Information and Computation 249 (2016) 138–159

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

A space-efficient simulation algorithm on probabilistic automata

Lijun Zhang [a,*], David N. Jansen [b]

[a] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[b] Radboud Universiteit, Model-Based System Development, Nijmegen, The Netherlands

ARTICLE INFO

Article history:
Received 19 December 2013
Received in revised form 31 August 2015
Available online 6 May 2016

Keywords:
Probabilistic automata
Simulation
Partition refinement
Decision algorithm

ABSTRACT

In the context of probabilistic automata, time-efficient algorithms for simulation have been proposed lately. The space complexity thereof is quadratic in the size of the transition relation, thus space requirements often become the practical bottleneck. In this paper, we propose a space-efficient algorithm for computing simulation based on partition refinement. Experimental evidence is given showing that not only the space efficiency is improved drastically: The experiments often require orders of magnitude less time. In practice, they are even faster than the (asymptotically) optimal algorithm by Crafa and Ranzato (2012).

1. Introduction

Probabilistic automata (PAs) [1] are a central model for concurrent systems exhibiting random phenomena. Not uncommon for concurrent system models, their verification often faces state space explosion problems. Simulation on PAs has been introduced in [2] to compare the stepwise behaviour of states in probabilistic automata, and has received much attention, see for example [3–7]. As in the non-probabilistic setting [8], bisimulation equivalence and simulation preorder are particularly important in compositional verification and model checking of probabilistic concurrent systems.

Bisimulation relations are equivalence relations, whereas simulation relations are preorders. Algorithms for computing bisimulation equivalences have been studied in [9,10] for the non-probabilistic setting. They solve the coarsest partition problem (CPP), to find the coarsest partition that does not contain a splitter, i.e. a block whose set of predecessors is not a union of partition blocks. The standard approach is partition refinement: One starts with an initial partition (a disjoint cover) and refines it as long as it contains splitters. Partition refinement has been extended to Markov chains (PAs without non-deterministic choices) in [11,12]. Because of the probabilistic choices, the splitting of predecessor blocks now depends on the transition probability to the splitter. Further, this framework is extended to compute bisimulations for PAs in [3], where both actions and transition probabilities need to be considered.

The framework is not sufficient to handle simulation relations in PAs, as simulations relations are preorders. In the non-probabilistic setting, a refined notion of *partition pair* is introduced for this purpose: in addition to a partition, one also records a relation (typically a partial order) over the blocks in the partition. In this way, simulation relations can be characterised by partition pairs. Then, the simulation problem can be reduced to a generalised coarsest partition problem (GCPP), which consists in determining the coarsest stable partition pair.

* Corresponding author.
E-mail addresses: zhanglj@ios.ac.cn (L. Zhang), dnjansen@cs.ru.nl (D.N. Jansen).

http://dx.doi.org/10.1016/j.ic.2016.04.002
0890-5401/

Table 1
Complexities of non-probabilistic simulation preorder decision algorithms.

Article		Time complexity	Space complexity
Henzinger et al.	1995 [19]	$\mathcal{O}(mn)$	$\mathcal{O}(n^2)$
Bustan/Grumberg	2000 [23]	$\mathcal{O}(n^2 n_\diamond^2(n_\diamond^2+m))$	$\mathcal{O}(n_\diamond^2+n\log n_\diamond)$
Tan/Cleaveland	2001 [22]	$\mathcal{O}(m\log n+mn_\sim)$	$\mathcal{O}(m+n_\sim^2)$
Gentilini et al.	2003 [13][a]	$\mathcal{O}(mn_\diamond^2)$	$\mathcal{O}(n_\diamond^2+n\log n_\diamond)$
Ranzato/Tapparo	2007 [25]	$\mathcal{O}(mn_\diamond)$	$\mathcal{O}(nn_\diamond)$

[a] Corrected by Glabbeek/Ploeger 2008 [24].

