

Meski A, Koutny M, Penczek P.

[Towards Quantitative Verification of Reaction Systems.](#)

In: 15th International Conference, UCNC. 2016, Manchester: Springer.

Copyright:

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-41312-9_12

Date deposited:

06/05/2016



This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](#)

Towards Quantitative Verification of Reaction Systems

Artur Męski¹, Maciej Koutny², and Wojciech Penczek^{1,3}

¹ Institute of Computer Science, PAS, Jana Kazimierza 5, 01-248 Warsaw, Poland
{meski,penczek@ipipan.waw.pl}

² School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
maciej.koutny@ncl.ac.uk

³ University of Natural Sciences and Humanities, ICS, Siedlce, Poland

Abstract. Reaction systems are a formal model for computational processes inspired by the functioning of the living cell. The key feature of this model is that its behaviour is determined by the interactions of biochemical reactions of the living cell, and these interactions are based on the mechanisms of facilitation and inhibition. The formal treatment of reaction systems is qualitative as there is no direct representation of the number of molecules involved in biochemical reactions.

This paper introduces reaction systems with discrete concentrations which are an extension of reaction systems allowing for quantitative modelling. We demonstrate that although reaction systems with discrete concentrations are semantically equivalent to the original qualitative reaction systems, they provide much more succinct representations in terms of the number of molecules being used. We then define the problem of reachability for reaction systems with discrete concentrations, and provide its suitable encoding in SMT, together with a verification method (bounded model checking) for reachability properties. Experimental results show that verifying reaction systems with discrete concentrations instead of the corresponding reaction systems is more efficient.

1 Introduction

Reaction systems (see, e.g., [5, 7, 8]) are a formal model for processes inspired by the functioning of living cells. The key feature of this model is that the functioning of the living cell is determined by the interactions of biochemical reactions, and these interactions are based on the mechanisms of facilitation and inhibition: the (products of the) reactions may facilitate or inhibit each other. Reaction system related research topics have been motivated by biological issues or by a need to understand computations/processes underlying the dynamic behaviour of reaction systems.

Following their introduction, a number of extensions of reaction systems were studied, e.g., reaction systems with time [9] and quantum and probabilistic reaction systems [13]. Mathematical properties of reaction systems were investigated in, e.g., [10–12, 17–20]. Examples of application of reaction systems to modelling

of systems include, e.g., [3, 6]. Recently, there has been an increasing interest in verification of reaction systems as described in, e.g., [1, 2, 15].

The formal treatment of basic reaction systems is qualitative as no direct representation of the number of molecules involved in biochemical reactions. This paper introduces reaction systems with discrete concentrations which are an extension of reaction systems allowing for quantitative modelling. We demonstrate that although reaction systems with discrete concentrations are semantically equivalent to the original qualitative reaction systems, they provide much more succinct representations in terms of the number of molecules being used.

There exist also other approaches that allow for modelling of complex dependencies of concentration levels and their changes, e.g. chemical reaction networks theory based on [14]. The formalism of reaction systems is much simpler and the processes of reaction systems depend on interactions with the environment.

We define the problem of state reachability for reaction systems with discrete concentrations, and provide its suitable encoding in SMT, together with a verification method (bounded model checking) for reachability properties. Experimental results show that verifying reaction systems with discrete concentrations instead of the corresponding reaction systems is more efficient.

2 Preliminaries

A *reaction system* is a pair $rs = (S, A)$, where S is a finite *background* set and A is a set of *reactions* over the background set. Each reaction in A is a triple $b = (R, I, P)$ such that R, I, P are nonempty subsets of S with $R \cap I = \emptyset$. The sets R, I , and P are respectively denoted by R_b, I_b , and P_b and called the *reactant*, *inhibitor*, and *product set* of reaction b .

A reaction $b \in A$ is *enabled* by $T \subseteq S$, denoted $en_b(T)$, if $R_b \subseteq T$ and $I_b \cap T = \emptyset$. The *result* of b on T is given by $res_b(T) = P_b$ if $en_b(T)$, and by $res_b(T) = \emptyset$ otherwise. Then the *result* of A on T is $res_A(T) = \bigcup\{res_b(T) \mid b \in A\} = \bigcup\{P_b \mid b \in A \text{ and } en_b(T)\}$.

Intuitively, T represents a state of a biochemical system being modelled by listing all present biochemical entities. A reaction b is enabled by T and can take place if all its reactants are present and none of its inhibitors is present in T .

Example 1. Let $(S, A) = (\{1, 2, 3, 4\}, \{a_1, a_2, a_3, a_4\})$ be a reaction system, where:

$$\begin{aligned} a_1 &= (\{1, 4\}, \{2\}, \{1, 2\}) & a_2 &= (\{2\}, \{3\}, \{1, 3, 4\}) \\ a_3 &= (\{1, 3\}, \{2\}, \{1, 2\}) & a_4 &= (\{3\}, \{2\}, \{1\}) \end{aligned}$$

In state $T = \{1, 3, 4\}$ reactions a_1, a_3 , and a_4 are enabled, while a_2 is not. Hence $res_A(T) = res_{a_1}(T) \cup res_{a_3}(T) \cup res_{a_4}(T) = \{1, 2\} \cup \{1, 2\} \cup \{1\} = \{1, 2\}$. \square

Entities in reaction systems are *non-permanent*, i.e., if entity x is present in the successor state T' of a current state T then it must have been produced (sustained) by a reaction enabled by T (thus $x \in res_A(T)$). Also, there are no conflicts between reactions enabled by T . Therefore there is no counting in

reaction systems, and so it is a qualitative model. This follows from the level of abstraction adopted for the basic model. However, in the broad framework of reaction systems (see, e.g., [7]) one considers models with aspects of counting.

A reaction system is a finite system in the sense that the size of each state is a priori limited (by the size of the background set), and the state transformations it describes are deterministic since there are no conflicts between enabled reactions. This changes once we decided to take account of the external environment which is necessary to reflect the fact that the living cell is an open system. Such an environment can be represented by a context automaton.

A *context automaton* over a finite set Ct , is a triple $ca = (Q, q_0, R)$, where Q is a finite set of *states*, $q_0 \in Q$ is the *initial state*, and $R \subseteq Q \times Ct \times Q$ is a *transition relation* labelled with elements of Ct .

