SMT-based bounded model checking for parametric reaction systems (Short Paper)

Artur Męski[1,2] and Wojciech Penczek[1]

[1] Institute of Computer Science, PAS, Jana Kazimierza 5, 01-248 Warsaw, Poland
{meski,penczek}@ipipan.waw.pl
[2] Vector GB Limited, London, WC2N 4JF, UK

1 Reaction Systems

Reaction systems [5] are a formal model for specifying and analysing computational processes in which reactions operate on sets of entities (molecules), providing a framework for dealing with qualitative aspects of biochemical systems. The model can capture in a very simple way the basic mechanisms underpinning the dynamic behaviour of a living cell. A key feature of reaction systems is that the latter results from the interactions of biochemical reactions based on the mechanisms of facilitation and inhibition, i.e., the products of reactions may facilitate or inhibit each other. The basic model of reaction systems consists of the reactions, states, and dynamic processes using (tuples of) finite sets, and so it directly captures the qualitative aspects of systems. This short article is based on our conference paper [8].

2 Applications and Verification of Reaction Systems

Examples of applications of reaction systems to modelling of systems include, e.g., [3, 4]. Verification of reaction systems was discussed in, e.g., [1, 2, 9]. The papers [7, 11] introduced reaction systems with discrete concentrations of entities and reactions operating on multisets of entities, resulting in a model allowing direct quantitative modelling. Although there exist other approaches that support modelling of complex dependencies of concentration levels and their changes, e.g., chemical reaction networks theory based on [6], reaction systems provide much simpler framework and the processes of reaction systems take into account interactions with the external environment. Discrete concentrations can be simulated in the original qualitative reaction systems, but reaction systems with discrete concentrations provide much more succinct representations in terms of the number of entities being used, and allow for more efficient verification [7]. The properties being verified are expressed in rsLTL which is a version of the standard linear-time temporal logic defined specifically for reaction systems.

3 Main Results for Parametric Reaction Systems

In practical applications, a reaction system may have only partially specified reactions, where reactant, inhibitors, or products might be initially unknown. In such situations, we propose to use parameters in place of the unspecified reaction parts. We then develop a reaction mining approach where the missing details are computed automatically. To develop such an approach, we introduce reaction systems with parameters. The main result [8] is a methodology which attempts to replace these parameters in such a way that the resulting reaction system satisfies a given *rs*LTL *formula* when operating in a given external *environment*. Intuitively, such a formula might correspond to a number of observations (runs) of the behaviour of a partially specified system. Moreover, the environment is specified using a *context automaton* which represents the influence of the bigger system in which the reaction system with discrete concentrations operates. We provide a suitable encoding of parametric reaction systems in SMT, and propose a synthesis procedure based on bounded model checking for solving the synthesis problem. Moreover, we show that the complexity of the non-emptiness problem of the parameter synthesis for rsLTL and parametric reaction systems is PSPACE-complete.

We also provide preliminary successful experimental results demonstrating the scalability of the new synthesis method.

4 Experimental evaluation

In this section we present the results of an experimental evaluation of the described approach. We test our method on a parametric version (PMUTEX) of the reaction system model for the mutual exclusion protocol introduced in [9]. The system consists of $n \geq 2$ processes competing for an exclusive access to the critical section. The background set of the reaction system[3]. modelling the mutual exclusion protocol is defined as $S = \bigcup_{i=1}^{n} S_i$, where the set of background entities corresponding to the i-th process is defined as: $S_i = \{out_i, req_i, in_i, act_i, lock, done, s\}$, where the entities *lock*, *done*, and s are shared amongst all the processes.

We start by defining the context automaton $\mathfrak{A}$. It ensures that initially all the processes are outside of their critical sections and are not requesting access, which is indicated by the presence of out_i for each $i \in \{1, \ldots, n\}$. Next, we assume $\mathfrak{A}$ may supply any subset of entities $C \subseteq \{act_1, ..., act_n\}$ such that $|C| \leq 2$, allowing at most two simultaneously active processes – we assume that if the context contains act_i then it is the i^{th} process' turn to perform an action. The i^{th} process requests access to its critical section by producing req_i. Then, it is possible for the process to enter the critical section when it is allowed to perform an action and the critical section is not locked (the *lock* entity is not present). In the case of entering a critical section, to avoid the situation where two processes enter

[3] For an introduction to reaction systems and the notation used in this paper, we refer the reader to [5]

their critical sections synchronously, the assumption on act_i is stricter: only one act_i for some $i \in \{1, \ldots, n\}$ is allowed to be present for the process to enter the critical section. When a process enters its critical section, the critical section is locked by production of the *lock* entity. The *lock* entity is preserved until the entity *done* appears, which is produced when a process leaves its critical section. Any reaction in the system may be inhibited by the s entity.

