Natural Computing (2024) 23:323–343
https://doi.org/10.1007/s11047-024-09989-y

Reaction mining for reaction systems

Artur Męski[1,2] · Maciej Koutny[3] · Łukasz Mikulski[1,4] · Wojciech Penczek[1]

Accepted: 18 April 2024 / Published online: 24 July 2024

Abstract
Reaction systems are a formal model for computational processing in which reactions operate on sets of entities (molecules) providing a framework for dealing with qualitative aspects of biochemical systems. This paper is concerned with reaction systems in which entities can have discrete concentrations, and so reactions operate on multisets rather than sets of entities. The resulting framework allows one to deal with quantitative aspects of reaction systems, and a bespoke linear-time temporal logic allows one to express and verify a wide range of key behavioural system properties. In practical applications, a reaction system with discrete concentrations may only be partially specified, and the possibility of an effective automated calculation of the missing details provides an attractive design approach. With this idea in mind, the current paper discusses parametric reaction systems with parameters representing unknown parts of hypothetical reactions. The main result is a method aimed at replacing the parameters in such a way that the resulting reaction system operating in a specified external environment satisfies a given temporal logic formula.This paper provides an encoding of parametric reaction systems in SMT, and outlines a synthesis procedure based on bounded model checking for solving the synthesis problem. It also reports on the initial experimental results demonstrating the feasibility of the novel synthesis method.

Keywords Reaction system, Parameterised reaction, Reaction mining, Synthesis, Temporal logic

1 Introduction

The seminal paper (Ehrenfeucht and Rozenberg 2007) introduced the *reaction systems* model for computational processes inspired by the functioning of a living cell. Reaction systems can capture in a very simple set-theoretic way the basic mechanisms underpinning the dynamic behaviour of a living cell. A key feature of reaction systems is that they can model 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 original reaction system model represents the reactions, states, and dynamic processes using (tuples of) finite sets, and so it directly captures the qualitative aspects of systems. Having said that, more involved structural concepts can be introduced on top of the basic ones.

Several papers on reaction systems have been motivated by biological issues or by the need to understand computations/processes underlying the dynamic behaviour of reaction systems (see, e.g., Ehrenfeucht et al. (2012, 2017)). There are also significant extensions of reaction systems, e.g., reaction systems with time (Ehrenfeucht and Rozenberg 2009), reaction systems with durations (Brijder et al. 2011), and quantum and probabilistic reaction systems (Hirvensalo 2012). Mathematical properties of reaction systems were investigated in, e.g., Alhazov et al. (2016); Formenti et al. (2014a, 2014b,

Artur Męski, Maciej Koutny, Łukasz Mikulski and Wojciech Penczek have contributed equally to this work.

✉ Artur Męski
meski@ipipan.waw.pl

Maciej Koutny
maciej.koutny@ncl.ac.uk

Łukasz Mikulski
lukasz.mikulski@mat.umk.pl

Wojciech Penczek
penczek@ipipan.waw.pl

[1] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

[2] Vector GB Limited, London, UK

[3] School of Computing, Newcastle University, Newcastle upon Tyne, UK

[4] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University in Toruń, Toruń, Poland

2014c); Salomaa (2012a, 2012b, 2013a, 2013b); Dennunzio et al. (2015a, 2015b).

Notable examples of applications of reaction systems to modelling of systems include, e.g., Azimi et al. (2014); Corolli et al. (2012), and the verification of reaction systems was adressed in, e.g., Azimi et al. (2016, 2015); Męski et al. (2015); Dennunzio et al. (2019); Ferrando et al. (2021); Brodo et al. (2023). The papers (Męski et al. 2016; Męski et al. 2017) 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 are other approaches supporting the modelling of complex dependencies of concentration levels and their changes, e.g., chemical reaction networks theory based on Horn and Jackson (1972) or non-linear biochemical control networks theory (Glass and Kauffman 1973), reaction systems provide much simpler framework (including 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 (Męski et al. 2016). The properties being verified can be expressed in rsLTL which is a version of the linear-time temporal logic tailored for reaction systems.

In practical applications, a reaction system with discrete concentrations may have only partially specified reactions, and a reaction mining i.e., a computationally effective method of filling in the missing details could provide an attractive design approach.

In the domain of discrete concurrent systems, which encompass reaction systems as well as closely related membrane systems (Paun and Rozenberg 2002) and tissue systems (Martín-Vide et al. 2003), *synthesis* usually refers to one of two methods for constructing a blueprint for a system's design. The first method includes transformations from system designs in one formalism to (behaviourally) equivalent designs in another formalism. The other method includes transformations of (desired) descriptions of system behaviour into (behaviourally) equivalent system designs. Both approaches have been applied in the fields of reaction systems, membrane systems, and tissue systems. Examples of the first approach include (Azimi et al. 2014) presenting the design a reaction system model for the heat shock response in such a way that the qualitative behavior of the result correlates well with the quantitative behavior of the corresponding ODE model of the heat shock response, and Męski et al. (2023) developing a reaction system model inspired by the mechanistic model of within-cell signal transduction networks of Zañudo et al. (2017). Examples of the second approach include (Kleijn et al. 2011; Kleijn and Koutny 2011) presenting automated synthesis from behavioural specifications into membrane systems which are a computational model carrying out calculations in cells using a biologically inspired abstraction, and Kleijn et al. (2014, 2012) discussing tissue systems which provide a computational abstraction of biological reactions and transport of molecules in a tissue. In both cases the problem addressed was that of synthesising systems from specifications of observed or desired behaviour given in the form of transition systems. The results obtained in Kleijn et al. (2011); Kleijn and Koutny (2011); Kleijn et al. (2014, 2012) were underpinned by strong connections between biologically inspired computational abstractions and Petri nets (Petri 1973) and were based on methods based on the theory of regions (Kleijn et al. 2013). Yet another strand of recently proposed work aimed at building reaction systems is to provide facilities allowing synthesising of complex designs using established programming constructs as well as the concepts related to asynchrony in real-life system behaviours (Koutny et al. 2021). A preliminary version of the current paper (Meski et al. 2018) extended the scope of synthesis to reaction systems in which entities can have discrete concentrations and reactions operate on multisets of entities, providing a succinct framework for dealing with quantitative aspects of biological systems. It introduced reaction systems with parameters representing the unknown parts of the reactions. The main result was a method to replace these parameters in such a way that the resulting reaction system operating in a given external environment satisfied a given temporal logic formula. Following the approach based on reaction mining, this paper discusses reaction systems with parameters representing the unknown parts of the reactions. The main result is a methodology which replaces 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, the rsLTL formula might correspond to a several observations of the behaviour of a partially specified system. The environment is given by a *context automaton* representing the influence of an encompassing system. The current paper provides an encoding of parametric reaction systems in SMT, and proposes a synthesis procedure based on bounded model checking. It also reports on the successful preliminary experimental results demonstrating the scalability of the new synthesis method.

The paper relies on the findings from the PhD thesis of one of the authors (Męski 2020), which have not been previously published anywhere else. It is important to emphasize that this paper not only introduces a method for parameter synthesis for reaction systems, but it also extends previous results for rsLTL published in Męski et al. (2017). Our paper provides a complexity result for rsLTL model checking as well as correctness analysis for the SMT encoding. Additionally, it shows how model checking for rsLTL can be translated into a corresponding problem for LTL. Last but not least, it proposes a method for reaction mining, for which a parametric extension of reaction systems is introduced.

The paper is organised as follows. The next section recalls the basic notations and definitions used by reaction systems with discrete concentrations. For self-containment, Sect. 3 recalls reaction systems with concentrations from Męski et al. (2016) and Męski et al. (2017) and shows bisimulation with the standard reaction systems demonstrating that discrete concentrations do not increase the expressive power, but allow for more concise models. The modal logic rsLTL for specifying properties of context-restricted systems is defined in Sect. 4. In the following section, an SMT-based encoding for both formalisms is provided together with a proof of its correctness. Section 6 introduces parametric reaction systems and their constrained variant together with accompanying context automata. Section 7 provides an SMT-based encoding for parametric reaction systems. Experimental evaluation of the proposed synthesis approach is presented in Sect. 8.

2 Preliminaries

2.1 Multisets

A *multiset* over a set X is a mapping $\mathbf{b} : X \to \mathbb{N}$, and the *empty* multiset $\varnothing_X$ (or simply $\varnothing$) is the constant function which always returns 0. The *carrier* of a multiset $\mathbf{b}$ is the set $carr(\mathbf{b}) = \{x \in X \mid \mathbf{b}(x) > 0\}$. By $\mathcal{B}(X)$ we denote the set of all multisets over X, hence $\mathbf{b} \in \mathcal{B}(X)$ means that $\mathbf{b}$ is a multiset over X. We define the union $\mathbb{M}$ of multisets in a standard way. For a finite set $\mathbf{B}$ of multisets over X, $\mathbb{M}(\mathbf{B})(x) = \max(\{\mathbf{b}(x) \mid \mathrm{b} \in \mathbf{B}\})$, for every $x \in X$. For two multisets, $\mathbf{b}$ and $\mathbf{b}'$, we denote $\mathbf{b} \leq \mathbf{b}'$ if $\mathbf{b}(x) \leq \mathbf{b}'(x)$, for every $x \in X$.

In what follows, we use the notation $x \mapsto i$ to indicate the multiplicity of an element x in a multiset over X. For example, $\{x \mapsto 2, y \mapsto 1\}$ is a multiset with two copies of x, one copy of y, and nothing else. If the multiplicity of an element is 1, we may also simply omit the value as in, e.g., $\{x \mapsto 2, y\}$.

2.2 Reaction systems

The basic definitions of reaction systems can be found in the seminal paper by Ehrenfeucht and Rozenberg (2007). However, we use the equivalent formulations originating from Meski et al. (2019).

Definition 1 (reaction system) A *reaction system* is a pair $\mathcal{R} = (S, A)$, where S is a finite *background* set and A is a set of *reactions* over S. A reaction is a triple

$$a = (R, I, P) = (R_a, I_a, P_a),$$

where R, I, P are nonempty subsets of S such that $R \cap I = \varnothing$, called respectively the *reactant*, *inhibitor*, and *product set*.

Definition 2 (behaviour of reaction system) Given a *reaction system* $\mathcal{R} = (S, A)$, a reaction $a \in A$ is *enabled* by $T \subseteq S$, denoted $en_a(T)$, if $R_a \subseteq T$ and $I_a \cap T = \varnothing$. The *result* of a on T is given by $res_a(T) = P_a$ if $en_a(T)$ holds, and by $res_a(T) = \varnothing$ otherwise. Moreover, the *result* of A on T is

$$res_A(T) = \bigcup\{res_a(T) \mid a \in A\} = \bigcup\{P_a \mid a \in A \,\text{and}\, en_a(T)\}.$$

The set T in Definition 2 represents the current state of a biochemical system being modelled by listing all present biochemical entities. A reaction is enabled and can take place if all its reactants are present and none of its inhibitors is present. The system reaches the next state $res_A(T)$ by executing all the reactions enabled 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 the state $T = \{1,3,4\}$ reactions a_1, a_3, and a_4 are enabled, while a_2 is not. Hence

$$res_A(T) = res_{a_1}(T) \cup res_{a_3}(T) \cup res_{a_4}(T) = \{1, 2\} \cup \{1, 2\} \cup \{1\} = \{1, 2\}.$$ □

A reaction system is a finite state system as each state is a subset of the finite background set, and its state transformations are deterministic since there are no conflicts between enabled reactions. This changes after introducing the external environment, represented by a context automaton, in order to reflect the fact that the living cell is an open system.

Definition 3 (context automaton) A *context automaton* over a set Σ is a triple $\mathfrak{A} = (\mathcal{Q}, \mathfrak{q}^{init}, R)$, where $\mathcal{Q}$ is a finite set of *states*, $\mathfrak{q}^{init} \in \mathcal{Q}$ is the *initial state*, and $R \subseteq \mathcal{Q} \times \Sigma \times \mathcal{Q}$ is a *transition relation* labelled with the elements of Σ.

We assume that for all $\mathfrak{q} \in \mathcal{Q}$ there exists $c \in \Sigma$ and $\mathfrak{q}' \in \mathcal{Q}$ such that $(\mathfrak{q}, c, \mathfrak{q}') \in R$.

Definition 4 (context-restricted reaction system) A *context-restricted reaction system* is a pair $\textsc{cr}\text{-}\mathcal{R} = (\mathcal{R}, \mathfrak{A})$ such that $\mathcal{R} = (S, A)$ is a reaction system, and $\mathfrak{A} = (\mathcal{Q}, \mathfrak{q}^{init}, R)$ is a *context automaton* over 2^S.

The dynamic behaviour of a context-restricted reaction system is captured by the state sequences of its interactive processes.

Definition 5 (interactive process) An *interactive process* in a context-restricted reaction system $\textsc{cr}\text{-}\mathcal{R} = (\mathcal{R}, \mathfrak{A})$, where $\mathcal{R} = (S, A)$ and $\mathfrak{A} = (\mathcal{Q}, \mathfrak{q}^{init}, R)$, is a triple $\pi = (\zeta, \gamma, \delta)$ such that:

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

Then the *state sequence* of π is

$$\tau = (W_0, \ldots, W_n) = (C_0 \cup D_0, \ldots, C_n \cup D_n).$$

Intuitively, the state sequence of π captures the observed behaviour of $\textsc{cr}\text{-}\mathcal{R}$ by recording the successive states of the evolution of the reaction system $\mathcal{R}$ in the environment represented by the context automaton $\mathfrak{A}$. Note that in Definition 5 it is required that for each context set C_i there exists in the automaton a transition from its current state.

3 Reaction systems with discrete concentrations

In practice, the *concentration levels* of the reactant and inhibitor molecules/entities can be more useful than simply using binary values representing their presence or absence. This can be addressed by introducing an explicit representation of the discrete concentration levels of entities using multisets rather than sets. In concrete terms, we will represent the k^{th} level of concentration of an entity x by including in the multiset of entities exactly k copies of x.

Definition 6 (reaction system with discrete concentrations (Męski et al. 2016)) A *reaction system with discrete concentrations* is a pair $\mathcal{C} = (S, A)$ such that:

- S is a finite *background* set; and
- A is a nonempty finite set of *c-reactions* over the background set, where a *c-reaction* is a triple $a = (\mathbf{r}, \mathbf{i}, \mathbf{p})$ such that $\mathbf{r}$, $\mathbf{i}$, $\mathbf{p}$ are multisets over S with $\mathbf{r}(e) < \mathbf{i}(e)$, for every $e \in carr(\mathbf{i})$.

The multisets $\mathbf{r} = \mathbf{r}_a$, $\mathbf{i} = \mathbf{i}_a$, and $\mathbf{p} = \mathbf{p}_a$ in Definition 6 are the *reactant*, *inhibitor*, and *product* concentration levels, respectively. Moreover, an entity e is an inhibitor of a whenever $e \in carr(\mathbf{i}_a)$.

Similarly as in the case of basic reaction systems recalled in Sect. 2.2, a c-reaction a is *enabled* by $\mathbf{t} \in \mathcal{B}(S)$ if it is facilitated by reactants (their concentrations are high enough) and not inhibited by inhibitors (their concentrations are below specified threshold).

Definition 7 [behaviour of reaction system with discrete concentrations (Męski et al. 2016)] Given a reaction system with discrete concentrations $\mathcal{C} = (S, A)$, a c-reaction $a \in A$ is *enabled* by $\mathbf{t} \in \mathcal{B}(S)$ (denoted by $en_a(\mathbf{t})$) if $\mathbf{r}_a \leq \mathbf{t}$ and $\mathbf{t}(e) < \mathbf{i}_a(e)$, for every $e \in carr(\mathbf{i}_a)$. The *result* of a on $\mathbf{t}$ is $res_a(\mathbf{t}) = \mathbf{p}_a$ if $en_a(\mathbf{t})$, and $res_a(\mathbf{t}) = \varnothing_S$ otherwise. Then the *result* of A on $\mathbf{t}$ is:
$res_A(\mathbf{t}) = \text{⩕}\{res_a(\mathbf{t}) \mid a \in A\} = \text{⩕}\{\mathbf{p}_a \mid a \in A \textit{ and } en_a(\mathbf{t})\}$.

The $\mathbf{t}$ in Definition 7 can be regarded as a *state* of a biochemical system, where $\mathbf{t}(e)$ is the *concentration level* of each entity e. In particular, $\mathbf{t}(e) = 0$ indicates the absence of e is in the current state, while $\mathbf{t}(e) = 1$ indicates that e is present in a current state at its lowest observable concentration level.

Definition 8 [context-restricted reaction system with discrete concentrations (Męski et al. 2017)] A *context-restricted reaction system with discrete concentrations* is a pair $\textsc{cr}\text{-}\mathcal{C} = (\mathcal{C}, \mathfrak{A})$ such that $\mathcal{C} = (S, A)$ is a reaction system with discrete concentrations, and $\mathfrak{A} = (\mathcal{Q}, \mathfrak{q}^{init}, R)$ is a *context automaton* over $\mathcal{B}(S)$.

To capture the dynamic behaviours a context-restricted reaction system with discrete concentrations, we next introduce interactive processes using state sequences.