In this paper, we first discuss the smallest quotient automaton induced by a simulation preorder for PAs. Then, we discuss how to incorporate the partition refinement scheme into the algorithm for deciding the simulation preorder for PAs. As in the non-probabilistic setting, we show first that simulation relations can be characterised by partition pairs, thus the problem can be reduced to GCPP as well. In PAs, states have in general non-trivial distributions instead of single states as successors. For the refinement step, we propose a graph-based analysis to refine the partition for PAs. As in [13], the relation over the partition is refined according to stability conditions. We arrive at an algorithm with space complexity $\mathcal{O}(n_\sim^2+n\log n_\diamond)$, where n is the number of states, and $n_\diamond$ and $n_\sim$ denote the number of simulation and bisimulation equivalence classes, respectively. We get, however, a rather excessive time complexity of $\mathcal{O}(mn_\diamond+m_\diamond^2 n_\diamond^4+m_\sim^2 n_\diamond^2)$, where m is the number of transitions, and $m_\diamond$ and $m_\sim$ denote the number of transitions in the simulation and bisimulation quotients, respectively. Similarly to algorithms for deciding simulation preorder for PAs [14], one can use parametric maximum flow techniques to improve the time complexity. However, more memory is then needed due to the storage of the networks and the maximum flow values of the corresponding networks across iterations. The combination of our first algorithm with parametric maximum flow techniques uses space $\mathcal{O}(m_\diamond^2+n\log n_\diamond)$ and time $\mathcal{O}(mn_\diamond+m_\sim^2 n_\diamond^2)$.

We have implemented both the space-efficient and time-efficient variants of the partition-refinement-based algorithm. Experimental results show that the space-efficient algorithm is very effective in memory, but also in time: Comparing to the original algorithm, not only the space-efficiency is improved drastically, but also often orders of magnitude less time are required. As in [15], both regular and random experiments show that the parametric maximum flow-based implementation does not perform better in general.

This paper is organised as follows. We present related work in Section 2. After introducing some notations in Section 3 and recalling some definitions in Section 4, we show in Section 5 that every probabilistic automaton has a smallest quotient automaton, and this quotient automaton can be obtained by simulation equivalence. In Section 6, we show that simulation relations can also be characterised by partition pairs. Using this, we develop a partition-refinement-based algorithm for computing the simulation preorder in Section 7. Finally, we report experimental results in Section 8 and conclude the paper in Section 9.

A preliminary version of the paper appeared in [16].

2. Related work

In the non-probabilistic setting, the decision algorithm for the simulation preorder has been studied extensively [17,18]. The efficient algorithm in [19] had a time complexity $\mathcal{O}(mn)$, and a space complexity $\mathcal{O}(n^2)$ due to the need of saving the simulation relations.[1] Since space could become the bottleneck in many applications [21], more space-efficient algorithms have been introduced. Tan and Cleaveland [22] combined the techniques in [19] with the bisimulation minimisation algorithm [9] and achieved a better time complexity $\mathcal{O}(m\log n+mn_\sim)$, where $n_\sim$ denotes the number of bisimulation equivalence classes. The corresponding space complexity is $\mathcal{O}(m+n_\sim^2)$. One characterises a simulation relation by a *partition pair*, i.e. a partition of the states and a partial order between the partition blocks. This needs space in $\mathcal{O}(n_\diamond^2+n\log n_\diamond)$: $\mathcal{O}(n\log n_\diamond)$ for the partition, by indicating to which class each state belongs, and $\mathcal{O}(n_\diamond^2)$ for the partial order. Then, the decision algorithm for deciding simulations can be reduced to a generalised coarsest partition problem (GCPP), which consists in determining the coarsest stable partition pair. The optimal space complexity $\mathcal{O}(n_\diamond^2+n\log n_\diamond)$ is achieved by an algorithm originally proposed by Bustan and Grumberg [23] and accelerated by [13,24]. A good compromise between time- and space-efficiency is provided by [25]. We summarise the mentioned algorithms in Table 1.

In the probabilistic setting, Baier et al. [3] introduced a polynomial decision algorithm for simulation preorder with time complexity $\mathcal{O}((mn^6+m^2n^3)/\log n)$ and space complexity $\mathcal{O}(m^2)$, by tailoring a network flow algorithm to the problem. Drastic improvements are possible by observing that the networks on which the maximum flows are calculated are very similar across iterations of the refinement loop [26,27,14]. Table 2 lists the complexities.

Finally, in recent work by Komuravelli et al. [6,7], the simulation checking problem for compositional systems is broken into subproblems, and a plain simulation algorithm is applied to verify simulation relations compositionally.

[1] To keep the space complexities comparable, we cite others' figures and give our own figures under the assumption that the memory required to store an identification of a state is constant. In a more detailed model, $\mathcal{O}(\log n)$ bits would be required to store such an identification, see for example [20].