Sepinoud Azimi | Cristian Gratie | Sergiu Ivanov | Ion Petre

Dependency Graphs and Mass Conservation in Reactions Systems

Turku Centre for Computer Science

TUCS Technical Report
No 1123, October 2014

Dependency Graphs and Mass Conservation in Reactions Systems

Sepinoud Azimi
Åbo Akademi University, Department of Computer Science
Joukahaisenkatu 3-5A, 20520 Turku, Finland
sazimi@abo.fi

Cristian Gratie
Åbo Akademi University, Department of Computer Science
Joukahaisenkatu 3-5 A, 20520 Turku, Finland
cgratie@abo.fi

Sergiu Ivanov
Laboratoire d'Algorithmique, Complexité et Logique,
Université Paris Est – Créteil Val de Marne, France
sergiu.ivanov@u-pec.fr

Ion Petre
Åbo Akademi University, Department of Computer Science
Joukahaisenkatu 3-5 A, 20520 Turku, Finland
ipetre@abo.fi

TUCS Technical Report
No 1123, October 2014

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 in 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.

TUCS Laboratory
Computational Biomodeling Laboratory (Combio Lab)

1 Introduction

Reaction systems is a framework inspired by the functioning of the living cells which was originally introduced in [6]. 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, whenever a resource is available, it is available in unlimited amount. This implies in particular that no competition for resources happens. The second 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 were conceived from the very beginning 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 [3], but they can generally be classified along two axes. The first direction 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., [5, 7, 8, 12, 13]). The other main line of research regards reaction systems as an instrument for biological modeling (e.g., [1, 2, 4]). Quite naturally, investigations along this line led to the study of model checking for reaction systems. For example, in [10], 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, [1] 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 [1]. We get a new insight into the connection between the internal structure of the reaction system and mass conservation by revealing a relation 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 show 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 system 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 [6] and [5].

Definition 2.1 ([6])**.** *Let S be a finite set. A* reaction a *in S is a triplet of finite nonempty 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 symbols (alternatively, elements or species) and $A \subseteq \mathrm{rac}(S)$. The set S is called the* background *(set) of $\mathcal{A}$.*

We use the following notations [1]:

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

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 ([6])**.** *Let $\mathcal{A} = (S, A)$ be a reaction system, $W \subseteq S$, and $a \in A$. 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:*

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

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

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

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

Definition 2.3 ([1]). *Let $\mathcal{A} = (S, A)$ be a reaction system. We say that a set $M \subseteq \operatorname{supp}(\mathcal{A})$ is* conserved *if for any $W \subseteq \operatorname{supp}(\mathcal{A})$, $M \cap W \neq \varnothing$ if and only if $M \cap \operatorname{res}_{\mathcal{A}}(W) \neq \varnothing$.*

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 elements 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 states tested for conservation.

Definition 2.4 (parameterized conservation). *Let $\mathcal{A} = (S, A)$ be a reaction system and $T \subseteq S$. 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 $M \cap \operatorname{res}_{\mathcal{A}}(W) \neq \varnothing$. We use $cons(\mathcal{A}, T)$ 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 $T = \operatorname{supp}(\mathcal{A})$.

Furthermore, it can be shown that in order to find 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. Indeed, consider the reaction system $\mathcal{A}' = (T, A')$ where:

$$A' = \{a' = (R_a, I_a \cap T, P_a \cap T) \mid a \in A \wedge R_a \subseteq T\}\,.$$

It is straightforward to see that for any $a \in A$ and $a' = (R_a, I_a \cap T, P_a \cap T)$ we have $\operatorname{en}_{\mathcal{A}}(W) = \operatorname{en}_{\mathcal{A}'}(W)$, given that $W \subseteq T$. Moreover, reactions $a \in A$ that do not have a corresponding reaction in A' are not enabled for $W \subseteq T$ because $R_{a'} \not\subseteq T$. Thus, we have $\operatorname{res}_{\mathcal{A}'}(W) = \operatorname{res}_{\mathcal{A}}(W) \cap T$, which leads to that $M \cap \operatorname{res}_{\mathcal{A}'}(W) = M \cap \operatorname{res}_{\mathcal{A}}(W)$, for any $M \subseteq T$, so the conserved sets with respect to T are the same in the two reaction systems, i.e. $cons(\mathcal{A}, T) = cons(\mathcal{A}', T)$. Since for $\mathcal{A}'$ we have that T is actually the background set, 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 $cons(\mathcal{A})$ instead of $cons(\mathcal{A}, S)$), but having all results implicitly applicable both for the parameterized conservation with respect to arbitrary sets T (by using the translation presented above) as well as for the original definition of mass conservation (by taking $T = \operatorname{supp}(\mathcal{A})$).

