Dependency Graphs and Mass Conservation in Reaction Systems ☆

Sepinoud Azimi[a], Cristian Gratie[a,*], Sergiu Ivanov[b], Ion Petre[a]

[a] *Turku Centre for Computer Science and Department of Information Technologies, Åbo Akademi University, Finland*
[b] *Université Paris Est – Créteil Val de Marne, France*

Abstract

Reaction systems is a new mathematical formalism inspired by the biological cell, which focuses on an abstract set-based representation of chemical reactions via facilitation and inhibition. In this article we focus on the property of mass conservation for reaction systems. We show that conservation of sets gives rise to a relation between the species, which we capture in the concept of the conservation dependency graph. We then describe an application of this relation to the problem of listing all conserved sets. We further give a sufficient negative polynomial criterion which can be used for proving that a set is not conserved. Finally, we present a simulator of reaction systems, which also includes an implementation of the algorithm for listing the conserved sets of a given reaction system.

Keywords: Reaction system, model checking, mass conservation, conserved set, conservation dependency graph, simulator

1. Introduction

Reaction systems is a framework inspired by the functioning of the living cells, which was originally introduced in [1]. This formalism focuses on reactions exclusively and only considers two basic ways in which they can interact: promotion and inhibition. Reaction systems are based upon two fundamental principles. The first one, referred to as the "threshold principle", states that,

☆This manuscript has been accepted for publication in Theoretical Computer Science. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all disclaimers that apply to the journal apply to this manuscript. A definitive version was subsequently published in Theoretical Computer Science (2015), DOI:10.1016/j.tcs.2015.02.014.

*Corresponding author

Email addresses: sazimi@abo.fi (Sepinoud Azimi), cgratie@abo.fi (Cristian Gratie), sergiu.ivanov@u-pec.fr (Sergiu Ivanov), ipetre@abo.fi (Ion Petre)

whenever a resource is available, it is available in unlimited amount. This implies in particular that no competition for resources takes place. The second principle, referred to as the "no-permanency principle", states that unless a resource is explicitly sustained by a process, it will vanish and thus it will not be present in the next state of the system.

One of the central features of reaction systems is that they are explicitly conceived as open-ended systems: the influence of the environment is represented as an inflow of resources (the context).

The research topics investigated in the domain of reaction systems are various [2], but they can generally be classified along two lines. The first line comprises the research focusing on the mathematical properties of reaction systems: the set functions they can implement, their state sequences, connections to Boolean functions, etc. (e.g., [3, 4, 5, 6, 7]). The second main line of research regards reaction systems as an instrument for biological modeling (e.g., [8, 9, 10]). Quite naturally, investigations along this second line led to the study of model checking for reaction systems. For example, in [11], the authors introduce a temporal logic to define and subsequently verify certain properties of reaction systems. They prove that the general model checking problem is PSPACE-complete. On the other hand, [8] starts with defining a series of biologically inspired properties for reaction systems and shows that checking some of them, while still intractable, is a problem of lower computational complexity.

In this paper we conduct a detailed study of the biologically inspired property of mass conservation in reaction systems, originally introduced and shown to be coNP-complete in [8]. We get a new insight into the connection between the internal structure of the reaction system and mass conservation by revealing a relation that the latter induces between the species, and we capture this relation by defining the conservation dependency graph. We then present an application of this graph to the inherently difficult problem of listing the conserved sets and show that, in certain cases, the algorithm we devise to solve this problem is capable of performing better than the naive exponential approach. We continue by regarding mass conservation from a yet another perspective and formulate a sufficient polynomial criterion which allows one to quickly decide that a given set of species is not conserved. Finally, we present the reaction system simulator we have developed with the goal of automating the process of running reaction systems, and which is also capable of building the conservation dependency graph of a reaction system and of using it to list the conserved sets.

This paper is structured as follows. In Section 2 we remind the basic notions of reaction systems, as well as the notion of mass conservation. In Section 3 we discuss the relationship between mass conservation and the inner structure of the reaction system, and introduce the conservation dependency graph. In Section 4 we describe the algorithm for listing the conserved sets, which is based on the conservation dependency graph. In Section 5 we provide a negative polynomial heuristics for mass conservation, as well as for a generalized conservation problem. Finally, in Section 6 we give a short presentation of our reaction systems simulator. We conclude the paper in Section 7 with a discussion of our work.

2. Preliminaries

In this section we remind the notion of a reaction system as well as some related concepts capturing the static structure and the dynamic aspects of the model. For the original introduction the reader is referred to [1] and [3].

Definition 2.1 ([1]). *Let S be a finite set, whose elements will be referred to as* species *(very often in the reaction systems literature they are also called* entities*). A* reaction *a in S is a triplet of finite sets $a = (R_a, I_a, P_a)$, where $R_a, I_a, P_a \subseteq S$ and $R_a \cap I_a = \varnothing$. We say that R_a, I_a, and P_a are the sets of* reactants*,* inhibitors*, and* products *of a, respectively. The set of all reactions in S is denoted by* $\mathrm{rac}(S)$.

A reaction system (RS) *is an ordered pair $\mathcal{A} = (S, A)$, where S is a finite set of species and $A \subseteq \mathrm{rac}(S)$. The set S is called the* background *(set) of $\mathcal{A}$.*

We use the following notations of [8]:

$$\mathcal{R} = \bigcup_{a \in A} R_a, \mathcal{P} = \bigcup_{a \in A} P_a, \text{ and } \mathrm{supp}(\mathcal{A}) = \mathcal{R} \cup \mathcal{P}.$$

The set $\mathrm{supp}(\mathcal{A})$ *will be called the* support set *of $\mathcal{A}$.*

The following definition introduces the result of a reaction and of a reaction system.

Definition 2.2 ([1]). *Let $\mathcal{A} = (S, A)$ be a reaction system, $W \subseteq S$ a set of species, and $a \in A$ a reaction. We say that a is* enabled *by W, denoted by* $\mathrm{en}_a(W)$*, if $R_a \subseteq W$ and $I_a \cap W = \varnothing$.*

(1) The result of a on W *is defined as follows:*

$$\mathrm{res}_a(W) = \begin{cases} P_a, & \text{if } \mathrm{en}_a(W), \\ \varnothing, & \text{otherwise.} \end{cases}$$

(2) The result of $\mathcal{A}$ on W *is defined as follows:*

$$\mathrm{res}_{\mathcal{A}}(W) = \bigcup_{a \in A} \mathrm{res}_a(W).$$

Next, we introduce a running example for this section, as well as Sections 3 and 4.

Example 2.1. *One of the best preserved defense mechanisms in the living cell is the* heat shock response*. Whenever the cell is exposed to environmental stress, its proteins start to misfold, which may eventually lead to cell death. The heat shock response mechanism causes an increase in the production of molecular chaperons called the heat shock response proteins (*hsp*). These chaperons bind to misfolded proteins and facilitate their refolding. A different group of proteins, the heat shock factors (*hsf*), control* hsp *expression by binding to the promoter site of the* hsp*-encoding gene (the heat-shock element* hse*) and thus activate the*

transcription of hsp. *A molecular model of such a mechanism is proposed in [12] and its corresponding reaction system based model is presented in [9]. In this paper we will consider the following simplified version of the model from [9]:*

$$
\begin{aligned}
S = {} & \{\mathrm{hsf}, \mathrm{hsp}, \mathrm{hse}, \mathrm{hsp{:}hsf}, \mathrm{hsf{:}hse}, \mathrm{d_I}\} \\
A = {} & \{(\{\mathrm{hsf}\}, \{\mathrm{hsp}\}, \{\mathrm{hsf}\}), \\
& \quad (\{\mathrm{hsp}, \mathrm{hsf}\}, \{\mathrm{d_I}\}, \{\mathrm{hsp{:}hsf}\}), \\
& \quad (\{\mathrm{hsp{:}hsf}\}, \{\mathrm{d_I}\}, \{\mathrm{hsp}, \mathrm{hsf}\}), \\
& \quad (\{\mathrm{hse}\}, \{\mathrm{hsf}\}, \{\mathrm{hse}\}), \\
& \quad (\{\mathrm{hsf}, \mathrm{hse}\}, \{\mathrm{hsp}\}, \{\mathrm{hsf{:}hse}\}), \\
& \quad (\{\mathrm{hsf{:}hse}\}, \{\mathrm{hsp}\}, \{\mathrm{hsf{:}hse}, \mathrm{hsp}\}), \\
& \quad (\{\mathrm{hsp}, \mathrm{hsf}, \mathrm{hse}\}, \{\mathrm{d_I}\}, \{\mathrm{hsp{:}hsf}, \mathrm{hse}\}), \\
& \quad (\{\mathrm{hsp}, \mathrm{hsf{:}hse}\}, \{\mathrm{d_I}\}, \{\mathrm{hsp{:}hsf}, \mathrm{hse}\})\}
\end{aligned}
$$

For this particular example, the support set contains all the species except for $\mathrm{d_I}$*:*

$$\mathrm{supp}(\mathcal{A}) = \{\mathrm{hsf}, \mathrm{hsp}, \mathrm{hse}, \mathrm{hsp{:}hsf}, \mathrm{hsf{:}hse}\}$$

We now recall the notion of mass conservation in reaction systems.

Definition 2.3 ([8])**.** *Let* $\mathcal{A} = (S, A)$ *be a reaction system. We say that a set* $M \subseteq \mathrm{supp}(\mathcal{A})$ *is* conserved *if for any* $W \subseteq \mathrm{supp}(\mathcal{A})$*,* $M \cap W \neq \varnothing$ *if and only if* $M \cap \mathrm{res}_{\mathcal{A}}(W) \neq \varnothing$*.*

The biological intuition behind a set M being conserved is that the entities in M represent different forms of the same species (or of several closely related species), e.g., a gene promoter region G and the same promoter region G bound to a transcription factor T. The fact that M is conserved is thus interpreted as follows: if the group of species encoded by M is present in the system (i.e. at least one entity from M is present), then it will be present in the next state as well (possibly via different entities of M). Furthermore, elements from M are not produced by the system in the next state unless some entity of M is already present in the current state.

Note that mass conservation has been defined with respect to the support set so as to exclude elements of the background set which can only be provided via the context, with the intuition that such species would inevitably hinder the satisfaction of conservation properties for reaction systems. We define here a generalization of mass conservation that allows one to consider a different set of elements that can be reasonably excluded from the sets tested for conservation.