Definition 9 (interactive process (Męski et al. 2017)) An *interactive process* in a context-restricted reaction system with discrete concentrations $\textsc{cr}\text{-}\mathcal{C} = (\mathcal{C}, \mathfrak{A})$, where $\mathcal{C} = (S, A)$ and $\mathfrak{A} = (\mathcal{Q}, \mathfrak{q}^{init}, R)$, is a triple $\pi = (\zeta, \gamma, \delta)$ such that:

- $\zeta = (\mathfrak{q}_0, \mathfrak{q}_1, \ldots, \mathfrak{q}_n)$, $\gamma = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n)$, and $\delta = (\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n)$
- $\mathfrak{q}_0, \mathfrak{q}_1, \ldots, \mathfrak{q}_n \in \mathcal{Q}$ with $\mathfrak{q}_0 = \mathfrak{q}^{init}$
- $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n, \mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n \in \mathcal{B}(S)$ with $\mathbf{d}_0 = \varnothing_{\mathcal{B}(S)}$

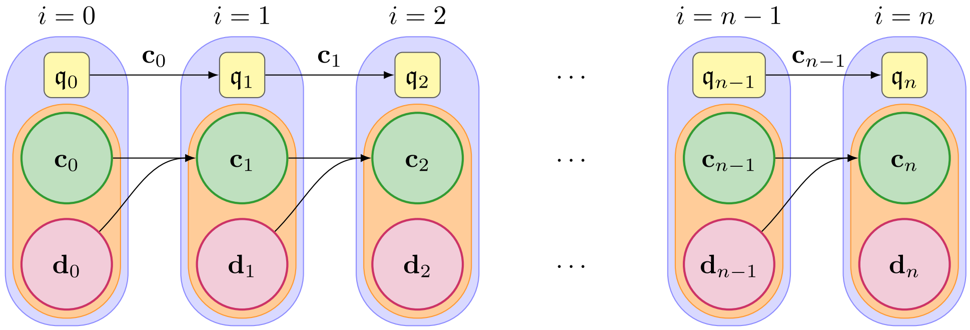

Fig. 1 Interactive process in context-restricted reaction system with discrete concentrations

- $(\mathfrak{q}_i, \mathbf{c}_i, \mathfrak{q}_{i+1}) \in R$, for every $i \in \{0, \ldots, n-1\}$
- $\mathbf{d}_i = res_A(\Lambda\!\!\!\setminus\{\mathbf{d}_{i-1}, \mathbf{c}_{i-1}\})$, for every $i \in \{1, \ldots, n\}$.

Then the *state sequence* of π is

$$\tau = (\mathbf{w}_0, \ldots, \mathbf{w}_n) = (\Lambda\!\!\!\setminus\{\mathbf{c}_0, \mathbf{d}_0\}, \ldots, \Lambda\!\!\!\setminus\{\mathbf{c}_n, \mathbf{d}_n\}).$$

A diagrammatic representation of an interactive process in a context-restricted reaction system with discrete concentrations is depicted in Fig. 1.

Note that a context-restricted reaction system with discrete concentrations $\textsc{cr}\text{-}\mathcal{C} = (\mathcal{C}, \mathfrak{A})$ comprises finitely many c-reactions. Since we also have finitely many multisets labelling on the arcs of its context automaton, $\textsc{cr}\text{-}\mathcal{C}$ is a finite state system. We can demonstrate this more precisely by showing that there exists a context-restricted reaction system which simulates the behaviour of $\textsc{cr}\text{-}\mathcal{C}$. Let $\#_{\textsc{cr}-\mathcal{C}}(e)$ be such that $\mathbf{w}(e) \leq \#_{\textsc{cr}-\mathcal{C}}(e)$, for all $e \in S$ and all the states occurring in the state sequences of the interactive processes in $\textsc{cr}\text{-}\mathcal{C}$. (Note that such a bound can be obtained by taking the maximum integer assigned to $e \in S$ in all the multisets of entities occurring in both $\mathcal{C}$ and $\mathfrak{A}$.)

To construct a reaction system that simulates $\textsc{cr}\text{-}\mathcal{C}$, for every multiset $\mathbf{t} \in \mathcal{B}(S)$, we define two sets of entities:

$$Gamma(\mathbf{t}) = \{e.i \mid e \in S \wedge \mathbf{t}(e) = i > 0\} \quad and$$
$$\Gamma_{all}(\mathbf{t}) = \{e.i \mid e \in S \wedge 1 \leq i \leq \mathbf{t}(e)\}.$$

The set Γ contains all the entities provided by the transition $\mathbf{t}$ together with their concentration, while Γ_{all} is the *downward-closure* of Γ (i.e., if $e.i \in \Gamma(\mathbf{t})$ and $i > 1$, then $e.1, \ldots, e.($ i-1$) \in \Gamma_{all}(\mathbf{t})$). In fact, Γ_{all} is a bijection from the set of all multisets over S to all the downward-closed sets of entities, 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 multisets and downward-closed sets. We define the context-restricted reaction system corresponding to $\textsc{cr}\text{-}\mathcal{C}$ as

$$\Theta(\textsc{cr}\text{-}\mathcal{C}) = (\mathcal{R}, \mathfrak{A}) = ((S', A'), (\mathcal{Q}, \mathfrak{q}^{init}, R'))$$

where:

- $S' = \{e.i \mid e \in S$ and $1 \leq i \leq \#_{\textsc{cr}-\mathcal{C}}(e)\}$,
- $A' = \{(\Gamma(\mathbf{r}), \Gamma(\mathbf{i}), \Gamma_{all}(\mathbf{p})) \mid (\mathbf{r}, \mathbf{i}, \mathbf{p}) \in A\}$, and
- $R' = \{(z, \Gamma_{all}(\mathbf{c}), z') \mid (z, \mathbf{c}, z') \in R\}$.

It is straightforward to see that $\Theta(\textsc{cr}\text{-}\mathcal{C})$ is well-defined.

Let us discuss the complexity of the translation. The sizes of the set of reactions, the set of states and the set of transitions remain the same. The representations of reactants and inhibitors are of the same order (i.e., multiplied by a constant allowing to specify the concentration level). However, the sizes of the background set as well as the representations of products and contexts are increased by the factor $\max\{\#_{\textsc{cr}-\mathcal{C}}(e) \mid e \in S\}$ in the worst case.

Let us look at a relationship between $\Theta(\textsc{cr}\text{-}\mathcal{C})$ and $\textsc{cr}\text{-}\mathcal{C}$. First, we observe that, by the definitions of A' and R', all sets of entities occurring in the interactive processes of $\Theta(\textsc{cr}\text{-}\mathcal{C})$ are downward-closed. Theorems 1 and 2 below demonstrate that all interactive processes of $\textsc{cr}\text{-}\mathcal{C}$ can be simulated by $\Theta(\textsc{cr}\text{-}\mathcal{C})$ and all interactive processes of $\Theta(\textsc{cr}\text{-}\mathcal{C})$ simulate those of $\textsc{cr}\text{-}\mathcal{C}$.

Theorem 1 (Męski et al. 2017) *If $\pi = (\zeta, \gamma, \delta)$ is an interactive process in $\textsc{cr}\text{-}\mathcal{C}$, then*

$$\pi' = (\zeta, \Gamma_{all}(\gamma), \Gamma_{all}(\delta))$$

is an interactive process in $\Theta(\textsc{cr}\text{-}\mathcal{C})$.

Theorem 2 (Męski et al. 2017) *If $\pi = (\zeta, \gamma, \delta)$ is an interactive process in $\Theta(\textsc{cr}\text{-}\mathcal{C})$, then*

$$\pi' = (\zeta, \Gamma_{all}^{-1}(\gamma), \Gamma_{all}^{-1}(\delta))$$

is an interactive process in $\textsc{cr}\text{-}\mathcal{C}$.

In this way, we have obtained a bisimulation between the interactive processes in $\Theta(\textsc{cr}\text{-}\mathcal{C})$ and $\textsc{cr}\text{-}\mathcal{C}$.

Remark 1 We can redefine Γ_{all} as:

$$\Gamma'_{all}(\mathbf{t}) = \Gamma(\mathbf{t}) \cup (\Gamma_{all}(\mathbf{t}) \cap \bigcup_{a \in A} \Gamma(\mathbf{r}_a) \cup \Gamma(\mathbf{i}_a))$$

and then only use those concentration levels which are important from the point of view of enabling the c-relations.

Note that syntactically context-restricted reaction system are a subclass of context-restricted reaction system with discrete concentrations such that all the concentration levels are binary, i.e., for all $\mathbf{t} \in \mathcal{B}(S)$ and $e \in carr(\mathbf{t})$, we have $\mathbf{t}(e) = 1$.

4 Linear-time temporal logic for reaction systems

In this section we recall from Męski et al. (2017) a linear-time temporal logic tailored to reaction systems and we evaluate its expressiveness.

Consider the following grammar of *multiset expressions* over a background set S:

$$\mathfrak{a} ::= true \mid e \sim c \mid e_1 \sim e_2 \mid \neg\mathfrak{a} \mid \mathfrak{a} \vee \mathfrak{a},$$

where $\sim \; \in \{<, \leq, =, \geq, >\}$, $e \in S$, and $c \in \mathbb{N}$. The set of all the multiset expressions over S is denoted by $BE(S)$.

Let $\mathbf{b}$ be a multiset over S. By $\mathbf{b} \models_b \mathfrak{a}$ we denote that $\mathfrak{a}$ holds in $\mathbf{b}$. The relation $\models_b$ is defined as follows:

$$\begin{array}{lll}
\mathbf{b} \models_b true & \text{for any} & \mathbf{b} \in \mathcal{B}(S), \\
\mathbf{b} \models_b e_1 \sim c & \text{iff} & \mathbf{b}(e_1) \sim c, \\
\mathbf{b} \models_b e_1 \sim e_2 & \text{iff} & \mathbf{b}(e_1) \sim \mathbf{b}(e_2), \\
\mathbf{b} \models_b \neg\mathfrak{a} & \text{iff} & \mathbf{b} \not\models_b \mathfrak{a}, \\
\mathbf{b} \models_b \mathfrak{a}_1 \vee \mathfrak{a}_2 & \text{iff} & \mathbf{b} \models_b \mathfrak{a}_1 \text{ or } \mathbf{b} \models_b \mathfrak{a}_2.
\end{array}$$

Note that the conjunction $\mathfrak{a}_1 \wedge \mathfrak{a}_2$ can be derived as $\neg(\neg\mathfrak{a}_1 \vee \neg\mathfrak{a}_2)$.

The language of *reaction systems linear-time temporal logic* (rsLTL, for short) is give by the following grammar, where $\mathfrak{a} \in BE(S)$:

$$\phi ::= \mathfrak{a} \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathsf{X}_{\mathfrak{a}}\phi \mid \phi\mathsf{U}_{\mathfrak{a}}\phi \mid \phi\mathsf{R}_{\mathfrak{a}}\phi$$

The logic captures constraints imposed on paths in the model of a context restricted reaction system with discrete concentrations. Intuitively, $\mathsf{X}_{\mathfrak{a}}\phi$ means 'following an action satisfying $\mathfrak{a}$, ϕ holds in the next state', $\phi_1\mathsf{U}_{\mathfrak{a}}\phi_2$ means 'ϕ_2 holds eventually, and ϕ_1 must hold at every preceding state, following only actions satisfying $\mathfrak{a}$', and $\phi_1\mathsf{R}_{\mathfrak{a}}\phi_2$ means 'following only actions satisfying $\mathfrak{a}$, ϕ_2 holds up to and including the first state where ϕ_1 holds'.

In order to define the semantics of the introduced operators, we need to define the model of a given cr-$\mathcal{C}$.

Definition 10 (Męski et al. 2017) Let cr-$\mathcal{C} = (\mathcal{C}, \mathfrak{A})$ be a context-restricted reaction system with discrete concentrations, where $\mathcal{C} = (S, A)$ and $\mathfrak{A} = (\mathcal{Q}, \mathfrak{q}^{init}, R)$. Then, the *model* for cr-$\mathcal{C}$ is a triple $\mathcal{M} = (\mathbb{W}, \mathtt{w}^{\mathtt{init}}, \longrightarrow)$, where:

- $\mathbb{W} = \mathcal{B}(S) \times \mathcal{Q}$ is the set of states,
- $\mathtt{w}^{\mathtt{init}} = (\varnothing, \mathfrak{q}^{\mathtt{init}})$ is the initial state,
- $\longrightarrow \subseteq \mathbb{W} \times \mathcal{B}(S) \times \mathbb{W}$ is the transition relation such that for all $\mathbf{w}, \mathbf{w}', \alpha \in \mathcal{B}(S)$, $\mathfrak{q}, \mathfrak{q}' \in \mathcal{Q}$: $\big((\mathbf{w}, \mathfrak{q}), \alpha, (\mathbf{w}', \mathfrak{q}')\big) \in \longrightarrow$ iff: $(\mathfrak{q}, \alpha, \mathfrak{q}') \in R \; and \; \mathbf{w}' = res_A(\mathbin{\wedge\!\!\wedge}\{\mathbf{w}, \alpha\})$.

Each element $(\mathtt{w}, \alpha, \mathtt{w}') \in \longrightarrow$ is denoted $\mathtt{w} \xrightarrow{\alpha} \mathtt{w}'$.

Assuming the notation as in Definition 10, the paths in rsLTL are defined as alternating sequences of states and actions (*context multisets*).

Definition 11 (Męski et al. 2017) A *path* is an infinite sequence $\sigma = (\mathtt{w}_0, \alpha_0, \mathtt{w}_1, \alpha_1, \ldots)$ of states and actions such that: $\mathtt{w}_\mathtt{i} \xrightarrow{\alpha_\mathtt{i}} \mathtt{w}_{\mathtt{i+1}}$ and $\alpha_i \in \mathcal{B}(S)$, for $i \geq 0$.

Let σ be a path as in Definition 11. By $\sigma_s(i)$ and $\sigma_a(i)$ we denote the i^{th} state $\mathtt{w}_\mathtt{i}$ and the i^{th} action α_i of the path σ, respectively. Let $\sigma_s(i) = (\mathbf{w}_i, q_i)$, for each $i \geq 0$. Then $\mathbf{w}_i$ is denoted by $\sigma_b(i)$, while q_i is denoted by $\sigma_{ca}(i)$. If $i \geq 0$, then σ^i denotes the suffix of σ such that

$$\sigma^i = (\sigma_s(i), \sigma_a(i), \sigma_s(i+1), \sigma_a(i+1), \ldots),$$

i.e., $\sigma_s^i(j) = \sigma_s(j+i)$ and $\sigma_a^i(j) = \sigma_a(j+i)$, for each $j \geq 0$. By $\Pi_{\mathcal{M}}$ we denote the set of all the paths of the model $\mathcal{M}$, whereas by $\Pi_{\mathcal{M}}(\mathtt{w})$ we denote the set of all the paths that start in $\mathtt{w} \in \mathbb{W}$, that is, $\Pi_{\mathcal{M}}(\mathtt{w}) = \{\sigma \in \Pi \mid \sigma_\mathtt{s}(0) = \mathtt{w}\}$.

Definition 12 (Męski et al. 2017) Let $\mathcal{M} = (\mathbb{W}, \mathtt{w}^{\mathtt{init}}, \longrightarrow)$ be a model as in Definition 10, and $\sigma \in \Pi_{\mathcal{M}}$ be a path of $\mathcal{M}$.

The fact that ϕ holds over σ is denoted by $\mathcal{M}, \sigma \models \phi$ (or $\sigma \models \phi$ if $\mathcal{M}$ is understood), where the relation $\models$ is defined as follows:

$$\begin{array}{lll}
\sigma \models \mathfrak{a} & \text{iff} & \sigma_b(0) \models_b \mathfrak{a}, \\
\sigma \models \phi_1 \vee \phi_2 & \text{iff} & \sigma \models \phi_1 \text{ or } \sigma \models \phi_2, \\
\sigma \models \phi_1 \wedge \phi_2 & \text{iff} & \sigma \models \phi_1 \text{ and } \sigma \models \phi_2, \\
\sigma \models \mathsf{X}_{\mathfrak{a}}\phi_1 & \text{iff} & \sigma_a(0) \models_b \mathfrak{a} \text{ and } \sigma^1 \models \phi_1, \\
\sigma \models \phi_1\mathsf{U}_{\mathfrak{a}}\phi_2 & \text{iff} & (\exists j \geq 0)\big(\sigma^j \models \phi_2 \\
 & & \text{and } (\forall 0 \leq l < j)(\sigma^l \models \phi_1 \text{ and } \sigma_a(l) \models_b \mathfrak{a})\big), \\
\sigma \models \phi_1\mathsf{R}_{\mathfrak{a}}\phi_2 & \text{iff} & (\forall j \geq 0)\big((\sigma^j \models \phi_2 \text{ and } (\forall 0 \leq l < j)(\sigma_a(l) \models_b \mathfrak{a})) \\
 & & \text{or } (\exists 0 \leq l < j)(\sigma^l \models \phi_1)\big).
\end{array}$$

Note that one can easily derive the following operators:

$$\alpha \Rightarrow \phi \overset{def}{\equiv} \neg\alpha \vee \phi, \quad \mathsf{G}_{\alpha}\phi \overset{def}{\equiv} \mathit{false}\mathsf{R}_{\alpha}\phi, \quad \mathsf{F}_{\alpha}\phi \overset{def}{\equiv} \mathit{true}\mathsf{U}_{\alpha}\phi.$$

Moreover, we assume $\alpha = \mathit{true}$ when α is unspecified for any of the rsLTL operators, e.g., $\mathsf{F}\phi$ is the same as $\mathsf{F}_t rue\phi$. The fragment of rsLTL where $\alpha = \mathit{true}$ for all the multiset expressions α is called LTL. And, if the exact concentration levels are irrelevant, we may simply write e for $e > 0$ and $\neg e$ for $e = 0$.

An rsLTL formula holds (holds existentially) in a model iff it holds in all the paths (at least one path) starting in the initial state. More precisely, $\mathcal{M} \models \phi$ iff $\sigma \models \phi$ for all $\sigma \in \Pi_{\mathcal{M}}(\mathtt{w}^{\mathtt{init}})$, and $\mathcal{M} \models_{\exists} \phi$ iff there exists $\sigma \in \Pi_{\mathcal{M}}(\mathtt{w}^{\mathtt{init}})$ such that $\sigma \models \phi$.

Given a context-restricted reaction system with discrete concentrations CR-$\mathcal{C}$ and an rsLTL formula ϕ, *rs* LTL *model checking* is the problem of deciding whether $\mathcal{M} \models \phi$, where $\mathcal{M}$ is the model for CR-$\mathcal{C}$. The existential rsLTL model checking problem is the problem of deciding whether $\mathcal{M} \models_{\exists} \phi$.

To carry out the complexity analysis of rsLTL, we need additional technical notions and notations.