It should be noted, though, that in this case the reaction system $\mathcal{A}'$ obtained as above may include reactions with empty inhibitor or product sets, even if $\mathcal{A}$ does not have them. Therefore, in this paper we relax the usual requirement that all the three sets defining reactions need to be non-empty [2, 3]. This is in line with the observation that reactions that can not be inhibited by the states taken into consideration (e.g. subsets of the support set) are crucial for mass conservation [1].

Finally, we recall one of the basic notations used in asymptotically estimating the complexity of an algorithm.

Definition 2.5. *Given two real-valued functions f and g defined on real numbers, we write $f(x) = O(g(x))$ if there exists a positive constant M and a real number x_0 such that $|f(x)| \leq M|g(x)|$, for all $x > x_0$.*

For further details on computational complexity we refer to the monograph [11].

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 a different representation of 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 [6]).

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.

Definition 3.2. *Let $G = (V, E)$ be a directed graph. We say that a node v is connected with 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 further introduction to graph theory, we refer the reader to [9].

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 $\mathcal{C}$.

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 it intersects every element of $\mathcal{C}$, i.e. $M \sqcap \mathcal{C} = \mathcal{C}$. We say that M is* disjoint *from $\mathcal{C}$ if it 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 it 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) ⇒ (1) follows from the fact that, for every set $W \subseteq S$, we have $next(W) \in CC_{G_b}(W)$.

We now prove that (1)⇒(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.

3.1 Conservation dependency graph

In what follows we aim to further investigate the properties of conserved sets in relation with 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 $\mathcal{C}$, $cover(\mathcal{C}) = \bigcup_{T \in \mathcal{C}} T$.*

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

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

(3) For every $y \in cover(\mathcal{C}_x)$ 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. $\{y\} \sqcap cons(\mathcal{A}) \subseteq \{x\} \sqcap cons(\mathcal{A})$.

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 at least one 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 the same as $x \in M$.

(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 essentially means that $M \cap cover(\mathcal{C}_x) = \varnothing$. 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 partitioin induced by the reactions on the state space. 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 of them. Note that there is also an interesting interplay between the two statements when there exists an x such that $cover(\mathcal{C}_x) = S$ and $\varnothing \in \mathcal{C}_x$. Indeed, the latter is equivalent to having $\mathcal{C}_x = \mathcal{C}_\varnothing$, which means that $cover(\mathcal{C}_\varnothing) = S$, so no element of S can be part of a conserved set. On the other hand, the former property leads to x being part of all nonempty conserved sets, which is trivially true since the only conserved set in this case is the empty set.

The more dramatic 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 $cover(\mathcal{C}_\varnothing) = S$.

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 $y \in cover(\mathcal{C}_x)$, 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 $E_{cd} = \{(x, y) \mid x \in S \wedge y \in cover(\mathcal{C}_x)\}$.*

Intuitively, every conserved set should satisfy all constraints that are encoded by the conservation dependency graph. Alternatively, we can focus on the dependency graph alone and consider all the sets that are consistent with the aforementioned constraints.

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 the cut (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 is a source set of the conservation dependency graph.*

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.

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 $u \notin S \wedge (u, v) \in E \Rightarrow v \notin S$.

Proof. The negation of either statement directly violates the definition of source sets by providing an edge (u, v) that goes from $V \setminus S$ to S. □

The statement of the previous proposition can be immediately generalized by induction to ancestors and descendants of nodes (u is an ancestor of v or, equivalently, v is a descendant of u, if is there exists a directed path from u to v).

