = t_k$. In case (2), there is some state u' such that $(u, a_{k+1}, u') \in \mathcal{T}$ and either $s_{k+1} = u'$ or $(s_{k+1}, c, u') \in \mathcal{T}$ for some action c . If $u = t_k$, we can take $t_{k+1} = u'$ and indeed $s_{k+1} \twoheadrightarrow_{\mathcal{T}} t_{k+1}$ and $t'' \twoheadrightarrow_{\mathcal{T}} t_{k+1}$. Otherwise, we can show as above that there is a state t_{k+1} such that $u' \twoheadrightarrow_{\mathcal{T}} t_{k+1}$ and $t'' \twoheadrightarrow_{\mathcal{T}} t_{k+1}$, based on $u \twoheadrightarrow_{\mathcal{T}} t_k$ and $t'' \twoheadrightarrow_{\mathcal{T}} t_k$. Since the path from u to t_k is one transition shorter than the path from s_k to t_k , this argument terminates. \square

Based on the lemma above, it is immediate that confluent transitions can always join again without having to take any non-confluent transitions. We state this property in terms of terminal SCCs—i.e., maximal strongly connected subgraphs without any outgoing transitions—of the system obtained when omitting all non-confluent transitions.

Corollary 23. *Consider an MDP M , a strongly probabilistically confluent set of transitions \mathcal{T} , and the subgraph of M obtained by keeping only the transitions that are in \mathcal{T} . Then, for every state s in this subgraph, there is a unique terminal SCC.*

Proof. This is an immediate consequence of the fact that $R = \{(s, s') \mid s \twoheadrightarrow_{\leftarrow \mathcal{T}} s'\}$ is an equivalence relation. After all, consider a state s with multiple outgoing \mathcal{T} -transitions, for instance to s' and s'' . Then, $s \twoheadrightarrow_{\leftarrow \mathcal{T}} s'$ and $s \twoheadrightarrow_{\leftarrow \mathcal{T}} s''$, and hence by transitivity of R also $s' \twoheadrightarrow_{\leftarrow \mathcal{T}} s''$. So, s' and s'' cannot be part of distinct terminal SCCs, as there is a state they can both reach. \square

As a last preparation for the main theorem of this section, we show that $\mu \rightsquigarrow_{\mathcal{T}} \nu$ implies that μ and ν assign the same probabilities to sets of states that are joinable by confluent transitions.

Lemma 24. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, and $\mathcal{T} \subseteq \Delta_M$ a strongly probabilistically confluent set of transitions of M . Also, let $R = \{(s, s') \mid s \twoheadrightarrow_{\leftarrow \mathcal{T}} s'\}$. Then, $\mu \rightsquigarrow_{\mathcal{T}} \nu$ implies $\mu \equiv_R \nu$.*

Proof. First of all, by Lemma 22 we find that R indeed is an equivalence relation.

Let $\mu \rightsquigarrow_{\mathcal{T}} \nu$, i.e., $\mu, \nu \neq \perp$ 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

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

Now let R' be the smallest equivalence relation that relates the states of every set S_i to each other and to their corresponding s_i . That is, for every S_i and for all $s, s' \in S_i$, let $(s, s') \in R'$ and $(s, s_i) \in R'$. Since $\mu(S_i) = \nu(s_i)$ for every S_i , clearly $\mu \equiv_{R'} \nu$.

Since a transition $(s, a, s_i) \in \mathcal{T}$ implies $s \twoheadrightarrow_{\leftarrow \mathcal{T}} s_i$, we find $(s, s_i) \in R$ for all $s \in S_i$ in every S_i . Also for every S_i and all $s, s' \in S_i$ we have $s \twoheadrightarrow_{\leftarrow \mathcal{T}} s'$, since they either coincide or can join at s_i (so also $(s, s') \in R$). Since R' is the smallest equivalence relation having these properties, we find that $R \supseteq R'$. Because of this, $\mu \equiv_{R'} \nu$ implies $\mu \equiv_R \nu$ (using Proposition 5.2.1.1 and 5.2.1.5 from [26]), which is what we wanted to show. \square

We are now able to prove the following theorem, stating that acyclic confluence reduction functions are correct with respect to probabilistic visible bisimulation. Note that acyclicity was introduced in Definition 7.

