Deciding Properties of Integral Relational Automata*

(Extended Abstract)

Kārlis Čerāns**
Institute of Mathematics and Computer Science
University of Latvia

Summary. This paper investigates automated model checking possibilities for CTL* formulae over infinite transition systems represented by relational automata (RA). The general model checking problem for CTL* formulae over RA is shown undecidable, the undecidability being observed already on the class of Restricted CTL formulae. The decidability result, however, is obtained for another substantial subset of the logic, called A-CTL*+, which includes all "linear time" formulae.

1 Introduction

The area of automated deciding of properties of infinite state systems has during the recent years received quite a number of impressive positive developments. For instance, in the theory of Timed Automata (TA) the decidability has been established for the reachability and infinite behaviour possibility problems, as well as other TA properties expressible in temporal logic TCTL [3, 4]. Also bisimulation equivalence for TA has been shown decidable [10]. In [1] it was showed that safety properties (and liveness without fairness constraints) are decidable for systems of automata communicating over unbounded lossy channels. The decidability of state (marking) reachability for Petri Nets is due already back to [19, 20], this result has been extended to decidability of linear time temporal logic formulae over PN's in [17]. The bisimulation equivalence has been proved decidable for various subclasses of Petri Nets in [13] (for the full class of Petri Nets the bisimulation equivalence is undecidable [18]). One should continue this list with decidability results on context-free processes ([8], [14]), and probably many others.

In this paper we consider deciding properties of infinite transition systems which are represented by so-called Relational Automata (RA). Each RA, besides its finite control structure, has a finite number of data variables, which can assume values from certain ordered domain, in this paper taken to be the integral numbers (with the ordering relation "<"). The operation repertoire of RA includes comparison of variable values on their ordering, as well as input/output and dummy operations.

RA were originally introduced in [6, 7] where they naturally emerged as a model for studying possibilities of automated complete test set generation for data processing programs. It seems that RA could be also applicable in modelling and analyzing real time programs for data stream processing (when mutual relations of data values can take part in determining the control of processing). When suitably combined with data abstraction facilities, RA can model phenomena like fairness³. Still, in this paper we do not focus on the possible applications of RA, but allow them to speak mostly on their own.

* Part of this work was performed while the author was visiting Department of Computing Science, Chalmers University of Technology, Göteborg, Sweden.
** Email: karlis@mii.lu.lv
³ This modelling makes heavy use of the integrality of the allowed variable value domain.
Our primary concern in this paper is to investigate the decidability of various verification problems over RA. In particular, we shall consider the possibilities of finding general procedures, which given an instance of RA, say P, and a temporal logic formula $\xi$ decides whether $\xi$ “holds” on P (i.e., whether P is a model for $\xi$). This problem in literature is often called “model checking” (due to [15]?). The logic used for describing properties of RA will be CTL* [15, 16], it contains modalities for specifying both the properties of linear behaviours of automata (“in next state”, “until”, “always”), and their transition system branching structure (quantifiers “for all (for some) computation paths”). The logic CTL* has showed its importance in describing and deciding properties of finite state systems.

The decidability of vertex reachability problem for RA has been shown already in [6, 7], see [5] for survey on this and related results. The problem of infinite behaviour set emptiness for RA (also admitting fairness constraints) has been shown decidable in [9]. The main contribution of this paper consists in showing (i) undecidability of the full CTL* model checking problem over RA (obtained already when considering the “purely branching” CTL* sublogic CTL), and (ii) designing a large subclass of CTL* for which a general deciding procedure exists (this subclass includes all A-CTL* formulae, and, so, also all “linear time” formulae)\(^4\). The general CTL* model checking undecidability result should be contrasted with the decidability of the similar model checking problem over the class of RA modified by allowing their variables to assume arbitrary rational values [11].

Partial preliminary reports on this work have appeared as [11] and [12], what contain some of the proofs only briefly outlined here. A full version of this paper is in preparation.

2 Relational Automata: The Basics

An (integral) relational automaton is a program with a finite number (say, $k$) of integral valued variables and the following allowed operations over them:

- $?x$ - input of a (new) integral value into the variable $x$,
- !x (!c) - output of the value of the variable $x$ (the constant $c$),
- $x < y$ ($x < c$, $c < y$) - a comparison of the values of two variables (the value of a variable with a constant), the operator produces an output flag "+" or "−" used to determine possible further control flow of the automaton,
- $x \leftarrow y$ ($x \leftarrow c$) - assignment of the value of the variable $y$ (the constant $c$) to the variable $x$,
- NOP - the dummy operation.

Two simple examples of relational automata can be found in Figure 1. The first automaton in every execution loop after retrieving values into the variables $x$ and $y$ compares them and places the largest one into the variable (memory cell) $z$. After the assignment, the computation in every branch is leading to a vertex with a NOP operator in it, which can be thought of as being some operator not reflected in the defined language (e.g. it could do some computations with the value of $z$). The second automaton illustrates the “fairness modelling” idea in RA: every infinite computation of it will output a value infinitely often.

\(^4\) The presented decidability results rely on such mathematical constructions as simulation relation between states of RA, monotonicity of validity sets of temporal formulae and their automata based representation [24], graphose representation and symbolic manipulation of linear inequality systems, finite minorability of considered inequality system set wrt. certain natural ordering, and others. The obtained undecidability result relies on a standard technique of modelling 2 counter machine behaviours.

\(^5\) In general, one can think of NOP as any operator which is not affecting the values of variables used further in computing the branching predicates.