Model Checking of Persistent Petri Nets

Eike Best and Javier Esparza
Institut für Informatik
Universität Hildesheim
Marienburger Platz 22
W-3200 Hildesheim, Germany
e-mail: {E.Best,esparza}@informatik.uni-hildesheim.de

Abstract

In this paper we develop a model checking algorithm which is fast in the size of the system. The class of system models we consider are safe persistent Petri nets; the logic is $\mathcal{S}_4$, i.e. propositional logic with a ‘some time’ operator. Our algorithm does not require to construct any transition system: We reduce the model checking problem to the problem of computing certain Parikh vectors, and we show that for the class of safe marked graphs these vectors can be computed – from the structure of the Petri net – in polynomial time in the size of the system.

1 Introduction

Model checking - the algorithmic determination of truth or falsehood of a modal or temporal logic formula, given a model - faces, when applied to concurrent systems, the state explosion problem: the size of the transition system (when finite) can at best be assumed exponential in the size of the underlying system. Therefore, in order to be able to verify properties of non-toy systems, the algorithms have to be able to accept as input graphs containing millions of nodes.

Much work has been done on how to palliate this problem, following two approaches. The first is to improve the efficiency of existing general algorithms: explicit knowledge about concurrency can be used in order to obtain condensed transition systems [9,11,18]. These techniques have the advantage of being generally applicable; however, it is very difficult to know a priori if they will be really effective.

The second approach attempts to take advantage of special properties of the underlying model in order to speed up the model checking algorithm. This could lead to efficient, albeit special purpose methods. Two examples of this line of work are [5,15]. However, these papers also require the construction of transition systems. In this contribution, we are more radical: we investigate the possibility of obtaining model checkers which do not require at all to construct the associated transition system, i.e. model checkers that work directly on the syntax. We consider the modal logic $\mathcal{S}_4$

1Partly supported by the Esprit Basic Research Action 3148 DEMON
[10] tailored for safe Petri nets (our syntax), and concentrate on a particular subclass of models, namely safe persistent Petri nets [13]. The logic can express properties such as reachability of a marking, liveness of a transition or mutual exclusion of a set of transitions. It also allows some "counting": properties such as "in order to reach a marking in which place s has one token, transition t has to occur 4 times" can be expressed as well.

(Safe) persistent nets are being currently used to model self-timed circuits [17]. In general, concurrent but deterministic systems (applications appear mainly in hardware design) can be modelled using persistent nets.

Given a safe persistent system $\Sigma$ and a formula $\phi$, we show how to reduce the model checking problem to a set of Linear Programming problems. For the subclass of safe $T$-systems, we prove that the model checker is polynomial in the size of $\Sigma$, although exponential in the length of $\phi$. Since formulae are usually short, while systems can be very large, this is a very satisfactory result; moreover, as shown in the paper, a model checker polynomial in both the size of $\Sigma$ and the length of $\phi$ can exist only if $P = NP$.

The paper is organised as follows. Section 2 introduces the logic. Section 3 discusses briefly the model checking problem. Section 4 introduces the models: persistent and strongly persistent systems. Section 5 presents some results on net processes; in particular, that strongly persistent systems have one single maximal process. The main theorem for the construction of the model checker is proved in Section 6. The model checker itself is described in Section 7. The particular case of $T$-systems is studied in Section 8. Some basic definitions are contained in an Appendix, although reading this paper is easier if the reader is familiar with the basic notions of Petri nets (otherwise, see [16]).

2 A Modal Logic for Safe Petri Nets

We define a simple modal logic over computations (more precisely occurrence sequences) of safe marked Petri nets, with the following basic propositions:

- Assertions of the form $s$, to be used with a model containing a place named $s$. The intended meaning is 'after the present computation, a token is on $s$'.

- Assertions of the form $t \leq 4$, to be used with a model containing a transition named $t$. The intended meaning is 'in the present computation $t$ occurs no more than 4 times'.

Our logic is $S_4$ [10], i.e., propositional logic augmented with the modal operator $\Diamond$, meaning 'it is possible that ... '.