A *context restricted reaction system* is a pair $crrs = (rs, ca)$ such that $rs = (S, A)$ is a reaction system, and $ca = (Q, q_0, R)$ is a *context automaton* over 2^S . The dynamic behaviour of $crrs$ is then captured by the state sequences of its interactive processes. An *interactive process* in $crrs$ is $\pi = (\zeta, \gamma, \delta)$, where:

- $\zeta = (z_0, z_1, \dots, z_n)$, $\gamma = (C_0, C_1, \dots, C_n)$, and $\delta = (D_0, D_1, \dots, D_n)$
- $z_0, z_1, \dots, z_n \in Q$ with $z_0 = q_0$
- $C_0, C_1, \dots, C_n, D_0, D_1, \dots, D_n \subseteq S$ with $D_0 = \emptyset$
- $(z_i, C_i, z_{i+1}) \in R$, for every $i \in \{0, \dots, n-1\}$
- $D_i = res_A(D_{i-1} \cup C_{i-1})$, for every $i \in \{1, \dots, n\}$.

Then the *state sequence* of π is $\tau = (W_0, \dots, W_n) = (C_0 \cup D_0, \dots, C_n \cup D_n)$.

Intuitively, the state sequence of π captures the observed behaviour of $crrs$ by recording the successive states of the evolution of the reaction system rs in the environment represented by the context automaton ca .

3 Reaction Systems with Discrete Concentrations

The enabling of some of biochemical reactions encountered in practical applications depends not only on the availability of the necessary reactants and the absence of inhibitors, but also on their concentration levels. To address this aspect in biochemical modelling, we will now introduce an extension of the basic reaction systems supporting an explicit representation of the discrete concentration levels of entities. The resulting model uses bags of entities, but otherwise it retains key features of the original framework. The main new idea that the k -th level of concentration of an entity x is represented by a bag containing k copies of x .

In what follows, a *bag* over a set X is any mapping $\mathbf{b} : X \rightarrow \{0, 1, \dots\}$, and the *empty* bag \emptyset_X is one which always returns 0. We denote this by $\mathbf{b} \in \mathcal{B}(X)$, where $\mathcal{B}(X)$ is the set of all bags over X . For a set \mathbf{B} of bags over X , $\mathbb{M}(\mathbf{B})$ is the bag over X such that $\mathbb{M}(\mathbf{B})(x) = \max(\{0\} \cup \{\mathbf{b}(x) \mid \mathbf{b} \in \mathbf{B}\})$, for every $x \in X$. For two bags, \mathbf{b} and \mathbf{b}' , we denote $\mathbf{b} \leq \mathbf{b}'$ if $\mathbf{b}(x) \leq \mathbf{b}'(x)$, for every $x \in X$. The *carrier* of a bag \mathbf{b} is the set $carr(\mathbf{b}) = \{x \in X \mid \mathbf{b}(x) > 0\}$.

A *reaction system with discrete concentrations* is a pair $rsc = (S, A)$, where S is a finite *background set* and A is a nonempty finite set of *c-reactions* over the background set. Each c-reaction in A is a triple $a = (\mathbf{r}, \mathbf{i}, \mathbf{p})$ such that $\mathbf{r}, \mathbf{i}, \mathbf{p}$ are bags over S with $\mathbf{r}(e) < \mathbf{i}(e)$, for every $e \in \text{carr}(\mathbf{i})$. The sets \mathbf{r}, \mathbf{i} , and \mathbf{p} are respectively denoted by $\mathbf{r}_a, \mathbf{i}_a$, and \mathbf{p}_a and called the *reactant, inhibitor, and product concentration levels* of c-reaction a . We would like to stress that an entity e is an inhibitor of a whenever $e \in \text{carr}(\mathbf{i}_a)$.

A c-reaction $a \in A$ is *enabled* by $\mathbf{t} \in \mathcal{B}(S)$, denoted $en_a(\mathbf{t})$, if $\mathbf{r}_a \leq \mathbf{t}$ and $\mathbf{t}(e) < \mathbf{i}_a(e)$, for every $e \in \text{carr}(\mathbf{i}_a)$. The *result* of a on \mathbf{t} is given by $res_a(\mathbf{t}) = \mathbf{p}_a$ if $en_a(\mathbf{t})$, and by $res_a(\mathbf{t}) = \emptyset_S$ otherwise. Then the *result* of A on \mathbf{t} is $res_A(\mathbf{t}) = \mathbb{M}\{res_a(\mathbf{t}) \mid a \in A\} = \mathbb{M}\{\mathbf{p}_a \mid a \in A \text{ and } en_a(\mathbf{t})\}$.

In the above, \mathbf{t} is a *state* of a biochemical system being modelled such that, for each entity $e \in S$, $\mathbf{t}(e)$ is the *concentration level* of e (e.g., $\mathbf{t}(e) = 0$ indicates that e is not present in the current state, and $\mathbf{t}(e) = 1$ indicates that e is present at its lowest concentration level). A c-reaction a is enabled by \mathbf{t} and can take place if the current concentration levels of all its reactants are at least as high as those specified by \mathbf{r}_a , and the current concentration levels of all its inhibitors (i.e., entities in the carrier of \mathbf{i}_a) are below the thresholds specified by \mathbf{i}_a .

A *context restricted reaction system with discrete concentrations* is a pair $crrsc = (rsc, ca)$ such that $rsc = (S, A)$ is a reaction system with discrete concentrations, and $ca = (Q, q_0, R)$ is a *context automaton* over $\mathcal{B}(S)$. The dynamic behaviour of $crrsc$ is then captured by the state sequences of its interactive processes. An *interactive process* in $crrsc$ is $\pi = (\zeta, \gamma, \delta)$, where:

- $\zeta = (z_0, z_1, \dots, z_n)$, $\gamma = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n)$, and $\delta = (\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_n)$
- $z_0, z_1, \dots, z_n \in Q$ with $z_0 = q_0$
- $\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n, \mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_n \in \mathcal{B}(S)$ with $\mathbf{d}_0 = \emptyset_{\mathcal{B}(S)}$
- $(z_i, \mathbf{c}_i, z_{i+1}) \in R$, for every $i \in \{0, \dots, n-1\}$
- $\mathbf{d}_i = res_A(\mathbb{M}\{\mathbf{d}_{i-1}, \mathbf{c}_{i-1}\})$, for every $i \in \{1, \dots, n\}$.