Definition 13 Let $\alpha \in BE(S)$. Then $|\alpha|$ – the size of α – is defined as follows:

- if $\alpha = \mathit{true}$, then $|\alpha| = 1$,
- if $\alpha = e_1 \sim c$ or $\alpha = e_1 \sim e_2$, where $e_1, e_2 \in S$ and $c \in \mathbb{N}$, then $|\alpha| = 2$,
- if $\alpha = \neg\alpha'$, then $|\alpha| = |\alpha'| + 1$,
- if $\alpha = \alpha' \vee \alpha''$, then $|\alpha| = |\alpha'| + |\alpha''|$.

Definition 14 Let ϕ be an rsLTL formula. Then $\mathsf{op}(\phi)$ – the number of operators used in ϕ – is defined as follows:

- if $\phi = \alpha$, where $\alpha \in BE(S)$, then $\mathsf{op}(\phi) = 0$,
- if $\phi = \mathsf{X}_{\alpha}\phi_1$, then $\mathsf{op}(\phi) = \mathsf{op}(\phi_1) + 1$,
- if $\phi \in \{\phi_1 \vee \phi_2, \phi_1 \wedge \phi_2, \phi_1 \mathsf{U}_{\alpha} \phi_2, \phi_1 \mathsf{R}_{\alpha} \phi_2\}$, then $\mathsf{op}(\phi) = \mathsf{op}(\phi_1) + \mathsf{op}(\phi_2) + 1$.

Definition 15 Let ϕ be an rsLTL formula. Then $\mathsf{mbe}(\phi)$ – the size of the largest expression $\alpha \in BE(S)$ with respect to $|\alpha|$ used in ϕ – is defined as follows:

- if $\phi = \alpha$, where $\alpha \in BE(S)$, then $\mathsf{mbe}(\phi) = 0$,
- if $\phi \in \{\phi_1 \vee \phi_2, \phi_1 \wedge \phi_2\}$, then $\mathsf{mbe}(\phi) = \mathit{max}(\{\mathsf{mbe}(\phi_1), \mathsf{mbe}(\phi_2)\})$,
- if $\phi = \mathsf{X}_{\alpha}\phi_1$, then $\mathsf{mbe}(\phi) = \mathit{max}(\{|\alpha|, \mathsf{mbe}(\phi_1)\})$,
- if $\phi \in \{\phi_1 \mathsf{U}_{\alpha} \phi_2, \phi_1 \mathsf{R}_{\alpha} \phi_2\}$, then $\mathsf{mbe}(\phi) = \mathit{max}(\{|\alpha|, \mathsf{mbe}(\phi_1), \mathsf{mbe}(\phi_2)\})$.

4.1 rsLTL as LTL

We now demonstrate how the model checking problem for rsLTL can be translated into LTL model checking.

Let ϕ be an rsLTL formula and CR-$\mathcal{C} = (\mathcal{C}, \mathfrak{A})$ be a context-restricted reaction system with discrete concentrations, where $\mathfrak{A} = (\mathcal{Q}, \mathfrak{q}^{init}, R)$ and $R = \{t_1, \ldots, t_m\}$. The aim of the translation is to define a context-restricted reaction system with discrete concentrations CR-$\mathcal{C}'$ and an LTL formula ϕ' such that $\mathcal{M} \models \phi$ iff $\mathcal{M}' \models \phi'$, where $\mathcal{M}$ and $\mathcal{M}'$ are the models for CR-$\mathcal{C}$ and CR-$\mathcal{C}'$, respectively.

The translation defines a context-restricted reaction system with discrete concentrations that for each context entity provided by the context automaton produces the corresponding entity. These entities indicate which context entities were provided immediately before the system transitioned to a given state, i.e., via which context the current state was reached. Then, the original rsLTL formula is translated into an LTL formula where all the multiset expressions restricting contexts are expressed as constraints on the states.

First, we define a set of entities corresponding to the transitions of $\mathfrak{A}$:

$$S_{\star} = \{\star_i \mid t_i \in R\}.$$

Next, we define the set of the entities that are used to distinguish the ordinary entities from the entities that were supplied by the context:

$$S_c = \{\tilde{e} \mid e \in S\}.$$

Note that this set can be made smaller by only selecting the entities that are used in the context automaton, i.e.,

$$S_c = \{\tilde{e} \mid (\exists t \in R)(t = (\mathfrak{q} \xrightarrow{\mathbf{c}} \mathfrak{q}') \wedge e \in carr(\mathbf{c}))\}$$

Let CR-$\mathcal{C}' = (\mathcal{C}', \mathfrak{A}')$, then $\mathcal{C}' = (S', A')$, where $S' = S \cup S_{\star} \cup S_c$. The set of reactions of $\mathcal{C}'$ is defined as $A' = A \cup A_c$ where the set A_c consists of the following reactions defined for each transition $t_i \in R$:

$$(\{\star_i \mapsto 1\}, \varnothing_{S'}, \{\tilde{e} \mapsto \mathbf{c}(e) \mid t_i = (\mathfrak{q} \xrightarrow{\mathbf{c}} \mathfrak{q}') \wedge e \in carr(\mathbf{c})\}).$$

The context automaton uses a modified transition relation and is defined as $\mathfrak{A}' = (\mathcal{Q}, \mathfrak{q}^{init}, R')$ where:

$$R' = \{\mathfrak{q} \xrightarrow{\mathbf{c}} \mathfrak{q}' \mid (\exists t_i \in R)(t_i = (\mathfrak{q}_i \xrightarrow{\mathbf{c}_i} \mathfrak{q}'_i) \wedge \mathbf{c} = ⩕(\mathbf{c}_i, \{\star_i \mapsto 1\}))\}.$$

Finally, we define the translation of multiset expressions interpreted over the context sets with $repl(\mathfrak{a})$ denoting the expression $\mathfrak{a}$, where each occurrence of $e \in S$ in $\mathfrak{a}$ is replaced with $\tilde{e}$. For an rsLTL formula ϕ we define the translation $tr^{\text{LTL}}(\phi)$ such that $tr^{\text{LTL}}(\phi)$ is an LTL formula.

- if $\phi = \mathfrak{a}$ and $\mathfrak{a} \in BE(S)$, then $tr^{\text{LTL}}(\phi) = \phi$,
- if $\phi = \phi_1 \vee \phi_2$, then $tr^{\text{LTL}}(\phi) = tr^{\text{LTL}}(\phi_1) \vee tr^{\text{LTL}}(\phi_2)$,
- if $\phi = \phi_1 \wedge \phi_2$, then $tr^{\text{LTL}}(\phi) = tr^{\text{LTL}}(\phi_1) \wedge tr^{\text{LTL}}(\phi_2)$,
- if $\phi = \mathsf{X}_{\mathfrak{a}}\phi_1$, then $tr^{\text{LTL}}(\phi) = \mathsf{X}(repl(\mathfrak{a}) \wedge tr^{\text{LTL}}(\phi_1))$,
- if $\phi = \phi_1 \mathsf{U}_{\mathfrak{a}} \phi_2$, then $tr^{\text{LTL}}(\phi) = (repl(\mathfrak{a}) \wedge tr^{\text{LTL}}(\phi_1))\mathsf{U} tr^{\text{LTL}}(\phi_2)$,
- if $\phi = \phi_1 \mathsf{R}_{\mathfrak{a}} \phi_2$, then $tr^{\text{LTL}}(\phi) = tr^{\text{LTL}}(\phi_1)\mathsf{R}\ (repl(\mathfrak{a}) \wedge tr^{\text{LTL}}(\phi_2))$.

The construction of $S_\star$, A_c, and $\mathfrak{A}'$ requires $\mathcal{O}(|R|)$ steps, while the set S_c can be built in $\mathcal{O}(|S|)$ steps. The translation of the formula ϕ runs in $\mathcal{O}(\mathsf{op}(\phi) \cdot \mathsf{mbe}(\phi))$, since for each temporal operator the associated expression $\mathfrak{a} \in BE(S)$ needs to be re-written using the entities of S_c, each rsLTL operator has a constant number of arguments, and $\mathsf{mbe}(\phi)$ is the largest $\mathfrak{a}$ used in ϕ. Therefore, the translation runs in $\mathcal{O}(|R| + |S| + \mathsf{op}(\phi) \cdot \mathsf{mbe}(\phi))$.

The verification method presented in Sect. 5 encodes context-restricted reaction systems with discrete concentrations and rsLTL directly and does not use the above translation.

4.2 Complexity analysis

In this section, our aim is to show that rsLTL model checking is PSPACE-complete. First, we define the reachability problem for context-restricted reaction system with discrete concentrations.

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

Lemma 1 *The reachability problem for context-restricted reaction systems with discrete concentrations is PSPACE-hard.*

Proof The proof is by reduction of a PSPACE-complete problem to the reachability problem for context-restricted reaction systems with discrete concentrations.

Let us take the problem of reachability of configurations of polynomial-space Turing machines, which is a PSPACE-complete problem (Papadimitriou 1994).

The presented reduction is similar to those in Formenti et al. (2014) and Dennunzio et al. (2016).

Let $TM = (Q, \Sigma, \Gamma, \delta, q_I, q_F)$ be a deterministic single-tape Turing machine, where $Q = \{q_1, \ldots, q_m\}$ is the set of states, $\Sigma = \{0, 1\}$ is the input alphabet, $\Gamma = \Sigma \cup \{\triangleright\}$ is the tape alphabet, and $q_I, q_F \in Q$ are, respectively, the initial and the accepting state of TM.

The transition function is defined as $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{-1, 0, 1\}$.

The input string always starts with the symbol $\triangleright$ that is never written or changed by TM, i.e., for all the transitions $\delta(q, \gamma) = (q, \gamma', d)$ we have $\gamma = \triangleright$ iff $\gamma' = \triangleright$. Moreover, if $\gamma = \triangleright$, then the tape head moves right, i.e., $d = 1$.

A *configuration* of TM is a tuple $C = (q, x, pos)$, where $q \in Q$ is a state, $x \in \Sigma^*$ is the tape content and pos is the head position. The *initial configuration* of TM is defined as $C_init = (q_I, \gamma_1 \ldots \gamma_N, 1)$, where the current state of TM is q_I, the tape head is at position 1, and $\gamma_1 \ldots \gamma_N \in \Sigma^*$ is an input string of length N. The initial tape head position is 1 as the position 0 is reserved for the symbol $\triangleright$ that is used to mark the boundary of the tape.

We also assume TM is polynomially space-bounded (Baier and Katoen 2008), i.e., there is a polynomial P such that for an input $\gamma_1 \ldots \gamma_N \in \Sigma^*$ the machine visits at most the first $P(N)$ cells of the tape. We assume $P(N) \geq N$.

The aim of this construction is to define CR-$\mathcal{C} = ((S, A), \mathfrak{A})$ that preserves the following property: a configuration C is reachable in TM from C_init iff

the state of CR-$\mathcal{C}$ corresponding to the configuration C is reachable in CR-$\mathcal{C}$.

We begin by introducing the background set $S = \{e, h, w_1, \ldots, w_{P(N)}\}$. The concentration levels of e are used to encode the states of Q: we define a bijection $\mathtt{c} : \mathtt{Q} \longrightarrow \{\mathtt{1}, \ldots, \mathtt{m}\}$ assigning concentration levels to the states, i.e., $\{e \mapsto \mathtt{c}(\mathtt{q})\}$ encodes $q \in Q$. The concentration of h denotes the position of the tape head (the concentration level values for h are taken from the set $\{0, \ldots, P(N) + 1\}$). The entities $w_1, \ldots, w_{P(N)}$ encode the symbols on the tape, i.e., tape contents.

To encode empty tape positions we use a concentration level $j \notin \Sigma$, i.e., $\{w_i \mapsto j\}$ encodes that the i^{th} tape position is empty.

Configurations of TM. Let $n \leq P(N)$. The configuration $C = (q, \gamma_1 \ldots \gamma_n, pos)$ is encoded in CR-$\mathcal{C}$ as follows:

$$conf_{\textsc{cr}-\mathcal{C}}(C) = \left(\{e \mapsto \mathtt{c}(\mathtt{q}), h \mapsto pos\} \cup \bigcup_{j \in \{1, \ldots, n\}} \{w_j \mapsto \gamma_j\} \right). \tag{1}$$

Next, we define the reactions of A that aim to emulate the steps of TM in CR-$\mathcal{C}$.

Transition function. For each transition $\delta(q,\gamma) = (r,\gamma',d)$ and each tape head position $pos \in \{1,\ldots,P(N)\}$ we define the following reaction:

$$(\{e \mapsto \mathtt{c}(\mathtt{q}), h \mapsto pos, w_{pos} \mapsto \gamma\},\\ \{e \mapsto (\mathtt{c}(\mathtt{q})+1), h \mapsto (pos+1), w_{pos} \mapsto (1-\gamma)\},\\ \{e \mapsto c(r), h \mapsto (pos+d), w_{pos} \mapsto \gamma'\}).$$

The reactants encode the concentration levels encoding the state, the head position, and the symbol for the transition to be enabled. The inhibitors are used to enforce exact concentration levels by not allowing concentrations higher than specified by the reactants. Finally, the products encode the successor state, the new head position, and the symbol written on the tape. The case for $pos = 0$ (tape boundary) is handled later.

Tape contents. For all $i, pos \in \{1,\ldots,P(N)\}$ such that $i \neq pos$, we define reactions that preserve the i^{th} symbol of the tape if the tape head is at a different position pos.

$$(\{w_i \mapsto 1, h \mapsto pos\}, \{h \mapsto (pos+1)\}, \{w_i \mapsto 1\}).$$

Tape boundary. If the tape reaches $\triangleright$, then the enforced move right is encoded using the following reaction:

$$(\{e \mapsto 1\}, \{h \mapsto 1\}, \{h \mapsto 1\}).$$

The reaction is enabled in any state $q \in Q$ as there is no upper bound on the concentration level of e.

We do not need to handle the remaining boundary of the tape since we assume *TM* visits only the first $P(N)$ cells of the tape; however if the head reaches the position $P(N)+1$ the computation halts since no transitions are enabled when the tape head at the position $P(N)+1$.

Context automaton. We define $\mathfrak{A} = (\{\mathfrak{q}_0, \mathfrak{q}_1\}, \mathfrak{q}_0, R)$ with the following transition relation:

$$R = \{\mathfrak{q}_0 \xrightarrow{C_i nit} \mathfrak{q}_1, \mathfrak{q}_1 \xrightarrow{\varnothing_S} \mathfrak{q}_1\}.$$

The role of the context automaton is to provide the encoded initial configuration as the initial context and allow for the subsequent computation steps by providing transitions with empty contexts.

The reduction runs in polynomial time since the encoding of the transition function requires $\mathcal{O}(P(N)\cdot|\delta|)$ reactions[1] and the encoding of the preservation of the tape contents requires $\mathcal{O}(P(N)^2)$ reactions. The construction of $\textsc{cr}\text{-}\mathcal{C}$ ensures that a configuration C is reachable in *TM* iff $conf_{\textsc{cr}-\mathcal{C}}(C)$ is reachable in $\textsc{cr}\text{-}\mathcal{C}$. Therefore, the reachability problem for context-restricted reaction systems with discrete concentrations is PSPACE-hard. □

The reachability of $\mathbf{d} \in \mathcal{B}(S)$ can be expressed in terms of existential rsLTL model checking using the following formula:

$$\mathsf{F}\left(\bigwedge_{e \in carr(\mathbf{d})} (e = \mathbf{d}(e))\right).$$

Therefore, from Lemma 1 we also get the following result.

Corollary 1 *The existential rsLTL model checking problem is PSPACE-hard.*

Let ϕ be an rsLTL formula, $\textsc{cr}\text{-}\mathcal{C}$ be a context-restricted reaction system with discrete concentrations, and $\mathcal{M}$ be the model for $\textsc{cr}\text{-}\mathcal{C}$. The existential decision problem yields *true* if $\mathcal{M} \models_{\exists} \phi$, and *false* otherwise. Since $\mathcal{M} \models \phi$ iff $\mathcal{M} \not\models_{\exists} \neg\phi$, the universal decision problem yields *true* iff the existential variant of the problem for $\neg\phi$ yields *false*.

Therefore, from the PSPACE-hardness of the existential rsLTL model checking problem we get PSPACE-hardness of its universal variant. This follows from the fact that the complement of a PSPACE-hard problem is also PSPACE-hard (Papadimitriou 1994).

Corollary 2 *The rsLTL model checking problem is PSPACE-hard.*

Now, we can give the complexity result of the model checking problem for rsLTL.

Lemma 2 *The rsLTL model checking problem is in PSPACE.*

Proof Since there is a polynomial translation of the rsLTL model checking problem to the LTL model checking problem, it is sufficient to show the LTL model checking problem for context-restricted reaction systems is in PSPACE.

The proof follows the same reasoning as the one for Lemma 5.47 in Baier and Katoen (2008). It gives a nondeterministic polynomial space-bounded algorithm solving the existential LTL model checking problem.

The algorithm nondeterministically guesses a path in $TS \otimes \mathcal{G}_{\phi}$, i.e., in the product of, respectively, a finite transition system and a generalised nondeterministic Büchi automaton for the verified LTL formula ϕ.

However, here *TS* is not given as the input and needs to be obtained from context-restricted reaction system. In fact, it only must be possible to obtain a successor state in polynomial space and a method for that is demonstrated in the proof of Lemma 6.2 for CTL in Męski et al. (2015). □

The next result follows directly from Corollary 2 and Lemma 2.

Theorem 3 *The rsLTL model checking problem is PSPACE-complete.*

[1] The size of the transition function δ is denoted by $|\delta|$.

4.3 Bounded semantics

Bounded model checking has proven its usefulness in a range of successful applications (Kroening and Strichman 2016). Motivated by this, we focus on the bounded model checking approach defined for finite prefixes of paths. Accordingly, we can limit ourselves to consider only a finite number of states and actions that belong to the prefix of the considered path and check whether specified formula holds there.

