Sergiu Ivanov | Vladimir Rogojin | Sepinoud Azimi | Ion Petre

WEBRSIM: A web-based reaction systems simulator

Turku Centre *for* Computer Science

TUCS Technical Report
No 1195, May 2018

WEBRSIM: A web-based reaction systems simulator

Sergiu Ivanov

IBISC, Université Évry, Université Paris-Saclay, France
sergiu.ivanov@univ-evry.fr

Vladimir Rogojin

Åbo Akademi University, Department of Information Technologies
vrogojin@abo.fi

Sepinoud Azimi

Åbo Akademi University, Department of Information Technologies
20126 Milano, Italy
sazimi@abo.fi

Ion Petre

Åbo Akademi University, Department of Information Technologies
ipetre@abo.fi

TUCS Technical Report

No 1195, May 2018

Abstract

We discuss in this paper WEBRSIM, the first web-based simulator for reaction systems. The simulator has an easy-to-use interface to input a reaction system and four functionalities: to compute the interactive process driven by a given context sequence, the behaviour graph of the reaction system, its conservation dependency graph, and all its conserved sets. WEBRSIM comes with a browser-based friendly interface and offers a fast software to support computational modeling with reaction systems.

Keywords: Reaction systems; web-based simulator; interactive processes; behaviour graph; conserved sets; conservation dependency graph; rewriting systems.

TUCS Laboratory
Computational Biomodeling Laboratory (Combio Lab)

1 Introduction

Reaction systems, first introduced in [13], is a qualitative framework inspired by two cellular regulation mechanisms, *facilitation* and *inhibition*, which control the interaction between biochemical reactions. Intuitively (and on a high level of abstraction), a biochemical reaction is enabled when all components needed to facilitate the reaction are present and all components that inhibit it are absent from the environment. Based on this intuition a reaction in the reaction systems formalism is defined as a triplet $a = (R_a, I_a, P_a)$ where R_a represents the set of reactants, I_a the set of inhibitors and P_a the set of products corresponding to reaction a. As a result of being triggered, the reaction transforms its set of reactants to the corresponding set of products.

The two main assumptions considered in the reaction systems framework are the following:

- *Threshold assumption*: an element is either abundantly present in the environment or it is absent from it. This implies that there is no counting in the reaction systems framework and as a result reaction systems are (at least in their basic version) a qualitative modelling framework.

- *No permanency assumption*: an element vanishes from the environment if no reaction is triggered to produce it back. This principle is supported by abstract biochemical considerations that maintaining an element in the environment is the result of an active (and energy-consuming) process. In the absence of such an explicit mechanism to maintain it, an element disappears from the environment.

Research done in the field of reaction systems has been very diverse, see for example [4, 6, 7, 8, 10, 12, 14, 20] for several recent contributions. The simple, yet expressive nature of this framework has attracted researchers from both theoretical and practical areas of science to focus on studying and analysing reaction systems. One of the main lines of such efforts focuses on modelling real world problems through reaction systems' framework. Such applications vary from biological modelling to number theory to quantum computing, see for example [4, 5, 9, 16, 20]. To study the properties of such models, especially the bio-inspired ones, a series of studies was initiated to formalise several properties of central interest in biomodeling and to study the computational complexity of deciding those properties, such as mass conservation, invariants, steady states, multistability, stationary processes, elementary fluxes, and periodicity: [1, 2, 3, 5].

The dynamics of a reaction system model can be observed through *interactive processes*. Intuitively, interactive processes describe the step-by-step evolution of a reaction system's model from one state to the next driven by external environment interventions. Many of the above mentioned properties of a reaction system's model can be captured through its interactive

processes. Reaction systems models corresponding to real world processes get quickly highly complex. That is why, even though it is straightforward to write an interactive process of a reaction system, doing this manually can very easily become tedious and highly error-prone. In this paper we introduce a reaction system simulator, `WEBRSIM` that automates the process of calculating states of a reaction system's model for a corresponding interactive process to overcome this issue. The simulator also provides the conserved sets of a given reaction system's model. The basic simulation engine behind `WEBRSIM`, `brsim`, was discussed in passing in [2].