Then the *state sequence* of π is $\tau = (\mathbf{w}_0, \dots, \mathbf{w}_n) = (\mathbb{M}\{\mathbf{c}_0, \mathbf{d}_0\}, \dots, \mathbb{M}\{\mathbf{c}_n, \mathbf{d}_n\})$.

A context restricted reaction system with discrete concentrations $crrsc = (rsc, ca)$ is a finite state system since it comprises finitely many c-reactions and finitely many bags labelling the arcs of its context automaton. More precisely, let $\#_{crrsc}(e)$ be the maximum integer assigned to $e \in S$ in all the bags of entities occurring in both rsc and ca . Then, $\mathbf{w}(e) \leq \#_{crrsc}(e)$, for all $e \in S$ and all states occurring in the state sequences of the interactive processes in $crrsc$. (Note that this bound can be improved by ignoring the reactant and inhibitor bags in c-reactions.) Moreover, the behaviour of $crrsc$ can be simulated by a suitable context restricted reaction system.

To construct such a system, for every $\mathbf{t} \in \mathcal{B}(S)$, we define two sets of entities, $\Gamma(\mathbf{t}) = \{e.i \mid e \in S \wedge \mathbf{t}(e) = i > 0\}$ and $\Gamma_{all}(\mathbf{t}) = \{e.i \mid e \in S \wedge 1 \leq i \leq \mathbf{t}(e)\}$. The $e.i$'s will be entities of the system we are going to construct. Note that $\Gamma_{all}(\mathbf{t})$ is a *downward-closed* set in the sense that if $e.i \in \Gamma_{all}(\mathbf{t})$ and $i > 1$, then $e.1, \dots, e.(i-1) \in \Gamma_{all}(\mathbf{t})$. In fact, Γ_{all} is a bijection from $\mathcal{B}(S)$ to all the downward-closed sets, and its inverse Γ_{all}^{-1} is given by $\Gamma_{all}^{-1}(Z)(e) = \max\{\{0\} \cup \{i \mid e.i \in Z\}\}$, for every $e \in S$. In what follows, Γ_{all} and Γ_{all}^{-1} will be applied

component-wise to sequences of respectively bags and downward-closed sets. For such $crrsc$, we define the corresponding context restricted reaction system as $\Theta(crrsc) = (rs, ca) = ((S', A'), (Q, q_0, R'))$, where: $S' = \{e.i \mid e \in S \text{ and } 1 \leq i \leq \#_{crrsc}(e)\}$, $A' = \{(\Gamma(\mathbf{r}), \Gamma(\mathbf{i}), \Gamma_{all}(\mathbf{p})) \mid (\mathbf{r}, \mathbf{i}, \mathbf{p}) \in A\}$, and $R' = \{(z, \Gamma_{all}(\mathbf{c}), z') \mid (z, \mathbf{c}, z') \in R\}$. It is straightforward to see that $\Theta(crrsc)$ is well-defined.

As to the complexity of the translation, the number of reactions, states and arrows remains the same. Moreover, the representations of reaction and inhibitors are of the same order. What changes is the size of the background set, in the worst case by the factor $\max\{\#_{crrsc}(e) \mid e \in S\}$ as well as the representations of products and contexts (again by the same factor).

We will now investigate a very close correspondence between $\Theta(crrsc)$ and $crrsc$. First, we observe that, by the definitions of A' and R' , all sets of entities occurring in the interactive processes of $\Theta(crrsc)$ are downward-closed. Then we obtain that all interactive processes of $crrsc$ can be simulated by $\Theta(crrsc)$.

Theorem 1. *If $\pi = (\zeta, \gamma, \delta)$ is an interactive process in $crrsc$, then $\pi' = (\zeta, \Gamma_{all}(\gamma), \Gamma_{all}(\delta))$ is an interactive process in $\Theta(crrsc)$.*

Proof. It suffices to show for \mathbf{w} in the state sequence of π , $\Gamma_{all}(res_A(\mathbf{w})) = res_{A'}(\Gamma_{all}(\mathbf{w}))$. Suppose $a = (\mathbf{r}, \mathbf{i}, \mathbf{p}) \in A$ and $a' = (\Gamma(\mathbf{r}), \Gamma(\mathbf{i}), \Gamma_{all}(\mathbf{p})) \in A'$. We first observe that a is enabled in \mathbf{w} (i.e., $\mathbf{r} \leq \mathbf{w}$ and $\mathbf{w}(e) < \mathbf{i}(e)$, for all $e \in carr(\mathbf{i})$) iff a' is enabled in $\Gamma_{all}(\mathbf{w})$ (i.e., $\Gamma(\mathbf{r}) \subseteq \Gamma_{all}(\mathbf{w})$ and $\Gamma(\mathbf{i}) \cap \Gamma_{all}(\mathbf{w}) = \emptyset$). Moreover, it is easy to check that $\Gamma_{all}(res_a(\mathbf{w})) = res_{a'}(\Gamma_{all}(\mathbf{w}))$. \square

Moreover, all interactive processes of $\Theta(crrsc)$ simulate those of $crrsc$.

Theorem 2. *If $\pi = (\zeta, \gamma, \delta)$ is an interactive process in $\Theta(crrsc)$, then $\pi' = (\zeta, \Gamma_{all}^{-1}(\gamma), \Gamma_{all}^{-1}(\delta))$ is an interactive process in $crrsc$.*

Proof. Similar to the proof of Theorem 1. \square

We have therefore obtained a one-to-one correspondence between the interactive processes of $\Theta(crrsc)$ and $crrsc$.

Remark 1. From the point of view of enabling c-reactions, not all concentration levels are important and, consequently, they do not need to be represented in the states of $\Theta(crrsc)$. To achieve the desired effect, all one needs to do is re-define Γ_{all} , in the following way: $\Gamma'_{all}(\mathbf{t}) = \Gamma(\mathbf{t}) \cup (\Gamma_{all}(\mathbf{t}) \cap \bigcup_{a \in A} \Gamma(\mathbf{r}_a) \cup \Gamma(\mathbf{i}_a))$.