Theorem 25. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, \mathcal{T} a strongly probabilistically confluent set of transitions from M and T an acyclic confluence reduction function under \mathcal{T} . Let $M_T = (S_T, \Sigma, P_T, s^0, \text{AP}, L_T)$ be the reduced MDP. Then, $M \equiv_{\text{pvb}} M_T$.*

Proof. In this proof, whenever we write that a transition is ‘confluent’, we mean that it is in \mathcal{T} . Similarly, a ‘confluent path’ in this proof is a path consisting only of transitions from \mathcal{T} .

Let $R = \{(s, s') \in S \times S \mid s \twoheadrightarrow_{\mathcal{T}} s'\}$ be the relation that relates all states that can join by traversing only confluent transitions. By Lemma 22, it is an equivalence relation. Let $\mathcal{R} = R \cap (S \times S_T)$, i.e., it restricts R to relating only states of the original MDP to the reduced one. Note that \mathcal{R} is still transitive, and that it is reflexive for the state of S_T . We first prove that \mathcal{R} is a probabilistic visible simulation for (M, M_T) . Note that $(s^0, s^0) \in \mathcal{R}$, since \mathcal{R} is reflexive for states in S_T and indeed $s^0 \in S_T$ due to the fact that reduction functions preserve initial states. For the additional conditions of probabilistic visible simulation, let $(s, s') \in \mathcal{R}$, so $s \in S$, $s' \in S_T$ and $s \twoheadrightarrow_{\mathcal{T}} s'$.

1. $L(s) = L_T(s')$ holds because all confluent transitions are invisible (and hence stuttering).
2. Let $a \in \text{en}(s)$. Since $(s, s') \in \mathcal{R}$ and hence also $(s, s') \in R$, there is at least one state in S that is reachable in M using confluent paths from both s and s' . Combining this fact with Corollary 23, we find that there is a unique terminal SCC of the subgraph of M obtained by keeping only the transitions that are in \mathcal{T} , that can be reached in M from s and s' by following only confluent transitions. Since reduction functions preserve at least one confluent transition in each state that has at least one such transition and T is acyclic, s' can still reach a subgraph of this terminal SCC in M_T . Let t be a state in this subgraph such that $T(t) = \text{en}(t)$, i.e., t is fully expanded. Such a state exists, due to acyclicity of the reduction.

So, there is a confluent path from s to t in M , and there is a confluent path from s' to t in M_T . Therefore, $(s, s'_i) \in \mathcal{R}$ for all states s'_i on the path from s' to t in M_T . Since all transitions on this path are confluent, the path is invisible, and it can be used to satisfy condition 2(b) of the definition of probabilistic visible simulation. We only still need to show that $a \in \text{en}(t)$ and that $P(s, a) \sqsubseteq_{\mathcal{R}} P_T(t, a)$. Since t is fully expanded, $P_T(t, a) = P(t, a)$, so we just need to prove that $P(s, a) \sqsubseteq_{\mathcal{R}} P(t, a)$.

Let $s_0 \xrightarrow{b_1} s_1 \xrightarrow{b_2} \dots \xrightarrow{b_n} s_n$ with $s_0 = s$ and $s_n = t$ be the confluent path from s to t . We show by induction on its length that either $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}$, or $P(s, a) \equiv_R P(s_i, a)$ for every $0 \leq i \leq n$ (note that we use \equiv_R instead of $\equiv_{\mathcal{R}}$). The first part coincides with condition 2(a) of Definition 10. The second part can be instantiated with $i = n$ to obtain $P(s, a) \equiv_R P(t, a)$ and thus also $P(s, a) \sqsubseteq_R P(t, a)$ (using Proposition 5.2.1.1 from [26]). As t is fully expanded, every state in the support of $P(t, a)$ is in S_T , so $P(s, a) \sqsubseteq_R P(t, a)$ implies $P(s, a) \sqsubseteq_{\mathcal{R}} P(t, a)$, which coincides with condition 2(b) of Definition 10 (using the confluent path from s' to t in M_T discussed above).