Other software supporting modeling with reaction system exist. The one closest to `WEBRSIM` was introduced in [18] in the form of a CPU- and GPU-based simulator for reaction systems: `HERESY`. The basic approach to simulating reaction systems is similar in `HERESY` and `WEBRSIM`, with the added feature of `HERESY` to parallelise the computation on Graphics Processing Units (GPU). The comparison done in [18] between `HERESY` and `brsim` shows that the running time in `brsim` is only marginally slower than the highly parallelised GPU-version of `HERESY` and much faster than its CPU version. `WEBRSIM` is highly user friendly as it comes with a web-based version that may be used through a standard browser, whereas `HERESY` requires some knowledge of Python programming to be able to invoke its graphical user interface. As `WEBRSIM` comes with the open source code [22] (as does `HERESY`), it also provides the possibility of tweaking the code to the need of more expert users. Where both `HERESY` and `WEBRSIM` provide interactive processes simulation for reaction systems, `WEBRSIM` takes one step further and implements an algorithm to compute the behaviour graph, the conserved sets, as well as the dependency graph.

Another useful software for modeling with reaction systems is the one in [17], that offers the possibility to do model checking of temporal properties, a feature not offered by `WEBRSIM`.

2 Preliminaries

In this section, we recall the notion of a reaction system, as well as some related concepts. For more details we refer to [13] and [11].

Definition 2.1 ([13])**.** *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 (in short, RS) *is an ordered pair $\mathcal{A} = (S, A)$, where S is a finite set of* symbols *(also called sometimes elements, species, or entities) and $A \subseteq \mathrm{rac}(S)$. The set S is called the* background *(set) of A.*

We use the following notations:

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

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

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

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

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

$$\mathrm{res}_{\mathcal{A}}(T) = \textstyle\bigcup_{a \in A} P_a.$$

(3) An interactive process *in* $\mathcal{A}$ *is a pair* $\pi = (\gamma, \delta)$, *where* $\gamma = (C_0, C_1, ..., C_n)$ *and* $\delta = (D_1, D_2, ..., D_n)$, $n \geq 1$, *are sequences of subsets of* S *with* $D_1 = \mathrm{res}_{\mathcal{A}}(C_0)$ *and, for each* $1 < i \leq n$, $D_i = \mathrm{res}_{\mathcal{A}}(C_{i-1} \cup D_{i-1})$.

2.1 Running example

The heat shock response is one of the highly conserved cellular defence mechanisms among eukaryotes against environmental stressors such as high temperatures, toxins, bacterial infection, etc. In this paper we consider a simplified model of the heat shock response of [19]. This model's set of reactions is in Table 1 and the model was proposed in [3].

When a cell is exposed to stress, proteins misfold (reaction (7) in Table 1) into complexes that disables certain cell functions which can eventually lead to cell death. To reverse such effects, the number of a special family of molecular chaperones, called heat shock proteins (hsp's), increases. These molecular chaperones bind to misfolded proteins and facilitate their correct refolding (reactions (8),(9) in Table 1).

Table 1: The molecular model of the eukaryotic heat shock response proposed in [19].

No.	Reaction	No.	Reaction
(1)	$3\,\mathsf{hsf} \rightleftarrows \mathsf{hsf_3}$	(6)	$\mathsf{hsp} + \mathsf{hsf_3{:}\,hse} \to \mathsf{hsp{:}\,hsf} + 2\,\mathsf{hsf} + \mathsf{hse}$
(2)	$\mathsf{hsf_3} + \mathsf{hse} \rightleftarrows \mathsf{hsf_3{:}\,hse}$	(7)	$\mathsf{prot} \to \mathsf{mfp}$
(3)	$\mathsf{hsf_3{:}\,hse} \to \mathsf{hsf_3{:}\,hse} + \mathsf{hsp}$	(8)	$\mathsf{hsp} + \mathsf{mfp} \rightleftarrows \mathsf{hsp{:}\,mfp}$
(4)	$\mathsf{hsp} + \mathsf{hsf} \rightleftarrows \mathsf{hsp{:}\,hsf}$	(9)	$\mathsf{hsp{:}\,mfp} \to \mathsf{hsp} + \mathsf{prot}$
(5)	$\mathsf{hsp} + \mathsf{hsf_3} \to \mathsf{hsp{:}\,hsf} + 2\,\mathsf{hsf}$		