Definition 2.4 (Parameterized conservation)**.** *Let* $\mathcal{A} = (S, A)$ *be a reaction system and* $T \subseteq S$ *a set of species. A set* $M \subseteq T$ *is* conserved with respect to T *if, for any* $W \subseteq T$*, it holds that* $M \cap W \neq \varnothing$ *if and only if* $M \cap \mathrm{res}_{\mathcal{A}}(W) \neq \varnothing$*. We use* $\mathrm{cons}(\mathcal{A}, T)$ *to refer to all sets that are conserved with respect to* T*.*

Note that the original definition of mass conservation (Definition 2.3) corresponds to parameterized conservation with respect to $T = \text{supp}(\mathcal{A})$.

Furthermore, it can be shown that in order to compute the conserved sets with respect to a given T we can, instead, find conserved sets with respect to the background set in a different RS. We start by defining the projection of a reaction system.

Definition 2.5. *Let $\mathcal{A} = (S, A)$ be a reaction system and $T \subseteq S$ a set of species. For a reaction $a \in A$ that satisfies $R_a \subseteq T$, we define its* projection *onto T as* $\text{proj}_T(a) = (R_a, I_a \cap T, P_a \cap T)$.

We define the projection *of $\mathcal{A}$ onto T as* $\text{proj}_T(\mathcal{A}) = (T, A')$ *where:*

$$A' = \{\text{proj}_T(a) \mid a \in A \wedge R_a \subseteq T\}$$

Lemma 2.1. *Let $\mathcal{A} = (S, A)$ be a reaction system, $T \subseteq S$ a set of species and $\mathcal{A}' = \text{proj}_T(\mathcal{A})$ the projection of $\mathcal{A}$ onto T. Then, for any set $W \subseteq T$, we have that $res_{\mathcal{A}'}(W) = res_{\mathcal{A}}(W) \cap T$.*

Proof. Consider an arbitrary reaction $a \in A$ and a set $W \subseteq T$. If $R_a \not\subseteq T$, then a has no corresponding reaction in $\mathcal{A}'$, since $\text{proj}_T(a)$ is not defined. But in this case note that a is not enabled for W in $\mathcal{A}$, since $W \subseteq T$ and $R_a \not\subseteq T$.

If $R_a \subseteq T$, then consider $a' = \text{proj}_T(a)$ and note that $en_a(W) = en_{a'}(W)$. Indeed, since $W \subseteq T$, we have that $W \cap I_a = \varnothing \Leftrightarrow W \cap (I_a \cap T) = \varnothing$. Since the reaction a' only produces $P_a \cap T$ rather than P_a, we obtain the desired result $res_{\mathcal{A}'}(W) = res_{\mathcal{A}}(W) \cap T$. □

Note that the meaning of Lemma 2.1 is that the projection of a reaction system onto a set T preserves the behavior of the result function with respect to T. This property enables us to reduce the problem of parameterized conservation to computing conserved sets with respect to the full background set.

Theorem 2.2. *For any reaction system $\mathcal{A} = (S, A)$ and any set of species $T \subseteq S$, we have that* $\text{cons}(\mathcal{A}, T) = \text{cons}(\text{proj}_T(\mathcal{A}), T)$.

Proof. Let $\mathcal{A}' = \text{proj}_T(\mathcal{A})$. Based on Lemma 2.1, we have that $M \cap \text{res}_{\mathcal{A}'}(W) = M \cap \text{res}_{\mathcal{A}}(W)$, for any $M \subseteq T$, so the conserved sets with respect to T are the same in the two reaction systems, i.e. $\text{cons}(\mathcal{A}, T) = \text{cons}(\mathcal{A}', T)$. □

Note that, for any reaction system $\mathcal{A} = (S, A)$, its projection onto a set $T \subseteq S$ uses T as background set. Based on this, we will consider, throughout the rest of this paper, only the problem of finding sets that are conserved with respect to the background set (denoted by $\text{cons}(\mathcal{A})$ instead of $\text{cons}(\mathcal{A}, S)$), but having all results implicitly applicable both for the parameterized conservation with respect to arbitrary sets T (by relying on Theorem 2.2) as well as for the original definition of mass conservation (by taking $T = \text{supp}(\mathcal{A})$ in Theorem 2.2).

It should be noted, though, that the projection $\mathcal{A}' = \text{proj}_T(\mathcal{A})$ may include reactions with empty inhibitor or product sets, even if $\mathcal{A}$ does not have them.

Therefore, in this paper we remove the usual requirement that all the three sets defining reactions need to be nonempty [9, 2]. This is in line with the observation that reactions that cannot be inhibited by the sets taken into consideration (e.g. subsets of the support set) are crucial for mass conservation [8].

Going back to Example 2.1, let us consider the projection of the reaction system onto its support set. This translates in this case to the removal of $\mathrm{d_I}$ from the inhibitor sets. We obtain $\mathrm{proj}_{\mathrm{supp}(\mathcal{A})}(\mathcal{A}) = (S', A')$, where:

$$\begin{aligned}
S' &= \{\mathrm{hsf}, \mathrm{hsp}, \mathrm{hse}, \mathrm{hsp{:}hsf}, \mathrm{hsf{:}hse}\} \\
A' &= \{(\{\mathrm{hsf}\}, \{\mathrm{hsp}\}, \{\mathrm{hsf}\}), \\
&\qquad (\{\mathrm{hsp}, \mathrm{hsf}\}, \{\}, \{\mathrm{hsp{:}hsf}\}), \\
&\qquad (\{\mathrm{hsp{:}hsf}\}, \{\}, \{\mathrm{hsp}, \mathrm{hsf}\}), \\
&\qquad (\{\mathrm{hse}\}, \{\mathrm{hsf}\}, \{\mathrm{hse}\}), \\
&\qquad (\{\mathrm{hsf}, \mathrm{hse}\}, \{\mathrm{hsp}\}, \{\mathrm{hsf{:}hse}\}), \\
&\qquad (\{\mathrm{hsf{:}hse}\}, \{\mathrm{hsp}\}, \{\mathrm{hsf{:}hse}, \mathrm{hsp}\}), \\
&\qquad (\{\mathrm{hsp}, \mathrm{hsf}, \mathrm{hse}\}, \{\}, \{\mathrm{hsp{:}hsf}, \mathrm{hse}\}), \\
&\qquad (\{\mathrm{hsp}, \mathrm{hsf{:}hse}\}, \{\}, \{\mathrm{hsp{:}hsf}, \mathrm{hse}\})\}
\end{aligned}$$

Notice that indeed we now have empty inhibitor sets for some of the reactions. This is consistent with the use of $\mathrm{d_I}$ in [2] as a so called "dummy inhibitor" meant only to ensure compliance with the more restrictive version of the definition of reaction systems.

3. From Mass Conservation Relations to Dependency Graphs

In this section we aim to gain a better understanding of mass conservation in reaction systems by relating it to the inner structure induced by reactions. We start by first translating the reactions to a graph that completely characterizes the behavior of the system.

Definition 3.1. *Let $\mathcal{A} = (S, A)$ be a reaction system. The* behavior graph *of $\mathcal{A}$ is defined as $G_b = (V_b, E_b)$, with $V_b = 2^S$ and $E_b = \big\{\big(W, \mathrm{res}_{\mathcal{A}}(W)\big) \mid W \subseteq S\big\}$.*

Note that the behavior graph is in fact the *phase space* of the reaction system, i.e. it contains all the possible states that the system can be in, with edges denoting the transition from one state to another. Moreover, the behavior graph only encodes the result function $\mathrm{res}_{\mathcal{A}}$. In particular, it is possible to have different reaction systems that translate to the same behavior graph (such systems are said to be functionally equivalent [1]). For the reaction system from Example 2.1 the behavior graph is depicted in Figure 1.

Consider now a conserved set M. For any state W we have that M either intersects both W and $\mathrm{res}_{\mathcal{A}}(W)$ or is disjoint from both of them. A similar property can be formulated for M with respect to the connected components of the behavior graph. Before showing how this can be achieved, we give several graph-theoretic definitions.

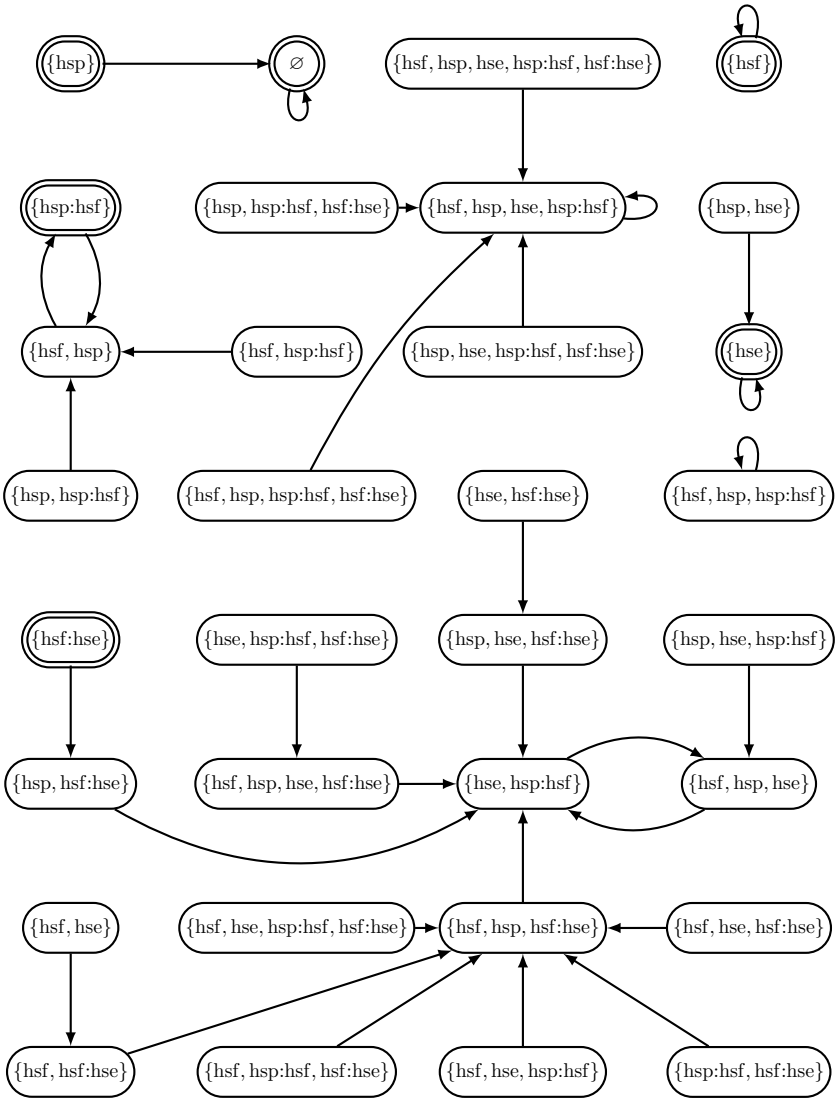


Figure 1: Behavior graph for the HSR model (Example 2.1). Double line borders correspond to singleton sets and to the empty set.