Base case. For s_0 , we immediately obtain $P(s, a) \equiv_R P(s_0, a)$ from the fact that $s_0 = s$ and the reflexivity of the $\equiv_{\mathcal{R}}$ relation.

Inductive case. Let either $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}$, or $P(s, a) \equiv_R P(s_i, a)$ for every $0 \leq i \leq k$ from some $k < n$. In case the first part of this disjunction is true, we are done. So, we assume $P(s, a) \equiv_R P(s_k, a)$ and prove either $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}$ or $P(s, a) \equiv_R P(s_{k+1}, a)$. We make a case distinction:

- (a) Let $a \in \Sigma_{\text{inv}}$ and $P(s_k, a) = \mathbf{1}_{s_{k+1}}$. Notice that $P(s, a) \equiv_R P(s_k, a)$, combined with the facts that $a \in \Sigma_{\text{inv}}$ and $P(s_k, a) = \mathbf{1}_{s_{k+1}}$, yields $(\text{target}(s, a), s_{k+1}) \in R$. Since there is a direct confluent path from s_{k+1} to t and one from s' to t , also $(s_{k+1}, s') \in R$. Finally, by transitivity of R we find $(\text{target}(s, a), s') \in R$ and since $s' \in M_T$, also $(\text{target}(s, a), s') \in \mathcal{R}$.
- (b) Let $a \notin \Sigma_{\text{inv}}$ or $P(s_k, a) \neq \mathbf{1}_{s_{k+1}}$. Then, by definition of confluence $P(s_k, a) \rightsquigarrow_{\mathcal{T}} P(s_{k+1}, a)$. Lemma 24 yields $P(s_k, a) \equiv_R P(s_{k+1}, a)$. By transitivity of \equiv_R , we find $P(s, a) \equiv_R P(s_{k+1}, a)$.

3. Let $s \xrightarrow{b_1 b_2 \dots}$ be an invisible path such that $(s_i, s') \in \mathcal{R}$ for every state s_i on the path. As shown above in part (2) of this proof, $(s, s') \in \mathcal{R}$ implies that there is a state t such that $T(t) = en(t)$, there is a confluent path from s to t in M and there is a confluent path from s' to t in M_T .

If s' has an outgoing confluent transition (so with an invisible action) to some state s'' , then $(s', s'') \in \mathcal{R}$ and by transitivity of \mathcal{R} also $(s_i, s'') \in \mathcal{R}$ for every i . Hence, the condition is met.

So, assume that s' does not have an outgoing confluent transition, and thus $t = s'$. Hence, there is a confluent path from s to s' . Note that 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 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 $s_k \xrightarrow{b_{k+1}} s_{k+1} \xrightarrow{b_{k+2}} \dots$. 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 the infinite invisible path $s'_j \xrightarrow{\tau \tau \dots}$, so $s'_j \xrightarrow{b} s^*$ for some b (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 $\text{target}(s'_j, b) = s'_{j+1}$. The second option is impossible, since $\text{target}(s'_j, b) = s^*$ is on the infinite path $s'_j \xrightarrow{\tau \tau \dots}$ 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 10. 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 $(v, s^*) \in \mathcal{R}^*$. It is easy to see from the proof that v corresponds to a state s_i on $s \xrightarrow{b_1 b_2 \dots}$ with $i > 0$, as required by Definition 10.

To see that \mathcal{R}^{-1} is a probabilistic visible simulation for (M_T, M) , we can use the same as 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 [24], gives the following corollary.

Corollary 26. *If T is an acyclic confluence reduction function for M , then M and M_T satisfy the same $\text{PCTL}^*_{\setminus X}$ -formulae.*

3.2.4. Weak confluence

Many weaker definitions of probabilistic confluence can be given. Here we provide one, based on the notion of action-based weak probabilistic confluence from [17].