Note that syntactically $crrs$ are a subclass of $crrsc$, such that all the concentration levels in $crrsc$ are limited to the value of at most one, that is, for any $\mathbf{t} \in \mathcal{B}(S)$ and for any $e \in carr(\mathbf{t})$ we have $\mathbf{t}(e) = 1$. Therefore, in the remainder of this paper we use $crrs$ and $crrsc$ interchangeably, depending on the concentration levels required.

4 Reachability testing

In this section we define the reachability problem for *crrsc* and provide its translation into a satisfiability modulo theory (SMT) with integer arithmetic.

Let $n \geq 0$ be an integer. A result $\mathbf{d} \in \mathcal{B}(S)$ is *n-reachable* in *crrsc* if there exists an interactive process $\pi = (\zeta, \gamma, \delta)$ in *crrsc* such that $\delta = (\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_n)$ and $\mathbf{d}_n = \mathbf{d}$. We say that \mathbf{d} is *reachable* in *crrsc* if there is $n \geq 0$ such that \mathbf{d} is *n-reachable* in *crrsc*.

Theorem 3. *The reachability problem for crrsc (crrs) is NP-hard.*

Proof. We show a reduction of 3-SAT to reachability in crrs. The proof is similar to that in [15] for rsCTL model checking. Let $PV = \{x_1, x_2, \dots, x_n\}$ be a set of propositional variables and $\beta(x_1, x_2, \dots, x_n)$ be a boolean formula in 3-CNF. We define the set of the negated propositional variables $\overline{PV} = \{\bar{x} \mid x \in PV\}$ and assume $\beta = c_1 \wedge c_2 \wedge \dots \wedge c_m$, where $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ with $l_{i,j} \in (PV \cup \overline{PV})$, for $1 \leq i \leq m$ and $1 \leq j \leq 3$. Moreover, for a clause c we define the set $vars(c) = \{1 \leq k \leq n \mid x_k \in PV \text{ is in } c\}$ and the set $\overline{vars}(c) = \{1 \leq k \leq n \mid \bar{x}_k \in \overline{PV} \text{ is in } c\}$. Next, we define the crrs which we use for the translation.

Let $\mathcal{V} = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ be the set of entities representing the propositional variables and their negations, and $\mathcal{C} = \{\hat{c}_1, \hat{c}_2, \dots, \hat{c}_m\}$ be the set of the entities that correspond to the clauses. The entity t is used to indicate that under the considered valuation the formula β is true. The entity h is used as the inhibitor of the reactions where no inhibitors are needed for the translation to work. This guarantees that the inhibitor set is non-empty. The background set is $S = \mathcal{V} \cup \mathcal{C} \cup \{t, h\}$, and we define the following sets of reactions:

- $P_i = \{(\{p_i\}, \{h\}, \{p_i\}), (\{\bar{p}_i\}, \{h\}, \{\bar{p}_i\})\}$ for $1 \leq i \leq n$
- $L_i = \{(\{p_k\}, \{\bar{p}_k\}, \{\hat{c}_i\}) \mid k \in vars(i)\} \cup \{(\{\bar{p}_k\}, \{p_k\}, \{\hat{c}_i\}) \mid k \in \overline{vars}(i)\}$
for $1 \leq i \leq m$
- $F = \{(\{\hat{c}_i\}, \{h\}, \{\hat{c}_i\}) \mid 1 \leq i \leq m\} \cup \{(\{\hat{c}_1, \hat{c}_2, \dots, \hat{c}_m\}, \{h\}, \{t\})\}$.

The set P_i contains the reactions responsible for preserving the valuations of the variables along the execution sequences. The reactions of L_i produce entities that indicate whether a single clause is satisfied, whereas the reactions of F that the entity t indicating that all the clauses are satisfied is produced. The set of all the reactions of the crrs is defined as $A = \bigcup_{i=1}^n P_i \cup \bigcup_{i=1}^m L_i \cup F$. Next, we define the context automaton $ca = (Q, q_0, R)$ where $Q = \{1, \dots, n+2\}$, $q_0 = 1$, and $R = \{(i, \{p_i\}, i+1) \mid 1 \leq i \leq n\} \cup \{(i, \{\bar{p}_i\}, i+1) \mid 1 \leq i \leq n\} \cup \{(i+1, \emptyset, i+2)\}$. Then, $rs = (S, A)$ and $crrs = (rs, ca)$. Any path from 1 to $n+1$ in ca corresponds to a valuation of the variables, where a choice of an edge from i to $i+1$ (for $1 \leq i \leq n$) represents a choice of the valuation of x_i (true for p_i , false for \bar{p}_i). When a chosen valuation satisfies a clause c_j (for $1 \leq j \leq m$), then \hat{c}_j is produced, and when \hat{c}_j for all $1 \leq j \leq m$ are produced, then t is produced (in ca this is allowed by the step from $n+1$ to $n+2$). Finally, β is satisfiable if D such that $t \in D$ is reachable in *crrs*. \square

In this paper we focus on the approach of bounded model checking [4], i.e., we test the reachability for all the interactive processes of a given length, and increase the length until the reachability is proved. In what follows we show how n -reachability problem can be encoded by an SMT formula. Due to lack of space, our presentation is quite dense, but it contains the complete encoding.

Let $crrsc = ((S, A), (Q, q_0, R))$ and $\pi = (\zeta, \gamma, \delta)$ be an interactive process in $crrsc$, where $\zeta = (z_0, z_1, \dots, z_n)$, $\gamma = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_n)$, and $\delta = (\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_n)$. Then, the i -th step of π is defined as $\pi_i = (z_i, c_i, d_i)$, where $0 \leq i \leq n$. To encode all the steps of π we introduce the following sets of positive integer variables used in the encoding: $\mathbf{P} = \bigcup_{i=0}^n \{\mathbf{p}_{i,1}, \dots, \mathbf{p}_{i,n}\}$, $\mathbf{P}^\mathcal{E} = \bigcup_{i=0}^n \{\mathbf{p}_{i,1}^\mathcal{E}, \dots, \mathbf{p}_{i,n}^\mathcal{E}\}$, and $\mathbf{Q} = \{q_0, \dots, q_n\}$. Then, π_i is encoded as $s_i = (\mathbf{q}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \bar{\mathbf{p}}_i)$, where \mathbf{q}_i encodes the state z_i of the context automaton, $\bar{\mathbf{p}}_i^\mathcal{E} = (\mathbf{p}_{i,1}^\mathcal{E}, \dots, \mathbf{p}_{i,n}^\mathcal{E})$ encodes the context set \mathbf{c}_i , and $\bar{\mathbf{p}}_i = (\mathbf{p}_{i,1}, \dots, \mathbf{p}_{i,n})$ encodes the result \mathbf{d}_i . With $\bar{\mathbf{p}}_i^\mathcal{E}[j]$ and $\bar{\mathbf{p}}_i[j]$ we denote, respectively, $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i,j}^\mathcal{E}$.