Definition 3.2. *Let $G = (V, E)$ be a directed graph. We say that a node v is connected to a node u if there is a (possibly degenerate) undirected path from u to v in G. Connectedness defined in this way is an equivalence relation. We refer to its equivalence classes as* connected components *and we use $CC_G(u)$ to denote the connected component that contains u. Furthermore, we denote the set of all connected components of G by $CCS_G = \{CC_G(u) \mid u \in V\}$.*

Note that our notion of connected components differs slightly from its standard use in the sense that we only refer to sets of nodes instead of induced subgraphs. For a further introduction to graph theory, we refer the reader to [13].

In the following definition we introduce a formal notation for saying that a given set M intersects all (or none) of the sets from a collection of sets $\mathcal{C}$.

Definition 3.3. *Let S be an arbitrary finite set and consider a set $M \subseteq S$ and a collection of sets $\mathcal{C} \subseteq 2^S$. We use the notation $M \sqcap \mathcal{C} = \{T \in \mathcal{C} \mid M \cap T \neq \varnothing\}$ to refer to the collection of those sets in $\mathcal{C}$ which intersect M.*

We say that M intersects *$\mathcal{C}$ if M intersects every element of $\mathcal{C}$, i.e. $M \sqcap \mathcal{C} = \mathcal{C}$. We say that M is* disjoint *from $\mathcal{C}$ if M is disjoint from every element of $\mathcal{C}$, i.e. $M \sqcap \mathcal{C} = \varnothing$. We say that M is* consistent *with $\mathcal{C}$ if M intersects $\mathcal{C}$ or is disjoint from $\mathcal{C}$.*

Proposition 3.1. *Let $\mathcal{A} = (S, A)$ be a reaction system and $G_b = (V_b, E_b)$ its behavior graph. For any set $M \subseteq S$, the following two statements are equivalent:*

(1) M is conserved,

(2) M is consistent with every connected component $\mathcal{C} \in CCS_{G_b}$.

Proof. The implication (2) $\Rightarrow$ (1) follows from the fact that, for every set $W \subseteq S$, we have $res_{\mathcal{A}}(W) \in CC_{G_b}(W)$.

We now prove that (1) $\Rightarrow$ (2). Let $\mathcal{C}$ be a connected component of the behavior graph G_b. Assume that M is a conserved set for which there exist $W_1, W_2 \in \mathcal{C}$ such that $M \cap W_1 = \varnothing$ and $M \cap W_2 \neq \varnothing$. Since W_1 and W_2 are in the same connected component, there is a path of nodes connecting them, i.e., there exist $V_1, \ldots, V_n$ such that $V_1 = W_1$, $V_n = W_2$ and, for all i with $1 \leq i \leq n-1$, $(V_i, V_{i+1}) \in E$ or $(V_{i+1}, V_i) \in E$. But then, in both cases, it must be that $M \cap V_i \neq \varnothing \Leftrightarrow M \cap V_{i+1} \neq \varnothing$, so we can via transitivity conclude that $M \cap W_1 \neq \varnothing \Leftrightarrow M \cap W_2 \neq \varnothing$, which contradicts our assumption and completes the proof. □

Note that, by the result presented in Proposition 3.1, the conservation of a given set M only depends on the connected components of the behavior graph and not on its edges or their direction. This means that even fairly different reaction systems may end up having the same conserved sets or, in other words, equivalence with respect to conserved sets is a lot weaker than functional equivalence.

For the heat shock response model, one conserved set is $M = \{\text{hsf:hse}, \text{hse}\}$. To see that indeed this is the case, by Proposition 3.1, note in Figure 1 that the connected components of $\varnothing$, $\{\text{hsf}\}$, $\{\text{hsp:hsf}\}$ and $\{\text{hsf}, \text{hsp}, \text{hsp:hsf}\}$ intersect M, whereas the other connected components of the behavior graph are disjoint from M.

3.1. Conservation dependency graph

In what follows we aim to further investigate the properties of conserved sets in relation with the connected components of the behavior graph.

Proposition 3.2. *Let $\mathcal{A} = (S, A)$ be a reaction system and $G_b = (V_b, E_b)$ its behavior graph. Consider an arbitrary element $x \in S$ and let $\mathcal{C}_x$ be the connected component that contains the singleton set $\{x\}$, i.e. $\mathcal{C}_x = CC_{G_b}(\{x\})$. Similarly, take $\mathcal{C}_\varnothing = CC_{G_b}(\varnothing)$. We denote, for any collection $\mathcal{C}$, $\mathrm{cover}(\mathcal{C}) = \bigcup_{T \in \mathcal{C}} T$.*

(1) If $x \in \mathrm{cover}(\mathcal{C}_\varnothing)$, then x is not contained in any conserved set of $\mathcal{A}$, i.e. $\{x\} \sqcap \mathrm{cons}(\mathcal{A}) = \varnothing$.

(2) If $\mathrm{cover}(\mathcal{C}_x) = S$, then x is contained in all nonempty conserved sets of $\mathcal{A}$, i.e. $\{x\} \sqcap \mathrm{cons}(\mathcal{A}) = \mathrm{cons}(\mathcal{A}) \setminus \{\varnothing\}$.

(3) For every $y \in \mathrm{cover}(\mathcal{C}_x)$ and for every conserved set M, if $x \notin M$, then $y \notin M$, (or, equivalently, $y \in M$ implies $x \in M$), i.e. $\{y\} \sqcap \mathrm{cons}(\mathcal{A}) \subseteq \{x\} \sqcap \mathrm{cons}(\mathcal{A})$.

Proof. (1) Let M be an arbitrary conserved set. Then, from Proposition 3.1, it follows that M must be consistent with $\mathcal{C}_\varnothing$. Since $\varnothing \in \mathcal{C}_\varnothing$ and $M \cap \varnothing = \varnothing$, it must be that $M \sqcap \mathcal{C}_\varnothing = \varnothing$. In particular, we must also have $M \cap \{x\} = \varnothing$, which means that $x \notin M$.

(2) Let M be a nonempty conserved set. Then, since $M \cap S \neq \varnothing$, it must be that M intersects a set from $\mathcal{C}_x$. But from Proposition 3.1 we know that M must be consistent with $\mathcal{C}_x$ and, thus, it must be that $M \sqcap \mathcal{C}_x = \mathcal{C}_x$. In particular, we must also have $M \cap \{x\} \neq \varnothing$, which is equivalent to $x \in M$.

(3) Let M be a conserved set such that $x \notin M$. Then $M \cap \{x\} = \varnothing$ and, since M must be consistent with $\mathcal{C}_x$, it must be that $M \sqcap \mathcal{C}_x = \varnothing$, which implies that $M \cap \mathrm{cover}(\mathcal{C}_x) = \varnothing$. In particular, this means that $y \notin M$. □

As we have seen, the conserved sets of a given RS only depend on the connected components of the behavior graph, i.e. on the partition induced by the reactions on the state space. In this context, Proposition 3.2 extracts properties of conserved sets by examining particular states and their connected components.

For example, the first two statements give us sufficient conditions for an element x to be in no conserved set, respectively in all nonempty conserved sets. Note that there is also an interesting interplay between the two statements when there exists an x such that $\mathrm{cover}(\mathcal{C}_x) = S$ and $\varnothing \in \mathcal{C}_x$. Indeed, the latter is equivalent to having $\mathcal{C}_x = \mathcal{C}_\varnothing$, which means that $\mathrm{cover}(\mathcal{C}_\varnothing) = S$, so no element of S can be part of a conserved set. On the other hand, the former property, $\mathrm{cover}(\mathcal{C}_x) = S$, which translates to x being part of all nonempty conserved sets, is still (trivially) true since the only conserved set in this case is the empty set.

The more important implication of the previous remark is that, for the standard definition of reaction systems, where empty inhibitor sets are not allowed in reactions, there can be no nonempty conserved set at all. Indeed, for such reaction systems it holds that $\mathrm{res}_{\mathcal{A}}(S) = \varnothing$, which leads to $\mathrm{cover}(\mathcal{C}_\varnothing) = S$.

For the reaction system of Example 2.1, we have 7 connected components. In particular, note that $\mathcal{C}_\varnothing = \{\varnothing, \{\text{hsp}\}\}$, which means that hsp cannot be part of any conserved set. Furthermore, we have that $\text{cover}(\mathcal{C}_{\text{hsf:hse}})$ is the full background set, so hsf:hse will be part of every nonempty conserved set.

The third claim of Proposition 3.2 defines a dependency relation between the elements of the reaction system with respect to mass conservation. The statement implies that, for a pair of species (x, y) such that $y \in \text{cover}(\mathcal{C}_x)$, a conserved set that does not contain x cannot contain y or, equivalently, any conserved set that contains y must contain x as well. We can capture this dependency between species in a directed graph.

Definition 3.4. *Let $\mathcal{A} = (S, A)$ be a reaction system and $G_b = (V_b, E_b)$ its behavior graph. The* conservation dependency graph $G_{cd} = (V_{cd}, E_{cd})$ *of $\mathcal{A}$ is given by $V_{cd} = S$ and $E_{cd} = \{(x, y) \mid x \in S \wedge y \in \text{cover}(\mathcal{C}_x)\}$.*

Consider again our running example. Based on the connected components of singleton sets, highlighted with double borders in Figure 1, we can compute the conservation dependency graph. The result is shown in Figure 2.

Intuitively, every conserved set should satisfy all the constraints that are encoded by the conservation dependency graph. Alternatively, we can focus on the conservation dependency graph alone and consider all the sets that are consistent with the aforementioned constraints. In what follows, we capture the constraints of Proposition 3.2 (3), for arbitrary directed graphs, via the concept of source sets.

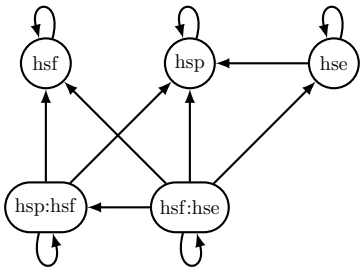


Figure 2: Conservation dependency graph for the HSR model (Example 2.1).