A family of proteins called heat shock transcription factors (hsf's) regulates the expression of hsp's. In a trimeric state (hsf_3) they bind to heat shock elements (hse's - the hsp-encoding gene promoter regions) and activate the transcription of hsp's (reactions (1)-(3) in Table 1). By binding to the hsf_3: hse's, hsf_3's, and hsf's and breaking down the complexes, the hsp's downregulate their expression which leads to stopping the expression activity (reactions (4)-(6)).

We use the reaction system model corresponding to the heat shock response model of Table 1 as the running example in this paper. This reaction system was first introduced in [3]. The reaction system model for the heat shock response is presented in Table 2. As discussed in [3], this RS-based model for the heat shock response is satisfactory in that its conserved sets (see below) correspond exactly to the mass conservation relations of the quantitative model of [19].

Table 2: Reaction system for heat shock response

No.	Reaction	No.	Reaction
1	$(\{\mathsf{hsf}\},\{\mathsf{hsp}\},\{\mathsf{hsf}_3\})$	11	$(\{\mathsf{hsp},\mathsf{hsf}_3\colon\mathsf{hse}\},\{\mathsf{mfp}\},\{\mathsf{hsp}\colon\mathsf{hsf},\mathsf{hse}\})$
2	$(\{\mathsf{hsp},\mathsf{hsf}\},\{\mathsf{mfp}\},\{\mathsf{hsp}\colon\mathsf{hsf}\})$	12	$(\{\mathsf{hsf}_3\colon\mathsf{hse},\mathsf{hsp},\mathsf{mfp}\},\{\mathsf{d}_\mathsf{I}\},\{\mathsf{hsf}_3\colon\mathsf{hse},\mathsf{hsp}\})$
3	$(\{\mathsf{hsf},\mathsf{hsp},\mathsf{mfp}\},\{\mathsf{d}_\mathsf{I}\},\{\mathsf{hsf}_3\})$	13	$(\{\mathsf{hse}\},\{\mathsf{hsf}_3\},\{\mathsf{hse}\})$
4	$(\{\mathsf{hsf}_3\},\{\mathsf{hse},\mathsf{hsp}\},\{\mathsf{hsf}\})$	14	$(\{\mathsf{hsp}\colon\mathsf{hsf},\mathsf{stress}\},\{\mathsf{nostress}\},\{\mathsf{hsp},\mathsf{hsf}\})$
5	$(\{\mathsf{hsp},\mathsf{hsf}_3\},\{\mathsf{mfp}\},\{\mathsf{hsp}\colon\mathsf{hsf},\mathsf{hsf}\})$	15	$(\{\mathsf{hsp}\colon\mathsf{hsf},\mathsf{nostress}\},\{\mathsf{stress}\},\{\mathsf{hsp}\colon\mathsf{hsf}\})$
6	$(\{\mathsf{hsf}_3,\mathsf{hsp},\mathsf{mfp}\},\{\mathsf{hse}\},\{\mathsf{hsf}\})$	16	$(\{\mathsf{prot},\mathsf{stress}\},\{\mathsf{nostress}\},\{\mathsf{prot},\mathsf{mfp}\})$
7	$(\{\mathsf{hsf}_3,\mathsf{hse}\},\{\mathsf{hsp}\},\{\mathsf{hsf}_3\colon\mathsf{hse}\})$	17	$(\{\mathsf{prot},\mathsf{nostress}\},\{\mathsf{stress}\},\{\mathsf{prot}\})$
8	$(\{\mathsf{hse},\mathsf{hsf}_3,\mathsf{hsp}\},\{\mathsf{mfp}\},\{\mathsf{hse}\})$	18	$(\{\mathsf{hsp},\mathsf{mfp}\},\{\mathsf{d}_\mathsf{I}\},\{\mathsf{hsp}\colon\mathsf{mfp}\})$
9	$(\{\mathsf{hsf}_3,\mathsf{hse},\mathsf{hsp},\mathsf{mfp}\},\{\mathsf{d}_\mathsf{I}\},\{\mathsf{hsf}_3\colon\mathsf{hse}\})$	19	$(\{\mathsf{mfp}\},\{\mathsf{hsp}\},\{\mathsf{mfp}\})$
10	$(\{\mathsf{hsf}_3\colon\mathsf{hse}\},\{\mathsf{hsp}\},\{\mathsf{hsf}_3\colon\mathsf{hse},\mathsf{hsp}\})$	20	$(\{\mathsf{hsp}\colon\mathsf{mfp}\},\{\mathsf{d}_\mathsf{I}\},\{\mathsf{hsp},\mathsf{prot}\})$