The entities of S are denoted by e_1, \dots, e_k , where $k = |S|$. For π_i we define the following functions that map background set entities to the corresponding variables of the encoding: for all $0 \leq i \leq n$ we define $\mathbf{t}_i : S \rightarrow \mathbf{P}_i$ and $\mathbf{t}_i^\mathcal{E} : S \rightarrow \mathbf{P}_i^\mathcal{E}$ such that $\mathbf{t}_i(e_j) = \mathbf{p}_{i,j}$, $\mathbf{t}_i^\mathcal{E}(e_j) = \mathbf{p}_{i,j}^\mathcal{E}$ for all $1 \leq j \leq k$. The function $\mathbf{e} : Q \rightarrow \{0, \dots, |Q| - 1\}$ maps states of the context automaton to the corresponding natural values used in the encoding. The set of the reactions that produce $e \in S$ is defined as $Prod(e) = \{a \in A \mid \mathbf{p}_a(e) > 0\}$.

To define the SMT encoding of the reachability problem for $crrsc$ we need auxiliary functions that correspond to elements of the encoding.

Result: $\text{Res}_{\mathbf{d}_i}(\bar{\mathbf{p}}_i) = \bigwedge_{e \in S} (\mathbf{t}_i(e) = \mathbf{d}_i(e))$ encodes a result $\mathbf{d}_i \in \mathcal{B}(S)$ as the conjunction of the variables with the corresponding concentration levels.

Context: $\text{Ct}_{\mathbf{c}_i}(\bar{\mathbf{p}}_i^\mathcal{E}) = \bigwedge_{e \in S} (\mathbf{t}_i^\mathcal{E}(e) = \mathbf{c}_i(e))$ encodes a bag $\mathbf{c}_i \in \mathcal{B}(S)$ of context entities.

Enabledness: $\text{En}_a(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}) = \bigwedge_{e \in S} (\mathbf{t}_i(e) \geq \mathbf{r}_a(e) \vee \mathbf{t}_i^\mathcal{E}(e) \geq \mathbf{r}_a(e)) \wedge \bigwedge_{e \in S} (\mathbf{t}_i(e) < \mathbf{i}_a(e) \wedge \mathbf{t}_i^\mathcal{E}(e) < \mathbf{i}(e))$ encodes the enabledness of a reaction a .

Entity concentration: Let f_1, f_2, f_3 be expressions over $\mathbf{P} \cup \mathbf{P}^\mathcal{E}$, then we introduce the *if-then-else* operator: $f_1 \rightarrow f_2 \mid f_3 = (f_1 \wedge f_2) \vee (\neg f_1 \wedge f_3)$. Let $e \in S$, then $Prod^{\text{sorted}}(e) = (a_1, a_2, \dots, a_m)$ is an ordered list of the reactions producing e , where $m = |Prod(e)|$ and $\mathbf{p}_{a_j} \leq \mathbf{p}_{a_{j+1}}$ for all $1 \leq j < m$. The produced concentration level for entity e and reaction a_j , $1 \leq j \leq m$, is encoded as: $\mathbf{C}_e^j(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \bar{\mathbf{p}}_{i+1}) = \text{En}_{a_j}(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}) \rightarrow \mathbf{t}_{i+1}(e) = \mathbf{p}_{a_j} \mid \mathbf{C}_e^{j+1}(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \bar{\mathbf{p}}_{i+1})$ if $j < m$, and $\text{En}_{a_j}(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}) \wedge \mathbf{t}_{i+1}(e) = \mathbf{p}_{a_j}$ if $j = m$. Finally, we define the complete entity concentration encoding for all the reactions. If $m = 0$, then $\mathbf{C}_e(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \bar{\mathbf{p}}_{i+1}) = (\mathbf{t}_{i+1}(e) = 0)$, otherwise $\mathbf{C}_e(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \bar{\mathbf{p}}_{i+1}) = \mathbf{C}_e^1(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \bar{\mathbf{p}}_{i+1}) \vee ((\bigwedge_{a \in Prod(e)} \neg \text{En}_a(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E})) \wedge \mathbf{t}_{i+1}(e) = 0)$.

Transitions of context automaton: The encoding of the transition relation of the context automaton is a disjunction of the encodings for each transition: $\text{Tr}_{ca}(\mathbf{q}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \mathbf{q}_{i+1}) = \bigvee_{(q, \mathbf{c}, q') \in R} (\mathbf{q} = \mathbf{e}(q) \wedge \text{Ct}_{\mathbf{c}}(\bar{\mathbf{p}}_i^\mathcal{E}) \wedge \mathbf{q}_{i+1} = \mathbf{e}_{i+1}(q'))$.

Step of interactive process: We build a conjunction of the produced concentration levels for all entities and the transition relation for the context automaton: $\text{St}(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \bar{\mathbf{p}}_{i+1}, \mathbf{q}_i, \mathbf{q}_{i+1}) = (\bigwedge_{e \in S} \mathbf{C}_e(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \bar{\mathbf{p}}_{i+1})) \wedge \text{Tr}_{ca}(\mathbf{q}_i, \bar{\mathbf{p}}_i^\mathcal{E}, \mathbf{q}_{i+1})$.