Definition 3.5. *Let $G = (V, E)$ be a directed graph. A set $S \subseteq V$ is a* source set *of G if $E \cap (V \setminus S) \times S = \varnothing$, i.e. all edges of G that cross S (if any) do so from S to $V \setminus S$. We denote the set of all source sets of G by $\sigma(G)$.*

It follows immediately from the definition that, for any graph $G = (V, E)$, both $\varnothing$ and V are source sets of G. The correspondence between the conserved sets of a reaction system and the source sets of its conservation dependency graph is given in Proposition 3.3.

Proposition 3.3. *Any conserved set M of a reaction system $\mathcal{A}$ is a source set in the conservation dependency graph of $\mathcal{A}$.*

Proof. The result follows from claim (3) of Proposition 3.2 and the definition of the conservation dependency graph. □

3.2. Computing the source sets of a directed graph

In this subsection we are concerned with the computation of source sets for general directed graphs. We start by investigating the interplay between source sets and the graph structure.

In what follows, we will say that a node u is an *ancestor* of a node v or, equivalently, that v is a *descendant* of u, if there exists a directed path from u to v.

Proposition 3.4. *Let $G = (V, E)$ be a directed graph and let S be an arbitrary source set of G.*

(1) The parent of a node that is in S is also in S, i.e. for every two nodes u and v we have $v \in S \wedge (u, v) \in E \Rightarrow u \in S$.

(2) The child of a node that is not in S cannot be in S either, i.e. for every two nodes u and v we have $u \notin S \wedge (u, v) \in E \Rightarrow v \notin S$.

(3) The ancestor of a node that is in S is also in S.

(4) The descendant of a node that is not in S is not in S either.

Proof. The negation of either (1) or (2) directly violates the definition of source sets by providing an edge (u, v) that goes from $V \setminus S$ to S. Furthermore, (3) and (4) can be proved by induction from (1) and (2), respectively. □

We are going to relate source sets to the strongly connected components of the graph under consideration.

Definition 3.6. *Let $G = (V, E)$ be a directed graph. Two nodes $u, v \in V$ are said to be* strongly connected *if there exist in G a directed path from u to v and a directed path from v to u. Strong connectedness defined in this way is an equivalence relation. We refer to its equivalence classes as* strongly connected components *and use $SCC_G(u)$ to refer to the strongly connected component that contains u. Furthermore, we denote the set of all strongly connected components of G by $SCCS_G$, i.e. $SCCS_G = \{SCC_G(u) \mid u \in V\}$.*

Note that, by definition, the strongly connected components of a directed graph are disjoint sets. It is not difficult to see by Proposition 3.4 that the source sets cannot split the strongly connected components of a graph.

Proposition 3.5. *Let $G = (V, E)$ be a directed graph, $C \in SCCS_G$ a strongly connected component of G and S a source set of G. If $C \cap S \neq \varnothing$, then $C \subseteq S$.*

Proof. Choose $u \in C \cap S$. From Proposition 3.4 it follows that all ancestors of u must be in S as well. In particular, this implies that $C \subseteq S$. □

Corollary 3.6. *Any source set of a graph G is a union of (disjoint) strongly connected components of G.*

Proof. Let S be an arbitrary source set of G. For every $u \in S$, we have by Proposition 3.5 that $SCC_G(u) \subseteq S$, so we can write $S = \bigcup_{u \in S} SCC_G(u)$. □

Thus, we have seen that all source sets are unions of strongly connected components. In order to see exactly which of such unions are source sets, we will refer to the condensation of G, i.e. the graph obtained by replacing each strongly connected component of G with a single node.

Definition 3.7. *Let $G = (V, E)$ be a directed graph. The* condensation *of G is the directed graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ whose nodes are the strongly connected components of G, i.e. $\widetilde{V} = SCCS_G$, and whose edges are defined as follows: $\widetilde{E} = \{(U, W) \in \widetilde{V} \times \widetilde{V} \mid U \neq W \wedge \exists u \in U, \exists w \in W, (u, w) \in E\}$, i.e. there is an edge (U, W) in $\widetilde{G}$ iff there is an edge in G from an element of U to an element of W.*

Note that the condensation graph is a directed acyclic graph (DAG). We illustrate the concept by showing, in Figure 3, the condensation of the conservation dependency graph for our running example. Note that for this particular case the condensation is almost identical to the original graph, only the nodes correspond now to singleton sets and the self-loops are no longer present. This very close similarity is due to the fact that the conservation dependency graph, shown in Figure 2, does not contain any cycles aside from self-loops.

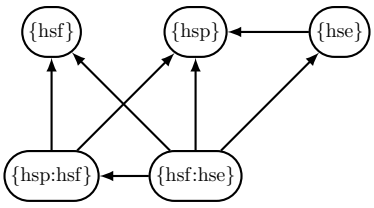


Figure 3: Condensation of the conservation dependency graph for the HSR model (Example 2.1).

Proposition 3.7. *Let $G = (V, E)$ be a directed graph and $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ its condensation. A set $S \subseteq V$ is a source set of G iff there exists a set $\widetilde{S} \subseteq \widetilde{V}$ such that $S = \mathrm{cover}(\widetilde{S})$ and $\widetilde{S}$ is a source set of $\widetilde{G}$.*

Proof. We start with the forward implication. We know already from Corollary 3.6 that there exists $\widetilde{S} \subseteq \widetilde{V}$ such that $S = \mathrm{cover}(\widetilde{S})$. Assume that $\widetilde{S}$ is not a source set in $\widetilde{G}$. Then there exist $U \in \widetilde{S}$ and $W \in \widetilde{V} \setminus \widetilde{S}$ such that $(W, U) \in \widetilde{E}$, which means that there exist $u \in U$ and $w \in W$ such that $(w, u) \in E$. But this contradicts the fact that S is a source set, since $u \in S$ and $w \in V \setminus S$. Thus, it must be that $\widetilde{S}$ is a source set of $\widetilde{G}$.

For the reverse implication, consider a source set $\widetilde{S}$ of the condensation graph $\widetilde{G}$ and let $S = \text{cover}(\widetilde{S})$. Assume that S is not a source set of G. Then there exist two nodes $u \in S$ and $w \in V \setminus S$ such that $(w, u) \in E$. Since strongly connected components are either fully contained in a source set or disjoint from it (by Proposition 3.5), it must be that $SCC_G(u) \in \widetilde{S}$ and $SCC_G(w) \in \widetilde{V} \setminus \widetilde{S}$. Furthermore, since $(w, u) \in E$, we have $(SCC_G(w), SCC_G(u)) \in \widetilde{E}$, which contradicts the fact that $\widetilde{S}$ is a source set of $\widetilde{G}$. Thus, it must be that S is a source set of G. □

The practical conclusion we can draw from Proposition 3.7 is that it suffices to have an algorithm for computing the source sets of directed acyclic graphs (DAG's) and use it on the condensation graph.

In what follows, we will use $G{\downarrow_S}$ to denote the restriction of the graph $G = (V, E)$ to a subset of nodes $S \subseteq V$, i.e. $G{\downarrow_S} = (S, E \cap (S \times S))$. We will also use $\text{desc}_G(S)$ to refer to the set containing all the nodes from S and all their descendants. Similarly, we will use $\text{anc}_G(S)$ to refer to the set containing all nodes from S and all their ancestors in G.

Theorem 3.8. *Let $G = (V, E)$ be a directed graph, $T \subseteq V$ an arbitrary set of nodes and $s \in V$ an arbitrary node from G.*

(1) A source set S includes a set T if and only if $S \setminus \text{anc}_G(T)$ is a source set of $G{\downarrow_{V \setminus \text{anc}_G(T)}}$ and, in addition, S contains all the ancestors of elements from T:

$$S \in \sigma(G) \wedge T \subseteq S \Leftrightarrow S \setminus \text{anc}_G(T) \in \sigma(G{\downarrow_{V \setminus \text{anc}_G(T)}}) \wedge \text{anc}_G(T) \subseteq S.$$

(2) A source set S does not intersect a set T if and only if S is a source set in the graph obtained from G by removing all elements of T and their descendants:

$$S \in \sigma(G) \wedge S \cap T = \varnothing \Leftrightarrow S \in \sigma(G{\downarrow_{V \setminus \text{desc}_G(T)}}).$$

(3) Given a node s, all source sets of G can be computed recursively by relying on subgraphs of G that do not contain s:

$$\sigma(G) = \sigma(G{\downarrow_{V \setminus \text{desc}_G(\{s\})}}) \cup \{S \cup \text{anc}_G(\{s\}) \mid S \in \sigma(G{\downarrow_{V \setminus \text{anc}_G(\{s\})}})\}.$$

Proof. (1) We have:

$$\begin{aligned}
&S \in \sigma(G) \wedge T \subseteq S \\
\Leftrightarrow &E \cap (V \setminus S) \times S = \varnothing \wedge \text{anc}_G(T) \subseteq S \\
\Leftrightarrow &E \cap (V \setminus S) \times (S \setminus \text{anc}_G(T)) = \varnothing \wedge \text{anc}_G(T) \subseteq S \\
\Leftrightarrow &E \cap \big((V \setminus \text{anc}_G(T)) \setminus (S \setminus \text{anc}_G(T)) \big) \times (S \setminus \text{anc}_G(T)) = \varnothing \wedge \text{anc}_G(T) \subseteq S \\
\Leftrightarrow &S \setminus \text{anc}_G(T) \in \sigma(G{\downarrow_{V \setminus \text{anc}_G(T)}}) \wedge \text{anc}_G(T) \subseteq S.
\end{aligned}$$

The first equivalence follows from the definition of conserved sets and Proposition 3.4. The second one follows from the definition of $\mathrm{anc}_G(T)$, as there can be no edges of G going into this set. The third equivalence relies on the relation $A \setminus B = (A \setminus X) \setminus (B \setminus X)$, which holds whenever $X \subseteq B \subseteq A$. Finally, the last step is a direct application of the definition of source sets.