Corollary 3.1. *Let $G = (V, E)$ be a directed graph and let S be an arbitrary source set of G.*

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

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

We are going to relate source sets with 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 (SCC's) *and use $SCC_G(u)$ to refer to the SCC that contains u. Furthermore, we denote the set of all SCC's of G by $SCCS_G$, i.e. $SCCS_G = \{SCC_G(u) \mid u \in V\}$.*

It is not difficult to see by Corollary 3.1 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. Then either $C \subseteq S$ or $C \cap S = \varnothing$.*

Proof. If $C \cap S = \varnothing$, we have nothing to prove. So assume $C \cap S \neq \varnothing$ and choose $u \in C \cap S$. From Corollary 3.1 it follows that all ancestors of u must be in S as well. In particular, this implies that $C \subseteq S$. □

Corollary 3.2. *Any source set of a graph G is a union of 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 SCC's. In order to see exactly which of such unions are source sets, we will refer to the condensation of G, the graph obtained by replacing each SCC 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 SCC's of G, i.e. $\widetilde{V} = SCCS_G$, and whose edges are defined as follows: $\widetilde{E} = \{(C_1, C_2) \in \widetilde{V} \times \widetilde{V} \mid \exists u \in C_1 \,.\, \exists v \in C_2 \,.\, (u, v) \in E\}$, i.e. there is an edge (C_1, C_2) in $\widetilde{G}$ iff there is an edge in G from an element of C_1 to an element of C_2.*

Proposition 3.6. *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 $S = cover(\widetilde{S})$ and $\widetilde{S}$ is a source set of $\widetilde{G}$.*

Proof. We start with the forward implication. We know already from Corollary 3.2 that there exists $\widetilde{S} \subseteq \widetilde{V}$ such that $S = cover(\widetilde{S})$. 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 $S = cover(\widetilde{S})$. 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 equal or disjoint, 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 3.6 is that it suffices to have an algorithm computing source sets for 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 the symbol $desc_G(S)$ to refer to the set containing all the nodes from S and all their descendants. Similarly, we will use $anc_G(S)$ to refer to the set containing all nodes from S and all their ancestors in G.

Theorem 3.1. *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.*

(i) Any set T can be expanded to a source set S of G by first adding to S all ancestors of T, and then taking the union with a source set of the rest of the graph. Moreover, every source set that includes T can be computed in this way:

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

(ii) 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:

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

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

$$\sigma(G) = \sigma(G{\downarrow_{V \setminus desc_G(\{s\})}}) \cup \{S \cup anc_G(\{s\}) \mid S \in \sigma(G{\downarrow_{V \setminus anc_G(\{s\})}})\}.$$

Proof. (i) We have:

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

The first equivalence follows from the definition of conserved sets and Corollary 3.1. The second follows from the definition of $anc_G(T)$, as there can be no edges of G going into this set. The last equivalence relies again on the definition of source sets and also on $A \setminus B = (A \setminus X) \setminus (B \setminus X)$, which holds whenever $X \subseteq B \subseteq A$.

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

$$\begin{aligned}
&S \in \sigma(G) \wedge S \cap T = \varnothing \\
\Leftrightarrow &E \cap (V \setminus S) \times S = \varnothing \wedge S \cap desc_G(T) = \varnothing \\
\Leftrightarrow &E \cap ((V \setminus desc_G(T)) \setminus S) \times S = \varnothing \wedge S \cap desc_G(T) = \varnothing \\
\Leftrightarrow &S \in \sigma(G{\downarrow}_{V \setminus desc_G(T)}).
\end{aligned}$$

Just as before, the first equivalence follows directly from the definition of source sets and from Corollary 3.1. The second equivalence relies on the definition of $desc_G(T)$, as there can be no edges of G going out of this set. Finally, we use the definition again to get the desired result.

(iii) The result follows from (i) and (ii) 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:

$$\begin{aligned}
&S \in \sigma(G) \wedge s \in S \Leftrightarrow S \setminus anc_G(\{s\}) \in \sigma(G{\downarrow}_{V \setminus anc_G(\{s\})}) \wedge anc_G(\{s\}) \subseteq S, \\
&S \in \sigma(G) \wedge s \notin S \Leftrightarrow S \in \sigma(G{\downarrow}_{V \setminus desc_G(\{s\})}).
\end{aligned}$$

These statements lead to the desired result. □

We can immediately apply the third claim of this lemma 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 3.3. *Let $G = (V, E)$ be a directed graph and let $s \in V$ be a source node. Then we have:*

$$\sigma(G) = \sigma(G{\downarrow}_{V \setminus desc_G(\{s\})}) \cup \{S \cup \{s\} \mid S \in \sigma(G{\downarrow}_{V \setminus \{s\}})\}.$$