Interactive process: To encode n steps of π we define the following formula:

$$\llbracket \pi \rrbracket_n = \text{Res}_{\emptyset_S}(\bar{\mathbf{p}}_0) \wedge \mathbf{e}(z_0) \wedge \bigwedge_{i=0}^{n-1} \text{St}(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_i^\varepsilon, \bar{\mathbf{p}}_{i+1}, \mathbf{q}_i, \mathbf{q}_{i+1}).$$

To perform the n -reachability test of $\mathbf{d} \in \mathcal{B}(S)$ in π we test the satisfiability of the formula $\llbracket \pi \rrbracket_n \wedge \bigvee_{i=0}^n \text{Res}_{\mathbf{d}}(\bar{\mathbf{p}}_i)$. Note that the n -reachability can also be defined for a pair $\rho = (\mathbf{x}, \mathbf{y})$ where $\mathbf{x}, \mathbf{y} \in \mathcal{B}(S)$. Then, ρ is n -reachable if there exists an interactive process $\pi = (\zeta, \gamma, \delta)$ in *crrsc* such that $\delta = (\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_n)$, and $\mathbf{x} \leq \mathbf{d}_n$, $\mathbf{d}_n(e) < \mathbf{y}(e)$, for every $e \in \text{carr}(\mathbf{y})$. In this case, the reachability test for ρ is encoded as $\llbracket \pi \rrbracket_n \wedge \bigvee_{i=0}^n \bigwedge_{e \in S} (\mathbf{t}_i(e) \geq \mathbf{x}(e) \wedge \mathbf{t}_i(e) < \mathbf{y}(e))$.

5 Experimental results

In this section we present the results of an experimental evaluation of the translation presented in Section 4. We compare the implementation for *crrsc* with an implementation for *crrs* by verifying the properties of the *crrs* obtained by applying the translation defined in Section 3 to *crrsc*.

To provide a fair comparison, both the verification tools were implemented in Python using similar techniques and use Z3 [16] for SMT solving. The implementation for *crrs* is based on the encoding from Section 4 which is optimised for *crrs* by using boolean variables instead of integer variables. The translation into SMT for *crrs* corresponds to the translation for *crrsc* – it is assumed that all concentration levels are equal to 1 when an entity is present, and equal to 0 otherwise. We also implement an incremental approach to SMT-solving, i.e., in a single SMT instance we increase the length of the encoded interactive processes by unrolling their encoding until the reachability is proved, instead of creating separate instances for each length tested.

When dealing with concentration levels we often need to perform incrementation and decrementation operations. For this we need additional notation (below we use the notation $e \mapsto i$ to indicate the multiplicity of an entity e in a bag of entities, e.g., $\{e \mapsto 1, f \mapsto 2\}$ is a bag with one copy of e , two copies of f , and nothing else).

Incrementation and decrementation operations: With \uparrow_e^g and \downarrow_e^g we denote the set of reactions encoding the operation of, respectively, incrementation and decrementation of concentration levels of $e \in S$ when $g \in S$ is present with a non-zero concentration. With M_e we denote the maximal allowed value of e . Then $\uparrow_e^g = \{(\{e \mapsto i, g \mapsto 1\}, \emptyset_S, \{e \mapsto i + 1\}) \mid 1 \leq i < M_e\}$ and $\downarrow_e^g = \{(\{e \mapsto i, g \mapsto 1\}, \emptyset_S, \{e \mapsto i - 1\}) \mid 2 < i \leq M_e\}$.

Permanency: $\diamond_e^{\mathbf{i}} = \{(\{e \mapsto i\}, \mathbf{i}, \{e \mapsto i\}) \mid 1 \leq i \leq M_e\}$ is a set of reactions ensuring permanency of $e \in S$ which can be inhibited by $\mathbf{i} \in \mathcal{B}(S)$.

We exploit the notation to use \uparrow_e^g , \downarrow_e^g , and $\diamond_e^{\mathbf{i}}$ in place of regular reactions ignoring that they are in fact sets of reactions. In the implementation for *crrsc* we introduce an optimisation where these reactions are encoded as *macro-reactions*, that is, as simple operations on integer variables that increment, decrement, or retain the value of the variable encoding concentration of e . Moreover, those macro-reactions are allowed only when no ordinary reaction is enabled.

5.1 Eukaryotic heat shock response

entity	description	entity	description
<i>hsp</i>	heat shock protein	<i>hsf3:hse</i>	<i>hsf3</i> bound with <i>hse</i>
<i>hsf</i>	heat shock factor	<i>hsp:mfp</i>	<i>hsp</i> bound with <i>mfp</i>
<i>hsf₂</i>	dimerised heat shock factor	<i>hsp:hsf</i>	complex consisting of <i>hsp</i> and <i>hsf</i>
<i>hsf₃</i>	trimerised heat shock factor	<i>temp</i>	temperature value
<i>hse</i>	heat shock element	<i>cool</i>	decreases the temperature
<i>mfp</i>	misfolded protein	<i>heat</i>	increases the temperature
<i>prot</i>	protein		

Table 1. Entities used in the heat shock response model.

Firstly, we test our implementation using the qualitative model of the eukaryotic heat shock response (HSR) introduced in [3]. HSR is an internal repair mechanism triggered when a cell is subjected to an environmental stressor – increased temperature that is not ideal for its functioning. A temperature exceeding the ideal temperature causes the proteins (*prot*) of a cell to misfold (*mfp*), which in turn may cause its malfunctioning. To facilitate refolding of the proteins, heat shock response proteins (*hsp*) are produced, which are molecular chaperones for the misfolded proteins. The production of *hsp* is initiated by heat shock factors (*hsf*) which are, dimerised (*hsf₂*), and then trimerised (*hsf₃*). Next, *hsf₃* activates *hsp* production by binding to the heat shock element (*hse*) which is the promoter-site of the gene encoding the heat shock proteins.

The original model of [3] used *stress* and *nostress* entities to distinguish between the presence and absence of the heat shock. We assume here that the heat shock appears at (and above) the temperature of 42 °C, and this is modelled using the *temp* entity. All the entities except *temp* remain at the concentration level of 1. We assume that the maximal value of the temperature modelled using the entity *temp* is 50.

The background set S for the rsc modelling HSR consists of the entities in Table 1. The set A_{ord} comprises the reactions in Table 2. We also define the set of reactions dealing with temperature $A_{temp} = \uparrow_{temp}^{heat} \cup \downarrow_{temp}^{cool} \cup \diamond_{temp}^i$, where $\mathbf{i} = \{heat \mapsto 1, cool \mapsto 1\}$. The rsc for HSR is defined as $rsc_{HSR} = (S, A_{ord} \cup A_{temp})$.