(2) We follow a similar approach and we have:

$$\begin{aligned} & S \in \sigma(G) \wedge S \cap T = \varnothing \\ \Leftrightarrow & E \cap (V \setminus S) \times S = \varnothing \wedge S \cap \mathrm{desc}_G(T) = \varnothing \\ \Leftrightarrow & E \cap ((V \setminus \mathrm{desc}_G(T)) \setminus S) \times S = \varnothing \wedge S \cap \mathrm{desc}_G(T) = \varnothing \\ \Leftrightarrow & S \in \sigma(G{\downarrow_{V \setminus \mathrm{desc}_G(T)}}). \end{aligned}$$

Just as before, the first equivalence follows directly from the definition of source sets and from Proposition 3.4. The second equivalence relies on the definition of $\mathrm{desc}_G(T)$, as there can be no edges of G going out of this set. Finally, we use the definition again to get the desired result.

(3) The result follows from (1) and (2) by noting that we can partition the source sets of G into those that contain s and those that do not contain it. We can thus write:

$$\begin{aligned} & S \in \sigma(G) \wedge s \in S \Leftrightarrow S \setminus \mathrm{anc}_G(\{s\}) \in \sigma(G{\downarrow_{V \setminus \mathrm{anc}_G(\{s\})}}) \wedge \mathrm{anc}_G(\{s\}) \subseteq S, \\ & S \in \sigma(G) \wedge s \notin S \Leftrightarrow S \in \sigma(G{\downarrow_{V \setminus \mathrm{desc}_G(\{s\})}}). \end{aligned}$$

These statements lead to the desired result. □

We can immediately apply the third claim of Theorem 3.8 to a source node of G (a node with no parents) and write an even simpler decomposition of the source sets of G into two parts.

Corollary 3.9. *Let $G = (V, E)$ be a directed graph and let $s \in V$ be a source node. Then we have:*

$$\sigma(G) = \sigma(G{\downarrow_{V \setminus \mathrm{desc}_G(\{s\})}}) \cup \{S \cup \{s\} \mid S \in \sigma(G{\downarrow_{V \setminus \{s\}}})\}.$$

We can translate the previous formal result into an algorithm for computing the source sets of a directed acyclic graph.

Algorithm 3.1 (source sets of a DAG)**.** *Let $G = (V, E)$ be a DAG. If the graph contains no nodes, return the empty set as the only source set. Otherwise choose a source node $s \in V$, compute the source sets of $G{\downarrow_{V \setminus \{s\}}}$ and $G{\downarrow_{V \setminus \mathrm{desc}_G(\{s\})}}$, then aggregate them according to Corollary 3.9 to obtain the source sets of G.*

Note that the fact that the graph is acyclic is required for the existence of the source node s.

4. Enumerating the Conserved Sets of a Reaction System

In this section we propose and discuss the advantages of an algorithm that relies on the conservation dependency graph to list all the conserved sets of a given reaction system.

4.1. An algorithm for enumerating all conserved sets

We provide here an algorithm for listing all conserved sets of a reaction system. The actual test for conservation relies on Proposition 3.1, whereas the candidate sets to be tested are computed using Algorithm 3.1, as well as the additional information coming from Proposition 3.2.

Algorithm 4.1 (compute all conserved sets)**.** *Let $\mathcal{A} = (S, A)$ be a reaction system.*

1. *Compute the behavior graph G_b of $\mathcal{A}$.*
2. *Compute the connected components of G_b and collect the following information:*
   (a) *compute $P_{out} = \mathrm{cover}(\mathcal{C}_\varnothing)$,*
   (b) *compute $P_{in} = \{x \in S \mid \mathrm{cover}(C_x) = S\}$.*
3. *Compute the conservation dependency graph G_{cd} of $\mathcal{A}$.*
4. *Compute the strongly connected components of G_{cd} and the condensation graph $\widetilde{G}_{cd}$. Collect the following additional information:*
   (a) *compute the set $\widetilde{P}_{out}$ of nodes from $\widetilde{G}_{cd}$ which contain elements of P_{out}, i.e. $\widetilde{P}_{out} = \{SCC_{G_{cd}}(x) \mid x \in P_{out}\}$,*
   (b) *compute the set $\widetilde{P}_{in}$ of nodes from $\widetilde{G}_{cd}$ which contain elements of P_{in}, i.e. $\widetilde{P}_{in} = \{SCC_{G_{cd}}(x) \mid x \in P_{in}\}$.*
5. *Compute the reduced condensation graph $\widetilde{G}'_{cd}$ by removing from $\widetilde{G}_{cd}$ all nodes from $\widetilde{P}_{out}$ with their descendants and all nodes from $\widetilde{P}_{in}$ with their ancestors:*
   (a) *compute the set $\widetilde{P}'_{out} = \mathrm{desc}_{\widetilde{G}_{cd}}(\widetilde{P}_{out})$,*
   (b) *compute the set $\widetilde{P}'_{in} = \mathrm{anc}_{\widetilde{G}_{cd}}(\widetilde{P}_{in})$,*
   (c) *compute the reduced condensation graph $\widetilde{G}'_{cd} = \widetilde{G}_{cd}{\downarrow}_{\widetilde{V}_{cd} \setminus (\widetilde{P}'_{out} \cup \widetilde{P}'_{in})}$.*
6. *Compute the source sets of $\widetilde{G}'_{cd}$ using Algorithm 3.1.*
7. *For each source set $\widetilde{T} \in \sigma(\widetilde{G}'_{cd})$, compute the corresponding source set of G_{cd} as $T = \mathrm{cover}(\widetilde{T} \cup \widetilde{P}'_{in})$.*
8. *For each source set T computed in step 7, test whether T is a conserved set by checking that it is consistent with all connected components of the behavior graph G_b.*
9. *Add $\varnothing$ to the list of conserved sets, if it was not obtained in the previous step.*

Theorem 4.1. *Algorithm 4.1 computes all conserved sets correctly.*

Proof. First, let us see that steps $1-3$ compute the data structures required for the rest of the algorithm, i.e. P_{in}, P_{out} and G_{cd}. Furthermore, steps $4-6$ translate to computing exactly the source sets of G_{cd} which include P_{in} and are disjoint from P_{out}. Indeed, in order to find all the source sets which are disjoint from P_{out}, we can rely on Proposition 3.7 to conclude that we also need to exclude the full strongly connected components of elements from P_{out}, then based on Theorem 3.8 (2) we must also exclude descendant strongly connected components. This translates to steps $4(a)$ and $5(a)$ of the algorithm, respectively. A similar justification holds for steps $4(b)$ and $5(b)$.

Now note that this algorithm relies on testing for conservation using Proposition 3.1, but only examines a reduced set of candidates by relying on Proposition 3.2 and Proposition 3.3.

The last step is required in order to ensure that the empty set is also included in the list of conserved sets since, whenever $P_{in} \neq \varnothing$, all the candidate sets tested in step 8 are nonempty. □

Remark that the decision problem for conserved sets is coNP-complete [8]. As such, we know already that we cannot test for conservation in polynomial time unless P = NP. On the other hand, we focus here on finding all conserved sets, which means that we can make use of aggregate information from the original reaction system in order to speed up the test for conservation. In particular, once we have the connected components of the behavior graph, we can simply forget about the reactions. Moreover, the analysis of the connected components of the empty set and singleton sets, together with the constraints encoded in the conservation dependency graph, enable us to reduce the actual number of candidates that we need to verify.

4.2. Efficiency of the algorithm

To understand the benefit of the strategy employed in Algorithm 4.1 and also the nature of the reactions systems for which it is efficient, we discuss in this subsection several examples.

We first consider the running example of the previous sections, the HSR model from Example 2.1. Recall that we have already seen the behavior graph in Figure 1, the conservation dependency graph in Figure 2, and the condensation of the conservation dependency graph in Figure 3. These graphs correspond to the execution of steps 1, 3 and 4 of the algorithm, respectively.

The result for step 2 is that $P_{out} = \{\text{hsp}\}$ and $P_{in} = \{\text{hsf:hse}\}$. Thus, when running step $5(a)$, we remove node {hsp} from the condensation graph (there are no descendants to remove in this case). Similarly, node {hsf:hse} is removed in step $5(b)$. The resulting graph is presented in Figure 4.

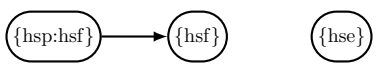


Figure 4: Reduced condensation graph for the HSR model (Example 2.1).

The reduced condensation graph has six source sets, which are shown below, together with the corresponding source sets of the original conservation dependency graph.

$$\begin{array}{ll}
\widetilde{T}_1 = \varnothing & T_1 = \{\text{hsf:hse}\} \\
\widetilde{T}_2 = \{\{\text{hsp:hsf}\}\} & T_2 = \{\text{hsf:hse}, \text{hsp:hsf}\} \\
\widetilde{T}_3 = \{\{\text{hsp:hsf}\}, \{\text{hsf}\}\} & T_3 = \{\text{hsf:hse}, \text{hsp:hsf}, \text{hsf}\} \\
\widetilde{T}_4 = \{\{\text{hse}\}\} & T_4 = \{\text{hsf:hse}, \text{hse}\} \\
\widetilde{T}_5 = \{\{\text{hsp:hsf}\}, \{\text{hse}\}\} & T_5 = \{\text{hsf:hse}, \text{hsp:hsf}, \text{hse}\} \\
\widetilde{T}_6 = \{\{\text{hsp:hsf}\}, \{\text{hsf}\}, \{\text{hse}\}\} & T_6 = \{\text{hsf:hse}, \text{hsp:hsf}, \text{hsf}, \text{hse}\}
\end{array}$$

Of the six source sets of the conservation dependency graph, only T_3, T_4 and T_6 are conserved. Note that T_3 corresponds to all the forms (free and occupied) of gene promoters (hse). Similarly, T_4 corresponds to the conservation of the total amount of heat shock factors (free, bound to the promoter or bound to the heat shock protein). The third conserved set, T_6, is in fact the union of the other two, so it does not bring additional information, since it is in general the case that the union of two conserved sets is also conserved.

Note that, out of the 32 possible sets, the algorithm enabled us to only test 6 candidates for conservation.

Example 4.1. *Consider the following reaction system:*

$$S = \{x, y, z\};$$
$$A = \{(\{x\}, \{z\}, \{y, z\}), (\{y\}, \varnothing, \{x\}), (\{x, z\}, \varnothing, \{y\}), (\{y\}, \{z\}, \{y\})\} \ .$$

The corresponding behavior graph G_b can be computed by finding $\operatorname{res}_{\mathcal{A}}(W)$ *for all $W \subseteq S$. The graph is illustrated in Figure 5.*

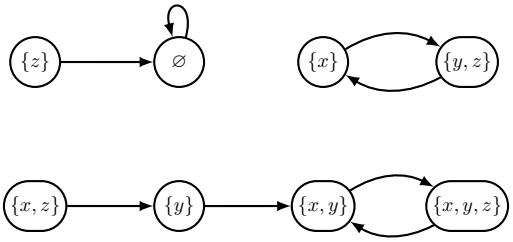
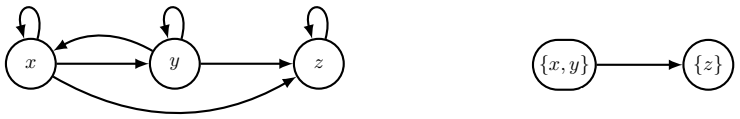


Figure 5: Behavior graph for the reaction system from Example 4.1.