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 $G{\downarrow}_{V \setminus desc_G(\{s\})}$, then aggregate them according to Corollary 3.3 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 is based on source sets and relies on propositions 3.1 and 3.3, but also on the heuristics 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.*
2. *Compute the connected components of G_b and analyze them.*
   - (a) *Compute $P = cover(\mathcal{C}_\varnothing)$.*
   - (b) *Compute $Q = \{x \in S \mid cover(C_x) = S\}$.*
3. *Compute the conservation dependency graph G_{cd} of $\mathcal{A}$.*
4. *Compute the strongly connected components of G_{cd} and the condensation graph $\widetilde{G}_{cd}$.*
5. *Adjust the condensation graph $\widetilde{G}_{cd}$ to account for P and Q.*
   - (a) *Remove SCC's that contain elements from P, together with their ancestors.*
   - (b) *Remove SCC's that contain elements from Q, together with their descendants.*
6. *Compute the source sets of the resulting graph using Algorithm 3.1.*
7. *For each source set T, test whether $T \cup Q$ is a conserved set by checking that it is consistent with all connected components of G_b.*

Theorem 4.1. *The algorithm computes all conserved sets correctly.*

Proof. We first show that steps $4 - 6$ and the input used for step 7 translate to computing exactly the source sets of G_{cd} which contain Q and are disjoint from P. In order to find all the source sets which are disjoint from P, we can rely on Proposition 3.6 to conclude that we also need to exclude the full SCC's of elements from P, then based on Theorem 3.1 (ii) we must also exclude descendant SCC's. This translates to step $5(a)$ in the algorithm. A similar justification holds for step $5(b)$.

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

Remark that the decision problem for conserved sets is coNP-complete [1]. As such, we know already that we cannot test for conservation in polynomial time unless $\mathsf{P} = \mathsf{NP}$. 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.

To understand the benefit of the heuristics employed in the algorithm and also the nature of the reactions systems for which it is effective, we consider some examples. We start with a simple example to illustrate how the algorithm actually works.

Example 4.1. *Consider the following simple reaction system:*

$$\begin{aligned}
&S = \{x, y, z\} \\
&A = \{(\{x\}, \{z\}, \{y, z\}), (\{y\}, \varnothing, \{x\}), (\{x, z\}, \varnothing, \{y\})\} \\
&\quad \cup \{(\{y\}, \{z\}, \{y\}), (\{y, z\}, \varnothing, \{x\})\} \; .
\end{aligned}$$

The corresponding behavior graph G_b can be computed by finding $\mathrm{res}_{\mathcal{A}}(W)$ for all $W \subseteq S$. We obtain the graph from Figure 1.

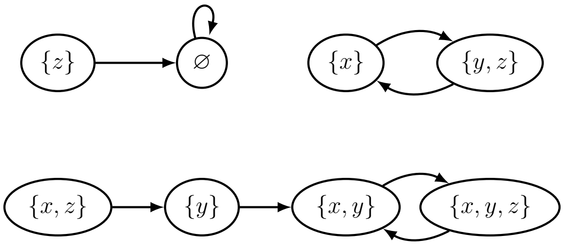

Figure 1: Behavior graph for the reaction system from Example 4.1.

The relevant connected components for our algorithm are:

$$\begin{aligned}
&\mathcal{C}_x = \{\{x\}, \{y, z\}\} \\
&\mathcal{C}_y = \{\{y\}, \{x, y\}, \{x, z\}, \{x, y, z\}\} \\
&\mathcal{C}_z = \{\{z\}, \varnothing\} = \mathcal{C}_\varnothing
\end{aligned}$$

The corresponding conservation dependency graph is presented in Figure 2.

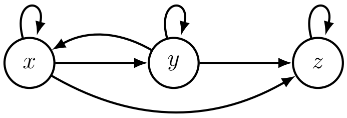

Figure 2: Conservation dependency graph for the reaction system from Example 4.1.

Note that this graph has only three source sets, namely $\varnothing$, $\{x, y\}$ and $S = \{x, y, z\}$. Thus, even if we are to consider the full conservation dependency graph, we would examine only 3 out of the total number of 8 candidates.

In fact, for this case we still need to consider the constraints coming from connected components of singleton sets and the empty set. This gives $P = \{z\}$ and $Q = \{y, z\}$. That is, all non-empty conserved sets must contain x and y and cannot contain z, which means that $M = \{x, y\}$ is the only non-empty conserved set for this reaction system. In particular, the algorithm will work on an empty graph at step 6.

While the previous example is rather simple, it still reveals the main improvements that come from the heuristics. On the other hand, consider also the case where the behavior and conservation dependency graphs do not provide any useful information, i.e. $P = \varnothing$, $Q = \varnothing$ 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 $P = Q = \varnothing$ 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 $\operatorname{res}_{\mathcal{A}}$ satisfies:

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

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 $M \cap \{x_i, x_j\} = \varnothing$, 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 also conserved.