Let A_i be the set of reactions of the i^{th} process, for $i \in \{1, \ldots, n\}$. Then, A_i consists of the following reactions:

– $(\{out_i, act_i\}, \{s\}, \{req_i\})$,
– $(\{out_i\}, \{act_i\}, \{out_i\})$,
– $(\{req_i, act_i, act_j\}, \{s\}, \{req_i\})$ for each $j \in \{1, \ldots, n\}$ such that $i \neq j$,
– $(\{req_i\}, \{act_i\}, \{req_i\})$,
– $(\{req_i, act_i\}, \{act_j \mid j \in \{1, \ldots, n\} \text{ and } j \neq i\} \cup \{lock\}, \{in_i, lock\})$,
– $(\{in_i, act_i\}, \{s\}, \{out_i, done\})$,
– $(\{in_i\}, \{act_i\}, \{in_i\})$.

In reaction systems, each reaction is a triple $b = (R, I, P)$, where the sets R, I, and P contain, respectively, the *reactants*, *inhibitors*, and *products* of the reaction b. Intuitively, for a reaction to be enabled, the current state must contain all the reactants and must not contain any of the inhibitors; when a reaction is enabled it produces its products.

Next, we assume here that the system is open and we allow for introducing new processes that participate in the communication to gain access to the critical section. Let us assume we are allowed to modify the behaviour of the additional process (here, the n^{th} process) only by introducing an additional reaction. Such an assumption could be justified by a mechanism that accepts new processes to participate in the protocol only if they contain the reactions of A_i for any $i \in \{1, \ldots, n\}$, while the remaining reactions could be performing some computation outside of the critical section.

Our aim is to violate the property of mutual exclusion by making the first and the n^{th} process enter their critical sections simultaneously. The additional (malicious) reaction uses the parameters of $P = \{\lambda_r, \lambda_i, \lambda_p\}$ and is defined as follows: $A_p = \{(\lambda_r, \lambda_i, \lambda_p)\}$. The set of reactions is defined as: $A = (\bigcup_{i=1}^{n} A_i) \cup A_p \cup \big\{(\{lock\}, \{done\}, \{lock\})\big\}$. Finally, we define the parametric recation system modelling Mutex as: CR-$\mathcal{CP}_M = ((S, P, A, \mathfrak{c}), \mathfrak{A})$, where: $\mathfrak{c} = \neg\lambda_p[in_n] \wedge \bigwedge_{\lambda \in P, e \in S \setminus S_n} \neg\lambda[e]$ constrains the additional reaction by requiring that it may produce only entities related to the n^{th} process and it cannot produce in_n, to avoid trivial solutions. Then, we need to synthesise a parameter valuation v of CR-$\mathcal{CP}_M$ such that the rsLTL property $\phi = \mathsf{F}(in_1 \wedge in_n)$ holds.

The verification tool was implemented in Python and uses Z3 4.5.0 [10] for SMT-solving. We implement an incremental approach, i.e. in a single SMT instance we increase the length of the encoded interactive processes by unrolling their encoding until witnesses for all the verified formulae are found. Then, the corresponding parameter valuation is extracted. The verification results presented in Fig. 1–2 compare four approaches: the implementation of the encoding