We now recall the definition of conserved sets for reaction systems models, as proposed in [1].

Definition 2.3 (Conserved sets, [1]). *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 $M \cap \mathrm{res}_\mathcal{A}(W) \neq \varnothing$.*

For our running example, a conserved set is $M = \{\mathsf{hse}, \mathsf{hsf}_3\colon\mathsf{hse}\}$. For an arbitrary set $W \subseteq \mathrm{supp}(\mathcal{A})$, let us suppose $\mathsf{hse} \in W \cap M$. We claim that at least one of the reactions (7), (8), (9), or (13) is enabled by W:

- if $\mathsf{hsf}_3 \notin W$, then reaction (13) is enabled,
- if $\mathsf{hsf}_3 \in W$ and $\mathsf{hsp} \notin W$, then reaction (7) is enabled,
- if $\{\mathsf{hsf}_3, \mathsf{hsp}\} \subseteq W$ and $\mathsf{mfp} \notin W$, then reaction (8) is enabled, and
- if $\{\mathsf{hsf}_3, \mathsf{hsp}, \mathsf{mfp}\} \subseteq W$, then reaction (9) is enabled.

The product sets of these reactions contain either hse or hsf_3: hse, so $\mathrm{res}_\mathcal{A}(W) \cap M \neq \varnothing$. Similarly we can argue that if $\mathsf{hsf}_3\colon\mathsf{hse} \in W \cap M$, then either of the

reactions (10), (11), and (12) are enabled and therefore $\mathrm{res}_{\mathcal{A}}(W) \cap M \neq \varnothing$ as well. As a result we can conclude that whenever $\mathrm{res}_{\mathcal{A}}(W) \cap M \neq \varnothing$, then $W \cap M \neq \varnothing$. Consequently, $W \cap M \neq \varnothing$ if and only if $\mathrm{res}_{\mathcal{A}}(W) \cap M \neq \varnothing$, i.e., M is a conserved set.

3 The Back-End of `WEBRSIM`

In this section we will recall the algorithms behind `WEBRSIM`, briefly discuss its design and Haskell-based implementation `brsim`, and show an example of how a concrete reaction system can be analysed with `brsim`.

3.1 Direct Simulation

The principal goal of `brsim` is automating the execution of reaction systems to avoid error-prone manual analysis. Due to the fact that, technically, reaction systems are a particular case of set rewriting, their simulation is straightforward. The worst-case time complexity of the simulation can be bounded in the following way:

1. Checking that a reaction $a = (R_a, I_a, P_a)$ is enabled on a set W can be done in $O(m \log |W|)$, where $m = \max(|R_a|, |I_a|)$ (e.g., see the documentation of the module `Data.Set` [21] for details and sharper bounds).

2. Filtering a set of reactions A to only keep the ones enabled on the set W can be done in $O(|A| \cdot m \log |W|)$, where $m = \max\{|R_a|, |I_a| : a = (R_a, I_a, P_a) \in A\}$ (extension of the definition from the previous paragraph).