The relevant connected components for our algorithm are:

$$\mathcal{C}_x = \{\{x\}, \{y, z\}\};$$
$$\mathcal{C}_y = \{\{y\}, \{x, y\}, \{x, z\}, \{x, y, z\}\};$$
$$\mathcal{C}_z = \{\{z\}, \varnothing\} = \mathcal{C}_{\varnothing}.$$



(a) (b)

Figure 6: (a) Conservation dependency graph and (b) its condensation, for the reaction system from Example 4.1.

The corresponding conservation dependency graph is presented in Figure 6a.

In step 5 *of the algorithm we rely on* $P_{out} = \{z\}$ *and* $P_{in} = \{x, y\}$ *(computed in steps* $2(a)$ *and* $2(b)$*, respectively) to obtain that all nonempty conserved sets must contain* x *and* y *and cannot contain* z*, which means that* $M = \{x, y\}$ *is the only nonempty conserved set for this reaction system. In particular, note that the algorithm will work on an empty graph at step* 6*.*

Furthermore, let us see that in this case the conservation dependency graph has only three source sets, namely $\varnothing$*,* $\{x, y\}$ *and* S*. Thus, even if we are to consider the full condensation of the conservation dependency graph, i.e. skip step* 5 *of the algorithm, we would still obtain a significant improvement since we need to examine only 3 out of the total number of 8 candidates.*

While the previous example is rather simple, it still reveals the main improvements that come from the approach. On the other hand, consider also the case where the behavior and conservation dependency graphs do not provide any useful information, i.e. $P_{out} = \varnothing$, $P_{in} = \varnothing$ and G_{cd} only has self-loops for each node, meaning that all sets are source sets.

Example 4.2. *Consider the reaction system* $\mathcal{A} = (S, A)$ *given by:*

$$\begin{aligned} S &= \{x_1, x_2, \ldots, x_n\} \\ A &= \{(\{x_i\}, \varnothing, \{x_i\}) \mid x_i \in S\} \end{aligned}$$

In this case the result function satisfies $\operatorname{res}_{\mathcal{A}}(W) = W$ *for all states* $W \subseteq S$*, i.e. all states are isolated and connected components contain a single state. In particular,* $\mathcal{C}_{\varnothing} = \{\varnothing\}$ *and* $\mathcal{C}_{x_i} = \{\{x_i\}\}$ *for all* $x_i \in S$*.*

Thus, for this example we have $P_{out} = P_{in} = \varnothing$ *and the conservation dependency graph has only self-loop edges* $E_{cd} = \{(x_i, x_i) \mid x_i \in S\}$*. This means that every subset of* S *is a source set, i.e. we need to examine all candidates. But in this case note that in fact all subsets of* S *are conserved.*

Example 4.3. *Consider the reaction system* $\mathcal{A} = (S, A)$ *given by:*

$$\begin{aligned} S &= \{x_1, x_2, \ldots, x_n\} \\ A &= \{(\{x_i\}, S \setminus \{x_i\}, \{x_i\}) \mid x_i \in S\} \\ &\quad \cup \{(\{x_i, x_j\}, \varnothing, S) \mid x_i, x_j \in S \wedge x_i \neq x_j\} \end{aligned}$$

The result function $\mathrm{res}_{\mathcal{A}}$ *satisfies:*

$$\mathrm{res}_{\mathcal{A}}(W) = \begin{cases} W, & \text{if } |W| \leq 1 \ , \\ S, & \text{if } |W| \geq 2 \ . \end{cases}$$

Just as in Example 4.2, this reaction system does not give useful information for reducing the number of candidates examined in Algorithm 4.1. Instead, we analyze the behavior graph in relation to Proposition 3.1.

Based on the result function, the behavior graph has $n+2$ *connected components in this case:* $\mathcal{C}_{\varnothing} = \{\varnothing\}$, $\mathcal{C}_{x_i} = \{\{x_i\}\}$ *for each* $x_i \in S$, *and one connected component containing all the other states, call it* $\mathcal{C}_S$.

We know that any conserved set M *must be consistent with all connected components of the behavior graph. For CC's that contain a single set, this holds trivially, so we only need to worry about* $\mathcal{C}_S$. *The empty set is always conserved, so we focus on* $M \neq \varnothing$. *Then* $M \cap S \neq \varnothing$, *so it must be that* $M \sqcap \mathcal{C}_S = \mathcal{C}_S$, *i.e.* M *intersects all the elements of* $\mathcal{C}_S$.

If more than two elements of S *are missing from* M, *then we can find* x_i, x_j *such that* $M \cap \{x_i, x_j\} = \varnothing$, *but* $\mathrm{res}_{\mathcal{A}}(\{x_i, x_j\}) = S$, *so* M *is not conserved. If at most one element is missing, on the other hand,* M *is consistent with* $\mathcal{C}_S$ *and, thus, it is conserved.*

Therefore, for this reaction system, the conserved sets are

$$\mathrm{cons}(\mathcal{A}) = \{\varnothing, S\} \cup \{S \setminus \{x_i\} \mid x_i \in S\},$$

for a total of $n+2$ *sets. However, the number of candidates that we need to examine is* 2^n.

Note that in Example 4.2 and Example 4.3 we end up examining all possible states, but there is a fundamental difference between the two. While for the former we actually do need to examine all sets since all are conserved, for the latter the number of conserved sets is $n+2$ out of the 2^n candidates. On the other hand, remark that the reaction system from Example 4.3 has a number of reactions quadratic in the number of species.

In what follows we aim to characterize the improvement provided by our algorithm over the naive approach. Let us first note that the test for conservation goes through the full behavior graph. Since the number of states of the behavior graph is exponential with respect to the number of species in the reaction system, the running time of our algorithm is not polynomially bounded. Moreover, we have seen in Example 4.2 that it is possible to have an exponential number of conserved sets for a given reaction system. Thus, the number of candidates that are tested for conservation is not polynomially bounded either. Therefore, we are going to assess the "quality" of our algorithm with respect to the number of non-conserved source sets, i.e. the candidate sets which are tested in addition to the conserved sets.

In the general case, one reaction (R_a, I_a, P_a) is enabled for all states W such that $R_a \subseteq W \subseteq S \setminus I_a$. Thus, a reaction that involves only a few species as reactants or inhibitors will have an impact on a significant subset of the edges

of the behavior graph. Put differently, the edges of G_b are strongly interrelated and breaking this interdependence, to decouple for example the singleton states from the rest of the graph, requires an increased number of reactions. Thus, we expect that reaction systems for which the number of reactions is linear in the number of species will reveal enough structural information in the conservation dependency graph so that the number of source sets is very close to the number of actual conserved sets. We formulate this idea as a conjecture below.

Conjecture. *Increasing the number of reactions linearly with respect to the number of species, the number of non-conserved source sets increases at most polynomially with respect to the number of species.*

4.3. Flexibility of the algorithm

Testing whether a set is conserved or not is a coNP-complete problem (see, e.g., [8]). Thus, even with our conjectured expectation that for a class of reaction systems we only need to examine a polynomial number of extra candidates, it is still likely that larger reaction systems will render the application of our algorithm infeasible. However, we show in this subsection that Algorithm 4.1 can be customized to take advantage of available resources, as well as additional knowledge about the reaction system or, more generally, about the model that the RS encodes.

First, let us notice that the computations which identify the actual conserved sets happen at the very end of the algorithm, essentially in step 8, relying on the already computed set of candidates and on Proposition 3.1 for the conservation test. For running the test, in addition to the candidates, we also need the behavior graph (more precisely, its connected components).

The computation of the candidates covers steps 4 to 7. This part of the algorithm uses the sets P_{out} and P_{in}, and the conservation dependency graph G_{cd} as inputs. The output consists of the source sets of G_{cd} which include P_{in} and do not intersect P_{out}. Another way to interpret this output is to consider that all three inputs are in fact sets of constraints and the obtained candidates are simply the sets of species that satisfy all the input constraints:

- P_{out} contains species that should not be part of any candidate set;
- P_{in} contains species that should be present in every candidate set;
- G_{cd} encodes constraints of the form "if x is in, then so is y" or, alternatively, "if y is not in, then neither is x".

For example, it only matters that we add to P_{in} species that are included in all conserved sets (or, more precisely, in all conserved sets that we are interested to compute).

Steps 2 and 3 of the algorithm compute the input constraints based on the behavior graph and rely on Proposition 3.2. The flow of the algorithm is schematically summarized in Figure 7, where each block (also referred to as *module*) is identified by its output and incoming edges stand for inputs.

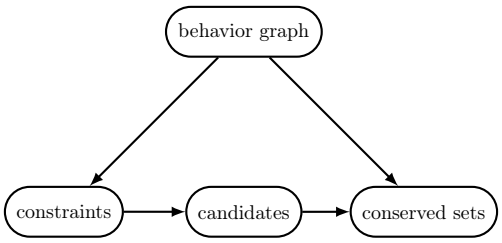


Figure 7: Schematic flowchart of Algorithm 4.1.

Note from the diagram that the behavior graph is required not only for computing the constraints to be satisfied by the candidate sets, but also for the conservation test. Thus, if the size of the reaction system is such that storing the behavior graph is impractical, we need to provide alternative implementations for the two modules. The test for conservation can rely on running the reaction system for each state, instead of the connected components of the behavior graph. Furthermore, we can also make use of the polynomial criterion from the next section in order to obtain a negative answer faster for the candidates. Moreover, note that each candidate set is tested separately, so this module is highly parallelizable.

For the constraints part, we can also rely on running the reaction system. Since we are under the assumption that the behavior graph is too large to fit in working memory, it may also be the case that it is not feasible to run the RS from an initial state up to identifying the first repeating state. Nevertheless, even by running the system a limited number of steps from particular initial states, we can partially recover the constraints that were obtained in Proposition 3.2 from the connected components of the behavior graph:

- any element of states that are reachable from $\varnothing$ will be in P_{out};
- any singleton state reachable from S will be included in P_{in};
- elements y of states reachable from singletons $\{x\}$ will contribute edges (x, y) for G_{cd}.