Definition 27 (Weak Probabilistic Confluence). *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, $\mathcal{T} \subseteq \Delta_M$ a set of transitions from M and $R = \{(s, s') \in S \times S \mid s \twoheadrightarrow_{\mathcal{T}} s'\}$ a relation over its states. Then, \mathcal{T} is weakly probabilistically confluent if $a \in \Sigma_{\text{inv}}$ for every transition $(s, a, \mu) \in \mathcal{T}$, and*

- *The relation R is an equivalence relation, and*
- *For every path $s \twoheadrightarrow_{\mathcal{T}} t$ and for every $a \in \Sigma$, $(s, a, \mu) \in \Delta_M$ implies $\exists t' \in S . t \twoheadrightarrow_{\mathcal{T}} t'$ such that either $P(t', a) \equiv_R \mu$ or $a \in \Sigma_{\text{inv}}$ and $\mu \equiv_R \mathbb{1}_{t'}$.*

Note that reflexivity and symmetry of R are immediate. Transitivity basically corresponds to requiring that two outgoing \mathcal{T} -transitions from the same state can always join again following only \mathcal{T} -transitions. This yields the very appealing property that, when only following \mathcal{T} -transitions, we always end up in a unique terminal strongly connected component (as we also used above with strong probabilistic confluence).

As expected, weak probabilistic confluence is implied by strong probabilistic confluence.

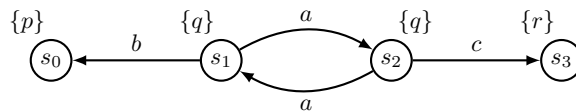
Proposition 28. *A strongly probabilistically confluent set of transitions is weakly probabilistically confluent.*

Proof. Let \mathcal{T} be a strongly confluent set of transitions. We need to prove two things. Firstly, we need to show that the relation $R = \{(s, s') \mid s \twoheadrightarrow_{\mathcal{T}} s'\}$ is an equivalence relation. This was already proven in Lemma 22. Secondly, we need to show that $s \twoheadrightarrow_{\mathcal{T}} t$ implies that for every a such that $P(s, a) \neq \perp$, there exists a state t' such that $t \twoheadrightarrow_{\mathcal{T}} t'$ and either $P(s, a) \equiv_R P(t', a)$ or $P(s, a) \equiv_R \mathbb{1}_{t'}$ and $a \in \Sigma_{\text{inv}}$.

For the second part, strong confluence guarantees that if $P(s, a) \neq \perp$, then on all confluent paths that start from s , a is never disabled (unless it is invisible). More formally, let $s = s_0 \xrightarrow{c_1} s_1 \xrightarrow{c_2} \dots \xrightarrow{c_n} s_n = t$ be a path from s to t such that all transitions are in \mathcal{T} . First, assume that $a \notin \Sigma_{\text{inv}}$. Then, we know by strong confluence that $P(s_{i-1}, a) \rightsquigarrow_{\mathcal{T}} P(s_i, a)$ for every $0 < i \leq n$, which by Lemma 24 implies that also $P(s_{i-1}, a) \equiv_R P(s_i, a)$. Then, transitivity of R gives the result.

If $a \in \Sigma_{\text{inv}}$, possibly at some point $P(s_i, a) = \mathbb{1}_{s_{i+1}}$. Since the above arguments apply until this point, $P(s, a) \equiv_R P(s_i, a)$. Moreover, since $s_{i+1} \twoheadrightarrow_{\mathcal{T}} t$, we find $\mathbb{1}_{s_{i+1}} \equiv_R \mathbb{1}_t$, so since $P(s_i, a) = \mathbb{1}_{s_{i+1}}$ also $P(s_i, a) \equiv_R \mathbb{1}_t$, and by transitivity of \equiv_R we obtain $P(s, a) \equiv_R \mathbb{1}_t$. \square

Although states that are connected by weakly confluent transitions can be shown to have the same observable behaviour, this fact is hard to use for state space reduction. For instance, consider this MDP:



Here, both a -transitions are weakly confluent, as the observable transitions are not disabled. Indeed, s_1 and s_2 are branching bisimilar in the sense that they have the same behaviour modulo invisible actions. However, a reduction function R that chooses for instance $R(s_1) = \{a\}$ would not be valid, as the observable b -transition is now not reachable anymore from s_1 and s_2 , while it was before. Hence, this reduction function based on a weakly confluent set is not sound, even though it is acyclic. A solution would be to merge all the equivalent states into one big state, but in practice this is much less convenient. For this reason, we will use strong probabilistic confluence in our comparison to ample sets.