3. Applying a set of enabled reactions A to a set W essentially consists in putting together all the product sets in A; the complexity of this operation does not depend on the size of W. Taking the union of all product sets in A can be done in $O(|A| \cdot k \log k)$, where $k = \max\{|P_a| : a = (R_a, I_a, P_a) \in A\}$ is the maximal size of a product set in A.

4. Finally, running the whole set of reactions A for t steps can be done in $O\Big(t|A| \cdot (m \log |S| + k \log k)\Big)$, where k and m are defined as in points 2 and 3, and S is the universe of species.

Thus, a rougher but easier to read upper bound on the worst-case complexity of a t-step simulation of a reaction system (S, A) is $O\Big(t|A| \cdot |S| \log |S|\Big)$.

3.2 Mass Conservation Analysis

Besides directly simulating reaction systems, `brsim` can also list all conserved sets. As mentioned in [1], deciding whether a set M is conserved in a given

reaction system is coNP-complete. It turns out that, in case one needs to enumerate *all* sets conserved by a given reaction system, it is possible to do better than just going through every possible set of species and checking whether it is conserved [2].

We will now briefly recall the central ideas of the algorithm for listing all conserved sets, presented in detail in [2]. The main steps are as follows:

1. *Build the behaviour graph G_b*: The behaviour graph of a reaction system (S, A) is the graph whose nodes are all subsets of S and which contains the edge (W, W') iff $res_A(W) = W'$.

2. *Build the conservation dependency graph G_{cd}*: The conservation dependency graph of a reaction system (S, A) is a graph whose vertices are species from S, and which contains the edge (x, y) if y appears in at least one of the sets of the connected component of G_b which also contains the singleton set $\{x\}$. The graph G_{cd} has an important property: if it contains the edge (x, y), then every conserved set that contains y must also contain x.

3. *Build the condensation $\widetilde{G}_{cd}$ of G_{cd}*: The condensation of a directed graph is the directed acyclic graph of its strongly connected components.

4. *Enumerate the source sets of $\widetilde{G}_{cd}$*: Given a directed acyclic graph $G = (V, E)$, a subset of its nodes $X \subseteq V$ is a source set if all edges in E involving a node $x \in X$ have x as the source node.

5. *Check which source sets are also conserved sets*: As shown in [2, Proposition 3.3], any set conserved by a given reaction system must be a source set of $\widetilde{G}_{cd}$. The converse implication is not necessarily true, meaning that one still needs to check conservation directly, but for a reduced number of candidates.

Subsection 4.2 of [2] showcases the performance of this algorithm on multiple examples, showing situations in which the number of candidates is considerably reduced, as well as situations in which all subsets of species must eventually be analysed.

Our simulator `brsim` includes an exact implementation of the algorithm, including the optimisations further reducing the number of candidates. Moreover, `brsim` can output the behaviour graph and the conservation dependency graph of a given reaction system.

3.3 Overview of the Implementation

`brsim` is written in Haskell and is distributed under GPLv3 [15]. The source code is freely available in an online Git repository [22].

`brsim` is a stand-alone command line application. It is principally intended to be run in batch mode, in which it reads the description of a reaction system and the context sequence from a file and outputs the sequence of states the reaction system traverses (the result sequence). `brsim` can be also run in interactive mode, in which it prompts the user to input the next context at every simulation step. Finally, `brsim` can be told to output the behaviour graph or the conservation dependency graph of the reaction system. Further technical details can be found on the project page [22], as well as by running `brsim help`.

Haskell was chosen as the implementation language for `brsim` mainly because it is a strictly typed, functional programming language, implying easy transposition of formal definitions into runnable code, as well as verifying some properties of the software at compile time via type checking. Moreover, Haskell has a rich ecosystem including libraries for graph manipulation and quick development of user interfaces. Finally, an overwhelming majority of Haskell's ecosystem is distributed under free or open-source software licences, meaning that `brsim` can be freely reused (provided the terms of the license are respected).

3.4 Example of Usage

In this section we will briefly show how `brsim` can be used to analyse the reaction system from our running example.