Note that, in addition to the constraints derived from the behavior graph (or by running the reaction system as suggested above), we can include other constraints as well, either by exploiting the particular structure of the reaction system under consideration, or by relying on available knowledge about the model that is described by the RS. For example, if we know a priori that a particular element x should be in all conserved sets (or, alternatively, if we are only interested to find those conserved sets that contain x) we can add this constraint by including x in P_{in}.

As long as all additional constraints that we put in are still provably satisfied by any conserved set of the input reaction system, we are going to obtain the full list of conserved sets as the output of the algorithm. If extra constraints

are added which come from the biological model or simply from the need to speed up the algorithm, then we will only obtain a subset of the conserved sets, namely those which satisfy all the constraints.

5. Negative Polynomial Heuristics for Formula Correspondence

In this section we give a simple polynomial (in size of the formulae and number of reactions) heuristics which can help decide whether a given set M is not conserved. The provided heuristics will be sufficient, but *not* necessary. We will provide two negative criteria for a more general problem first, and then show how they can be applied to mass conservation.

Note that, since deciding whether a given set of species M is conserved in a reaction system is coNP-complete [8], we could not expect to give such a polynomial criterion which would be both sufficient *and* necessary. This section shows, nevertheless, that analyzing some static properties of the reaction system may help conclude that M is not conserved in polynomial time, without enumerating all subsets of species.

We recall first that a *Boolean formula* φ is said to be over an alphabet S if all its variable names are from S. In the following we assume all Boolean formulae to be given in a disjunctive normal form. A subset $W \subseteq S$ is said to *satisfy* the Boolean formula φ over S if the expression for φ contains a conjunction $x_1 \wedge \ldots \wedge x_n \wedge \bar{y}_1 \wedge \ldots \wedge \bar{y}_m$ such that

(1) $\{x_i \mid 1 \leq i \leq n\} \subseteq W$, and

(2) $\{y_j \mid 1 \leq i \leq m\} \cap W = \varnothing$.

By convention, we write $\varphi(W) = 1$, or simply $\varphi(W)$, if the subset W satisfies φ, and $\varphi(W) = 0$ otherwise. For more details about the relationship between reaction systems and Boolean functions we refer to [2] and [1].

The paper [8] generalizes mass conservation in the form of two *formula correspondence problems*. Given a reaction system $\mathcal{A} = (S, A)$ and two Boolean formulae ϕ and ψ over S, the formula correspondence problems consist in deciding whether the following relations hold for every set $W \subseteq \mathrm{supp}(\mathcal{A})$:

$$\begin{array}{rcl} \phi(W) & \Rightarrow & \psi(\mathrm{res}_{\mathcal{A}}(W)), \\ \phi(W) & \Leftrightarrow & \psi(\mathrm{res}_{\mathcal{A}}(W)). \end{array}$$

It is shown in [8] that deciding either of these questions is coNP-complete.

We can parameterize the formula correspondence problems for a subset T of the background set in the same way as we parameterized mass conservation in Section 2. In such a case, we would define the formulae ϕ and ψ over T, and would require $\phi(W) \Rightarrow \psi(\mathrm{res}_A(W))$ (respectively, $\phi(W) \Leftrightarrow \psi(\mathrm{res}_A(W))$) for all subsets W of T, instead of the support of $\mathcal{A}$. It turns out that, just as with parameterized mass conservation, checking formula correspondence against a subset $T \subseteq S$ can be reduced to testing the same formulae against the projection $\mathcal{A}' = \mathrm{proj}_T(\mathcal{A})$. Indeed, recall that, for any set $W \subseteq T$, we have $\mathrm{res}_{\mathcal{A}'}(W) =$

$\mathrm{res}_{\mathcal{A}}(W) \cap T$. Since ψ is over T as well, the elements from the potentially nonempty $\mathrm{res}_{\mathcal{A}}(W) \setminus T$ will have no influence upon the satisfiability of ψ, i.e. $\psi(\mathrm{res}_{\mathcal{A}}(W)) \Leftrightarrow \psi(\mathrm{res}_{\mathcal{A}}(W) \cap T)$. This implies that formula correspondence in $\mathcal{A}$ with respect to T holds if and only if it holds in $\mathcal{A}'$ with respect to its full background set.

Seeing that conventional formula correspondence can be expressed as parameterized correspondence over $T = \mathrm{supp}(\mathcal{A})$ is a matter of remarking that ϕ and ψ in the conventional formulation can be restricted to the alphabet $\mathrm{supp}(\mathcal{A})$, without losing generality. Indeed, having ϕ include $\bar{x}$, with $x \in S \setminus \mathrm{supp}(\mathcal{A})$, for example, is redundant since we are only checking the correspondence against the subsets of $\mathrm{supp}(\mathcal{A})$ anyway. If, on the other hand, ϕ employs x in its non-negated form, then no subset of $\mathrm{supp}(\mathcal{A})$ will satisfy ϕ. Similar arguments can be given for ψ and the result set $\mathrm{res}_{\mathcal{A}}(W)$.

In view of the fact that any case of parameterized formula correspondence, and, in particular, the conventional correspondence, is reducible to formula correspondence over the full background set, we only focus on the latter problem.

For a given conjunction $\phi_1 = x_1 \wedge \ldots \wedge x_n \wedge \bar{y}_1 \wedge \ldots \wedge \bar{y}_m$, we will use the following shortcut notations:

$$\begin{array}{rcl} \mathrm{pos}(\phi_1) & = & \{x_1, \ldots, x_n\}, \\ \mathrm{neg}(\phi_1) & = & \{y_1, \ldots, y_n\}. \end{array}$$

Suppose now that the first formula ϕ over S is given in a disjunctive normal form, $\phi = \bigvee_{i=1}^{n} \phi_i$, and consider a reaction $a = (R_a, I_a, P_a)$ over the same alphabet S. We would like to know the conditions for a to be enabled on at least one subset satisfying ϕ.

Lemma 5.1. *For a reaction system $\mathcal{A} = (A, S)$, a reaction $a = (R_a, I_a, P_a) \in A$, and a Boolean formula $\phi = \bigvee_{i=1}^{n} \phi_i$, both over the same alphabet S, the following conditions are equivalent:*

(1) $\exists W \subseteq S, \phi(W) \wedge \mathrm{en}_a(W)$, *and*

(2) $\exists i \in \{1, \ldots, n\}, R_a \cap \mathrm{neg}(\phi_i) = I_a \cap \mathrm{pos}(\phi_i) = \varnothing$.

Proof. (1)⇒(2): Suppose there exists a subset W which both satisfies ϕ and enables a. This means that $R_a \subseteq W$ and $I_a \cap W = \varnothing$, but also that ϕ contains a conjunction ϕ_i such that $\mathrm{pos}(\phi_i) \subseteq W$ and $\mathrm{neg}(\phi_i) \cap W = \varnothing$. Therefore, $I_a \cap \mathrm{pos}(\phi_i) \subseteq I_a \cap W = \varnothing$ and $R_a \cap \mathrm{neg}(\phi_i) \subseteq W \cap \mathrm{neg}(\phi_i) = \varnothing$, which lead to $\Rightarrow I_a \cap \mathrm{pos}(\phi_i) = \varnothing$ and $R_a \cap \mathrm{neg}(\phi_i) = \varnothing$, respectively.

(2)⇒(1): Suppose that ϕ contains a conjunction ϕ_i such that $R_a \cap \mathrm{neg}(\phi_i) = I_a \cap \mathrm{pos}(\phi_i) = \varnothing$ and consider the set $W = R_a \cup \mathrm{pos}(\phi_i)$. Clearly, $\phi_i(W)$ holds, because $\mathrm{pos}(\phi_i) \subseteq W$, and because $R_a \cap \mathrm{neg}(\phi_i) = \mathrm{pos}(\phi_i) \cap \mathrm{neg}(\phi_i) = \varnothing$. On the other hand, we also know that $R_a \subseteq W$ and $I_a \cap R_a = I_a \cap \mathrm{pos}(\phi_i) = \varnothing$, so the reaction a is enabled on $R_a \cup \mathrm{pos}(\phi_i)$. We have therefore successfully constructed a set satisfying the statement (1). □

We will write $\mathrm{en}_a(\phi) = 1$, or just $\mathrm{en}_a(\phi)$, to refer to the fact that a is enabled on a set satisfying ϕ.

The following two observations give negative heuristic criteria for formula correspondence. Both cases are formulated in the setting of a reaction system $\mathcal{A} = (S, A)$ and two Boolean formulae $\varphi = \bigvee_{i=1}^{n} \phi_i$ and $\psi = \bigvee_{j=1}^{m} \psi_j$ over S.

Lemma 5.2. *If $\mathcal{A}$ contains a reaction a such that $\mathrm{en}_a(\phi)$, but $P_a \cap \mathrm{neg}(\psi_j) \neq \varnothing$, for all $1 \leq j \leq m$, then there exists a subset $W \subseteq S$ for which $\phi(W) \not\Rightarrow \psi(\mathrm{res}_{\mathcal{A}}(W))$.*

Proof. Since we know that $\mathrm{en}_a(\phi)$, there exists such a subset $W \subseteq S$ that $\phi(W)$ and $\mathrm{en}_a(W)$. The hypothesis that the product set of a intersects all $\mathrm{neg}(\psi_j)$ means that $\mathrm{res}_{\mathcal{A}}(W)$ intersects all $\mathrm{neg}(\psi_j)$ as well, and therefore $\psi(\mathrm{res}_{\mathcal{A}}(W))$ does not hold. □

Verifying the condition of the previous lemma requires going through both ϕ and ψ for every reaction of $\mathcal{A}$. The time complexity of such a procedure is in $O\big(|\phi_1| \cdot |\psi| \cdot (N_R + N_I + N_P)\big)$, where $|\phi|$ is the number of atomic terms in the disjunctive normal form of ϕ, while N_R, N_I, and N_P are the total sizes of the reactant, inhibitor, and product sets of the reactions in $\mathcal{A}$:

$$N_R = \sum_{a \in A} |R_a|, \quad N_I = \sum_{a \in A} |I_a|, \quad N_P = \sum_{a \in A} |P_a|.$$

Lemma 5.3. *Consider the set $B = \{b \mid b \in A, \mathrm{en}_b(\phi)\}$ and take the union of the products of the reactions in this set: $\bar{P} = \bigcup_{b \in B} P_b$. If, for any conjunction ψ_j of Ψ, it is true that $\mathrm{pos}(\psi_j) \not\subseteq P$, then there exists a subset $W \subseteq S$ for which $\phi(W) \not\Rightarrow \psi(\mathrm{res}_{\mathcal{A}}(W))$.*

Proof. Consider a set W such that $\phi(W)$. Then, by definition of the set B, $\mathrm{res}_{\mathcal{A}}(W) \subseteq \bar{P}$. But, since formula ψ contains *no* conjunction ψ_j such that $\mathrm{pos}(\psi_j) \subseteq \bar{P}$, this means that $\mathrm{res}_{\mathcal{A}}(W)$ satisfies no conjunction of ψ and therefore does not satisfy ψ. □

Verifying the condition of this lemma requires going through ϕ for every reaction in the system ($O\big(|\phi| \cdot (N_R + N_I)\big)$ steps), putting together the product sets of certain reactions ($O(N_P)$ steps), and then checking if the non-negated variables of a conjunction of ψ form a subset of this union ($O(|\psi| \cdot N_P)$ steps). The time complexity of such a procedure can therefore be estimated to belong to $O\big(|\phi| \cdot (N_R + N_I) + |\psi| \cdot N_P\big)$.