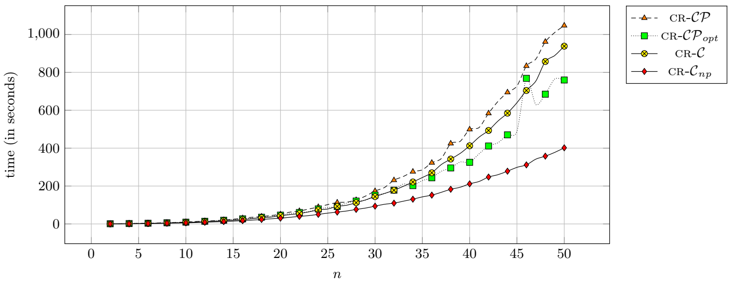

Fig. 1. Synthesis results for PMUTEX: execution time

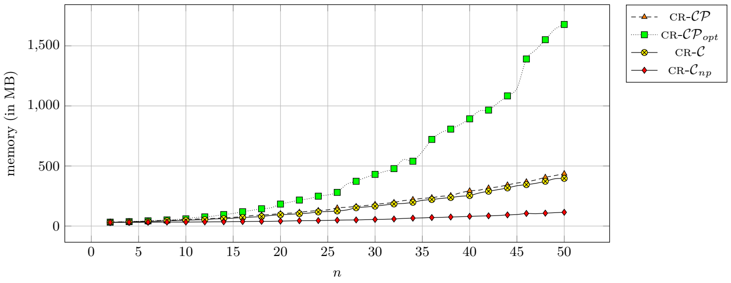

Fig. 2. Synthesis results for PMUTEX: memory consumption

from [8] (CR-$\mathcal{CP}$) and its extension (CR-$\mathcal{CP}_{opt}$) that optimises the obtained parameter valuations by using OptSMT provided with Z3. Then, we also use the same encoding for verification of the rsLTL property (CR-$\mathcal{C}$), i.e. we replace all the parameters with the obtained parameter valuations and test the formula ϕ in the same way as in [11]. Next, we compare our results with the ones obtained using the non-parametric method (CR-$\mathcal{C}_{np}$) of [11]. Our experimental implementation provides a valuation $\mathtt{v}$ which allows to violate the mutual exclution property, where $\lambda_r^{\leftarrow\mathtt{v}} = \{out_n\}$, $\lambda_i^{\leftarrow\mathtt{v}} = \{s\}$, and $\lambda_p^{\leftarrow\mathtt{v}} = \{req_n, done\}$ for all the values $n \geq 2$ tested. This valuation was obtained using CR-$\mathcal{CP}_{opt}$.

When using CR-$\mathcal{CP}_{opt}$, the memory consumption increases. However, the method might require less time to calculate the result than CR-$\mathcal{CP}$. The difference in time and memory consumption between the parametric (CR-$\mathcal{CP}$) and the non-parametric (CR-$\mathcal{C}$) approach is minor. However, CR-$\mathcal{C}_{np}$ is the most efficient of all the approaches tested. This suggests that our parameter synthesis method might possibly be improved by optimising the encoding used.

5 Final remarks

We have developed a method which allows for calculating parameter valuations for partially defined reactions of reaction systems and demonstrated how the presented method can be used for synthesis of an attack in which we inject an additional instruction represented by a reaction, where we use rsLTL to express the goal of the attack. Assuming there is a finite set of allowed concentration levels for the parameters, the presented method also allows for enumerating all the possible parameter valuations for fixed-length processes. This can be achieved by adding an additional constraint blocking the parameter valuation obtained in the previous step. Our method focuses only on existential observations which can be obtained from simulations or experiments performed on the system. However, when we consider some widely accepted laws governing the system under investigation, those should be formulated as universal observations. In our future work we are going to tackle the problem of universal observations.

References

1. Azimi, S., Gratie, C., Ivanov, S., Manzoni, L., Petre, I., Porreca, A.E.: Complexity of model checking for reaction systems. Theoretical Computer Science 623, 103–113 (2016)
2. Azimi, S., Gratie, C., Ivanov, S., Petre, I.: Dependency graphs and mass conservation in reaction systems. Theoretical Computer Science 598, 23–39 (2015)
3. Azimi, S., Iancu, B., Petre, I.: Reaction system models for the heat shock response. Fundamenta Informaticae 131(3-4), 299–312 (2014)
4. Corolli, L., Maj, C., Marini, F., Besozzi, D., Mauri, G.: An excursion in reaction systems: From computer science to biology. Theoretical Computer Science 454, 95–108 (2012)
5. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. Fundamenta Informaticae 75(1-4), 263–280 (2007)
6. Horn, F., Jackson, R.: General mass action kinetics. Archive for Rational Mechanics and Analysis 47(2), 81–116 (1972)
7. Męski, A., Koutny, M., Penczek, W.: Towards quantitative verification of reaction systems. In: Unconventional Computation and Natural Computation: 15th International Conference, UCNC 2016, Manchester, UK, July 11-15, 2016, Proceedings. pp. 142–154 (2016)
8. Męski, A., Koutny, M., Penczek, W.: Reaction mining for reaction systems. In: Unconventional Computation and Natural Computation - 17th International Conference, UCNC 2018, Fontainebleau, France, June 25-29, 2018, Proceedings. pp. 131–144 (2018)
9. Męski, A., Penczek, W., Rozenberg, G.: Model checking temporal properties of reaction systems. Information Sciences 313, 22–42 (2015)
10. de Moura, L., Bjørner, N.: Z3: An efficient SMT solver. In: Proceedings of the 14th International Conference on Tools and Algorithms for Construction and Analysis of Systems. pp. 337–340. TACAS 2008, Springer-Verlang (2008)
11. Męski, A., Koutny, M., Penczek, W.: Verification of linear-time temporal properties for reaction systems with discrete concentrations. Fundam. Inform. 154(1-4), 289–306 (2017)