Reachability in Resource-Bounded Reaction Systems*

Alberto Dennunzio[1], Enrico Formenti[2], Luca Manzoni[1], and Antonio E. Porreca[1]

[1] Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli Studi di Milano-Bicocca, Viale Sarca 336/14, 20126 Milano, Italy
{dennunzio, luca.manzoni, porreca}@disco.unimib.it
[2] Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France
enrico.formenti@unice.fr

Abstract. Reaction systems, a formalism describing biochemical reactions in terms of sets of reactants, inhibitors, and products, are known to have a **PSPACE**-complete configuration reachability problem. We show that the complexity of the problem remains unchanged even for some classes of resource-bounded reaction systems, where we disallow either inhibitors or reactants. We also prove that the complexity decreases to **NP** in the specific case of inhibitorless reaction systems using only *one* reactant per reaction.

1 Introduction

During the last decades, many new computing models have been introduced. Each one was meant to more clearly illustrate some features or provide new settings for developing new computing technologies. In most cases, nature has been the main source of inspiration. In 2004, Ehrenfeucht and Rozenberg introduced *reaction systems* (RS in the following) as an abstract model of chemical reactions in living cells [5,6]. Indeed, in living cells, a biochemical reaction takes place only whenever *reactants* are present and *inhibitors* are missing. Hence, a reaction can be represented by a triple (R, I, P) where R is the set of reactants, I the inhibitor and P is the set of products which are left once the reaction is finished. Of course, one has to require that $R \cap I = \varnothing$. Informally, a RS is a (finite) collection of reactions.

The simple definition of the model contrasts with its computing capabilities. In fact, RS are capable of simulating any space-bounded Turing machine computation (many constructions have been provided, one is also given in Section 3). Moreover, they provide new examples of natural problems in higher levels of the polynomial hierarchy [8,7]. As a third argument in favour of studying RS, one may advocate

* This work has been supported by Fondo d'Ateneo (FA) 2013 of Università degli Studi di Milano-Bicocca: "Complessità computazionale in modelli di calcolo bioispirati: Sistemi a membrane e sistemi a reazioni" and by the Italian MIUR PRIN 2010-2011 grant "Automata and Formal Languages: Mathematical and Applicative Aspects" H41J12000190001.

that they are a reference model for other finite systems. Indeed, in [7], it is proved that RS provide lower complexity bounds for Boolean automata networks (BAN). We remark that in the context of BAN, the complexity of relatively few problems is known (see for instance [14]).

This paper pursues the study of complexity problems for RS in the same vein as [8] and subsequent papers [7,9,3]. The new focus is on resource-bounded computation. The idea is to take a classical and important reference problem, namely the *reachability problem*, and try to see how its complexity varies according to constraints that are put on reactions. The constraints we impose consist in limiting the maximum number of reactants and inhibitors involved in each reaction; this changes how much of the current state can be "observed" by a single reaction. In principle, the resulting dynamical behaviours of reaction systems are less rich than for unrestricted systems, although this does not necessarily reduce the complexity of the reachability problem.

From [11], it is known that result functions (state transition functions) of RS can be completely classified into five classes of functions over lattices, which correspond to specific limitations on the number of reactants and inhibitors allowed in each reaction of the corresponding RS. Theorems 1 and 2 prove that the reachability problem is **PSPACE**-complete for three out of the five classes. The class of result functions computed by RS with no reactants and no inhibitors corresponds to constant functions, making the reachability problem very simple. Concerning the fifth class, we only succeeded in proving that reachability is in **NP** (Theorem 4) but we suspect that it is also **NP**-hard. Indeed, a slight variant of the reachability problem is **NP**-complete for this class. The proof of this last result is also of some interest in its own. Indeed, it uses the Prime Number Theorem to precisely evaluate the complexity of the reduction.

2 Basic Notions

This section briefly recalls the basic notions about RS as introduced in [6]. We remark that in this paper the set of reactants and inhibitors of a reaction are allowed to be empty, unlike what is often required in literature. The reason for this generalised definition is that, as it will be shown later (Corollary 1), the reachability problem for "minimal" RS [13], having exactly one reactant and one inhibitor per reaction, is already **PSPACE**-complete.

Definition 1. *Consider a finite set S, whose elements are called* entities. *A* reaction *a over S is a triple (R_a, I_a, P_a) of subsets of S. The set R_a is the set of* reactants, *I_a the set of* inhibitors, *and P_a is the nonempty set of* products. *The set of all reactions over S is denoted by* $\mathrm{rac}(S)$.