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, by proving that every ample set reduction also is a confluence reduction. In addition to this, we discuss the aspects that differentiate ample sets from confluence. To show that these are the only differences, we provide variations to the concepts that make them coincide. The choice of which concept is varied in each situation, is to a large extent arbitrary. Restricting confluence or relaxing ample sets is not the issue here, the objective is to prove that we have identified the essential differences. However, the variations are made in such a way that the resulting notions are useful in practice. Restrictions of confluence rule out features that are plausibly hard to implement in practice, and relaxed features of ample sets are such that they have been used in practice.

4.1. Why confluence is strictly more powerful

The starting point of our investigation is given by Theorem 29. 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. In the following, “confluence” refers to the notion of strong confluence of Definition 19.

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 (the *nontrivial transitions*); otherwise, $\bar{A}(s)$ is the empty set (Definition 7).

Theorem 29. *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 strongly 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 $(s, a, \mathbb{1}_t) \in \mathcal{T}_A^*$ 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 $(s, a, \mathbb{1}_t) \in \mathcal{T}_A^*$ 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 18. 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 trivial) 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.

Also note that, if $(s, b, \mu) \in \mathcal{T}_A^*$ too, then for confluence it has to be mimicked by a confluent transition. Indeed, since $(s, b, \mu) \in \mathcal{T}_A^*$ and $a \in \text{en}(s)$, by construction also the b -transition from t is in \mathcal{T}_A^* . \square

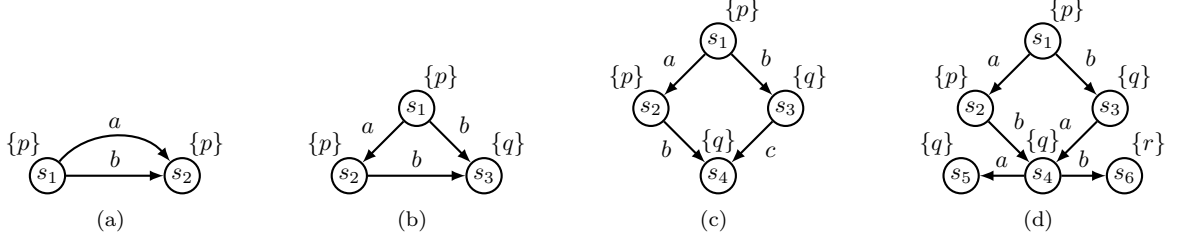


Figure 6: Confluence triumphs over ample sets.

This result obviously holds for weaker notions of confluence (probabilistic confluence from [17] and weak probabilistic confluence), which are even more powerful than strong probabilistic confluence. 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, some of which are illustrated by the following examples.

Example 30. Consider the MDPs in Figure 6 (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, some confluence reduction is allowed to omit their outgoing b -transitions, removing six transitions and two states.

In Figure 6(a), also the b -transition is invisible. Due to the part $b \in \Sigma_{\text{inv}}$ and $P(s, b) = \mathbf{1}_t$ of the disjunction in Definition 19, 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 $\mathcal{T} = \{(s_1, a, \mathbf{1}_{s_2})\}$. 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 6(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 $\mathcal{T} = \{(s_1, a, \mathbf{1}_{s_2})\}$ as the underlying confluent set for confluence reduction, due to the part $S_i = \{s_i\}$ of the disjunction in Definition 16 (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, ample set 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 6(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 ample set 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 6(c), the a -transition can be considered confluent since the diamond shape is closed perfectly (taking $\mathcal{T} = \{(s_1, a, \mathbf{1}_{s_2}), (s_3, c, \mathbf{1}_{s_4})\}$). 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 6(d), the outgoing a -transition from s_1 is confluent since the diamond shape of independence is present (taking $\mathcal{T} = \{(s_1, a, \mathbf{1}_{s_2}), (s_3, a, \mathbf{1}_{s_4})\}$). 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 ample set 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, which diminishes the difference [4, 17].

