Information Processing Letters 112 (2012) 320–328

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Model checking probabilistic systems against pushdown specifications

Clemens Dubslaff *, Christel Baier, Manuela Berg

Institut für Theoretische Informatik, Technische Universität Dresden, Dresden, Germany

ARTICLE INFO

Article history:
Received 21 August 2011
Received in revised form 5 December 2011
Accepted 19 January 2012
Available online 24 January 2012
Communicated by A. Muscholl

Keywords:
Formal methods
Probabilistic model checking
Pushdown systems
Context-free specifications
Markov chains

ABSTRACT

Model checking is a fully automatic verification technique traditionally used to verify finite-state systems against regular specifications. Although regular specifications have been proven to be feasible in practice, many desirable specifications are non-regular. For instance, requirements which involve counting cannot be formalized by regular specifications but using pushdown specifications, i.e., context-free properties represented by pushdown automata. Research on model-checking techniques for pushdown specifications is, however, rare and limited to the verification of non-probabilistic systems.

In this paper, we address the probabilistic model-checking problem for systems modeled by discrete-time Markov chains and specifications that are provided by deterministic pushdown automata over infinite words. We first consider finite-state Markov chains and show that the quantitative and qualitative model-checking problem is solvable via a product construction and techniques that are known for the verification of probabilistic pushdown automata. Then, we consider recursive systems modeled by probabilistic pushdown automata with an infinite-state Markov chain semantics. We first show that imposing appropriate compatibility (visibility) restrictions on the synchronizations between the pushdown automaton for the system and the specification, decidability of the probabilistic model-checking problem can be established. Finally we prove that slightly departing from this compatibility assumption leads to the undecidability of the probabilistic model-checking problem, even for qualitative properties specified by deterministic context-free specifications.

1. Introduction

In the traditional model-checking approach (see, e.g., [8,5]), an operational system model is verified against a formal specification provided by some propositional temporal formula. Formulae of linear temporal logic (LTL) impose *regular* constraints on the runs of the system. The most prominent LTL model-checking approach relies on a representation of LTL formulae by finite-state automata [22].

Many relevant safety and liveness properties are indeed regular, but there are also interesting non-regular system properties. Examples are properties that involve some kind of counting, such as the property stating that in each finite run of the system, the number of request actions is greater or equal than the number of acknowledgements. Another example are pre-/post conditions for recursive programs stating that whenever condition a holds in the moment where some recursive procedure P is invoked, then condition b holds when returning from procedure P. More examples can be found in [15] and [4]. Research on model-checking techniques for non-regular specifications is, however, rare.

Verification techniques for pushdown specifications, i.e., context-free specification provided by some pushdown automaton, have been first addressed in [19] and were significantly enhanced by Kupferman, Piterman and Vardi in [15]. In the latter paper, a model-checking algorithm for finite-state systems and non-deterministic pushdown tree

* Corresponding author.
E-mail addresses: dubslaff@tcs.inf.tu-dresden.de (C. Dubslaff), baier@tcs.inf.tu-dresden.de (C. Baier), manu@tcs.inf.tu-dresden.de (M. Berg).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2012.01.006

automata with parity acceptance conditions is provided that runs in time exponentially in the size of the system and specification. For procedural systems modeled by pushdown automata, the model-checking problem against non-deterministic pushdown tree automata with parity acceptance conditions becomes undecidable. Decidability results can though be obtained by imposing some syntactic restrictions on the pushdown automata formalizing the systems and the specifications. *Visibly pushdown automata* (*VPAs*) were first introduced in [4]. Although more expressive than finite automata, the class of languages described by VPAs enjoys the same closure properties as regular languages. A convenient formalism to describe VPA specifications is the temporal logic CaRet that extends LTL with modalities for calls and returns [3].

The purpose of this paper is to study verification of probabilistic systems with an operational semantics based on discrete-time Markov chains against pushdown specifications. Typical examples for such systems contain unreliable components that behave faulty with some small probabilities (e.g., channels that might lose messages) or rely on a protocol with randomized actions (e.g., tossing a coin to break symmetry or to generate input samples or fingerprints) [13,17]. In the linear-time setting, the task of the *quantitative model-checking problem* for a Markov chain $\mathcal{M}$ is to compute the probability that a given measurable linear-time property φ holds. When also taking a probability interval $I \subseteq [0,1]$ as input, the quantitative model-checking problem can be rephrased as decision problem asking whether the probability measure of the set of runs in $\mathcal{M}$ satisfying φ belongs to I. The *qualitative model-checking problem* for Markov chains and linear-time properties asks whether the given property holds almost surely, i.e., with probability 1. For specifications provided by LTL formulae and finite-state Markov chains, the model-checking problem is well understood. Both the qualitative and quantitative version are solvable using an automata-based approach [21,5] or an incremental approach that modifies the given Markov chain by adding components for the subformulae iteratively [10].