Definition 16 (Męski et al. 2017) A path $\sigma = (\mathtt{w}_0, \alpha_0, \mathtt{w}_1, \alpha_1, \ldots)$ is a $(k,\, l)$*-loop* (or k*-loop*) if there exist $k \geq l > 0$ such that $\mathtt{w}_{\mathtt{l}-1} = \mathtt{w}_\mathtt{k}$ and $\sigma = (\mathtt{w}_0, \alpha_0, \ldots, \alpha_{\mathtt{l}-2}, \mathtt{w}_{\mathtt{l}-1})(\alpha_{\mathtt{l}-1}, \mathtt{w}_\mathtt{l}, \alpha_\mathtt{l}, \ldots, \alpha_{\mathtt{k}-1}, \mathtt{w}_\mathtt{k})^{\omega}$.

Note that considering a sufficiently large and finite fragment of a path we may conclude that a given formula holds in that path. To address this issue, the bounded semantics for rsLTL is defined for finite path prefixes, and we define a satisfiability relation for a given path considering only its first k states and $k-1$ actions.

Definition 17 (Męski et al. 2017) The fact that a formula ϕ holds in a path σ with bound $k \in \mathbb{N}$ is denoted by $\sigma \models^k \phi$. Then, $\sigma \models^k \phi$ if:

- σ is a $(k,\, l)$-loop for some $0 < l \leq k$ and $\sigma \models \phi$, or
- $\sigma \models_{nl} \phi$, where:

$$
\begin{array}{lll}
\sigma \models_{nl} \mathfrak{a} & \text{iff} & \sigma_s(0) \models_b \mathfrak{a}, \\
\sigma \models_{nl} \phi_1 \wedge \phi_2 & \text{iff} & \sigma \models_{nl} \phi_1 \text{ and } \sigma \models_{nl} \phi_2, \\
\sigma \models_{nl} \phi_1 \vee \phi_2 & \text{iff} & \sigma \models_{nl} \phi_1 \text{ or } \sigma \models_{nl} \phi_2, \\
\sigma \models_{nl} \mathsf{X}_{\mathfrak{a}}\phi & \text{iff} & k > 0,\ \sigma_a(0) \models_b \mathfrak{a}, \text{ and } \sigma^1 \models_{nl} \phi, \\
\sigma \models_{nl} \phi_1 \mathsf{U}_{\mathfrak{a}} \phi_2 & \text{iff} & (\exists 0 \leq j \leq k)\big(\sigma^j \models_{nl} \phi_2 \\
 & & \text{and } (\forall 0 \leq l < j)(\sigma^l \models_{nl} \phi_1 \text{ and } \sigma_a(l) \models_b \mathfrak{a})\big) \\
\sigma \models_{nl} \phi_1 \mathsf{R}_{\mathfrak{a}} \phi_2 & \text{iff} & (\exists 0 \leq j \leq k)\big(\sigma^j \models_{nl} \phi_1 \text{ and } ((\forall 0 \leq l \leq j)(\sigma^l \models_{nl} \phi_2) \\
 & & \text{and } (\forall 0 \leq l < j)(\sigma_a(l) \models_b \mathfrak{a}))\big)
\end{array}
$$

For the sake of completeness, we now recall the theorem related with the compatibility of the bounded semantics with the general semantics of rsLTL.

Theorem 4 *(Męski et al. 2017) Let ϕ be an rsLTL formula and $\mathcal{M}$ be a model. Then, $\mathcal{M} \models_{\exists} \phi$ iff there exists $k \in \mathbb{N}$ such that $\mathcal{M} \models^k_{\exists} \phi$.*

5 SMT-based encoding

In this section, we explain how the bounded model checking problem for rsLTL can be mapped to Satisfiability Modulo Theory (SMT) (Kroening and Strichman 2016) within the context of integer arithmetic theory. SMT presents a broader perspective on the Boolean satisfiability problem by allowing certain functions and predicate symbols to have interpretations based on the foundational theory they stem from.

Let cr-$\mathcal{C} = (\mathcal{R}, \mathfrak{A})$ be a context-restricted reaction system with discrete concentrations where $\mathcal{R} = (S, A)$ and $\mathfrak{A} = (\mathcal{Q}, \mathtt{q}^{init}, R)$, and let $\mathcal{M}$ be the model for cr-$\mathcal{C}$ and an rsLTL formula ϕ. For an integer $k \geq 0$ we construct a formula $[\mathcal{M}, \phi, k]$ such that $\mathcal{M} \models^k_{\exists} \phi$ iff $[\mathcal{M}, \phi, k]$ is satisfiable. We encode all the paths of the model $\mathcal{M}$ that are bounded with k. The entities of S are denoted by $e_1, \ldots, e_m$, where $m = |S|$. For each $i \in \{0, \ldots, k\}$ we introduce the following sets of positive integer variables used in the encoding:

$$\mathbf{P}_i = \{\mathtt{p}_{\mathtt{i},1}, \ldots, \mathtt{p}_{\mathtt{i},\mathtt{m}}\} \ \ \textit{and} \ \ \mathbf{P}^{\mathcal{E}}_\mathtt{i} = \{\mathtt{p}^{\mathcal{E}}_{\mathtt{i},1}, \ldots, \mathtt{p}^{\mathcal{E}}_{\mathtt{i},\mathtt{m}}\}.$$

Let σ be a path of $\mathcal{M}$. Then, $\mathtt{p}_{\mathtt{i},1}, \ldots, \mathtt{p}_{\mathtt{i},\mathtt{m}}$ and $\mathtt{p}^{\mathcal{E}}_{\mathtt{i},1}, \ldots, \mathtt{p}^{\mathcal{E}}_{\mathtt{i},\mathtt{m}}$ encode $\sigma_b(i)$ and $\sigma_a(i)$, respectively. We also introduce the variables $\mathtt{q}_0, \ldots, \mathtt{q}_\mathtt{k}$ which are used to encode the locations of $\mathfrak{A}$. The location $\sigma_{ca}(i)$ of the context automaton is then encoded with $\mathtt{q}_\mathtt{i}$. Then, we define $\overline{\mathbf{p}}_i = (\mathtt{p}_{\mathtt{i},1}, \ldots, \mathtt{p}_{\mathtt{i},\mathtt{m}})$ and $\overline{\mathbf{p}}^{\mathcal{E}}_i = (\mathtt{p}^{\mathcal{E}}_{\mathtt{i},1}, \ldots, \mathtt{p}^{\mathcal{E}}_{\mathtt{i},\mathtt{m}})$. With $\overline{\mathbf{p}}_i[j]$ and $\overline{\mathbf{p}}^{\mathcal{E}}_i[j]$ we denote, respectively, $\mathtt{p}_{\mathtt{i},\mathtt{j}}$ and $\mathtt{p}^{\mathcal{E}}_{\mathtt{i},\mathtt{j}}$. Then, we also define $\mathbf{P} = \bigcup_{i=0}^{k} \mathbf{P}_i$ and $\mathbf{P}^{\mathcal{E}} = \bigcup_{i=0}^{k} \mathbf{P}^{\mathcal{E}}_i$.

We define the subsequent functions that associate entities within the background set with the corresponding variables used in the encoding: for all $0 \leq i \leq k$ we define $\mathtt{t}_\mathtt{i} : \mathtt{S} \to \mathbf{P}_\mathtt{i}$ and $\mathtt{t}^{\mathcal{E}}_\mathtt{i} : \mathtt{S} \to \mathbf{P}^{\mathcal{E}}_\mathtt{i}$ such that $\mathtt{t}_\mathtt{i}(\mathtt{e}_\mathtt{j}) = \mathtt{p}_{\mathtt{i},\mathtt{j}}$, $\mathtt{t}^{\mathcal{E}}_\mathtt{i}(\mathtt{e}_\mathtt{j}) = \mathtt{p}^{\mathcal{E}}_{\mathtt{i},\mathtt{j}}$ for all $1 \leq j \leq m$. To map states of the context automaton to the corresponding natural values used in the encoding we use the function $\mathtt{e} : \mathcal{Q} \to \{\mathtt{0}, \ldots, |\mathcal{Q}| - \mathtt{1}\}$. The set of the reactions that are capable of producing $e \in S$ is $\mathit{Prod}(e) = \{a \in A \mid \mathbf{p}_a(e) > 0\}$. Let f_1, f_2, f_3 be expressions over $\mathbf{P} \cup \mathbf{P}^{\mathcal{E}}$, then we define the *if-then-else* operator:

$$f_1 \to f_2 \mid f_3 = (f_1 \wedge f_2) \vee (\neg f_1 \wedge f_3).$$

To define the SMT encoding of the paths we need auxiliary functions that correspond to elements of the encoding.

The encoding of the reactions is defined in two steps: (i) we define a formula encoding the condition for when a reaction is enabled, and (ii) a formula encoding what that reaction produces when it is enabled.

The enabledness of a reaction $a \in A$ is encoded as follows:

$$\mathsf{En}_a(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}^{\mathcal{E}}_i) = \bigwedge_{e \in S} \big(\mathtt{t}_\mathtt{i}(\mathtt{e}) \geq \mathbf{r}_\mathtt{a}(\mathtt{e}) \vee \mathtt{t}^{\mathcal{E}}_\mathtt{i}(\mathtt{e}) \geq \mathbf{r}_\mathtt{a}(\mathtt{e})\big)$$
$$\wedge \bigwedge_{e \in S} \big(\mathtt{t}_\mathtt{i}(\mathtt{e}) < \mathbf{i}_\mathtt{a}(\mathtt{e}) \wedge \mathtt{t}^{\mathcal{E}}_\mathtt{i}(\mathtt{e}) < \mathbf{i}(\mathtt{e})\big).$$

The formula encodes the conditions for $a \in A$ to be enabled, i.e., in the current state and in the context, the concentration levels of the reactants specified in $\mathbf{r}_a$ need to be sufficient and the concentration levels of all its inhibitors need to be below the threshold specified by $\mathbf{i}_a$.

To encode the produced entity concentration level of an entity $e \in S$ we take all the reactions that have e in their products, i.e., all the reactions of $Prod(e)$ and order them with respect to the produced concentration levels of e. Let $a_1, a_2, \ldots, a_w \in Prod(e)$ and assume $\mathbf{p}_{a_j} \leq \mathbf{p}_{a_{j+1}}$ for all $1 \leq j < w$. First, we encode the produced concentration level of entity e where $j \in \{1, \ldots, w\}$ when there exist reactions producing e, i.e., $w > 0$, and at least one such reaction is enabled (the remaining cases are handled later). The encoding is defined as follows:

$$\mathsf{C}_e^j(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}, \overline{\mathbf{p}}_{i+1}) = \begin{cases} \mathsf{En}_{a_j}(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}) \rightarrow (\mathtt{t}_{\mathtt{i+1}}(\mathtt{e}) = \mathbf{p}_{\mathtt{a_j}}) \mid \mathsf{C}_{\mathtt{e}}^{\mathtt{j-1}}(\overline{\mathbf{p}}_{\mathtt{i}}, \overline{\mathbf{p}}_{\mathtt{i}}^{\mathcal{E}}, \overline{\mathbf{p}}_{\mathtt{i+1}}) & \text{if } 1 < j \leq w, \\ \mathsf{En}_{a_j}(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}) \wedge (\mathtt{t}_{\mathtt{i+1}}(\mathtt{e}) = \mathbf{p}_{\mathtt{a_j}}) & \text{if } j = 1. \end{cases}$$

In the definition of $\mathsf{C}_e^j(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}, \overline{\mathbf{p}}_{i+1})$ with the use of the *if-then-else* operator we always encode the highest available concentration of e if the corresponding reaction producing e with that concentration level is enabled. Finally, we define the complete entity concentration encoding for all the reactions:

$$\mathsf{C}_e(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}, \overline{\mathbf{p}}_{i+1}) = \begin{cases} \mathtt{t}_{\mathtt{i+1}}(\mathtt{e}) = 0 & \text{if } w = 0, \\ \mathsf{C}_e^w(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}, \overline{\mathbf{p}}_{i+1}) \vee \left(\left(\bigwedge_{a \in Prod(e)} \neg \mathsf{En}_a(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}) \right) \wedge (\mathtt{t}_{\mathtt{i+1}}(\mathtt{e}) = 0) \right) & \text{if } w > 0. \end{cases}$$

In the above, we handle the remaining cases: (1) when $w = 0$, i.e., if there are no reactions producing e, and (2) there are reactions producing e or (3) none of them are enabled. In (1) and (3) the entity e is produced with the concentration level equal 0. In (2) the encoding of $\mathsf{C}_e^w(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}, \overline{\mathbf{p}}_{i+1})$ ensures production of e and selection of its maximal concentration level by starting the encoding from a_w, i.e., the reaction producing the highest concentration level of e.

Let $\mathbf{c} \in \mathcal{B}(S)$ be a multiset of context entities. The encoding of $\mathbf{c}$ is defined as follows:

$$\mathsf{Ct}_{\mathbf{c}}(\overline{\mathbf{p}}_i^{\mathcal{E}}) = \bigwedge_{e \in S} (\mathtt{t}_{\mathtt{i}}^{\mathcal{E}}(\mathtt{e}) = \mathbf{c}(\mathtt{e})).$$

The encoding of the transition relation of the context automaton is a disjunction of the encodings for each transition:

$$\mathsf{Tr}_{\mathfrak{A}}(\mathtt{q}_{\mathtt{i}}, \overline{\mathbf{p}}_{\mathtt{i}}^{\mathcal{E}}, \mathtt{q}_{\mathtt{i+1}}) = \bigvee_{(\mathtt{q}, \mathbf{c}, \mathtt{q}') \in \mathbf{R}} (\mathtt{q}_{\mathtt{i}} = \mathtt{e}(\mathtt{q}) \wedge \mathsf{Ct}_{\mathbf{c}}(\overline{\mathbf{p}}_{\mathtt{i}}^{\mathcal{E}}) \wedge \mathtt{q}_{\mathtt{i+1}} = \mathtt{e}(\mathtt{q}')).$$

We build a conjunction by combining the generated concentration levels for all entities and the transition relation of the context automaton. This conjunction serves to encode the transition relation of the model:

$$\mathsf{Tr}_{\text{CR}-\mathcal{C}}(\overline{\mathbf{p}}_i, \mathtt{q}_{\mathtt{i}}, \overline{\mathbf{p}}_{\mathtt{i}}^{\mathcal{E}}, \overline{\mathbf{p}}_{\mathtt{i+1}}, \mathtt{q}_{\mathtt{i+1}}) = \left(\bigwedge_{e \in S} \mathsf{C}_e(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_i^{\mathcal{E}}, \overline{\mathbf{p}}_{i+1}) \right) \wedge \mathsf{Tr}_{\mathfrak{A}}(\mathtt{q}_{\mathtt{i}}, \overline{\mathbf{p}}_{\mathtt{i}}^{\mathcal{E}}, \mathtt{q}_{\mathtt{i+1}}).$$

To encode the initial state of the model where all the concentration levels are set to zero and the context automaton is in its initial state we define the following formula:

$$\mathsf{Init}(\overline{\mathbf{p}}_i, \mathtt{q}_{\mathtt{i}}) = \bigwedge_{\mathtt{e} \in \mathtt{S}} (\mathtt{t}_{\mathtt{i}}(\mathtt{e}) = 0) \wedge (\mathtt{q}_{\mathtt{i}} = \mathtt{e}(\mathtt{q}^{\mathtt{init}})).$$

We encode the paths of $\mathcal{M}$ that are bounded with k by unrolling the transition relation up to k and combining it with the encoding of the initial state of the model as follows:

$$\mathsf{Paths}^k = \mathsf{Init}(\overline{\mathbf{p}}_0, \mathtt{q}_0) \wedge \bigwedge_{\mathtt{i}=0}^{\mathtt{k}-1} \mathsf{Tr}_{\text{CR}-\mathcal{C}}(\overline{\mathbf{p}}_{\mathtt{i}}, \mathtt{q}_{\mathtt{i}}, \overline{\mathbf{p}}_{\mathtt{i}}^{\mathcal{E}}, \overline{\mathbf{p}}_{\mathtt{i+1}}, \mathtt{q}_{\mathtt{i+1}}).$$

Following the fixed point encoding for LTL in Biere et al. (2006), we translate the rsLTL formulae into an SMT encoding.

In rsLTL, instead of employing propositional variables found in conventional LTL formulas, we utilize multiset expressions. Let $\mathfrak{a}$ be a multiset expression. By $enc_i^b(\mathfrak{a})$ and $enc_i^{ct}(\mathfrak{a})$ we denote the encoding of $\mathfrak{a}$ using the variables of $\overline{\mathbf{p}}_i$ and $\overline{\mathbf{p}}_i^{\mathcal{E}}$, respectively. The former refers to states of $\mathcal{C}$, while the latter refers to actions (or contexts). We are defining a translation into SMT , hence the encodings of multisets can be defined in a straightforward way. We introduce an integer variable $\mathtt{L}$, which is utilized to handle (k, l)-loops as follows. When $\mathtt{L} = \mathtt{l}$ holds for a path then the path is a (k, l)-loop:

$$\mathsf{Loops}^k = \neg(\mathtt{L} = 0) \wedge \bigwedge_{\mathtt{i}=1}^{\mathtt{k}} ((\mathtt{L} = \mathtt{i}) \Rightarrow \mathsf{E}(\overline{\mathbf{p}}_{\mathtt{i-1}}, \mathtt{q}_{\mathtt{i-1}}, \overline{\mathbf{p}}_{\mathtt{k}}, \mathtt{q}_{\mathtt{k}})),$$

where E encodes the equivalence of two states of the model:

$$\mathsf{E}(\overline{\mathbf{p}}_i, \mathtt{q}_{\mathtt{i}}, \overline{\mathbf{p}}_{\mathtt{j}}, \mathtt{q}_{\mathtt{j}}) = (\mathtt{q}_{\mathtt{i}} = \mathtt{q}_{\mathtt{j}}) \wedge \bigwedge_{\mathtt{c}=1}^{\mathtt{m}} (\overline{\mathbf{p}}_{\mathtt{i}}[\mathtt{c}] = \overline{\mathbf{p}}_{\mathtt{j}}[\mathtt{c}]).$$

The encoding of an rsLTL formula ϕ at the position $i \in \{0, \ldots, k\}$ is defined as $[\![\phi]\!]_i^k$. We start with introducing the encoding for propositional formulae:

$[\![\phi]\!]_i^k$	$0 \leq i \leq k$
$[\![\mathfrak{a}]\!]_i^k$	$enc_i^b(\mathfrak{a})$
$[\![\phi_1 \wedge \phi_2]\!]_i^k$	$[\![\phi_1]\!]_i^k \wedge [\![\phi_2]\!]_i^k$
$[\![\phi_1 \vee \phi_2]\!]_i^k$	$[\![\phi_1]\!]_i^k \vee [\![\phi_2]\!]_i^k$

We follow up with the encoding for temporal formulae. Here we follow the fixed point encoding for CTL Clarke et al. (1999) and introduce an auxiliary translation $\langle\langle \phi \rangle\rangle_i^k$ to cope with the until and release operators.

