Simulation Is Decidable for One-Counter Nets
(Extended Abstract)

Parosh Aziz Abdulla\(^1\) and Karlis Ėrāns\(^2\)

\(^1\) Dept. of Computer Systems, P.O. Box 325, S-751 05 Uppsala, Sweden, parosh@docs.uu.se
\(^2\) Institute of Mathematics and Computer Science, University of Latvia, karlis@cclu.lv

Abstract. We prove that the simulation preorder is decidable for the class of one-counter nets. A one-counter net consists of a finite-state machine operating on a variable (counter) which ranges over the natural numbers. Each transition can increase or decrease the value of the counter. A transition may not be performed if this implies that the value of the counter becomes negative. The class of one-counter nets is computationally equivalent to the class of Petri nets with one unbounded place, and to the class of pushdown automata where the stack alphabet is restricted to one symbol. To our knowledge, this is the first result in the literature which gives a positive answer to the decidability of simulation preorder between pairs of processes in a class whose elements are neither finite-state nor allow finite partitioning of their state spaces.

1 Introduction

Several criteria for comparison of program behaviour have been introduced in the literature during the past few years. Examples are the notions of bisimulation equivalence and simulation preorder which are refinements of trace equivalence and trace inclusion in language theory. One area of interest has been to design algorithms to verify such properties for infinite-state systems. Classes of infinite-state systems to which a considerable research effort has been devoted are those of pushdown processes, context-free processes and Petri nets. The decidability of bisimulation has been shown for different variants of context-free processes [BBK93,CHS92,CHM93]. In [GH94] the simulation preorder is shown to be undecidable for context-free processes. In [Sti96] bisimulation was shown to be decidable for normed pushdown processes. Simulation and bisimulation properties have also been studied for Petri nets [Jan95,JE96,Esp95,JM95,Jan97]. In [Jan95], both simulation and bisimulation were shown to be undecidable for Petri nets, assuming that the nets have at least two unbounded places. This result left open the problem of whether simulation and bisimulation are decidable for Petri nets with one unbounded place. This class of Petri nets is interesting also because it is equivalent to the class of pushdown automata where the stack alphabet is restricted to one symbol. In this paper, we show that the simulation preorder is decidable in this case. Jančar [Jan97] shows that bisimulation is decidable for nets with one unbounded place. Thus, our results contribute
to delineating the border between decidable and undecidable problems in the context of infinite-state systems.

We consider the problem of deciding simulation for a class of infinite-state systems which we call one-counter nets. A one-counter net consists of a finite-state machine operating on a variable (counter) which ranges over the natural numbers. Each transition can increase or decrease the value of the counter. A transition may not be performed if this implies that the value of the counter becomes negative. A configuration $\gamma$ of a net is a pair $(q, x)$ where $q$ is a state of the finite-state machine and $x$ is the value of the counter. We show that given nets $N_1$ and $N_2$, and configurations $\gamma_1$ and $\gamma_2$ in $N_1$ and $N_2$, it is decidable whether $\gamma_1$ is simulated by $\gamma_2$. The class of one-counter nets is computationally equivalent to the class of Petri nets with one unbounded place, and the class of pushdown automata where the stack alphabet is restricted to one symbol\(^1\).

To perform our decidability proof we notice that the simulation relation is upward closed with respect to the counter of $N_2$: if $(q, x)$ is simulated by $(r, y)$ and $y \leq y'$ then $(q, x)$ is simulated\(^2\) by $(r, y')$. In fact our decidability proof consists of two steps. The first step which is the crucial step and the most difficult one shows that the simulation relation satisfies a much stronger property than upward closedness, namely the following. Given nets $N_1$ and $N_2$ and states $q$ and $r$ in $N_1$ and $N_2$ respectively, there is a rational number $\rho$ (which we call the quality of $(q,r)$) which defines the border between simulation and non-simulation modulo some constant $c$. More precisely, we prove that $(q, x)$ is simulated by $(r, y)$ if $\rho \cdot x + c < y$ and $(q, x)$ is not simulated by $(r, y)$ if $\rho \cdot x - c \geq y$.

The main idea in proving the existence of qualities is to "decompose" the nets into a finite number of deterministic subnets. The quality of a pair $(q,r)$ is first computed in each of these deterministic subnets. We take the quality of $(q,r)$ to be the minimum of its qualities in the deterministic subnets. To prove the correctness of our definition of qualities, we describe the simulation relation as a game between two players $A$ (representing $N_1$) and $B$ (representing $N_2$). Different game-theoretic descriptions of simulation and bisimulation relations have been given e.g. in [Sti95,JM95]. We design a number of (quite complicated) strategies for the players $A$ and $B$, which we use to show that qualities indeed satisfy the above property (i.e., they define borders separating counter values which give simulation and non-simulation respectively). In the second (simpler) step of the proof, we show that the existence of qualities implies that each pair of configurations which is related by simulation has a semi-linear witness (a semi-linear simulation relation containing the pair) which can be found and checked in a finite amount of time.

In [Jan97] it is shown that also bisimulation can be decided by providing semi-linear witnesses. In fact [Jan97] considers one-counter machines which are more general than one-counter nets, in the sense that they allow for zero-testing\(^1\).

\(^1\) The fact that the stack alphabet contains only one symbol implies implicitly that emptiness of the stack cannot be checked (corresponding to the fact that we do not perform zero-testing of the counter value). In most models of push-down automata, stack emptiness is implemented by having at least two symbols in the stack alphabet, using one symbol to represent the bottom of the stack.

\(^2\) However, the simulation relation is obviously not upward closed with respect to the counter of $N_1$. Hence, the relation cannot be characterized by ideals (used e.g. in [AJ93]).