To define a crrrs for rsc_{HSR} we use the context automaton $ca_{HSR} = (Q, q_0, R)$ where $Q = \{0, 1\}$, $q_0 = 0$ and $R = \{(0, \{hsf \mapsto 1, prot \mapsto 1, hse \mapsto 1, temp \mapsto 35\}, 1), (1, \{cool \mapsto 1\}, 1), (1, \{heat \mapsto 1\}, 1), (1, \emptyset_S, 1)\}$. Then, the crrsc for rsc_{HSR} is defined as $crrsc_{HSR} = (rsc_{HSR}, ca_{HSR})$. The context set specified in ca_{HSR} for the transition from 0 (the initial state) corresponds to the initial context set used in [3] as the minimal set of entities needed in HSR, together with the *temp* entity indicating a temperature that does not cause the heat shock.

We test the efficiency of our implementation by verifying the reachability of the following results of $crrsc_{HSR}$: $\rho_1 = (\mathbf{x}_1, \mathbf{y}_1)$ where $\mathbf{x}_1 = \{hsp:hsf \mapsto 1, hse \mapsto 1, prot \mapsto 1\}$, $\mathbf{y}_1 = \{temp \mapsto 42\}$. and $\rho_2 = (\mathbf{x}_2, \mathbf{y}_2)$ where $\mathbf{x}_2 = \{mfp \mapsto 1\}$,

reactants	inhibitors	products
$hsf \mapsto 1$	$hsp \mapsto 1$	$hsf_3 \mapsto 1$
$hsf \mapsto 1, hsp \mapsto 1, mfp \mapsto 1$	\emptyset_S	$hsf_3 \mapsto 1$
$hsf_3 \mapsto 1$	$hsp \mapsto 1, hse \mapsto 1$	$hsf \mapsto 1$
$hsp \mapsto 1, hsf_3 \mapsto 1, mfp \mapsto 1$	$hse \mapsto 1$	$hsf \mapsto 1$
$hsf_3 \mapsto 1, hse \mapsto 1$	$hsp \mapsto 1$	$hsf_3:hse \mapsto 1$
$hsp \mapsto 1, hsf_3 \mapsto 1, mfp \mapsto 1, hse \mapsto 1$	\emptyset_S	$hsf_3:hse \mapsto 1$
$hse \mapsto 1$	$hsf_3 \mapsto 1$	$hse \mapsto 1$
$hsp \mapsto 1, hsf_3 \mapsto 1, hse \mapsto 1$	$mfp \mapsto 1$	$hse \mapsto 1$
$hsf_3:hse \mapsto 1$	$hsp \mapsto 1$	$hsp \mapsto 1, hsf_3:hse \mapsto 1$
$hsp, mfp, hsf_3:hse \mapsto 1$	\emptyset_S	$hsp \mapsto 1, hsf_3:hse \mapsto 1$
$hsf \mapsto 1, hsp \mapsto 1$	$mfp \mapsto 1$	$hsp:hsf \mapsto 1$
$hsp:hsf \mapsto 1, temp \mapsto 42$	\emptyset_S	$hsf \mapsto 1, hsp \mapsto 1$
$hsp:hsf \mapsto 1$	$temp \mapsto 42$	$hsp:hsf \mapsto 1$
$hsp \mapsto 1, hsf_3 \mapsto 1$	$mfp \mapsto 1$	$hsp:hsf \mapsto 1$
$hsp \mapsto 1, hsf_3:hse \mapsto 1$	$mfp \mapsto 1$	$hse \mapsto 1, hsp:hsf \mapsto 1$
$temp \mapsto 42, prot \mapsto 1$	\emptyset_S	$mfp \mapsto 1, prot \mapsto 1$
$prot \mapsto 1$	$temp \mapsto 42$	$prot \mapsto 1$
$hsp \mapsto 1, mfp \mapsto 1$	\emptyset_S	$hsp:mfp \mapsto 1$
$mfp \mapsto 1$	$hsp \mapsto 1$	$mfp \mapsto 1$
$hsp:mfp \mapsto 1$	\emptyset_S	$hsp \mapsto 1, prot \mapsto 1$

Table 2. Reactions of the heat shock response model (curly brackets are omitted).

	ρ_1		ρ_2	
	time [s]	memory [MB]	time [s]	memory [MB]
crrs	17.32	25.08	38.78	28.38
crrsc	0.35	24.87	0.93	24.99
improvement	49.48 \times	1.01 \times	41.69 \times	1.13 \times

Table 3. Results for the heat shock response model.

$y_2 = \emptyset_S$. Reachability of ρ_1 proves that it is possible to enter the state where HSR may become stable, while reachability of ρ_2 proves that it is possible for the proteins to eventually misfold. The verification results⁴ are summarised in Table 3. In terms of n -reachability, ρ_1 is proved for $n = 4$, while ρ_2 for $n = 9$. There is no noticeable improvement in memory consumption for the verification of crrsc over crrs. However, there is a significant difference in the execution times in favour of crrsc, e.g., for ρ_1 the verification for crrsc is 49.48 times faster.

⁴ The experimental results were obtained using a system equipped with 3.7GHz Intel Xeon E5 processor and 12GB of memory, running Mac OS X 10.11.3.

5.2 Scalable chain

Here we introduce an abstract system that executes reactions incrementing concentration levels of m molecules up to a maximal concentration level k . The background set is defined as the set of the molecules combined with entities used in the context sets: $S = \{e_1, e_2, \dots, e_m, inc, dec\}$. The *inc* and *dec* entities cause, respectively, incrementation or decrementation of concentration levels. We define the following sets of reactions: $\mathcal{P} = \{(\{e_i \mapsto k\}, \emptyset_S, \{e_{i+1} \mapsto 1\}) \mid 1 \leq i < m\}$, $\mathcal{O} = \{\uparrow_{e_i}^{inc}, \downarrow_{e_i}^{dec} \mid 1 \leq i \leq m\}$, $\mathcal{F} = \{(\{e_m \mapsto k\}, \{dec \mapsto 1\}, \{e_m \mapsto k\})\}$. The reactions of \mathcal{P} take care of the production of the subsequent molecules, while their concentration levels are changed by the reactions of \mathcal{O} . The reaction of \mathcal{F} ensures persistency of the “final” molecule e_m when it reaches the concentration of k , unless *dec* is present. The rsc for the scalable chain system is defined as $rsc_{sc} = (S, \mathcal{P} \cup \mathcal{O} \cup \mathcal{F})$. Next, we define the context automaton $ca_{sc} = (Q, q_0, R)$ where $Q = \{0, 1\}$, $q_0 = 0$, and the set R consists of the following transitions: $(0, \{e_1 \mapsto 1, inc \mapsto 1\}, 1)$, $(1, \{inc \mapsto 1\}, 1)$, $(1, \{dec \mapsto 1\}, 1)$. Finally, we define $crrsc_{sc} = (rsc_{sc}, ca_{sc})$. Time and memory consumption results are presented