To formulate a heuristic criterion for mass conservation, we will rewrite this problem in Boolean formulae. The arguments in Section 2 allow us to consider mass conservation over the full background set. For a set $M \subseteq S$, the sets W satisfying the condition $M \cap W \neq \varnothing$ are exactly the sets satisfying the following Boolean formula:

$$\phi = \bigvee_{x \in M} x.$$

The property of M being conserved can then be written as follows (cf. [8]):

$$\forall W \subseteq S, \phi(W) \Leftrightarrow \phi(\mathrm{res}_{\mathcal{A}}(W)).$$

Applying the statement of Lemma 5.2 to this particular instance of the formula correspondence problem is ineffective, because no conjunction in ψ contains negated variables. However, instantiating the statement of Lemma 5.3 (and that of Lemma 5.1) yields the following negative heuristics for mass conservation.

Corollary 5.4. *Consider a reaction system $\mathcal{A} = (S, A)$, a subset of species $M \subseteq S$, and a subset of reactions $B = \{b \mid b = (R_b, I_b, P_b) \in A, M \setminus I_b \neq \varnothing\}$. If it is true that $M \cap \bigcup_{b \in B} P_b = \varnothing$, then M is* not *conserved in $\mathcal{A}$.*

Example 5.1. *Consider the reaction system $\mathcal{A} = (S, A)$, with $S = \{x, y\}$ and*

$$A = \{(\{x\}, \{y\}, \{y\}), (\{y\}, \{x\}, \{x\})\}.$$

$\mathcal{A}$ conserves no nonempty sets, because $\mathrm{res}_{\mathcal{A}}(S) = \varnothing$. Take $M_1 = \{x\}$, then $B_1 = \{(\{x\}, \{y\}, \{y\})\}$. Since $M_1 \cap \bigcup_{b \in B_1} P_b = \{x\} \cap \{y\} = \varnothing$, the previous corollary allows us to correctly conclude that M_1 is not conserved. Consider, on the other hand, the set $M_2 = \{x, y\}$. Then $B_2 = A$, $M_2 \cap \bigcup_{b \in B_2} P_b = \{x, y\} \cap \{x, y\} = \{x, y\} \neq \varnothing$, and the criterion does not allow us to conclude whether M_2 is conserved or not.

6. Reaction System Simulator

Even though it is relatively easy to write out an interactive process of a reaction system given a context sequence, doing this by hand quickly becomes tedious and error-prone. To automate the task, we developed a reaction system simulator, `brsim`. This is a stand-alone software tool which reads the description of a reaction system and a sequence of contexts from a file, runs the system with the supplied contexts, and then outputs the sequence of results. The simulator includes the option of annotating the evolution, in which case, for each evolution step, it will show the previous result, the context added at the current step, as well as the reactions enabled in the current state. Interactively running the reaction system is also supported, in which case the simulator will ask for the new context at each step.

Besides being able to run a reaction system for a given context sequence, the simulator can also show its conservation dependency graph as well as compute and list the conserved sets using an implementation of the algorithm shown in Section 4.

The source code of the simulator is licensed under GPLv3 and is available at [14]. We also provide a web interface to `brsim` at [15].

The input format of the simulator is similar to the notations conventionally used to write reactions. For example, a reaction system containing the reactions $(\{a\}, \{b, x\}, \{a\})$ and $(\{b\}, \{a, x\}, \{b\})$ would be described as follows:

```
a, b x, a
b, a x, b
```

The context sequence $C_0 = \{a, b\}$, $C_1 = \varnothing$, $C_2 = \{a, x\}$ would be represented in the following way:

```
a b
.
a x
```

For further details about using the simulator as a stand-alone application or via its web interface we refer the reader to [14, 15].

7. Conclusion

In this paper we focused on the biologically inspired property of mass conservation in reaction systems and unveiled the conservation dependency relation it induces between the species. It turned out that relying on the conservation dependency graph makes it possible to design an algorithm for listing the conserved sets which, in certain cases, performs better than the naive approach. Because conserved sets can well be exponential in number (cf. [8]), we cannot expect to build an algorithm which would always work in subexponential time. Yet, the fact that using the conservation dependency graph allows reducing the number of computational steps in some cases serves as an example of how observing certain *structural* properties of a reaction system can help to answer difficult questions more efficiently.

Several bibliographical references indicate that, in various biochemical networks, the number of reactions is linear in the number of involved species. For example, the metabolic networks of the bacteria *E. coli*, *H. influenzae*, *H. pylori*, and *G. sulfurreducens* employ 660, 296, 291, and 588 genes, and 720, 488, 288, and 523 reactions respectively [16]. It is equally noteworthy that the stoichiometric matrices of biochemical networks are generally sparse, since only a few species usually participate in a reaction [17]. In view of these observations, the conjecture proposed in Section 4 would imply that, for reaction system models of real-life biochemical networks, Algorithm 4.1 will only produce a polynomial number of *non-conserved* candidate sets (in terms of the number of species).

In Section 5 we also provided a sufficient polynomial criterion which can be used to prove that a given set of species is not conserved. The criterion is built around a different series of observations revealing yet other connections between the inner structure of the reaction system and the sets it conserves. Because deciding the conservation of a set is coNP-complete, we could not hope to have a sufficient *and* necessary criterion which would also be polynomial.

While we do show an important application of the conservation dependency graph to listing the conserved sets of a reaction system, we expect that a number of other properties of this graph remain to be further explored. A promising research direction would be that of establishing in which way the conservation

dependency graph is related to other conservation properties, like invariant sets, or the formula correspondence problems (see [8] for the definitions).

Lastly, in Section 6, we presented the simulator `brsim` which automates the process of running a reaction system with a given sequence of contexts, but also supports listing the conserved sets using an implementation of Algorithm 4.1. Since it is possible to both run the simulator as a stand-alone application and work with it via a web interface, we hope that it will be useful to the actively growing community of researchers working in the domain of reaction systems.

Acknowledgments

Sepinoud Azimi, Cristian Gratie and Ion Petre gratefully acknowledge support from Academy of Finland through projects 267915 and 272559. This work was partially done during Sergiu Ivanov's visit at the Computational Biomodeling Laboratory, Turku, Finland in Spring 2014.

References

[1] A. Ehrenfeucht, G. Rozenberg, Reaction systems, Fundamenta Informaticae 75 (1) (2007) 263–280.

[2] R. Brijder, A. Ehrenfeucht, M. G. Main, G. Rozenberg, A tour of reaction systems, International Journal of Foundations of Computer Science 22 (7) (2011) 1499–1517.

[3] A. Ehrenfeucht, M. Main, G. Rozenberg, Functions defined by reaction systems, International Journal of Foundations of Computer Science 22 (01) (2011) 167–178.

[4] E. Formenti, L. Manzoni, A. E. Porreca, Cycles and global attractors of reaction systems, in: H. Jürgensen, J. Karhumäki, A. Okhotin (Eds.), Descriptional Complexity of Formal Systems, Vol. 8614 of Lecture Notes in Computer Science, Springer, 2014, pp. 114–125.

[5] E. Formenti, L. Manzoni, A. E. Porreca, Fixed points and attractors of reaction systems, in: A. Beckmann, E. Csuhaj-Varjú, K. Meer (Eds.), Language, Life, Limits, 10th Conference on Computability in Europe, CiE 2014, Vol. 8493 of Lecture Notes in Computer Science, Springer, 2014, pp. 194–203.

[6] A. Salomaa, Functions and sequences generated by reaction systems, Theoretical Computer Science 466 (2012) 87–96.

[7] A. Salomaa, Functional constructions between reaction systems and propositional logic, International Journal of Foundations of Computer Science 24 (1) (2013) 147–159.

[8] S. Azimi, C. Gratie, S. Ivanov, L. Manzoni, I. Petre, A. E. Porreca, Complexity of model checking for reaction systems, Tech. Rep. 1122, Turku Centre for Computer Science (2014).

[9] S. Azimi, B. Iancu, I. Petre, Reaction system models for the heat shock response, Fundamenta Informaticae 131 (3) (2014) 299–312.

[10] L. Corolli, C. Maj, F. Marini, D. Besozzi, G. Mauri, An excursion in reaction systems: From computer science to biology, Theoretical Computer Science 454 (2012) 95–108.

[11] A. Meski, W. Penczek, G. Rozenberg, Model checking temporal properties of reaction systems, Tech. Rep. 1028, ICS PAS, Warsaw (2014).

[12] I. Petre, A. Mizera, C. L. Hyder, A. Meinander, A. Mikhailov, R. I. Morimoto, L. Sistonen, J. E. Eriksson, R.-J. Back, A simple mass-action model for the eukaryotic heat shock response and its mathematical validation, Natural Computing 10 (1) (2011) 595–612.

[13] A. Gibbons, Algorithmic Graph Theory, Cambridge University Press, 1985.

[14] The web interface of the reaction system simulator, `http://combio.abo.fi/research/reaction-systems/reaction-system-simulator/`.

[15] The source code of the reaction system simulator, `https://github.com/scolobb/brsim`.

[16] B. O. Palsson, Systems biology, Cambridge university press, 2006.

[17] L. Edsberg, Introduction to computation and modeling for differential equations, John Wiley & Sons, 2013.