4.2. Making confluence and ample sets coincide

To show that the differences discussed above are indeed the only differences between confluence and ample sets, we remove them and show that the resulting notions indeed coincide. As a first step, we precisely prohibit all the liberal aspects of confluence that make the reductions in Figure 6(a), 6(b) and 6(c) work. When looking at Figure 2(a), this implies changing the double lines to single lines (and hence not allowing ‘shortcuts’ anymore). 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 ample set reduction to optimise Figure 6(d). Note that we do this safely, i.e., Theorem 25 is never compromised in the process, as all these notions will still be confluent in the sense used in that theorem.

Restricted confluence. First of all, we strengthen equivalence up to \mathcal{T} -steps to force it to 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 6(b) anymore.

Definition 31 (Restricted 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*

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

When symbolic analysis is carried out for ample sets and similar methods, the relations that are extracted are usually assumed symmetric: if a and b are independent, then they do not disable each other. This is much due to the way algorithms for generating them often work (though not always, see for instance [16]). The above stronger version of up-to-equivalence features this same symmetry.

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 6(a).

Definition 32 (Restricted 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 restrictedly probabilistically confluent if all its transitions have invisible actions, and for every $(s, a, \mathbb{1}_t) \in \mathcal{T}$ and every $b \in \text{en}(s)$ ($b \neq a$), it holds that*

- $P(s, b) \rightsquigarrow_{\mathcal{T}}^* P(t, b)$ and, if $(s, b, P(s, b)) \in \mathcal{T}$, then also $(t, b, P(t, b)) \in \mathcal{T}$.

We call a reduction function with an underlying restricted confluent set a restricted confluence reduction function.

We add the restriction $b \neq a$, as without it, confluent transitions would not commute with themselves anymore. Since in the original definition every confluent transition also already commuted with itself, this does not weaken the concept. Hence, Definition 32 is a true restriction of Definition 19.

Finally, we saw in Figure 6(c) 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 ample set 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 6(c) are not allowed anymore.

Definition 33 (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 [27] and [28], 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 34 (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);*
- $\sum_{s^* \in S} P(s', a)(s^*) \cdot P(s^*, b)(t) = \sum_{s^* \in S} P(s', b)(s^*) \cdot P(s^*, a)(t)$, 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 35. In Example 13 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 36. *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*

$$\{a, b\} \subseteq \text{en}(s') \implies P(s', b) \rightsquigarrow_{\mathcal{T}}^* P(\text{target}(s', a), b)$$

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 $(s', a, t) \in \mathcal{T}$.

Due to the implication assumed by this lemma, $P(s', b) \rightsquigarrow_{\mathcal{T}}^* P(t, b)$. Now, from this and the fact that \mathcal{T} only contains a -transitions, using 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 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 18 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 37 (Relaxed ample sets). *A set $A(s)$ is a relaxed ample set if it meets the criteria of Definition 12, 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 38. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP. Then, $T: S \rightarrow 2^\Sigma$ is an acyclic action-separable restricted 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 acyclic action-separable restricted 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 confluent $a \in \Sigma_{\text{inv}}$.

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 36, 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 restricted confluence $P(s, a_1) \rightsquigarrow_{\mathcal{T}}^* P(\text{target}(s, a), a_1)$. Now, by definition of $\rightsquigarrow_{\mathcal{T}}^*$ 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 confluent 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 $P(s', b) \rightsquigarrow_{\mathcal{T}}^* P(\text{target}(s', a), b)$. ($b \neq a$ 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 29: 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 18. (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 18, 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 restricted confluence reduction function. \square

Note that an action-separable restricted confluence reduction function is just a special case of the liberal definition of confluence, used in Theorem 25, 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 (except that we preserve divergences), the following corollary is also immediate.