To run a reaction system, `brsim` needs its list of reactions. The input format is very close to the typical way in which reactions are written. For example, the reaction $(\{a, b\}, \{e, f\}, \{c, d\})$ will be transcribed as follows: `a b, e f, c d`. Table 3 shows how to transcribe the reaction system from the running example.

A similar syntax is used to supply `brsim` with a context sequence to drive the activity of the reaction system. Table 4 shows how do transcribe the context sequence from the running example.

We can now run the simulator using the following command:

```
brsim run example.rs -context=example.ctx
```

The output of this command is shown in Table 5.

To see which sets are conserved by the reaction system from the running example, the following command should be issued to `brsim`:

```
brsim show conserved-sets example.rs
```

The output produced for the reaction system from the running example is shown in Table 6. The dot in the first line of the listing stands for the empty set (which is always trivially conserved).

Finally, `brsim` can be told to show the behaviour graph and the conservation dependency graph of the reaction system from the running example using the following respective commands:

```
hsf , hsp , hsf3
hsf hsp mfp , dI , hsf3
hsf3 , hse hsp , hsf
hsf3 hsp mfp , hse , hsf
hsf3 hse , hsp , hsf3 : hse
hsf3 hse hsp mfp , dI , hsf3 : hse
hse , hsf3 , hse
hse hsf3 hsp , mfp , hse
hsf3 : hse , hsp , hsf3 : hse hsp
hsf3 : hse hsp mfp , dI , hsf3 : hse hsp

hsp hsf , mfp , hsp : hsf
hsp : hsf stress , nostress , hsp hsf
hsp : hsf nostress , stress , hsp : hsf
hsp hsf3 , mfp , hsp : hsf
hsp hsf3 : hse , mfp , hsp : hsf hse
prot stress , nostress , prot mfp
prot nostress , stress , prot
hsp mfp , dI , hsp : mfp
mfp , hsp , mfp
hsp : mfp , dI , hsp prot
```

Table 3: The input file `example.rs` describing the reaction system from the running example.

```
hsf prot hse nostress
nostress
nostress
nostress
nostress
nostress
```

Table 4: The input file `example.ctx` describing the context sequence from the running example.

```
hse hsf3 prot
hsf3:hse prot
hsf3:hse hsp prot
hse hsp:hsf prot
hse hsp:hsf prot
hse hsp:hsf prot
```

Table 5: The output of `brsim run example.rs -context=example.ctx`.

```
.
hse hsf3:hse
```

Table 6: The output of `brsim show conserved-sets example.rs`.

```
brsim show behaviour-graph example.rs
brsim show cons-dep-graph example.rs
```

Figure 1 shows the conservation dependency graph of the reaction system from the running example as output by `brsim` and rendered using a circular layout filter. We do not show the behaviour graph here for reasons of space: it contains 2048 nodes (since the reaction system has 11 species).

4 The Web-Service

From the very beginning, we developed `brsim` with portability and ease of use in mind. To improve user experience even further, we decided to also deploy an online version, which is freely available here [23]. This page includes two examples giving some typical usage patterns.

The online version of `brsim` covers an essential subset of the functionality: simulating reaction systems, enumerating all conserved sets, and showing the behaviour and the conservation dependency graphs. As before, reactions are described in a format close to how they are usually written. For example, `hse hsf3 hsp, mfp, hse` would stand for the reaction $(\{\mathsf{hse}, \mathsf{hsf}_3, \mathsf{hsp}\}, \{\mathsf{mfp}\}, \{\mathsf{hse}\})$. To run a simulation, the reactions and the context sequence should be given in the two text fields available on the page. The context sequence is not required for enumerating the conserved sets or for showing the conservation dependency graph. Obviously, since the behaviour graph's size is exponential in the size of the background set, its calculation is only possible for modest-sized models.

