Information Sciences 313 (2015) 22–42

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Model checking temporal properties of reaction systems

Artur Męski [a,b,*], Wojciech Penczek [a,e], Grzegorz Rozenberg [c,d]

[a] Institute of Computer Science, PAS, Jana Kazimierza 5, 01-248 Warsaw, Poland
[b] University of Łódź, FMCS, Banacha 22, 90-238 Łódź, Poland
[c] LIACS, Leiden University, P.O. Box 9512, 2300 RA, The Netherlands
[d] Department of Computer Science, University of Colorado at Boulder, 430 UCB, Boulder, CO 80309-0430, USA
[e] University of Natural Sciences and Humanities, ICS, Siedlce, Poland

ARTICLE INFO

Article history:
Received 13 November 2014
Received in revised form 12 February 2015
Accepted 22 March 2015
Available online 28 March 2015

Keywords:
Reaction system
Model checking
Temporal logic

ABSTRACT

This paper defines a temporal logic for reaction systems (rsCTL). The logic is interpreted over the models for the context restricted reaction systems that generalise standard reaction systems by controlling context sequences. Moreover, a translation from the context restricted reaction systems into boolean functions is defined in order to be used for a symbolic model checking for rsCTL over these systems. The model checking for rsCTL is proved to be PSPACE-complete. The proposed approach to model checking was implemented and experimentally evaluated using four benchmarks.

1. Introduction

Model checking [9] is a method that allows for checking whether or not the system in question satisfies a given formula specifying either a desired or an undesired property of that system. Typically, the verified properties are expressed using some modal logic formalism. This method is fully automatic and to be used in practice it does not require expert knowledge of verification techniques.

The basic idea behind the original motivation of reaction systems as models of processes inspired by the functioning of the living cell (see, e.g., [5,12,13]) is that this functioning is based on the interactions of individual biochemical reactions. Moreover, these interactions are driven by two mechanisms: facilitation/acceleration and inhibition/retardation.

This paper introduces Computation Tree Logic for reaction systems (rsCTL), which is a logic for specifying properties of reaction systems, together with a method for verifying these properties. This is the first paper providing a verification method for reaction systems.

Because the processes of reaction systems are guided by the context sequences (which model an interaction with the environment), to enable the verification we introduce a generalisation of reaction systems which allows to specify context entities generating all the context sequences for the processes of the given reaction system. Moreover, we describe an encoding of the model for reaction systems into boolean formulae that can be used for the symbolic model checking approach. We also provide some complexity results for the problem of model checking reaction systems.

In this paper we extend the results of our technical report [21] with a theoretical and experimental analysis of the proposed approach to verification of reaction systems. Following the introduction of reaction systems [13], their extensions were also studied: [14] introduced time in reaction systems, [18] dealt with quantum and probabilistic reaction systems,

* Corresponding author at: Institute of Computer Science, PAS, Jana Kazimierza 5, 01-248 Warsaw, Poland. Fax: +48 22 380 05 10.
E-mail address: meski@ipipan.waw.pl (A. Męski).

http://dx.doi.org/10.1016/j.ins.2015.03.048
0020-0255/

and [22] approached reaction systems with an automata semantics. However, none of the proposed extensions dealt with restrictions on the context sequences of reaction systems. A significant part of the research on reaction systems is focused on their mathematical properties. Fixed points, attractors, and cycles for reaction systems were investigated in [15,16]. Occurrence and convergence problems were tackled in [17,28,29]. Different classes and properties of reaction systems related to their state sequences were studied in [26,27]. Another strand of research focuses on applying reaction systems to modelling of systems. In [11] the authors modelled the gene regulation mechanism for lactose operon of Escherichia coli and explored modelling possibilities for several well-known computer science problems. A reaction systems model of the eukaryotic heat shock response, which we use in this paper as a benchmark, was described in [4] together with properties of its expected behaviour. There is also an increasing interest in verification of reaction systems. In [2] the authors define several biologically inspired properties, including a mass conservation property, together with the corresponding verification problems. These properties can be expressed in the logic considered in the technical report [21], in which the logic was called RSTL, and this paper (rsCTL). Some of the problems defined in the paper are of lower complexity than for the model checking of rsCTL. The paper focuses mostly on complexity considerations for decision problems related to verification and does not develop any language for specifying properties of reaction systems. In [3] the authors investigate the property of mass conservation using dependency graphs, and introduce a simulator for reaction systems.