$\llbracket\phi\rrbracket_i^k$	$0 \leq i < k$
$\llbracket\mathsf{X}_{\mathfrak{a}}\phi_1\rrbracket_i^k$	$\llbracket\phi_1\rrbracket_{i+1}^k \wedge enc_i^{ct}(\mathfrak{a})$
$\llbracket\phi_1\mathsf{U}_{\mathfrak{a}}\phi_2\rrbracket_i^k$	$\llbracket\phi_2\rrbracket_i^k \vee \left(\llbracket\phi_1\rrbracket_i^k \wedge (\llbracket\phi_1\mathsf{U}_{\mathfrak{a}}\phi_2\rrbracket_{i+1}^k \wedge enc_i^{ct}(\mathfrak{a}))\right)$
$\llbracket\phi_1\mathsf{R}_{\mathfrak{a}}\phi_2\rrbracket_i^k$	$\llbracket\phi_2\rrbracket_i^k \wedge \left(\llbracket\phi_1\rrbracket_i^k \vee (\llbracket\phi_1\mathsf{R}_{\mathfrak{a}}\phi_2\rrbracket_{i+1}^k \wedge enc_i^{ct}(\mathfrak{a}))\right)$
	$i = k$
$\llbracket\mathsf{X}_{\mathfrak{a}}\phi_1\rrbracket_i^k$	$\bigvee_{j=1}^k((\mathtt{L} = \mathtt{j}) \wedge \llbracket\phi_1\rrbracket_{\mathtt{j}}^{\mathtt{k}}) \wedge \mathtt{enc}_{\mathtt{i}}^{\mathtt{ct}}(\mathfrak{a})$
$\llbracket\phi_1\mathsf{U}_{\mathfrak{a}}\phi_2\rrbracket_i^k$	$\llbracket\phi_2\rrbracket_i^k \vee \left(\llbracket\phi_1\rrbracket_i^k \wedge (\bigvee_{j=1}^k((\mathtt{L} = \mathtt{j}) \wedge \langle\langle\phi_1\mathsf{U}_{\mathfrak{a}}\phi_2\rangle\rangle_{\mathtt{i+1}}^{\mathtt{k}})\right.$ $\left.\wedge enc_i^{ct}(\mathfrak{a}))\right)$
$\llbracket\phi_1\mathsf{R}_{\mathfrak{a}}\phi_2\rrbracket_i^k$	$\llbracket\phi_2\rrbracket_i^k \wedge \left(\llbracket\phi_1\rrbracket_i^k \vee (\bigvee_{j=1}^k((\mathtt{L} = \mathtt{j}) \wedge \langle\langle\phi_1\mathsf{R}_{\mathfrak{a}}\phi_2\rangle\rangle_{\mathtt{i+1}}^{\mathtt{k}})\right.$ $\left.\wedge enc_i^{ct}(\mathfrak{a}))\right)$
$\langle\langle\phi\rangle\rangle_i^k$	$0 \leq i < k$
$\langle\langle\phi_1\mathsf{U}_{\mathfrak{a}}\phi_2\rangle\rangle_i^k$	$\llbracket\phi_2\rrbracket_i^k \vee \left(\llbracket\phi_1\rrbracket_i^k \wedge (\langle\langle\phi_1\mathsf{U}_{\mathfrak{a}}\phi_2\rangle\rangle_{i+1}^k \wedge enc_i^{ct}(\mathfrak{a}))\right)$
$\langle\langle\phi_1\mathsf{R}_{\mathfrak{a}}\phi_2\rangle\rangle_i^k$	$\llbracket\phi_2\rrbracket_i^k \wedge \left(\llbracket\phi_1\rrbracket_i^k \vee (\langle\langle\phi_1\mathsf{R}_{\mathfrak{a}}\phi_2\rangle\rangle_{i+1}^k \wedge enc_i^{ct}(\mathfrak{a}))\right)$
	$i = k$
$\langle\langle\phi_1\mathsf{U}_{\mathfrak{a}}\phi_2\rangle\rangle_i^k$	$\llbracket\phi_2\rrbracket_i^k$
$\langle\langle\phi_1\mathsf{R}_{\mathfrak{a}}\phi_2\rangle\rangle_i^k$	$\llbracket\phi_2\rrbracket_i^k$

We consider separately the cases for $\llbracket\phi\rrbracket_i^k$ when $i = k$. When (k, j)-loop exists ($\mathtt{L} = \mathtt{j}$), the additional transitions for $j \in \{1, \ldots, k\}$ need to be encoded. Unlike the LTL encoding in (Biere et al. 2006), we necessitate that all transitions are subject to constraints imposed by the parameter $\mathfrak{a}$, which is encoded using $enc_i^{ct}(\mathfrak{a})$.

Finally, we reduce the bounded model checking problem for rsLTL to the satisfiability checking, i.e., to verify if $\mathcal{M} \models_{\exists}^k \phi$ we check the satisfiability of the following formula:

$$[\mathcal{M}, \phi, k] = \mathsf{Paths}^k \wedge \mathsf{Loops}^k \wedge \llbracket\phi\rrbracket_0^k.$$

We conclude this section with the following theorem, confirming the correctness of the presented procedure.

Theorem 5 *Let* CR-$\mathcal{C} = (\mathcal{C}, \mathfrak{A})$ *be a context-restricted reaction system with discrete concentrations*, $\mathcal{M}$ *be its model and* ϕ *an rsLTL formula. For any* $k \in \mathbb{N}$, *the formula* $[\mathcal{M}, \phi, k]$ *is satisfiable iff* $\mathcal{M} \models_{\exists}^k \phi$.

Proof Let $k \in \mathbb{N}$.

Then $[\mathcal{M}, \phi, k]$ is satisfiable iff there exists a valuation of the variables used in the encoding such that the formula is satisfied. The valuation then represents the path prefix of a path in $\mathcal{M}$ for which the formula ϕ holds. We first show that Paths^k encodes path prefixes of paths in $\mathcal{M}$. There exists a path σ in $\mathcal{M}$ and σ^k is its prefix of length k iff there exists a valuation representing σ^k that satisfies Paths^k.

Let $i \in \{0, \ldots, k-1\}$. We observe the formula $\mathsf{Tr}_{\text{CR}-\mathcal{C}}$ is satisfied for the valuation encoding $\sigma_s(i)$, $\sigma_a(i)$ and $\sigma_s(i+1)$ iff $\sigma_s(i) \xrightarrow{\sigma_a(i)} \sigma_s(i+1)$. This follows from the encoding of C_e and $\mathsf{Tr}_{\mathfrak{A}}$. Let us recall that $\sigma_s(i) = (\sigma_b(i), \sigma_{ca}(i))$. For an entity $e \in S$ it is clear from the construction that the formula C_e is satisfied iff the valuation encodes the concentration level of the entity e in $\sigma_b(i+1)$ that is produced by the reactions enabled in $\sigma_b(i)$ with the context $\sigma_a(i)$.

The encoding C_e is applied to all $e \in S$, i.e., the valuation must encode in $\sigma_b(i+1)$ the concentration levels of all the entities of S.

The formula $\mathsf{Tr}_{\mathfrak{A}}$ is satisfied iff the valuation encodes a transition $(\mathfrak{q}, \mathbf{c}, \mathfrak{q}') \in R$ such that $\mathfrak{q} = \sigma_{ca}(i)$, $\mathbf{c} = \sigma_a(i)$ and $\mathfrak{q}' = \sigma_{ca}(i+1)$. Then, the formula Paths^k is satisfied iff the valuation encodes a path prefix σ^k such that the state $\sigma_s(0)$ of $\mathcal{M}$ is the initial state and $\sigma_s(i) \xrightarrow{\sigma_a(i)} \sigma_s(i+1)$ for all $i \in \{0, \ldots, k-1\}$.

It remains to be shown that $\mathsf{Loops}^k \wedge \llbracket\phi\rrbracket_0^k$ restricts the valuation so that the formula $[\mathcal{M}, \phi, k]$ is satisfiable iff it encodes a path prefix of a path in which ϕ holds. To show this we apply the same reasoning as for LTL in (Biere et al. 2006, Theorem 3.1), which follows by induction on the structure of the LTL formula. □

6 Parametric reaction systems

In *parametric reaction systems* reactions can be defined partially, i.e., reactants, inhibitors, and products can be replaced with *parameters*.

Definition 18 (parametric reaction system) A *parametric reaction system* is a triple $\mathcal{P} = (S, P, A)$, where:

- S is a finite *background* set,
- P is a finite set of elements called *parameters*, and
- A is a nonempty finite set of *parametric reactions* over S, where each parametric reaction is a triple $a = (\mathfrak{r}, \mathfrak{i}, \mathfrak{p})$ such that $\mathfrak{r}, \mathfrak{i}, \mathfrak{p} \in \mathcal{B}(S) \cup P$.

The elements $\mathfrak{r}$, $\mathfrak{i}$, and $\mathfrak{p}$ are respectively denoted by $\mathfrak{r}_a$, $\mathfrak{i}_a$, and $\mathfrak{p}_a$ and called the *reactants*, *inhibitors*, and *products* of parametric reaction a.

Definition 19 Let $\mathcal{P} = (S, P, A)$ be a parametric reaction system. A *parameter valuation* of $\mathcal{P}$ is a function $\mathtt{v} : \mathtt{P} \cup \mathcal{B}(\mathtt{S}) \to \mathcal{B}(\mathtt{S})$ such that $\mathtt{v}(\mathfrak{b}) = \mathfrak{b}$ if $\mathfrak{b} \in \mathcal{B}(S)$.

We also write $\mathfrak{b}^{\leftarrow\mathtt{v}}$ for $\mathtt{v}(\mathfrak{b})$. The set of all the parameter valuations for $\mathcal{P}$ is denoted by $\mathtt{PV}_{\mathcal{P}}$. Let $\mathtt{v} \in \mathtt{PV}_{\mathcal{P}}$. For any subset $X \subseteq A$ of reactions of $\mathcal{P}$ we define:

$$X^{\leftarrow\mathtt{v}} = \{(a_{\mathfrak{r}}^{\leftarrow\mathtt{v}}, a_{\mathfrak{i}}^{\leftarrow\mathtt{v}}, a_{\mathfrak{p}}^{\leftarrow\mathtt{v}}) \mid a \in X\}.$$

By $\mathcal{P}^{\leftarrow\mathtt{v}}$ we denote the structure $(S, A^{\leftarrow\mathtt{v}})$ where all the parameters in A are substituted according to the parameter valuation $\mathtt{v}$. We say that $\mathtt{v} \in \mathtt{PV}_{\mathcal{P}}$ is a *valid parameter* *valuation* if $\mathcal{P}^{\leftarrow \mathtt{v}}$ yields a reaction system with discrete concentrations.

Definition 20 (context-restricted parametric reaction system) A *context-restricted parametric reaction system* is a pair $\textsc{cr}\text{-}\mathcal{P} = (\mathcal{P}, \mathfrak{A})$ such that $\mathcal{P} = (S, P, A)$ is a parametric reaction system and $\mathfrak{A} = (\mathcal{Q}, \mathtt{q}^{init}, R)$ is a context automaton over S.

For $\mathtt{v} \in \mathtt{PV}_{\mathcal{P}}$ we define $\textsc{cr}\text{-}\mathcal{P}^{\leftarrow \mathtt{v}} = (\mathcal{P}^{\leftarrow \mathtt{v}}, \mathfrak{A})$.

Example 2 We consider a simple parametric reaction system for a simplified abstract genetic regulatory system based on Ehrenfeucht et al. (2012).

The system contains two (abstract) genes x and y expressing proteins X and Y, respectively, and a protein complex Q formed by X and Y.

The background set is defined as $S = \{x, \widehat{x}, X, y, \widehat{y}, Y, h, Q\}$, where $\widehat{x}$ and $\widehat{y}$ denote RNA polymerase attached to the promoter of genes x and y, respectively. Here h is used as an abstract inhibitor.

Finally, the set of parametric reactions consists of the following subsets:

- $A_x = \{(\{x\}, \{h\}, \{x\}), (\{x\}, \{h\}, \{\widehat{x}\}), (\{x, \widehat{x}\}, \{h\}, \{X\})\}$,
- $A_y = \{(\{y\}, \{h\}, \lambda_1), (\lambda_2, \{h\}, \{\widehat{y}\}), (\{y, \widehat{y}\}, \{h\}, \lambda_3)\}$,
- $A_Q = \{(\{X, Y\}, \{h\}, \{Q\})\}$.

Notice that the reactions of A_y use parameters $\lambda_1, \lambda_2, \lambda_3$ to define expression of the protein Y.

Suppose that we investigate the processes starting from the states that already contain x and y. This leads to the following definition of the context automaton: $\mathfrak{A} = (\{\mathtt{q}_0, \mathtt{q}_1\}, \mathtt{q}_0, R)$, where: $R = \{\mathtt{q}_0 \xrightarrow{\{x,y\}} \mathtt{q}_1, \mathtt{q}_1 \xrightarrow{\varnothing} \mathtt{q}_1, \mathtt{q}_1 \xrightarrow{\{h\}} \mathtt{q}_0)\}$. When the context set contains the entity h, $\mathfrak{A}$ reverts back to the initial location, while for the empty context the automaton remains in $\mathtt{q}_1$.

Finally, the *context-restricted parametric reaction system* is defined as $\textsc{cr}\text{-}\mathcal{P} = ((S, P, A), \mathfrak{A})$, where: $P = \{\lambda_1, \lambda_2, \lambda_3\}$ and $A = A_x \cup A_y \cup A_Q$. □

We focus on the synthesis of a parameter valuation, given n observations of the behaviour of the system that are expressed with rsLTL formulae.

Let $\textsc{cr}\text{-}\mathcal{P} = (\mathcal{P}, \mathfrak{A})$ be a *context-restricted parametric reaction system* and $F = \{\phi_1, \ldots, \phi_n\}$ be a set of rsLTL formulae. The aim of *parameter synthesis* for *context-restricted parametric reaction system* is to find a valid parameter valuation $\mathtt{v}$ of $\textsc{cr}\text{-}\mathcal{P}$ such that:

$$\left(\mathcal{M}(\textsc{cr}\text{-}\mathcal{P}^{\leftarrow \mathtt{v}}) \models_{\exists} \phi_1\right) \wedge \ldots \wedge \left(\mathcal{M}(\textsc{cr}\text{-}\mathcal{P}^{\leftarrow \mathtt{v}}) \models_{\exists} \phi_n\right).$$

Each formula of F corresponds to an interactive process observed in the analysed system via, e.g., experiments or simulations. Therefore, for each such process we expect an individual path in $\mathcal{M}(\textsc{cr}\text{-}\mathcal{P}^{\leftarrow \mathtt{v}})$ and we solve the n model checking problem instances for rsLTL in one instance. However, the parameter valuation v is shared among all instances, which allows us to calculate v for which all the properties of F are satisfied.

Example 3 Let us assume we performed an experiment on the system from Example 2 where protein Y was expressed, and we collected the following observations related to the expression of Y:

- when the current state contains y, then y and $\widehat{y}$ are present in the next state:
  $\phi_1^c = \mathsf{G}_{\neg h}(y \Rightarrow \mathsf{X}(y \wedge \widehat{y}))$,
- when y and $\widehat{y}$ are present, then Y is finally produced:
  $\phi_2^c = \mathsf{G}_{\neg h}((y \wedge \widehat{y}) \Rightarrow \mathsf{F}Y)$,
- the entities y, $\widehat{y}$, and Y are eventually produced:
  $\phi^r = (\mathsf{F}_{\neg h} y) \wedge (\mathsf{F}_{\neg h} \widehat{y}) \wedge (\mathsf{F}_{\neg h} Y)$.

These observations are made assuming h is not provided in the context set. Additionally, we observe that the protein Q is not present in the first three steps of the execution and then, after an arbitrary number of steps, it is finally produced:

$$\phi^d = \neg Q \wedge \mathsf{X}(\neg Q \wedge \mathsf{X}(\neg Q \wedge \mathsf{F}Q)).$$

The observations are related to a single interactive process (or an experiment), therefore we constrain the problem using the conjunction of all the observations.

Finally, the observations are expressed using the following rsLTL formula:

$$\phi_y = \phi^r \wedge \phi_1^c \wedge \phi_2^c \wedge \phi^d.$$

We perform parameter synthesis for $F = \{\phi_y\}$, that is, we obtain a valid parameter valuation $\mathtt{v}$ such that $\mathcal{M}(\textsc{cr}\text{-}\mathcal{P}^{\leftarrow \mathtt{v}}) \models_{\exists} \phi$. In fact, it may be possible to obtain more than one such valuation. A parameter valuation $\mathtt{v}_1$ such that

$$\lambda_1^{\leftarrow \mathtt{v}_1} = \{y\},\ \lambda_2^{\leftarrow \mathtt{v}_1} = \{y\},\ \lambda_3^{\leftarrow \mathtt{v}_1} = \{Y\}$$

is valid and satisfies the requirements of our observations. A parameter valuation $\mathtt{v}_2$ such that

$$\lambda_1^{\leftarrow \mathtt{v}_2} = \{X, y\},\ \lambda_2^{\leftarrow \mathtt{v}_2} = \{x, \widehat{x}, y\},\ \lambda_3^{\leftarrow \mathtt{v}_2} = \{X, y, \widehat{y}, Y, Q\}$$

is an another example of a valid valuation which satisfies the requirements. □

Example 4 We introduce an additional *unknown* into the system declared in Example 2. That is, we add a parameter λ_4 and re-define the reactions of A_x in such a way that one of them uses the newly introduced parameter:

$$A_x = \{(\{x\},\{h\},\{x\}),(\lambda_4,\{h\},\{\widehat{x}\}),(\{x,\widehat{x}\},\{h\},\{X\})\}.$$

Let us assume that in another experiment we observed when the current state contains x, then x and $\widehat{x}$ are found in the next state. This is expressed with the formula $\phi_x = \mathsf{G}_{\neg h}(x \Rightarrow \mathsf{X}(x \wedge \widehat{x}))$. Next, we perform parameter synthesis for $F = \{\phi_x, \phi_y\}$, where ϕ_y is the formula from Example 3. We use two rsLTL formulae in F since our observations were gathered in two separate experiments and may be related to separate interactive processes.

