Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

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, Finland*
[b] *Formal Methods and Tools, Faculty of EEMCS, University of Twente, The Netherlands*

A R T I C L E I N F O

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

A B S T R A C T

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.

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* [1–3] 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

* Corresponding author.
E-mail addresses: henri.hansen@gmail.com (H. Hansen), timmer@cs.utwente.nl (M. Timmer).

0304-3975/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.tcs.2013.07.014

Please cite this article in press as: H. Hansen, M. Timmer, A comparison of confluence and ample sets in probabilistic and non-probabilistic branching time, Theoretical Computer Science (2013), http://dx.doi.org/10.1016/j.tcs.2013.07.014

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 $\mathsf{LTL}_{\setminus X}$ and $\mathsf{CTL}^*_{\setminus X}$ [6–9]. 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, $\mathsf{CTL}^*_{\setminus X}$). There is not as much work on weaker variants of confluence, though in [10] a variant is explored that makes no distinction between visible and invisible actions and does not require acyclicity. The variant preserves deadlocks much in the same way as weaker versions of ample and stubborn sets [8].

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

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

Ample sets and confluent transitions are defined and detected quite differently: ample sets are defined by first giving an independence relation for the action labels, whereas confluence is a property of a set of (invisible) transitions in the final state space. Even so, the underlying ideas are similar on the intuitive level. 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 $\mathsf{PCTL}^*_{\setminus X}$. 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

Please cite this article in press as: H. Hansen, M. Timmer, A comparison of confluence and ample sets in probabilistic and non-probabilistic branching time, Theoretical Computer Science (2013), http://dx.doi.org/10.1016/j.tcs.2013.07.014