The paper is organised as follows. In Section 2 we recall the basic notions of reaction systems, while in Section 3 we define two generalisations of reaction systems. Two basic illustrative examples are given in Section 4. In Section 5 we define the syntax of the logic for reaction systems together with a model used to define the semantics of the logic, based on which, in Section 6 we define a model checking method for reaction systems and prove its complexity. In Section 7 we present the encoding for the model defined in Section 5 – this encoding can be used for symbolic model checking of reaction systems. In Section 8 we introduce a tool which is an experimental implementation of the model checking method for reaction systems. We use the tool to evaluate our approach on four reaction systems revealing various practical aspects of the tackled verification problems. The last section of the paper provides concluding remarks.

2. Reaction systems

Reaction systems are a formal model for processes instigated by the functioning of living cell. Research topics in this research area are motivated either by biological issues or by a need to understand computations/processes underlying the dynamic behaviour of reaction systems. By now reaction systems became an interesting and novel model of computation.

In this section we recall some basic notions of reaction systems that are used in this paper. First of all, we recall the notion of a reaction.

Definition 2.1. A *reaction* is a triplet $b = (R, I, P)$ such that R, I, P are finite nonempty sets with $R \cap I = \emptyset$.

The sets R, I, P are called the *reactant set of b*, the *inhibitor set of b*, and *the product set of b*, respectively – they are also denoted by P_b, I_b, and P_b, respectively. The requirement that all three sets R, I, and P are nonempty is motivated by biological considerations: there is no creation from nothing ($R \neq \emptyset$), each reaction may be inhibited ($I \neq \emptyset$), and if a reaction takes place then this creates a "material" effect – something is produced ($P \neq \emptyset$).

If $R, I, P \subseteq Z$ for a finite set Z, then we say that b is a *reaction in Z*. We use $rac(Z)$ to denote the set of all reactions in Z.

The above formal notion of a reaction corresponds closely to the basic intuition behind a biochemical reaction. Such a reaction will take place if all of its reactants are present and none of its inhibitors is present, and if it takes place it produces its set of products.

Definition 2.2. Let Z be a finite set, and let $T \subseteq Z$.

1. We say that $b \in rac(Z)$ is *enabled* by T, denoted $en_b(T)$, if $R_b \subseteq T$, and $I_b \cap T = \emptyset$. The *result* of b on T, denoted by $res_b(T)$, is defined by: $res_b(T) = P_b$ if $en_b(T)$, and $res_b(T) = \emptyset$ otherwise.
2. For $B \subseteq rac(Z)$, the result of B on T, denoted by $res_B(T)$, is defined by $res_B(T) = \bigcup\{res_b(T) | b \in B\}$.

The intuition underlying the above definition is that T formalises a state of a biochemical system under consideration: it is simply the set of all biochemical entities present in the given state. A reaction b is enabled by T (can take place at T) if all the reactants of b are present in T and none of the inhibitors of b is present in T. Therefore we require that $R_b \cap I_b = \emptyset$ – in this way we do not consider "trivial reactions", i.e., reaction that are never enabled.

Then the result of a set of reactions B is *cumulative*, i.e., it is the union of results of the individual reactions from B – clearly, $res_B(T) = \bigcup\{res_b(T) | b \in B \text{and} en_b(T)\}$.

We are ready now to define the notion of *reaction system*.

Definition 2.3. A *reaction system*, *rs* for short, is an ordered pair $\mathcal{R} = (S, A)$, where S is a finite set and $A \subseteq rac(S)$.

The set S is the *background set* of $\mathcal{R}$. The elements of S are called entities, each subset of S is called a *state* of $\mathcal{R}$, and A is the *set of reactions from* $\mathcal{R}$.