The implementation of the online instance of the simulator is quite straightforward: when the user clicks on one of the four buttons "Simulate", "Behaviour graph", "Conservation dependency graph", or "Conserved sets", the

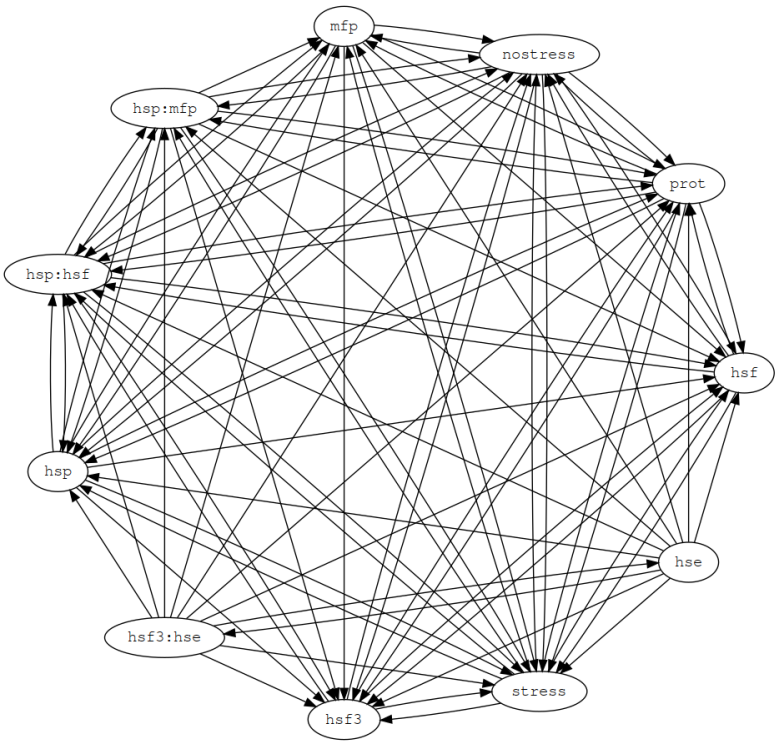

Figure 1: The conservation dependency graph of the reaction system from the running example as constructed by `brsim` and rendered using a circular layout filter.

web server executes the corresponding command of `brsim`. The result, when ready, is rendered on the page. The online interface was mainly written in basic PHP and JavaScript, the staple technologies of web programming.

5 Discussion

Reaction systems is the latest addition to the burgeoning field of computational modelling frameworks for complex systems. It is a qualitative framework focusing on the cause-effects of various events in a given system: a reaction can only be triggered once its reactants were produced (by other reactions) and none of its inhibitors are present in the environment. In particular, it gives the modeller the ability to trace explicitly why a certain event (reaction) was eventually triggered. Its two main principles, the non-permanency assumption and the threshold assumption, distinguish it quite drastically from other modelling frameworks, both quantitative (say ODE-based, or Petri net-based), or qualitative (say Boolean networks). In our experience, see [3], [4], [5], the modeller is forced in reaction systems to be explicit about the nature of all interactions in the system, rather than rely on

the interplay between numerical parameters of the system, which necessarily brings a plus of clarity in the model's assumptions.

Having an easy-to-use computer support for reaction systems is a crucial ingredient in developing the research on this topic, both the theoretical line exploring their mathematical properties, as well as the applied one exploring their modelling expressivity in various domains. This is exactly where `WEBRSIM` contributes, by offering a user-friendly, browser-based software for reaction systems. The software takes a simple textual input and it gives a simple output directly in the browser. It hides from the user the technical details of the implementation through its web-based service. At the same time, the source code of the back-end engine is openly available at [22] for the experienced users to experiment with.

`WEBRSIM` is highly efficient, even with its current hosting on a standard web server. For example, we ran the simulation for the reaction system model of the Erbb signalling as the benchmark of [18]. This model comprised of 6720 reactions and the simulation in `WEBRSIM` was successfully performed in less than three seconds. Detailed experiments done in [18] show that in fact `WEBRSIM`'s computational efficiency is only marginally less than that of a highly parallelised version of `HERESY` [18], running on a dedicated GPU hardware with thousands of processors. `WEBRSIM` thus gives access to the standard user to a highly efficient simulation and analysis platform for 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. *Theor. Comput. Sci.*, 623:103–113, 2016.