A parameter valuation $\mathtt{v}$ such that $\lambda_1^{\leftarrow\mathtt{v}} = \{X,y\}$, $\lambda_2^{\leftarrow\mathtt{v}} = \{y\}$, $\lambda_3^{\leftarrow\mathtt{v}} = \{\widehat{x},Y\}$, $\lambda_4^{\leftarrow\mathtt{v}} = \{Q\}$ is valid and satisfies the requirements of our observations. □

The parameter valuation $\mathtt{v}_2$ obtained in Example 3 and the valuation from Example 4 do not result in the same reactions as the original ones from Ehrenfeucht et al. (2012). This might be undesired, depending on the application of the synthesis and the knowledge of the system under analysis. To address this issue, in the following section we introduce parameter constraints which allow for providing additional restrictions on the parameters used in the synthesis.

6.1 Parameter constraints

In some cases restricting parameter valuations using only rsLTL formulae may prove to be less efficient than constraining the valuation using specialised constraints for the parameters of a parametric reaction system.

Definition 21 The grammar of the *parameter constraints* for $\mathcal{P} = (S,P,A)$ is defined as follows:

$$\mathfrak{c} ::= true \mid \lambda[e] \sim c \mid \lambda[e] \sim \lambda[e] \mid \neg\mathfrak{c} \mid \mathfrak{c} \vee \mathfrak{c},$$

where $\lambda \in P$, $e \in S$, $c \in \mathbb{N}$, and $\sim\ \in \{<,\leq,=,\geq,>\}$.

The set of all the parameter constraints for $\mathcal{P}$ is denoted by $PC(\mathcal{P})$. Intuitively, $\lambda[e]$ can be used to refer to the concentration of $e \in S$ in the multisets corresponding to the valuations of λ.

Definition 22 Let $\mathtt{v}$ be a parameter valuation of $\mathcal{P}$. The fact that a parameter constraint $\mathfrak{c}$ holds in $\mathtt{v}$ is denoted by $\mathtt{v} \models_{\mathtt{p}} \mathfrak{c}$ and defined as follows:

$$\begin{array}{lll}
v \models_p true & & \textit{for every } \mathtt{v}, \\
\mathtt{v} \models_{\mathtt{p}} \lambda[\mathtt{e}] \sim \mathtt{c} & \text{iff} & \lambda^{\leftarrow\mathtt{v}}(e) \sim c, \\
v \models_p \lambda_1[e_1] \sim \lambda_2[e_2] & \text{iff} & \lambda_1^{\leftarrow\mathtt{v}}(e_1) \sim \lambda_2^{\leftarrow\mathtt{v}}(e_2), \\
\mathtt{v} \models_{\mathtt{p}} \neg\mathfrak{c} & \text{iff} & v \not\models_p \mathfrak{c}, \\
v \models_p \mathfrak{c}_1 \vee \mathfrak{c}_2 & \text{iff} & \mathtt{v} \models_{\mathtt{p}} \mathfrak{c}_1 \ \mathtt{or}\ \mathtt{v} \models_{\mathtt{p}} \mathfrak{c}_2.
\end{array}$$

Definition 23 (constrained parametric reaction system) A *constrained parametric reaction system* is a tuple $\mathcal{CP} = (S,P,A,\mathfrak{c})$ such that $\mathcal{P} = (S,P,A)$ is a parametric reaction system and $\mathfrak{c} \in PC(\mathcal{P})$.

For $\mathtt{v} \in \mathtt{PV}_{\mathcal{P}}$, we then define $\mathcal{CP}^{\leftarrow\mathtt{v}} = \mathcal{P}^{\leftarrow\mathtt{v}}$. A parameter valuation $\mathtt{v} \in \mathtt{PV}_{\mathcal{P}}$ is *valid* in $\mathcal{CP}$ if it is valid in $\mathcal{P}$ and $\mathtt{v} \models_{\mathtt{p}} \mathfrak{c}$.

Definition 24 (constrained context-restricted parametric reaction system) A *constrained context-restricted parametrised reaction system* is a pair $\text{cr-}\mathcal{CP} = (\mathcal{CP},\mathfrak{A})$ such that $\mathcal{CP} = (S,P,A,\mathfrak{c})$ is a constrained parametric reaction system and $\mathfrak{A}$ is a context automaton over S.

We also denote $\text{cr-}\mathcal{CP}^{\leftarrow\mathtt{v}} = (\mathcal{CP}^{\leftarrow\mathtt{v}},\mathfrak{A})$.

Example 5 Let us consider the system used in Example 4. We might want to assume that the parameters used in the reactions of A_y do not use any of the entities used in the reactions of A_x. Let $E = \{x,\widehat{x},X\}$ be the set of entities we want to exclude and that are used in A_x. The desired restriction can be expressed using the following parameter constraint:

$$\bigwedge_{i\in\{1,\ldots,3\}}\ \bigwedge_{e\in E}(\lambda_i[e] = 0).$$

Similarly,

$$\bigwedge_{e\in(S\setminus E)}(\lambda_4[e] = 0).$$

can express that the parameter λ_4 used in the reaction of A_x is only allowed to use the entities of E.□

Example 6 It is possible to constrain multisets corresponding to parameters. Suppose $\lambda_1,\lambda_2,\lambda_3 \in P$. To constrain $\lambda_1^{\leftarrow\mathtt{v}}$ to be a sub-multiset of $\lambda_2^{\leftarrow\mathtt{v}}$ (i.e., $\lambda_1^{\leftarrow\mathtt{v}} \subseteq \lambda_2^{\leftarrow\mathtt{v}}$, for all $\mathtt{v}$), we define:

$$submset(\lambda_1,\lambda_2) = \bigwedge_{e\in S}(\lambda_1[e] \leq \lambda_2[e]).$$

To constrain $\lambda_3^{\leftarrow\mathtt{v}}$ to be the intersection of $\lambda_1^{\leftarrow\mathtt{v}}$ and $\lambda_2^{\leftarrow\mathtt{v}}$ (i.e., $\lambda_1^{\leftarrow\mathtt{v}} \cap \lambda_2^{\leftarrow\mathtt{v}} = \lambda_3^{\leftarrow\mathtt{v}}$, for all $\mathtt{v}$), we define:

$$\begin{aligned}intersect(\lambda_1,\lambda_2,\lambda_3) = \bigwedge_{e\in S}\Big(&((\lambda_1[e] > \lambda_2[e]) \wedge (\lambda_3[e] = \lambda_2[e]))\\ &\vee((\lambda_1[e] \leq \lambda_2[e]) \wedge (\lambda_3[e] = \lambda_1[e]))\Big).\end{aligned}$$

□

The parameter synthesis problem for *constrained context-restricted parametric reaction systems* is defined similarly as for context-restricted parametric reaction systems. Let $\text{cr-}\mathcal{CP} = (\mathcal{CP},\mathfrak{A})$ be a *constrained context-restricted parametric reaction system*, and $F = \{\phi_1,\ldots,\phi_n\}$ be a set of rsLTL formulae. The aim of *parameter synthesis* for $\textsc{cr}\text{-}\mathcal{CP}$ is to find a valid parameter valuation $\mathtt{v}$ of $\textsc{cr}\text{-}\mathcal{CP}$ such that:

$$(\mathcal{M}(\textsc{cr}\text{-}\mathcal{CP}^{\leftarrow \mathtt{v}}) \models_{\exists} \phi_1) \wedge \ldots \wedge (\mathcal{M}(\textsc{cr}\text{-}\mathcal{CP}^{\leftarrow \mathtt{v}}) \models_{\exists} \phi_n). \tag{2}$$

Next, we define the *nonemptiness checking* problem, which is a decision problem related to the problem of parameter synthesis. The nonemptiness checking problem consists in checking if there exists a valuation $\mathtt{v}$ such that the condition (2) holds.

6.2 Complexity analysis

Theorem 6 *The nonemptiness checking problem for constrained context-restricted parametric reaction systems and* $\textsc{rsltl}$ *is* $\textsc{pspace}$*-complete.*

Proof The lower bound follows directly from Corollary 1, hence the problem is $\textsc{pspace}$-hard.

For the upper bound we need to show the problem is in $\textsc{pspace}$. To show this we define a nondeterministic space-bounded algorithm and use Lemma 2. Algorithm 1 presents an outline for the nonemptiness checking procedure.

Algorithm 1 Nondeterministic procedure for nonemptiness checking

```
guess $\mathtt{v} \in \mathtt{PV}_{\mathcal{P}}$
if $\mathtt{v} \models_p \mathfrak{c}$ then
    $R := true$
    for all $\phi \in F$ do
        $R := (\mathcal{M}(\textsc{cr}\text{-}\mathcal{CP}^{\leftarrow \mathtt{v}}) \models_{\exists} \phi) \wedge R$
    end for
    if $R = true$ then return $true$
    end if
end if
```

First, the algorithm nondeterministically generates a valuation $\mathtt{v} \in \mathtt{PV}_{\mathcal{P}}$. If $\mathtt{v}$ is valid in $\textsc{cr}\text{-}\mathcal{CP}$, then it proceeds to verifying the $\textsc{rsltl}$ formulae. For all the formulae $\phi \in F$ the algorithm performs existential $\textsc{rsltl}$ model checking in $\mathcal{M}(\textsc{cr}\text{-}\mathcal{CP}^{\leftarrow \mathtt{v}})$. From Lemma 2 and the fact that $\textsc{pspace}$ is closed under complementation, i.e., $\textsc{pspace} = co\textsc{pspace}$, the existential variant of the $\textsc{rsltl}$ model checking problem is also in $\textsc{pspace}$. The nonemptiness checking algorithm requires the space needed by the algorithm for $\textsc{rsltl}$ model checking. Since all the $|F|$ model checking instances are constructed independently and the algorithm only stores the overall result R, the algorithm requires space for at most one instance at any given time. Additionally, the algorithm requires space $\mathcal{O}(|A| \cdot |S|)$ to store the valuation $\mathtt{v}$ and $\mathcal{O}(1)$ for the verification result R. Therefore, the problem remains in $\textsc{pspace}$ and given the lower bound we conclude the problem is $\textsc{pspace}$-complete. □

In the following section we show how the synthesis problem can be solved using an incremental approach, which amounts to checking

$$(\mathcal{M}(\textsc{cr}\text{-}\mathcal{CP}^{\leftarrow \mathtt{v}}) \models^k_{\exists} \phi_1) \wedge \ldots \wedge (\mathcal{M}(\textsc{cr}\text{-}\mathcal{CP}^{\leftarrow \mathtt{v}}) \models^k_{\exists} \phi_n)$$

for $k \geq 0$, by increasing the value of k until a valid parameter valuation is found.

7 SMT-based encoding for parametric reaction systems

In this section, we provide a translation of the parameter synthesis problem for *constrained context-restricted parametric reaction systems* and $\textsc{rsltl}$ into $\textsc{smt}$ with the integer arithmetic theory.

Let $\textsc{cr}\text{-}\mathcal{CP} = ((S, P, A, \mathfrak{c}), (\mathcal{Q}, \mathfrak{q}^{init}, R))$ be a *constrained context-restricted parametric reaction system*, and $F = \{\phi_1, \ldots, \phi_n\}$ be a set of $\textsc{rsltl}$ formulae. We encode the model $\mathcal{M}_{\textsc{cr}-\mathcal{CP}^{\leftarrow \mathtt{v}}}$, where $\mathtt{v}$ is a valid parameter valuation of $\textsc{cr}\text{-}\mathcal{CP}$.

Let $k \geq 0$ be an integer, then for each $f \in \{1, \ldots, n\}$ we encode any possible path prefix of $\mathcal{M}_{\textsc{cr}-\mathcal{CP}^{\leftarrow \mathtt{v}}}$. The encoded path prefixes are bounded with k. That is, for each formula ϕ_f we encode a separate path prefix representing its witness. The entities of S are denoted by $e_1, \ldots, e_m$, where $m = |S|$. For each $\phi_f \in F$ and $i \in \{0, \ldots, k\}$, we introduce sets of positive integer variables:

$$\mathbf{P}_{f,i} = \{\mathtt{p}_{\mathtt{f,i,1}}, \ldots, \mathtt{p}_{\mathtt{f,i,m}}\}, \quad \mathbf{P}^{\mathcal{E}}_{\mathtt{f,i}} = \{\mathtt{p}^{\mathcal{E}}_{\mathtt{f,i,1}}, \ldots, \mathtt{p}^{\mathcal{E}}_{\mathtt{f,i,m}}\}, \quad \text{and } \mathbf{Q}_f = \{\mathtt{q}_{\mathtt{f},0}, \ldots, \mathtt{q}_{\mathtt{f,k}}\}.$$

Let $\mathtt{ta} : \mathtt{A} \rightarrow \{1, \ldots, |\mathtt{A}|\}$ be a bijection mapping all the reactions to integers. For each $a \in A$, we introduce the set of variables encoding the products:

$$\mathbf{P}^{p}_{f,i,a} = \{\mathtt{p}^{\mathtt{p}}_{\mathtt{f,i,ta(a),1}}, \ldots, \mathtt{p}^{\mathtt{p}}_{\mathtt{f,i,ta(a),m}}\}.$$

Let $\sigma.f$ be a path of $\mathcal{M}(\textsc{cr}\text{-}\mathcal{CP}^{\leftarrow \mathtt{v}})$, then

$$\overline{\mathbf{p}}_{f,i} = (\mathtt{p}_{\mathtt{f,i,1}}, \ldots, \mathtt{p}_{\mathtt{f,i,m}}) \quad \text{and} \quad \overline{\mathbf{p}}^{\mathcal{E}}_{\mathtt{f,i}} = (\mathtt{p}^{\mathcal{E}}_{\mathtt{f,i,1}}, \ldots, \mathtt{p}^{\mathcal{E}}_{\mathtt{f,i,m}})$$

are used to encode $(\sigma.f)_b(i)$ and $(\sigma.f)_a(i)$, respectively. With $\overline{\mathbf{p}}_{f,i}[j]$ and $\overline{\mathbf{p}}^{\mathcal{E}}_{f,i}[j]$ we denote, respectively, $\mathtt{p}_{\mathtt{f,i,j}}$ and $\mathtt{p}^{\mathcal{E}}_{\mathtt{f,i,j}}$. For each $i \geq 1$, we define:

$$\overline{\mathbf{p}}^{p}_{f,i} = (\mathtt{p}^{\mathtt{p}}_{\mathtt{f,i,1,1}}, \ldots, \mathtt{p}^{\mathtt{p}}_{\mathtt{f,i,1,m}}, \ldots, \mathtt{p}^{\mathtt{p}}_{\mathtt{f,i,|A|,1}}, \ldots, \mathtt{p}^{\mathtt{p}}_{\mathtt{f,i,|A|,m}}).$$

The following functions map the background set entities to the corresponding variables of the encoding: for all $i \in \{0, \ldots, k\}$, we define $\mathtt{t}_{\mathtt{f,i}} : \mathtt{S} \rightarrow \mathbf{P}_{\mathtt{f,i}}$ and $\mathtt{t}^{\mathcal{E}}_{\mathtt{f,i}} : \mathtt{S} \rightarrow$

$\mathbf{P}^{\mathcal{E}}_{\mathtt{f,i}}$ such that $\mathtt{t_{f,i}(e_j)} = \mathtt{p_{f,i,j}}$ and $\mathtt{t}^{\mathcal{E}}_{\mathtt{f,i}}\mathtt{(e_j)} = \mathtt{p}^{\mathcal{E}}_{\mathtt{f,i,j}}$, for all $j \in \{1, \ldots, m\}$. For all $i \in \{0, \ldots, k\}$ and $a \in A$, we define $\mathtt{t}^{\mathtt{p}}_{\mathtt{f,i,a}} : \mathtt{S} \rightarrow \mathbf{P}^{\mathtt{p}}_{\mathtt{f,i,a}}$ such that: $\mathtt{t}^{\mathtt{p}}_{\mathtt{f,i,a}}\mathtt{(e_j)} = \mathtt{p}^{\mathtt{p}}_{\mathtt{f,i,ta(a),j}}$, for all $j \in \{1, \ldots, m\}$.

The bijection $\mathtt{e} : \mathcal{Q} \rightarrow \{\mathtt{1}, \ldots, |\mathcal{Q}|\}$ maps the states of the context automaton to the integers used in the encoding. Let $\mathtt{tp} : \mathtt{P} \rightarrow \{\mathtt{1}, \ldots, |\mathtt{P}|\}$ be a bijection mapping all the parameters to their corresponding integers. We define:

$$\overline{\mathbf{p}}^{par} = (\mathtt{p}^{\mathtt{par}}_{\mathtt{1,1}}, \ldots, \mathtt{p}^{\mathtt{par}}_{\mathtt{1,m}}, \ldots, \mathtt{p}^{\mathtt{par}}_{\mathtt{|P|,1}}, \ldots, \mathtt{p}^{\mathtt{par}}_{\mathtt{|P|,m}}).$$

For each parameter $\lambda \in P$, we define:

$$\mathbf{P}^{par}_{\lambda} = \{\mathtt{p}^{\mathtt{par}}_{\mathtt{tp(\lambda),1}}, \ldots, \mathtt{p}^{\mathtt{par}}_{\mathtt{tp(\lambda),m}}\}$$

and $\mathtt{pm}_{\lambda} : \mathtt{S} \rightarrow \mathbf{P}^{\mathtt{par}}_{\lambda}$ such that $\mathtt{pm}_{\lambda}\mathtt{(e_j)} = \mathtt{p}^{\mathtt{par}}_{\mathtt{tp(\lambda),j}}$. Let $a \in A$ and $\mathfrak{s} \in \{\mathfrak{r}_a, \mathfrak{i}_a, \mathfrak{p}_a\}$. Then, $re^{\mathfrak{s}}(e_j)$ denotes $\mathtt{pm}_{\mathfrak{s}}\mathtt{(e_j)}$ if $\mathfrak{s} \in P$, and $\mathfrak{s}(e_j)$ otherwise. To define the SMT encoding of the paths we need auxiliary functions that correspond to elements of the encoding.