Corollary 39. *In the non-probabilistic setting, confluence reduction is able to reduce more than ample set reduction. With some adjustments (as in Definitions 31, 32, 33, 34 and 37), the two notions coincide.*

5. Practical Implications of the Theory

To further reduce the number of states, we adapt the probabilistic confluence reduction technique of [17], which uses the method of representative states, as introduced in [4]. A highly similar construction was used in [22] 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 pick one of them as a representative for the set and omit the other states. To make sure that all visible transitions are enabled immediately from the representative, the representative has to be chosen from the terminal strongly connected component (TSCC) of the subgraph spanned by confluent transitions. The representative can easily be found using a variant of Tarjan's algorithm for strongly connected components, as explained in detail in [4, 5]: The algorithm follows confluent transitions until it detects a TSCC and then picks a state to serve as representative for all the states that can reach it via confluent transitions.

We now introduce the technicalities needed for the use of a representation map with MDPs in such a way as to preserve probabilistic visible bisimulation.

Definition 40 (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*

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

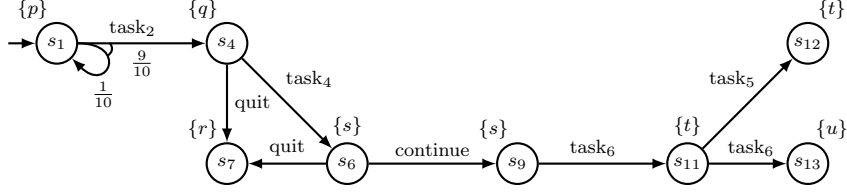


Figure 7: A quotient MDP under a representation map.

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 41. *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an MDP, and \mathcal{T} a strongly confluent set of transitions from M . Then, there exists a representation map for M under \mathcal{T} .*

Proof. By Corollary 23, we know that for each state s there is a unique terminal SCC of the subgraph of M obtained by keeping only the transitions that are in \mathcal{T} , that can be reached from s . For each terminal SCC, we choose one state t to be the representative of all states s that can reach that SCC by following only \mathcal{T} -transitions. Now, by construction indeed $s \rightarrow_{\mathcal{T}} \phi_{\mathcal{T}}(s)$. Also, if $(s, a, \mathbb{1}_{s'}) \in \mathcal{T}$, then s can reach the same terminal SCCs as s' when following \mathcal{T} -transitions. Since both have a unique such terminal SCC, these must coincide, and hence s and s' indeed have the same representative. \square

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

Definition 42 (Quotient MDP). *Let $M = (S, \Sigma, P, s^0, \text{AP}, L)$ be an 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 43. Consider again the MDP in Figure 1 and the strongly confluent set provided in Example 21. As stated there, $\mathcal{T} = \{(s_0, \text{task}_1, s_1), (s_2, \text{task}_1, s_3), (s_3, \text{task}_3, s_4), (s_5, \text{task}_3, s_6), (s_8, \text{task}_5, s_9), (s_{10}, \text{task}_5, s_{11})\}$. In the absence of cycles in \mathcal{T} , there is only one possible representation map under \mathcal{T} :

$$\phi_{\mathcal{T}}(s_0) = s_1 \quad \phi_{\mathcal{T}}(s_2) = s_4 \quad \phi_{\mathcal{T}}(s_3) = s_4 \quad \phi_{\mathcal{T}}(s_5) = s_6 \quad \phi_{\mathcal{T}}(s_8) = s_9 \quad \phi_{\mathcal{T}}(s_{10}) = s_{11}$$

and $\phi_{\mathcal{T}}(s) = s$ for all other states s . The quotient MDP under this representation map is shown in Figure 7. \square

The following theorem states that the use of representation maps to reduce a state space preserves probabilistic visible bisimulation.

