Journal of Computer and System Sciences 80 (2014) 1298–1322

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Verifying of interface assertions for infinite state Mealy machines

Manfred Broy

Institut für Informatik, Technische Universität München, D-80290 München, Germany

ARTICLE INFO

Article history:
Received 18 August 2012
Received in revised form 12 February 2014
Accepted 3 March 2014
Available online 17 March 2014

Keywords:
Interactive Systems
I/O-machines
Specification
Verification
Invariants

ABSTRACT

This paper aims at techniques and methods for the verification of logical assertions about the interface behavior of generalized I/O-state machines. The *interface behavior* of such machines is specified by *interface assertions* formulated in predicate logic. Interface assertions specify the interface behavior of state machines in terms of the streams of messages produced via their input and output channels. The verification of interface assertions for state machines is carried out with the help of generalized invariants. Such invariants are proved for state machines in terms of stable assertions. Nontrivial liveness properties such as fairness lead to specifications and also to state machines defining sets of computations that specify sets of output streams that are not limit closed. Elementary examples for such cases are described and the implied complications are analyzed. Furthermore, verification methods are provided for these cases and demonstrated by small examples.

Contents

E-mail address: broy@in.tum.de.
URL: http://wwwbroy.informatik.tu-muenchen.de.

http://dx.doi.org/10.1016/j.jcss.2014.03.002
0022-0000/

1. Introduction and motivation

Invariants are a standard concept for proving properties about imperative programs with while-loops. Originally developed for proving assertions about loops in programs, they have been carried over to state machines. For state machines, invariants are predicates about states that are valid for the set of all states reachable in computations of the machines that start from specified sets of initial states. Notions of state invariants are a useful technique, too, to prove safety properties.

State machines with non-terminating infinite computations show nontrivial liveness properties such as fairness conditions that bring in new challenges for verification. Fairness leads to computations that go even beyond Church's thesis of computability (see, for instance, [6–11,13,14]), since fairness leads to unbounded non-determinism (see also [1] and [2]). This relates also to the well-known lemma (König's lemma) stating that for each tree that is only finitely branching and that contains an infinite number of nodes there exists an infinite path. In König's lemma a tree is a directed acyclic graph where each node besides one node, called the root, has exactly one predecessor. The root has no predecessor and to each node x there exists a unique finite path from the root. So, if we assume that in each ("finite") step of computation only a finite number of choices are possible, since the tree is finitely branching, then to choose between elements from an infinite set (which is – at least implicitly – the case under the condition of fairness) by a sequence of finite choices necessarily includes the existence of a non-terminating path. As is well known, to choose between an infinite set of elements an infinite number of finite choices is necessary. Since fairness, a concept often found in models of concurrent computations, requires to exclude the infinite non-terminating branch, by fairness assumptions the ground of Turing computability is left (for details see [5]).

We study state machines with input and output also known as Mealy machines and Moore machines. These are machines, where each state transition in a given state is triggered by some input and generates by the transition a new state and some output. This way we define computations of state machines with input and output, which characterize on infinite sequence of states and histories of input and output. Given an initial state and an infinite stream of input messages machines with input and output generated infinite sequences of states and infinite streams of output messages. In the so-called *interface abstraction* we hide the chain of states in computations and only observe their pairs of input and output streams which we call *interface histories*. Talking only about interface properties we can formulate so-called *interface assertions*, which describe logical properties of the interface histories. They specify the interface behavior of state machines with input and output.

In contrast to techniques used for the machines with input and output in [18], where fairness is achieved by an explicit concept to specify fair computations or by Büchi-automata, that are in particular used to express fairness, we work with state machines with infinite state spaces and state transitions with infinite choices, which allow expressing fairness.

In the following, we study techniques for proving interface assertions about state machines with infinite state spaces. We show how to prove properties by generalized techniques based on stable interface assertions and invariants. Actually, we use assertions and invariant techniques to specify and to prove both safety and liveness properties for sets of interface histories associated with the computations of state machines with input and output. Safety properties of sets of histories are properties that can be specified by considering only all finite prefixes of histories. Liveness properties are properties that in contrast cannot be expressed in terms of finite prefixes but only for the infinite histories. We study, in particular, state machines with liveness properties such as fairness assumptions. Such liveness properties are more difficult to deal with