Therefore, for this reaction system, the conserved sets are

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

for a total of $n+2$ sets. However, the number of candidates 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 in the former we actually do need to examine all sets since all are conserved, in 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 quadratic number of reactions.

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 conjecture that most 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 to make the computation of conserved sets effective.

4.2 Flexibility of the algorithm

In developing the algorithm presented in this section we relied as a first step on the computation of the behavior graph and we implicitly assumed that this was feasible. Note, however, that the number of possible states for a reaction system is exponential in its number of species and, as such, storing the behavior graph may be impractical. In this subsection we discuss that our algorithm is flexible in the way that the full behavior graph could be replaced in the input of the algorithm with a different, smaller graph; the consequence is of course in terms of higher running time.

First, remark that checking whether a given set is conserved does not need the behavior graph and can be performed instead by running the system on all possible states. Of course this trades time for space, as we lose the advantage of having cached the results in the behavior graph. On the other hand the task of checking the conservation condition is highly parallelizable, since individual states can be considered separately.

Second, note that the reduction in the number of candidates relies on three elements: the set P of species that can not be part of any conserved set, the set Q of species that are known to be part of all conserved sets, and the conservation dependency graph G_{cd}. However, the algorithm can work also with partial versions of these.

For example, whenever it is not practical to store the full behavior graph, we can instead compute partial versions of P, Q and G_{cd} by running the RS on the empty set, singleton sets and the background set. From these, we can extract the following:

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

Moreover, note that we can even augment the algorithm with constraints that do not come from the RS, but from the actual problem we are interested to solve. For example, if we are only looking for conserved sets that contain a particular element x, we can tune the algorithm by adding x to (either the full, or a partial version of) Q.

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 a coNP-complete [1], we could not expect to give such a polynomial criterion which would be both sufficient *and* necessary. This section shows however that analysing 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 variables 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 [3] and [6].

The paper [1] 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 [1] 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 $\phi(W) \Rightarrow \psi(res_A(W))$ (respectively, $\phi(W) \Leftrightarrow \psi(res_A(W))$) for all subsets of T, instead of the support of $\mathcal{A}$. It turns out that, just as with parameterized mass conservation, checking formula correspondence against $T \subseteq S$ can be reduced to testing the same formulae against the subsets of the background set of the reaction system $\mathcal{A}' = (T, A')$ where:

$$A' = \{(R_a, I_a \cap T, P_a \cap T) \mid a \in A \wedge R_a \subseteq T\}.$$

Indeed, remember 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 non-empty $\mathrm{res}_{\mathcal{A}}(W) \setminus T$ 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 however 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.

Consider a 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, $\phi = \bigvee\limits_{i=1}^{n} \phi_i$, 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 conjunction ϕ_i over S, the following conditions are equivalent:*

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

(2) $R_a \cap \mathrm{neg}(\phi_i) = I_a \cap \mathrm{pos}(\phi_i) = \varnothing$.

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

(2)⇒(1): Suppose that, for the reaction a, it is true 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 use the shorthand notation $en_a(\phi_i) = 1$, or simply $en_a(\phi_i)$, to refer to the fact that a can be enabled on a subset satisfying the conjunction ϕ_i.

It is now possible to formulate a similar statement for a general formula given in a disjunctive normal form.

Lemma 5.2. *For a reaction system $\mathcal{A} = (A, S)$, a reaction $a = (R_a, I_a, P_a) \in A$, and a Boolean formula $\phi = \bigvee\limits_{i=1}^{n} \phi_i$, 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) $\exists i \in \{1, \ldots, n\} \,.\, R_a \cap \mathrm{neg}(\phi_i) = I_a \cap \mathrm{pos}(\phi_i) = \varnothing$.

As before, 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 $\varphi = \bigvee\limits_{i=1}^{n} \phi_i$ and $\psi = \bigvee\limits_{j=1}^{m} \psi_j$ over S.