Theorem 44. *Let M be an MDP, and \mathcal{T} a strongly confluent set of transitions from M . If ϕ is a representation map for M under \mathcal{T} , then $M \equiv_{\text{pvb}} M_{\phi}$.*

Proof. 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 we have $s \rightarrow_{\mathcal{T}} \phi(s)$, and since \mathcal{T} is a confluent set, 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, since $s \rightarrow_{\mathcal{T}} s'$ and all \mathcal{T} -transitions are invisible.
2. Let $a \in \text{en}(s)$. Using the same inductive argument as in the proof of Theorem 25 (with the only extra information that states connected by confluent transitions have the same representative), we can show that either $a \in \Sigma_{\text{inv}} \wedge (\text{target}(s, a), s') \in \mathcal{R}$, or $P(s, a) \rightsquigarrow_{\mathcal{T}} P(s_1, a) \rightsquigarrow_{\mathcal{T}} \dots \rightsquigarrow_{\mathcal{T}} P(s', a)$. In the first case, condition 2(a) of Definition 10 follows immediately, so we assume the sequence of $\rightsquigarrow_{\mathcal{T}}$ -steps. Hence, $a \in \text{en}(s')$ in M , and by definition of the quotient, this also implies that $a \in \text{en}(s')$ in M_ϕ .

To show that condition 2(b) holds, we don't need an invisible path in the quotient (we can just take the empty path). So, we just need to prove that $P(s, a) \sqsubseteq_{\mathcal{R}} P_\phi(s', a)$. To do so, 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. 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 25, 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 $\phi(v) = s'$ (since it is on $s \xrightarrow{b_1 b_2 \dots}$), 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 representation maps. 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 M_ϕ and the second from the definition of w .

3. Let $s \xrightarrow{b_1 b_2 \dots}$ be an infinite invisible path of M_ϕ such that $(s_i, s'_i) \in \mathcal{R}^{-1}$ 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 44 is useful, not only for confluence reduction, but also for ample set reduction. After all, from Theorem 29 we know that every ample set reduction is a confluence reduction. The representation map approach serves as an alternative implementation of the cycle condition of ample sets. The cycle condition is satisfied in the sense that the quotient MDP never indefinitely ignores any behaviour of the original MDP.

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 in branching time. 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 and ample sets have to be modified 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.

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. Future work could focus more on the relative merits of the two notions in practice and potentially on the improvement of the syntactical heuristics.

A natural question is, whether there are similar results that could be proven for weaker semantics, like reductions that preserve (probabilistic) $\text{LTL}_{\setminus X}$. For most part, the answer is obvious: confluence reduction preserves branching time properties, so it also preserves $\text{LTL}_{\setminus X}$. 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, and might then reduce more.

One class of open and interesting questions remains, however. When aiming to prove Theorem 38, we worked mostly by *restricting* confluence. It is sensible to ask, if we could have proven the theorem by relaxing the ample set conditions such as the notion of independence 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 $\text{LTL}_{\setminus X}$? 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), by the EU under FP7 grant 214755 (QUASIMODO), and the Finnish Foundation for Technology Promotion. We thank Stefan Blom and Mariëlle Stoelinga 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] A. Hartmanns, M. Timmer, On-the-fly confluence detection for statistical model checking, in: Proceedings of the 5th NASA Formal Methods Symposium (NFM), volume 7871 of *Lecture Notes in Computer Science*, Springer, 2013 (to appear).
- [22] 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.
- [23] C. Baier, J.-P. Katoen, Principles of Model Checking, MIT Press, 2008.
- [24] M. Größer, Reduction Methods for Probabilistic Model Checking, Ph.D. thesis, Technische Universität Dresden, 2008.
- [25] S. Evangelista, C. Pajault, Solving the ignoring problem for partial order reduction, *International Journal on Software Tools for Technology Transfer* 12 (2010) 155–170.
- [26] M. I. A. Stoelinga, Alea jacta est: verification of probabilistic, real-time and parametric systems, Ph.D. thesis, University of Nijmegen, 2002.
- [27] S. Katz, D. Peled, Defining conditional independence using collapses, *Theoretical Computer Science* 101 (1992) 337–359.
- [28] 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.