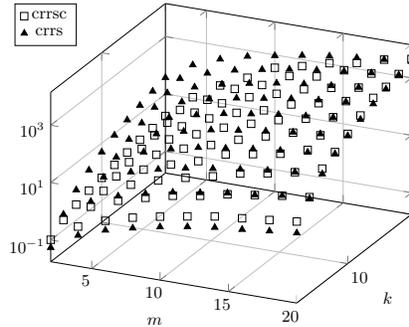


Fig. 1. Time (in seconds)

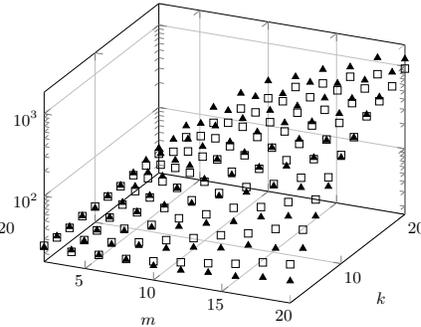


Fig. 2. Memory (in MB)

in Fig. 1–2. The verified reachability property is proved for $n = m \cdot k - 1$. In most cases there is an observable advantage of the implementation for *crrs* when the value of k is relatively large compared to m , e.g., for $m = 8$ and $k = 20$ the results for *crrs* are 5.6 times better. For $m = 10$ and $k = 14$ the verification of *crrs* proved to be 1.6 times more efficient as it only consumed 1334 seconds, compared to 2155 seconds for *crssc*. However, for $m = 20$ and $k = 16$ *crrs* was only 1.2 times better. We attribute this inconsequence to the heuristics of the SMT-solver used. The *crrs* implementation appears to be more memory-efficient when dealing with larger concentration level values. It appears that when the verified system is highly-dependent on a large domain of concentration levels, then the *crrs* will most likely be more suitable.

6 Concluding remarks

In this paper, we introduced reaction systems with discrete concentrations which support quantitative modelling. Although the formalism is not more expressive than the standard reaction systems, the experimental results we obtained demonstrate that expressing concentration levels in an explicit way allows for some improvements in the efficiency of verification, and opens up possibilities for introducing different optimisations.

In our future work we plan to extend this approach to provide a comprehensive framework for verifying quantitative properties of reaction systems.

Acknowledgements

The study is cofounded by the European Union from resources of the European Social Fund. Project PO KL „Information technologies: Research and their interdisciplinary applications”, Agreement UDA-POKL.04.01.01-00-051/10-00.

References

1. Azimi, S., Gratie, C., Ivanov, S., Manzoni, L., Petre, I., Porreca, A.E.: Complexity of model checking for reaction systems. Tech. Rep. 1122, TUCS (2014)
2. Azimi, S., Gratie, C., Ivanov, S., Petre, I.: Dependency graphs and mass conservation in reaction systems. Tech. Rep. 1123, TUCS (2014)
3. Azimi, S., Iancu, B., Petre, I.: Reaction system models for the heat shock response. *Fundam. Inf.* 131(3-4), 299–312 (2014)
4. Biere, A., Cimatti, A., Clarke, E.M., Zhu, Y.: Symbolic model checking without BDDs. In: *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems*. pp. 193–207. TACAS '99, Springer-Verlag, London, UK (1999)
5. Brijder, R., Ehrenfeucht, A., Main, M.G., Rozenberg, G.: A tour of reaction systems. *Int. J. Found. Comput. Sci.* 22(7), 1499–1517 (2011)
6. 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)
7. Ehrenfeucht, A., Kleijn, J., Koutny, M., Rozenberg, G.: Reaction systems: A natural computing approach to the functioning of living cells. *A Computable Universe, Understanding and Exploring Nature as Computation* pp. 189–208 (2012)
8. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. *Fundamenta Informaticae* 75(1-4), 263–280 (2007)
9. Ehrenfeucht, A., Rozenberg, G.: Introducing time in reaction systems. *Theoretical Computer Science* 410(4-5), 310–322 (2009)
10. Formenti, E., Manzoni, L., Porreca, A.E.: Cycles and global attractors of reaction systems. In: *Descriptional Complexity of Formal Systems - 16th International Workshop, DCFS 2014*. pp. 114–125. LNCS (2014)
11. Formenti, E., Manzoni, L., Porreca, A.E.: Fixed points and attractors of reaction systems. In: *Language, Life, Limits - 10th Conference on Computability in Europe, CiE 2014*. LNCS, vol. 8493, pp. 194–203. Springer (2014)

12. Formenti, E., Manzoni, L., Porreca, A.E.: On the complexity of occurrence and convergence problems in reaction systems. *Natural Computing* pp. 1–7 (2014)
13. Hirvensalo, M.: On probabilistic and quantum reaction systems. *Theoretical Computer Science* 429(C), 134–143 (2012)
14. Horn, F., Jackson, R.: General mass action kinetics. *Archive for Rational Mechanics and Analysis* 47(2), 81–116 (1972)
15. Męski, A., Penczek, W., Rozenberg, G.: Model checking temporal properties of reaction systems. *Information Sciences* 313, 22–42 (2015)
16. 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)
17. Salomaa, A.: Functions and sequences generated by reaction systems. *Theoretical Computer Science* 466, 87–96 (2012)
18. Salomaa, A.: On state sequences defined by reaction systems. In: *Logic and Program Semantics*. pp. 271–282 (2012)
19. Salomaa, A.: Functional constructions between reaction systems and propositional logic. *Int. J. of Foundations of Computer Science* 24(1), 147–160 (2013)
20. Salomaa, A.: Minimal and almost minimal reaction systems. *Natural Computing* 12(3), 369–376 (2013)