[2] Sepinoud Azimi, Cristian Gratie, Sergiu Ivanov, and Ion Petre. Dependency graphs and mass conservation in reaction systems. *Theor. Comput. Sci.*, 598:23–39, 2015.

[3] Sepinoud Azimi, Bogdan Iancu, and Ion Petre. Reaction system models for the heat shock response. *Fundamenta Informaticae*, 131(3-4):299–312, 2014.

[4] Sepinoud Azimi, Charmi Panchal, Eugen Czeizler, and Ion Petre. Reaction systems models for the self-assembly of intermediate filaments. *Ann. Univ. Buchar*, 62(2):9–24, 2015.

[5] Sepinoud Azimi, Charmi Panchal, Andrzej Mizera, and Ion Petre. Multistability, limit cycles, and period-doubling bifurcation with reaction

systems. *International Journal of Foundations of Computer Science*, 28(08):1007–1020, 2017.

[6] Robert Brijder, Andrzej Ehrenfeucht, and Grzegorz Rozenberg. A note on causalities in reaction systems. *Electronic Communications of the EASST*, 30:1–10, 2010.

[7] Robert Brijder, Andrzej Ehrenfeucht, and Grzegorz Rozenberg. Reaction systems with duration. In *Computation, cooperation, and life*, pages 191–202. Springer, 2011.

[8] 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.

[9] 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.

[10] Andrzej Ehrenfeucht, Michael Main, and Grzegorz Rozenberg. Combinatorics of life and death for reaction systems. *International Journal of Foundations of Computer Science*, 21(03):345–356, 2010.

[11] Andrzej Ehrenfeucht, Ion Petre, and Grzegorz Rozenberg. *Reaction Systems: A Model of Computation Inspired by the Functioning of the Living Cell*, chapter Chapter 1, pages 1–32. WORLD SCIENTIFIC, 2017.

[12] Andrzej Ehrenfeucht and Grzegorz Rozenberg. Events and modules in reaction systems. *Theoretical Computer Science*, 376(1):3–16, 2007.

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

[14] Andrzej Ehrenfeucht and Grzegorz Rozenberg. Introducing time in reaction systems. *Theoretical Computer Science*, 410(4):310–322, 2009.

[15] Free Software Foundation. GNU General Public License. `http://www.gnu.org/licenses/gpl.html`.

[16] Mika Hirvensalo. Quantum computing. In *Encyclopedia of Sciences and Religions*, pages 1922–1926. Springer, 2013.

[17] Artur Meski, Wojciech Penczek, and Grzegorz Rozenberg. Model checking temporal properties of reaction systems. *Information Sciences*, 313:24–42, 2015.

[18] Marco S Nobile, Antonio E Porreca, Simone Spolaor, Luca Manzoni, Paolo Cazzaniga, Giancarlo Mauri, and Daniela Besozzi. Efficient simulation of reaction systems on graphics processing units. *Fundamenta Informaticae*, 154(1-4):307–321, 2017.

[19] Ion Petre, Andrzej Mizera, Claire L. Hyder, Annika Meinander, Andrey Mikhailov, Richard I. Morimoto, Lea Sistonen, John E. Eriksson, and Ralph-Johan Back. A simple mass-action model for the eukaryotic heat shock response and its mathematical validation. *Natural Computing*, 10(1):595–612, 2011.

[20] Arto Salomaa. Applications of the chinese remainder theorem to reaction systems with duration. *Theoretical Computer Science*, 598:15–22, 2015.

[21] Documentation of `Data.Set`. `http://hackage.haskell.org/package/containers-0.5.11`

[22] GitHub — scolobb/brsim: A Basic Reaction System Simulator. `https://github.com/scolobb/brsim`.

[23] Web interface for Basic Reaction System Simulator. `http://combio.abo.fi/research/reaction-systems/reaction-system-simulator/`.

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 978-952-12-3715-7
ISSN 1239-1891