Verification of parametric concurrent systems with prioritised FIFO resource management

Ahmed Bouajjani · Peter Habermehl · Tomáš Vojnar

Published online: 25 January 2008
© Springer Science+Business Media, LLC 2008

Abstract We consider the problem of parametric verification over a class of systems of processes competing for access to shared resources. We suppose the access to the resources to be controlled according to a FIFO-based policy with a possibility of distinguishing low-priority and high-priority resource requests. We propose a model of the concerned systems based on extended automata with queues. Over this model, we address verification of properties expressed in LTL\,X enriched with global process quantification and interpreted on finite as well as fair behaviours of the given systems. In addition, we examine parametric verification of process deadlockability too. By reducing the parametric verification problems to finite-state model checking, we establish several decidability results for different classes of the considered properties and systems (including the special case of systems with the pure FIFO resource management). Furthermore, we show that parametric verification against formulae with local process quantification is undecidable in the given context.

Keywords Formal verification · Parameterised verification · Infinite-state system verification · Cut off · Model checking · Parameterised networks of processes · Resource sharing

This work was supported in part by the European Commission (FET project ADVANCE, contract No. IST-1999-29082), the French government by the project ACI Sécurité Informatique, the Czech Grant Agency (project 102/07/0322), the Czech-French Barrande project 2-06-27, and the Czech Ministry of Education within the project MSM 0021630528 “Security-Oriented Research in Information Technology”.

A. Bouajjani · P. Habermehl
LIAFA, University Paris 7—Denis Diderot/CNRS, Case 7014, 2, place Jussieu, 75251 Paris Cedex 05, France
A. Bouajjani
e-mail: Ahmed.Bouajjani@liafa.jussieu.fr
P. Habermehl
e-mail: Peter.Habermehl@liafa.jussieu.fr

T. Vojnar (✉)
FIT, Brno University of Technology, Božetřeščova 2, 61266 Brno, Czech Republic
e-mail: vojnar@fit.vutbr.cz
1 Introduction

Managing concurrent access to shared resources is a fundamental problem that appears in many contexts, e.g., operating systems, multithreaded programs, control software, etc. The critical properties to ensure are typically (1) mutual exclusion when exclusive access is required, (2) absence of starvation (a process that requires a resource will eventually get it), and (3) absence of deadlocks.

Many different instances of the above problem can be defined depending on the assumptions on the allowed actions for access to resources and the policies for managing the access to these resources.

In this work, we consider systems with a finite number of resources shared by a set of identical processes. These processes can request a set of resources, get access and use the requested resources, and release the used resources. The requests can be of a low-priority or a high-priority level. The access to the resources is managed by a locker according to a FIFO-based policy taking into account the priorities of the requests—i.e. a waiting high-priority request can overtake waiting low-priority ones. As a special case allowing for an optimised treatment, we then examine the situation when no high-priority requests are used, and the locker behaves according to the pure FIFO discipline.

As mentioned later in related work, the above framework is, in particular, inspired by a need to verify the use of shared resources in some of Ericsson’s ATM switches. However, the operations for access to shared resources and the resource management policies used are quite natural in general in concurrent applications dealing with shared resources.

Verification of the described systems can, of course, be carried out using finite-state model-checking if we fix the number of processes. However, a precise number of processes present in such a system in practice is usually not known in advance, and it is thus crucial to verify that the system behaves correctly for any number of them. This yields a nontrivial parametric verification problem as we have to deal with an infinite number of system instances.

The aim of this paper is to study decidability of the described problem for a significant class of properties including the three most important ones given above.

For an abstract description of the concerned systems, we define a model based on extended automata with queues recording the identities of the waiting processes for each resource. Then, we address the verification problem for families of such systems with an arbitrary number of processes (called RTR families—RTR stands for request-take-release) against formulae of the temporal logic $\text{LTL}\backslash X$ with global process quantification. We consider two interpretation domains for the logic: the set of finite behaviours (which is natural for safety properties), and the set of fair behaviours (in order to cover liveness properties). In addition, we consider the parametric verification problem of process deadlockability too.

We adopt the approach of finding cut-off bounds to show that many interesting parametric verification problems in the given context can be reduced to finite-state model checking. This means that given a class of formulae, we prove that deciding whether all systems of a family satisfy a formula is equivalent to deciding whether some finite number of systems in the family (each of them having a fixed number of processes) satisfies this formula.

When establishing our results, we consider the question whether it is possible to find cut-off bounds that do not depend on the structure of the involved processes and the formula at hand, but only on the number of resources and the number of processes quantified in the formula. Indeed, these numbers are relatively small, especially in comparison to the size of process control automata.

We show that for RTR families where the pure FIFO resource management is used (i.e. no high-priority access to resources is required), parametric verification of finite as well as