To encode the initial state of the model for $\phi_f \in F$, we define

$$\mathsf{Init}(\overline{\mathbf{p}}_{f,i}, \mathtt{q_{f,i}}) = \left(\bigwedge_{\mathtt{e} \in \mathtt{S}} \mathtt{t_{f,i}(e)} = \mathtt{0}\right) \wedge \mathtt{q_{f,i}} = \mathtt{e(q^{init})},$$

where all the concentration levels are set to zero, and the context automaton is in its initial state.

With $\mathsf{PC}(\overline{\mathbf{p}}^{par})$ we encode the parameter constraints, require that the concentration levels of the reactants are always lower than the concentration levels of the inhibitors, and ensure that all the multisets corresponding to the parameters are non-empty, i.e., for each parameter, at least one entity must a have positive concentration level:

$$\begin{aligned}\mathsf{PC}(\overline{\mathbf{p}}^{par}) = {} & enc_{\mathfrak{c}}(\overline{\mathbf{p}}^{par}) \\ & \wedge \left(\bigwedge_{a \in A} \bigwedge_{e \in S} re^{\mathfrak{i}_a}(e) > 0 \Rightarrow (re^{\mathfrak{r}_a}(e) < re^{\mathfrak{i}_a}(e))\right) \wedge \\ & \left(\bigwedge_{\lambda \in P} \bigvee_{e \in S} \mathtt{pm}_{\lambda}\mathtt{(e)} > \mathtt{0}\right),\end{aligned}$$

where $enc_{\mathfrak{c}}(\overline{\mathbf{p}}^{par})$ is the encoding of $\mathfrak{c}$ and it is defined over the variables of $\overline{\mathbf{p}}^{par}$. The encoding follows directly from the semantics of parameter constraints.

The parametric reactions $a \in A$ are encoded with

$$\begin{aligned}& \mathsf{Rct}_a(\overline{\mathbf{p}}_{f,i}, \overline{\mathbf{p}}^{\mathcal{E}}_{f,i}, \overline{\mathbf{p}}^{p}_{f,i+1}, \overline{\mathbf{p}}^{par}) = \bigwedge_{e \in S}((\mathtt{t_{f,i}(e)} \geq \mathtt{re}^{\mathfrak{r}_a}\mathtt{(e)} \\ & \vee \mathtt{t}^{\mathcal{E}}_{\mathtt{f,i}}\mathtt{(e)} \geq \mathtt{re}^{\mathfrak{r}_a}\mathtt{(e)}) \wedge \\ & (\mathtt{t_{f,i}(e)} < \mathtt{re}^{\mathfrak{i}_a}\mathtt{(e)} \wedge \mathtt{t}^{\mathcal{E}}_{\mathtt{f,i}}\mathtt{(e)} < \mathtt{re}^{\mathfrak{i}_a}\mathtt{(e)}) \wedge (\mathtt{t}^{\mathtt{p}}_{\mathtt{f,a,i+1}}\mathtt{(e)} = \mathtt{re}^{\mathfrak{p}_a}\mathtt{(e)})).\end{aligned}$$

With the following formula we encode the selection of the maximal concentration levels produced for each entity by all the reactions:

$$\begin{aligned}& \mathsf{Results}(\overline{\mathbf{p}}_{f,i}, \overline{\mathbf{p}}^{p}_{f,i}) \\ & \quad = \left(\bigwedge_{e \in S} \mathtt{t_{f,i+1}(e)} = \mathtt{max}(\{\mathtt{0}\} \cup \bigcup_{\mathtt{a} \in \mathtt{A}} \{\mathtt{t}^{\mathtt{p}}_{\mathtt{f,a,i+1}}\mathtt{(e)}\})\right).\end{aligned}$$

We encode the local state changes of $\mathcal{CP}$ with the following function:

$$\begin{aligned}& \mathsf{Tr}_{\mathcal{CP}}(\overline{\mathbf{p}}_{f,i}, \overline{\mathbf{p}}^{\mathcal{E}}_{f,i}, \overline{\mathbf{p}}^{p}_{f,i+1}, \overline{\mathbf{p}}_{f,i+1}, \overline{\mathbf{p}}^{par}) \\ & \quad = \left(\bigwedge_{a \in A} \mathsf{Rct}_a(\overline{\mathbf{p}}_{f,i}, \overline{\mathbf{p}}^{\mathcal{E}}_{f,i}, \overline{\mathbf{p}}^{p}_{f,i+1}, \overline{\mathbf{p}}^{par})\right) \wedge \\ & \quad\quad \mathsf{Results}(\overline{\mathbf{p}}_{f,i+1}, \overline{\mathbf{p}}^{p}_{f,i+1}).\end{aligned}$$

To encode a multiset $\mathbf{c} \in \mathcal{B}(S)$ of context entities we define the following function:

$$\mathsf{Ct}_{\mathbf{c}}(\overline{\mathbf{p}}^{\mathcal{E}}_{f,i}) = \bigwedge_{e \in S} \mathtt{t}^{\mathcal{E}}_{\mathtt{f,i}}\mathtt{(e)} = \mathbf{c}\mathtt{(e)}$$

The encoding of the transition relation of the context automaton is a disjunction of the encoded transitions:

$$\begin{aligned}& \mathsf{Tr}_{\mathfrak{A}}(\mathtt{q_{f,i}}, \overline{\mathbf{p}}^{\mathcal{E}}_{\mathtt{f,i}}, \mathtt{q_{f,i+1}}) \\ & \quad = \bigvee_{(\mathfrak{q}, \mathbf{c}, \mathfrak{q}') \in \mathbf{R}} \left(\mathtt{q_{f,i}} = \mathtt{e}(\mathfrak{q}) \wedge \mathsf{Ct}_{\mathbf{c}}(\overline{\mathbf{p}}^{\mathcal{E}}_{\mathtt{f,i}}) \wedge \mathtt{q_{f,i+1}} = \mathtt{e}(\mathfrak{q}')\right).\end{aligned}$$

The transition relation of the model for CR-$\mathcal{CP}$ is a conjunction of the encoded transition relations for $\mathcal{CP}$ and $\mathfrak{A}$:

$$\begin{aligned}& \mathsf{Tr}_{\text{CR}-\mathcal{CP}}(\overline{\mathbf{p}}_{f,i}, \mathtt{q_{f,i}}, \overline{\mathbf{p}}^{\mathcal{E}}_{\mathtt{f,i}}, \overline{\mathbf{p}}^{\mathtt{p}}_{\mathtt{f,i+1}}, \overline{\mathbf{p}}_{\mathtt{f,i+1}}, \overline{\mathbf{p}}^{\mathtt{par}}) = \\ & \mathsf{Tr}_{\mathcal{CP}}(\overline{\mathbf{p}}_{f,i}, \overline{\mathbf{p}}^{\mathcal{E}}_{f,i}, \overline{\mathbf{p}}^{p}_{f,i+1}, \overline{\mathbf{p}}_{f,i+1}, \overline{\mathbf{p}}^{par}) \wedge \mathsf{Tr}_{\mathfrak{A}}(\mathtt{q_{f,i}}, \overline{\mathbf{p}}^{\mathcal{E}}_{\mathtt{f,i}}, \mathtt{q_{f,i+1}}).\end{aligned}$$

Finally, to encode the paths of $\mathcal{M}_{\text{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}$ that are bounded with k, we unroll the transition relation up to k and combine it with the encoding of the initial state of the model:

$$\begin{aligned}\mathsf{Paths}^k_f = {} & \mathsf{Init}\left(\overline{\mathbf{p}}_{f,0}, \mathtt{q_{f,0}}\right) \\ & \wedge \bigwedge_{i=0}^{k-1} \mathsf{Tr}_{\text{CR}-\mathcal{CP}}(\overline{\mathbf{p}}_{f,i}, \mathtt{q_{f,i}},, \overline{\mathbf{p}}^{\mathcal{E}}_{\mathtt{f,i}}, \overline{\mathbf{p}}^{\mathtt{p}}_{\mathtt{f,i+1}}, \overline{\mathbf{p}}_{\mathtt{f,i+1}}, \overline{\mathbf{p}}^{\mathtt{par}}).\end{aligned}$$

The encoded RSLTL formula ϕ_f at a position $i \in \{0, \ldots, k\}$ is denoted by $[\![\phi_f]\!]^k_i$. To encode the formula $[\![\phi_f]\!]^k_i$, we use our translation presented in Sect. 5. However, for each formula $\phi_f \in F$, we use independent sets of encoding variables corresponding to its path, i.e., the variables indexed with f. The encoding Loops^k_f for the loop positions is defined for each formula $\phi_f \in F$. Finally, we perform the synthesis of the parameter valuation $\mathtt{v}$ by testing the satisfiability of the following formula:

$$[\mathcal{M}_{\text{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}, F, k] = \bigwedge_{\phi_f \in F} \left(\mathsf{Paths}^k_f \wedge \mathsf{Loops}^k_f \wedge [\![\phi_f]\!]^k_0\right) \wedge \mathsf{PC}(\overline{\mathbf{p}}^{par}). \quad (3)$$

The presented encoding differs from that for the context-restricted reaction systems with discrete concentrations and rsLTL (Sect. 5) in the way the transition relation is encoded. Here, we use an additional step that encodes the concentration levels of each entity produced by the individual reactions.

These results are then used to select the maximal concentration level produced for a given entity. This is required because some reactions produce parameters for which we do not have concrete values at the encoding stage. Therefore, it is not possible to use the technique demonstrated in Sect. 5, where the reactions are ordered and effectively only that producing the maximal concentration level is enabled.

We can then establish the correctness of the proposed encoding for a given valid parameter valuation.

Theorem 7 *Let* cr-$\mathcal{CP} = (\mathcal{CP}, \mathfrak{A})$ *be a constrained context-restricted parametric reaction system,* $\mathtt{v} \in \mathrm{PV}_{\mathrm{CR}-\mathcal{CP}}$ *be a valid parameter valuation,* $\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}$ *be its model, and* F *be a set of rsLTL formulae.*

For any $k \in \mathbb{N}$*, the formula* $[\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}, F, k]$ *is satisfiable iff* $\bigwedge_{\phi \in F}(\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}} \models^k_\exists \phi)$.

Proof Since we assume a valid parameter valuation $\mathtt{v}$, to obtain a *constrained context-restricted parametric reaction system* we can perform the substitution of all the parameters that occur in context-restricted parametric reaction system.

Then, the proof is similar to the one of Theorem 5 for context-restricted reaction systems with discrete concentrations and rsLTL.

The formula $\mathsf{PC}(\overline{\mathbf{p}}^{par})$ applies only to the encoding of parameters and after performing the substitution we can simply assume that it is *true* as it does not constrain anything.

The formula $[\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}, F, k]$ encodes $|F|$ bounded model checking instances (similar to what was described in Sect. 5).

Let us consider $\phi_f \in F$. We assume an arbitrary $k \in \mathbb{N}$ and focus on the satisfiability of Paths^k_f, since the encoding of this formula differs from the corresponding one in the encoding for cr-$\mathcal{C}$.

There exists a path $\sigma.f$ in $\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}$ and $(\sigma.f)^k$ is its prefix of length k iff there exists a valuation that represents $(\sigma.f)^k$ which satisfies Paths^k_f.

Let $i \in \{0, \ldots, k-1\}$. We observe the formula $\mathsf{Tr}_{\mathrm{CR}-\mathcal{CP}}$ is satisfied for the valuation encoding $(\sigma.f)_s(i)$, $(\sigma.f)_a(i)$ and $(\sigma.f)_s(i+1)$ iff $(\sigma.f)_s(i) \xrightarrow{(\sigma.f)_a(i)} (\sigma.f)_s(i+1)$. This follows from the encoding of $\mathsf{Tr}_{\mathcal{CP}}$ and $\mathsf{Tr}_{\mathfrak{A}}$. Let us first recall that $(\sigma.f)_s(i) = \big((\sigma.f)_b(i), (\sigma.f)_{ca}(i)\big)$. The formula $\mathsf{Tr}_{\mathcal{CP}}$ is satisfied iff the valuation satisfies $\mathsf{Results}$ and Rct_a for each $a \in A$. The formula Rct_a encodes the produced concentration levels for all the entities $e \in S$ by $a \in A$ using the intermediate variables of $\mathbf{P}^p_{f,i,a}$. Then, $\mathsf{Results}$ ensures the maximal produced concentration levels for each entity and each reaction are encoded using the variables of $\mathbf{P}_{f,i+1}$. It follows from the construction that $\mathsf{Results}$ and Rct_a are satisfied iff the valuation encodes the concentration levels of the entities in the successor state $(\sigma.f)_b(i+1)$ that are produced by the reactions from the state $(\sigma.f)_b(i)$ combined with the context $(\sigma.f)_a(i)$.

The formula $\mathsf{Tr}_{\mathfrak{A}}$ is satisfied iff the valuation encodes a transition $(\mathfrak{q}, \mathbf{c}, \mathfrak{q}') \in R$ such that $\mathfrak{q} = (\sigma.f)_{ca}(i)$, $\mathbf{c} = (\sigma.f)_a(i)$ and $\mathfrak{q}' = (\sigma.f)_{ca}(i+1)$. Then, the formula Paths^k_f is satisfied iff the valuation encodes a path prefix $(\sigma.f)^k$ such that the state $(\sigma.f)_s(0)$ of $\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}$ is the initial state and $(\sigma.f)_s(i) \xrightarrow{(\sigma.f)_a(i)} (\sigma.f)_s(i+1)$ for all $i \in \{0, \ldots, k-1\}$. The rest of the proof for Loops^k_f and $[\![\phi_f]\!]^k_0$ follows as for Theorem 5.

Now it is easy to see that $\mathsf{Paths}^k_f \wedge \mathsf{Loops}^k_f \wedge [\![\phi_f]\!]^k_0$ is satisfiable iff $\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}} \models^k_\exists \phi_f$. Finally, we conclude that $[\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}, F, k]$ is satisfiable iff $\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}} \models^k_\exists \phi_f$ for all $\phi_f \in F$. □

In practice, the encoding $[\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}}, F, k]$ is intended to be satisfiable for any valid parameter valuation $\mathtt{v}$ such that $\bigwedge_{\phi \in F}(\mathcal{M}_{\mathrm{CR}-\mathcal{CP}^{\leftarrow \mathtt{v}}} \models^k_\exists \phi)$. The constraints that enforce the valuation to be valid are expressed using the encoding of PC. We extract the valuation of the parameters of P when the formula (3) is satisfiable. For the satisfied formula we obtain its model, i.e., the valuations of the variables used in the formula. Let $V(\mathtt{p})$ denote the valuation of a variable $\mathtt{p}$ used in our encoding. The parameter valuations are defined as follows: $\lambda^{\leftarrow \mathtt{v}}(e) = V(\mathtt{pm}_\lambda(\mathtt{e}))$ for each $e \in S$ and $\lambda \in P$.

8 Experimental evaluation

In this section, we present the results of experimental evaluation of the translation presented in Sect. 7. We tested the method on a parametric version (pmutex) of the reaction system model introduced in Męski et al. (2015) for a mutual exclusion protocol.

The system we consider consists of $n \geq 2$ processes competing for an exclusive access to the critical section. The background set of the context-restricted reaction system with discrete concentrations modelling the mutual exclusion protocol is defined as $S = \bigcup_{i=1}^n S_i$, with the set of background entities corresponding to the i-th process given by:

$$S_i = \{out_i, req_i, in_i, act_i, lock, done, s\},$$

where the entities $lock$, $done$, and s are shared amongst all the processes.

We start by defining the context automaton $\mathfrak{A}$. 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 that $\mathfrak{A}$ may supply any $C \subseteq \{act_1, \ldots, act_n\}$ such that $|C| \leq 2$, allowing at most two simultaneously active processes – we assume that if the context contains act_i then the i^{th} process is to perform an action. This leads to the following definition of the context automaton: $\mathfrak{A} = (\{\mathfrak{q}_0, \mathfrak{q}_1\}, \mathfrak{q}_0, R)$, where:

$$R = \{\mathfrak{q}_0 \xrightarrow{\{out_1, \ldots, out_n\}} \mathfrak{q}_1\} \cup$$
$$\{\mathfrak{q}_1 \xrightarrow{C} \mathfrak{q}_1 \mid C \subseteq \{act_1, \ldots, act_n\} \text{ and } |C| \leq 2\}.$$

We allow at most two active processes at a time to avoid encoding in the context automaton all the 2^n transitions with the subsets of $\{act_1, \ldots, act_n\}$.

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 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 the critical section, the critical section is locked, i.e., the $lock$ entity is produced. The $lock$ entity is preserved until the entity $done$ appears, which is produced when the process leaves the critical section. Any reaction in the system may be inhibited by the s entity.

This version of the mutual exclusion protocol implementation differs from that presented in Męski et al. (2015) by the use of concentration levels. Each process after requesting access to its critical section must wait at least one step before it is allowed to gain access and, after entering the critical section, the process performs computations which take two steps.