Proposition 5.1. *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 $en_a(\phi)$, 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 $\mathrm{neg}(\psi_j)$ too, and therefore $\psi(\mathrm{res}_{\mathcal{A}}(W))$ does not hold. □

Verifying the condition of the previous proposition 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 a disjunctive normal 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 $\mathcal{A}$:

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

Proposition 5.2. *Consider the set $B = \{b \mid b \in A, en_b(\phi)\}$ and take the union of the products of the reactions in this set: $\bar{P} = \bigcup\limits_{b \in B} P_b$. If, for any conjunction ψ_j, 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 reaction $b \in B$. Since $\mathrm{en}_b(\phi)$, there exists a set W such that $\phi(W)$ and also $\mathrm{en}_b(W)$, so $\mathrm{res}_{\mathcal{A}}(W) \neq \varnothing$. But $\mathrm{res}_{\mathcal{A}}(W) \subseteq \bar{P}$ and we know that formula ψ contains *no* conjunction ψ_j such that $\mathrm{pos}(\psi_j) \subseteq \bar{P}$. Therefore, for *no* ψ_j it is true that $\mathrm{pos}(\psi_j) \subseteq \mathrm{res}_{\mathcal{A}}(W)$, which means that $\mathrm{res}_{\mathcal{A}}(W)$ does not satisfy ψ. □

Verifying the condition of this proposition requires going through ϕ for every reaction in the system ($O\big(|\phi| \cdot (N_R + N_I)$ 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. [1]):

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

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

Corollary 5.1. *Consider a reaction system $\mathcal{A} = (S, A)$, a subset of species $M \subseteq S$, and a subset of reactions $B = \big\{b \mid b = (R_b, I_b, P_b) \in A, M \setminus I_b \neq \varnothing\big\}$. If it is true that $M \cap \bigcup_{b \in B} P_b = \varnothing$, then M is* not *conserved in $\mathcal{A}$.*

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. [1]), 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 helps to relatively quickly answer questions which would otherwise need an exponential amount of time.

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 [1] 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.

References

[1] Sepinoud Azimi, Cristian Gratie, Sergiu Ivanov, Luca Manzoni, Ion Petre, and Antonio E. Porreca. Complexity of model checking for reaction systems. Technical Report 1122, Turku Centre for Computer Science, 2014.

[2] Sepinoud Azimi, Bogdan Iancu, and Ion Petre. Reaction system models for the heat shock response. *Fundamenta Informaticae*, 131(3):299–312, 2014.

[3] Robert Brijder, Andrzej Ehrenfeucht, Michael G. Main, and Grzegorz Rozenberg. A tour of reaction systems. *International Journal of Foundations of Computer Science*, 22(7):1499–1517, 2011.

[4] Luca Corolli, Carlo Maj, Fabrizio Marini, Daniela Besozzi, and Giancarlo Mauri. An excursion in reaction systems: From computer science to biology. *Theoretical Computer Science*, 454:95–108, 2012.

[5] Andrzej Ehrenfeucht, Michael Main, and Grzegorz Rozenberg. Functions defined by reaction systems. *International Journal of Foundations of Computer Science*, 22(01):167–178, 2011.

[6] Andrzej Ehrenfeucht and Grzegorz Rozenberg. Reaction systems. *Fundamenta Informaticae*, 75(1):263–280, 2007.

[7] Enrico Formenti, Luca Manzoni, and Antonio E. Porreca. Cycles and global attractors of reaction systems. In Helmut Jürgensen, Juhani Karhumäki, and Alexander Okhotin, editors, *Descriptional Complexity of Formal Systems*, volume 8614 of *Lecture Notes in Computer Science*, pages 114–125. Springer, 2014.

[8] Enrico Formenti, Luca Manzoni, and Antonio E. Porreca. Fixed points and attractors of reaction systems. In Arnold Beckmann, Erzsébet Csuhaj-Varjú, and Klaus Meer, editors, *Language, Life, Limits, 10th Conference on Computability in Europe, CiE 2014*, volume 8493 of *Lecture Notes in Computer Science*, pages 194–203. Springer, 2014.

[9] Alan Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, 1985.

[10] Artur Meski, Wojciech Penczek, and Grzegorz Rozenberg. Model checking temporal properties of reaction systems. Technical Report 1028, University of Warsaw, 2014.

[11] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1993.

[12] Arto Salomaa. Functions and sequences generated by reaction systems. *Theoretical Computer Science*, 466:87–96, 2012.

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

[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`.

Turku
Centre *for*
Computer
Science

Joukahaisenkatu 3-5 A, 20520 TURKU, Finland | www.tucs.fi

University of Turku
Faculty of Mathematics and Natural Sciences
- Department of Information Technology
- Department of Mathematics

Turku School of Economics
- Institute of Information Systems Sciences

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

ISBN 978-952-12-3123-0
ISSN 1239-1891