For systems with an infinite-state Markov chain semantics and regular specifications, solving the model-checking problem is more involved. However, there are systems for which algorithms could be provided. Examples are systems modeled by probabilistic lossy channel systems [1], probabilistic vector addition systems [2] or probabilistic pushdown automata [11,6,23]. For the latter, i.e., probabilistic pushdown automaton (pPDA), and ω-regular specifications (including requirements expressed by LTL formulae), both the qualitative and the quantitative model-checking problem for pPDA and regular specifications turned out to be decidable.

In this paper, we address the "opposite" problem which asks for model-checking algorithms when the system is modeled by a Markov chain and the specification given by a pushdown automaton. To the best of our knowledge, this is the first attempt to study the model-checking problem for probabilistic system models and context-free specifications. The main contribution of this paper is to show the decidability of the qualitative and quantitative model-checking problem for

(1) finite-state Markov chains and context-free linear-time properties specified by deterministic pushdown automata (DPDAs), and
(2) recursive systems modeled by visibly pPDAs and deterministic context-free linear-time properties specified by visibly DPDAs.

In both cases, we consider DPDAs over infinite words (briefly called ω-DPDAs) and deal with Muller and Büchi acceptance conditions. To establish (1), we employ a product construction and provide a reduction to the model-checking problem for systems modeled by pPDA. Although the product construction is standard, some care is needed to treat ε-transitions in the ω-DPDA properly and to ensure that the product meets indeed the syntactic conditions of a pPDA. For (2), we adapt known concepts for verifying systems modeled by visibly PDA against visibly PDA-specifications [4] to the probabilistic setting. This requires an adequate definition of the visibility condition for pPDA and also relies on a reduction to the model-checking problem for pPDAs and regular specifications.

Given that two-stack automata have Turing power, one cannot expect a general decidability result of the model-checking problem where both the system and the requirement are modeled by some kind of PDAs. In fact, we show that if the visibility conditions of the pPDA for the system and the DPDA for the requirements are not compatible, even the qualitative model-checking problem becomes undecidable.

Outline. The remainder of the paper is organized as follows. Section 2 summarizes some basic concepts on ω-automata, Markov chains and pushdown systems and introduces the notations used throughout the paper. In Section 3, we present our results on model checking finite-state Markov chains against ω-DPDAs. The model-checking problem for systems modeled by pPDAs and ω-DPDA-specifications is presented in Section 4. The paper ends with some concluding remarks in Section 5.

2. Preliminaries

We denote the set of finite (respectively, infinite) words over an alphabet Σ by Σ^* (respectively, Σ^ω). If $\sigma \in \Sigma^* \cup \Sigma^\omega$, then σ_i denotes the $(i{+}1)$th symbol of σ and $|\sigma|$ its length. If $U_1, \ldots, U_k$ are sets then $\cdot|_{U_i} : U_1 \times \cdots \times U_k \to U_i$ denotes the projection function given by $u|_{U_i} = u_i$ for each tuple $u = (u_1, \ldots, u_k) \in U_1 \times \cdots \times U_k$. For sequences $\pi = \pi_0\ \pi_1 \ldots$ of tuples $\pi_j \in U_1 \times \cdots \times U_k$ the projection function is applied elementwise, i.e., $\pi|_{U_i} = \pi_0|_{U_i}\ \pi_1|_{U_i} \ldots$.

2.1. ω-Automata

We briefly summarize our notations for finite-state and pushdown automata over infinite words. For further details we refer to [20]. A *non-deterministic Büchi automaton* (*NBA*) is a tuple $\mathcal{B} = (Q, Q_0, \Sigma, \delta, Acc)$, where Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, Σ is an input alphabet, $\delta : Q \times \Sigma \to 2^Q$ is the transition function, and $Acc \subseteq Q$ is the acceptance condition of $\mathcal{B}$. The size