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

- $(\{out_i{\mapsto}1, act_i{\mapsto}1\}, \{s{\mapsto}1\}, \{req_i{\mapsto}1\})$,
- $(\{out_i{\mapsto}1\}, \{act_i{\mapsto}1\}, \{out_i{\mapsto}1\})$,
- $(\{req_i{\mapsto}1, act_i{\mapsto}1, act_j{\mapsto}1\}, \{s{\mapsto}1\}, \{req_i{\mapsto}1\})$ for each $j \in \{1, \ldots, n\}$ s.t. $i \neq j$,
- $(\{req_i{\mapsto}1\}, \{act_i{\mapsto}1\}, \{req_i{\mapsto}2\})$,
- $(\{req_i{\mapsto}2, act_i{\mapsto}1\}, \{act_j{\mapsto}1 \mid j \in \{1, \ldots, n\}$ and $j \neq i\} \cup \{lock{\mapsto}1\}, \{in_i{\mapsto}3, lock{\mapsto}1\})$,
- $(\{in_i{\mapsto}3, act_i{\mapsto}1\}, \{s{\mapsto}1\}, \{in_i{\mapsto}2\})$,
- $(\{in_i{\mapsto}2, act_i{\mapsto}1\}, \{s{\mapsto}1\}, \{in_i{\mapsto}1\})$,
- $(\{in_i{\mapsto}1, act_i{\mapsto}1\}, \{s{\mapsto}1\}, \{out_i{\mapsto}1, done{\mapsto}1\})$,
- $(\{in_i{\mapsto}1\}, \{s{\mapsto}1\}, \{in_i{\mapsto}1\})$.

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 that we are allowed to modify the behaviour of an additional process (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 $i \in \{1, \ldots, n\}$, 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:

$$A = \left(\bigcup_{i=1}^{n} A_i\right) \cup A_p \cup \{(\{lock{\mapsto}1\}, \{done{\mapsto}1\}, \{lock{\mapsto}1\})\}.$$

Finally, we define the *constrained context-restricted parametric reaction system* modelling PMUTEX as: CR-$\mathcal{C}_M = ((S, P, A, \mathfrak{c}), \mathfrak{A})$, where:

$$\mathfrak{c} = \left((\lambda_p[in_n] = 0) \wedge \bigwedge_{\lambda \in P, e \in S \setminus S_n} (\lambda[e] = 0) \right).$$

The constraint $\mathfrak{c}$ constrains the additional reaction by requiring that it may produce only entities related to the n^{th} process and it cannot produce in_n. This is to avoid trivial solutions. Then, we need to synthesise a parameter valuation v of CR-$\mathcal{CP}_M$ which gives the rsLTL property $\phi = \mathsf{F}((in_1 > 0) \wedge (in_n > 0))$, i.e., $\mathcal{M}(\text{CR-}\mathcal{CP}_M^{\leftarrow \mathrm{v}}) \models_{\exists} \phi$.

The verification tool was implemented in Python and it used Z3 4.12.2 (Moura et al. 2008) for SMT-solving. We implemented 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. 2–3 compare the implementation of the encoding from Sect. 7 (CR-$\mathcal{CP}$) and its extension (CR-$\mathcal{CP}_{opt}$)

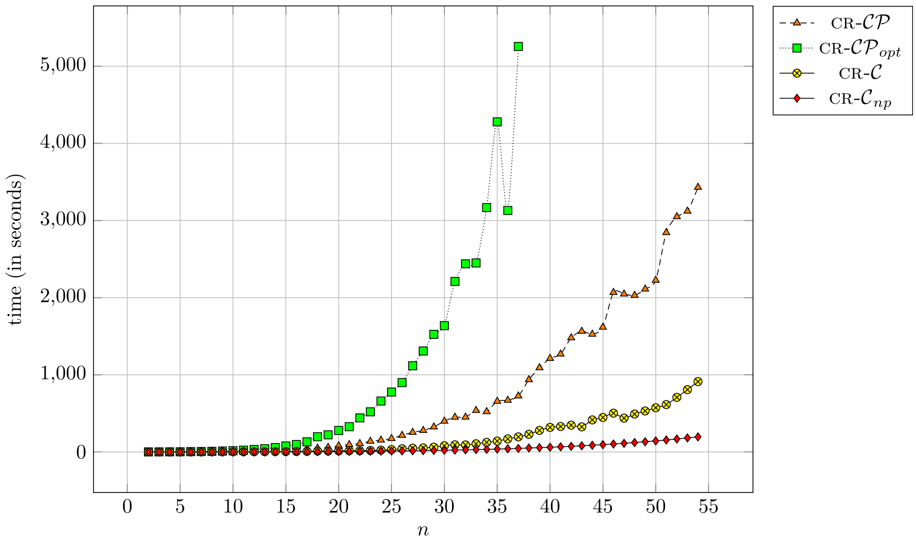

Fig. 2 Synthesis results for PMUTEX: execution time

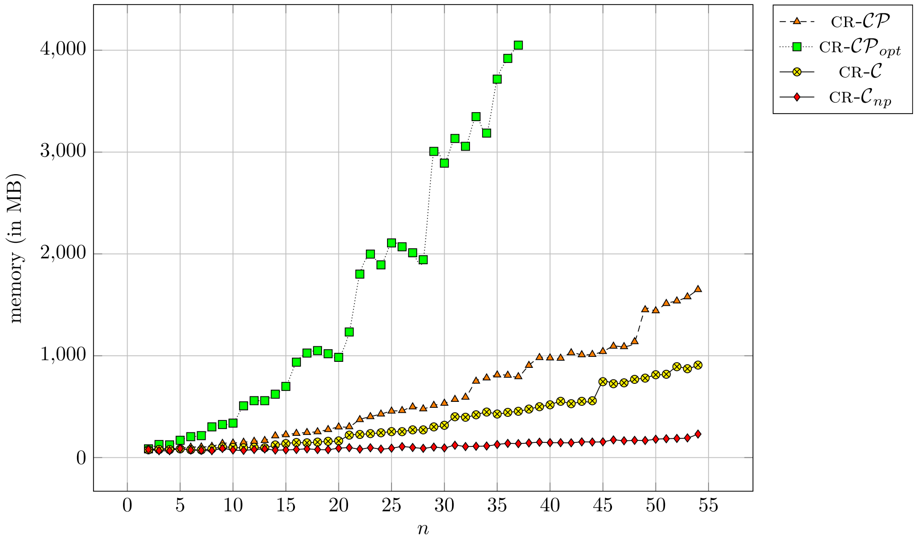

Fig. 3 Synthesis results for PMUTEX: memory consumption

that optimises the obtained parameter valuations by using OptSMT provided with Z3. We also use the same encoding for the 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 it is possible with the method defined in Sect. 5. Next, we compare our results with those obtained using the non-parametric method (CR-$\mathcal{C}_{np}$) defined in Sect. 5.

The experimental implementation provides a valuation v which allows to violate the mutual exclusion property, where $\lambda_r^{\leftarrow v} = \{out_n \mapsto 1\}$, $\lambda_i^{\leftarrow v} = \{s \mapsto 1\}$, and $\lambda_p^{\leftarrow v} = \{req_n \mapsto 2, done \mapsto 1\}$, for all the values $n \geq 2$ tested. This valuation was obtained using CR-$\mathcal{CP}_{opt}$.

The difference in time and memory consumption between the parametric (CR-$\mathcal{CP}$) and the non-parametric (CR-$\mathcal{C}$) approach is minor, while CR-$\mathcal{C}_{np}$ is the most efficient of all the approaches tested. The difference between the performance of CR-$\mathcal{C}_{np}$ and CR-$\mathcal{C}$ shows the potential for optimisations of our parametric encoding as CR-$\mathcal{C}$ uses a simpler encoding. However, the performance of CR-$\mathcal{CP}_{opt}$ shows that the cost of minimising parameter valuations can be high and for $n = 37$ our implementation consumed 4050MB of memory, while the variant without the optimisation required only 794MB. For this reason we did not obtain results for $n > 37$.

9 Concluding remarks

In this paper, we introduced a method for reaction mining that allows for calculating parameter valuations for partially defined reactions of reaction systems. We then demonstrated how the proposed method can be used for the synthesis of an attack in which one injects an additional instruction represented by a reaction, and used rsLTL to express the goal of such an attack. Furthermore, we extended our results for rsLTL (Męski et al. 2017) showing that rsLTL model checking is PSPACE-complete, and provided a translation of rsLTL model checking to LTL model checking for reaction systems.

Assuming there is a finite set of allowed concentration levels for the parameters, the method introduced here 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 preceding step.

When dealing with parameter synthesis, the parameters could be associated with the model (Alur et al. 1993; Hune et al. 2002) or with the formalism used to express its properties (Knapik et al. 2015; Jones et al. 2012). We focused on the synthesis of the parameters which appear in the reaction system. One could also consider extending this approach to include parameters in the context automaton. This might allow one to synthesise the behaviour of the environment which would lead to the satisfaction of the verified rsLTL property. Having said that, in the implementation of our approach, when the verified formula is satisfied, we also obtain the witness which contains the entire context sequence generated by the context automaton. This sequence represents the behaviour of the environment which leads to satisfaction of the rsLTL formula.

Note also that parameters could be introduced in the CTL or rsLTL formulae leading to parametric variants of both logics, e.g., by introducing parameters in place of the families of sets of entities (or the multiset expressions for rsLTL).

Author contributions All authors have contributed equally.

Funding This research was funded in part, by the Polish National Center for Research and Develeopment (NCBR) the Luxembourg National Research Fund (FNR), under the PolLux/FNR-CORE project SpaceVote (POLLUX-XI/14/SpaceVote/2023). For the purpose of open access, and in fulfilment of the obligations arising from the grant agreement, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

Data availibility Our experimental results can be reproduced by cloning the repository with Reactics (https://github.com/arturmeski/reactics/) and running the shell script from the `experiments/reaction_mining_mutex` directory.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval Not applicable.

References

Ehrenfeucht A, Rozenberg G (2007) Reaction systems. Fundam Infor 75(1–4):263–280

Ehrenfeucht A, Kleijn J, Koutny M, Rozenberg G (2012) Reaction systems: a natural computing approach to the functioning of living cells. A computable universe, understanding and exploring nature as computation, pp 189–208

Ehrenfeucht A, Kleijn J, Koutny M, Rozenberg G (2017) Evolving reaction systems. Theor Comput Sci 682:79–99

Ehrenfeucht A, Rozenberg G (2009) Introducing time in reaction systems. Theor Comput Sci 410(4–5):310–322

Brijder R, Ehrenfeucht A, Rozenberg G (2011) Reaction systems with duration. In: Computation, cooperation, and life - essays dedicated to gheorghe paun on the occasion of his 60th birthday. LNCS 6610:191–202

Hirvensalo M (2012) On probabilistic and quantum reaction systems. Theor Comput Sci 429:134–143

Alhazov A, Aman B, Freund R, Ivanov S (2016) Simulating R systems by P systems. In: Membrane computing, 17th international conference, CMC 2016, Milan, Italy, pp 51–66

Formenti E, Manzoni L, Porreca AE (2014a) Cycles and global attractors of reaction systems. In: Descriptional complexity of formal systems - 16th international workshop, DCFS 2014. LNCS, pp 114–125

Formenti E, Manzoni L, Porreca AE (2014b) 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

Formenti E, Manzoni L, Porreca AE (2014c) On the complexity of occurrence and convergence problems in reaction systems. Nat Comput 14:185–191

Salomaa A (2012a) Functions and sequences generated by reaction systems. Theor Comput Sci 466:87–96

Salomaa A (2012b) On state sequences defined by reaction systems. In: Logic and program semantics, pp 271–282

Salomaa A (2013a) Functional constructions between reaction systems and propositional logic. Int J Found Comput Sci 24(1):147–160

Salomaa A (2013b) Minimal and almost minimal reaction systems. Nat Comput 12(3):369–376

Dennunzio A, Formenti E, Manzoni L (2015a) Reaction systems and extremal combinatorics properties. Theor Comput Sci 598:138–149

Dennunzio A, Formenti E, Manzoni L, Porreca AE (2015b) Ancestors, descendants, and gardens of Eden in reaction systems. Theor Comput Sci 608:16–26

Azimi S, Iancu B, Petre I (2014) Reaction system models for the heat shock response. Fundam Inf 131(3–4):299–312

Corolli L, Maj C, Marini F, Besozzi D, Mauri G (2012) An excursion in reaction systems: from computer science to biology. Theor Comput Sci 454:95–108

Azimi S, Gratie C, Ivanov S, Manzoni L, Petre I, Porreca AE (2016) Complexity of model checking for reaction systems. Theor Comput Sci 623:103–113

Azimi S, Gratie C, Ivanov S, Petre I (2015) Dependency graphs and mass conservation in reaction systems. Theor Comput Sci 598:23–39

Męski A, Penczek W, Rozenberg G (2015) Model checking temporal properties of reaction systems. Inf Sci 313:22–42

Dennunzio A, Formenti E, Manzoni L, Porreca AE (2019) Complexity of the dynamics of reaction systems. Inf Comput 267:96–109

Ferrando A, Malvone V (2021) Towards the verification of strategic properties in multi-agent systems with imperfect information. arXiv preprint arXiv:2112.13621

Brodo L, Bruni R, Falaschi M (2023) Verification of reaction systems processes. In: Challenges of Software Verification, pp 243–264

Męski A, Koutny M, Penczek W (2016) Towards quantitative verification of reaction systems. In: Unconventional computation and natural computation: 15th international conference, UCNC 2016, Manchester, UK, Proceedings, pp 142–154

Męski A, Koutny M, Penczek W (2017) Verification of linear-time temporal properties for reaction systems with discrete concentrations. Fundam Inform 154(1–4):289–306

Horn F, Jackson R (1972) General mass action kinetics. Arch Ration Mech Anal 47(2):81–116

Glass L, Kauffman SA (1973) The logical analysis of continuous, non-linear biochemical control networks. J Theor Biol 39(1):103–129

Paun G, Rozenberg G (2002) A guide to membrane computing. Theor Comput Sci 287(1):73–100

Martín-Vide C, Paun G, Pazos J, Rodríguez-Patón A (2003) Tissue P systems.Theor Comput Sci 296(2): 295–326

Męski A, Koutny M, Mikulski L, Penczek W (2023) Model checking for distributed reaction systems with rsCTLK (submitted)

Zañudo JGT, Scaltriti M, Albert R (2017) A network modeling approach to elucidate drug resistance mechanisms and predict combinatorial drug treatments in breast cancer. Cancer Converg 1(1):1–25

Kleijn J, Koutny M, Pietkiewicz-Koutny M, Rozenberg G (2011) Classifying boolean nets for region-based synthesis. In: Desel J, Yakovlev A (eds) Proceedings of the workshop applications of region theory 2011, Newcastle upon Tyne, UK, CEUR Workshop Proceedings, vol 725, pp 5–21

Kleijn J, Koutny M (2011) Membrane systems with qualitative evolution rules. Fundam Inform 110(1–4):217–230

Kleijn J, Koutny M, Pietkiewicz-Koutny M (2014) Tissue systems and petri net synthesis. Trans Petri Nets Other Model Concurr 9:124–146

Kleijn J, Koutny M, Pietkiewicz-Koutny M, Rozenberg G (2012) Membrane systems and petri net synthesis. In: Ciobanu G (ed) Proceedings 6th workshop on membrane computing and biologically inspired process Calculi, MeCBIC 2012, Newcastle, UK, 8th September 2012. EPTCS, vol 100, pp 1–13

Petri CA (1973) Concepts of net theory. In: Mathematical foundations of computer science: proceedings of symposium and summer school, Strbské Pleso, High Tatras, Czechoslovakia, September 3-8, 1973, pp 137–146

Kleijn J, Koutny M, Pietkiewicz-Koutny M, Rozenberg G (2013) Step semantics of boolean nets. Acta Inform 50(1):15–39

Koutny M, Pietkiewicz-Koutny M, Yakovlev A (2021) Asynchrony and persistence in reaction systems. Theor Comput Sci 881:97–110

Meski A, Koutny M, Penczek W (2018) Reaction mining for reaction systems. In: Unconventional computation and natural computation - 17th international conference, UCNC 2018, Fontainebleau, France, Proceedings, pp 131–144

Męski A (2020) Model checking for reaction and multi-agent systems. PhD thesis, PhD thesis. Institute of Computer Science, Polish Academy of Sciences

Meski A, Koutny M, Penczek W (2019) Model checking for temporal-epistemic properties of distributed reaction systems. School of Computing Technical Report Series

Papadimitriou CH (1994) Computational complexity. Addison-Wesley

Formenti E, Manzoni L, Porreca AE (2014) Cycles and global attractors of reaction systems. In: International workshop on descriptional complexity of formal systems, Springer, pp 114–125

Dennunzio A, Formenti E, Manzoni L, Porreca AE (2016) Reachability in resource-bounded reaction systems. In: International conference on language and automata theory and applications, Springer, pp 592–602

Baier C, Katoen J (2008) Principles of model checking. MIT Press

Kroening D, Strichman O (2016) Decision procedures - an algorithmic point of view, 2nd edn. Texts in Theoretical Computer Science, An EATCS Series

Biere A, Cimatti A, Clarke EM, Zhu Y (1999) Symbolic model checking without BDDs. In: Proceedings of the 5th international conference on tools and algorithms for construction and analysis of systems. TACAS ’99, pp 193–207

Biere A, Heljanko K, Junttila TA, Latvala T, Schuppan V (2006) Linear encodings of bounded LTL model checking. Log Methods Comput Sci 2(5):1

Clarke E, Grumberg O, Peled D (1999) Model checking. MIT Press

Moura L, Bjørner N (2008) Z3: An efficient SMT solver. In: Proceedings of the 14th international conference on tools and algorithms for construction and analysis of systems. TACAS, pp 337–340

Alur R, Henzinger TA, Vardi MY (1993) Parametric real-time reasoning. In: Proceedings of the twenty-fifth annual ACM symposium on theory of computing, San Diego, CA, USA, pp 592–601

Hune T, Romijn J, Stoelinga M, Vaandrager F (2002) Linear parametric model checking of timed automata. J Log Algebra Program 52–53:183–220

Knapik M, Męski A, Penczek W (2015) Action synthesis for branching time logic: theory and applications. ACM Trans Embedded Comput Syst 14(4):64–16423

Jones AV, Knapik M, Penczek W, Lomuscio A (2012) Group synthesis for parametric temporal-epistemic logic. In: International conference on autonomous agents and multiagent systems, AAMAS 2012, (3 Volumes), pp 1107–1114

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.