Definition 2. *A* reaction system (RS) *is a pair $\mathcal{A} = (S, A)$ where S is a finite set, called the* background set, *and $A \subseteq \mathrm{rac}(S)$.*

Given a *state* $T \subseteq S$, a reaction a is said to be *enabled* in T when $R_a \subseteq T$ and $I_a \cap T = \varnothing$. The *result function* $\mathrm{res}_a \colon 2^S \to 2^S$ of a, where 2^S denotes the

Class of RS	Subclass of $2^S \to 2^S$	(via k-simulation)
$\mathcal{RS}(\infty, \infty)$	all	$\mathcal{RS}(1, 1)$
$\mathcal{RS}(0, \infty)$	antitone	$\mathcal{RS}(0, 1)$
$\mathcal{RS}(\infty, 0)$	monotone	$\mathcal{RS}(2, 0)$
$\mathcal{RS}(1, 0)$	additive	$\mathcal{RS}(1, 0)$
$\mathcal{RS}(0, 0)$	constant	$\mathcal{RS}(0, 0)$

Fig. 1. Functions computed by restricted classed of RS.

power set of S, is defined as $\text{res}_a(T) = P_a$ if a is enabled in T, and $\text{res}_a(T) = \varnothing$ otherwise. The definition of res_a naturally extends to sets of reactions: given $T \subseteq S$ and $A \subseteq \text{rac}(S)$, define $\text{res}_A(T) = \bigcup_{a \in A} \text{res}_a(T)$. The result function $\text{res}_{\mathcal{A}}$ of a RS $\mathcal{A} = (S, A)$ is res_A, i.e., the result function on the whole set of reactions. In this way, any RS $\mathcal{A} = (S, A)$ induces a discrete dynamical system where the state set is 2^S and the next state function is $\text{res}_{\mathcal{A}}$. The set of reactions of $\mathcal{A}$ enabled in a state T is denoted by $\text{en}_{\mathcal{A}}(T)$.

The *orbit* or *state sequence* of a given state T of a RS $\mathcal{A}$ is defined as the sequence of states obtained by iterations of $\text{res}_{\mathcal{A}}$ starting from T, namely the sequence $(T, \text{res}_{\mathcal{A}}(T), \text{res}^2_{\mathcal{A}}(T), \ldots)$. Being finite systems, RS only admit ultimately periodic orbits, i.e., orbits ending up in a cycle.

We now recall the classification of RS in terms of number of resources employed per reaction [11].

Definition 3. *Let $r, i \in \mathbb{N}$. The class $\mathcal{RS}(r, i)$ consists of all RS having at most r reactants and i inhibitors for reaction. We also define the unbounded classes $\mathcal{RS}(\infty, i) = \bigcup_{r=0}^{\infty} \mathcal{RS}(r, i)$, $\mathcal{RS}(r, \infty) = \bigcup_{i=0}^{\infty} \mathcal{RS}(r, i)$, and $\mathcal{RS}(\infty, \infty) = \bigcup_{r=0}^{\infty} \bigcup_{i=0}^{\infty} \mathcal{RS}(r, i)$.*

We remark that this classification does not include the number of products as a parameter, since RS can always be assumed to be in *singleton product normal form* [2]: any reaction $(R, I, \{p_1, \ldots, p_m\})$ can be replaced by the set of reactions $(R, I, \{p_1\}), \ldots, (R, I, \{p_m\})$, since they produce the same result.

Several of the above defined classed have a characterisation in terms of functions over the Boolean lattice 2^S [11]. Recall that a function $f \colon 2^S \to 2^S$ is *antitone* if $X \subseteq Y$ implies $f(X) \supseteq f(Y)$, *monotone* if $X \subseteq Y$ implies $f(X) \subseteq f(Y)$, *additive* (or an *upper-semilattice endomorphism*) if $f(X \cup Y) = f(X) \cup f(Y)$. We say that the RS $\mathcal{A} = (S, A)$ computes the function $f \colon 2^S \to 2^S$ if $\text{res}_{\mathcal{A}} = f$. Furthermore, we say that the RS $\mathcal{A} = (S', A)$ computes a function $f \colon 2^S \to 2^S$ via k-simulation if $S \subseteq S'$ and $\text{res}^k_{\mathcal{A}}(T) \cap S = f(T)$ for all $T \subseteq S$. The (distinct) classes of functions computed by restricted classes of RS are illustrated in Figure 1. These results show that, in a sense, the classes $\mathcal{RS}(1, 1)$, $\mathcal{RS}(0, 1)$, and $\mathcal{RS}(2, 0)$ capture the expressiveness of the whole classes $\mathcal{RS}(\infty, \infty)$, $\mathcal{RS}(0, \infty)$, and $\mathcal{RS}(\infty, 0)$, respectively (i.e., they simulate the more generic RS with a polynomial slowdown).

We conclude this section by recalling the formulation of the problem addressed in this paper.

Definition 4. *The* reachability problem *for the class $\mathcal{RS}(i, r)$, with i and r possibly infinite, consists of deciding, given $\mathcal{A} \in \mathcal{RS}(i, r)$ and two of its states T, U, whether U is reachable from T, i.e., whether* $\mathrm{res}_{\mathcal{A}}^{t}(T) = U$ *for some $t \geq 0$.*

For the notions of complexity theory, such as the definitions of the classes of problems **NP** and **PSPACE**, we refer the reader to any relevant textbook, such as [12].

3 Inhibitorless Classes $\mathcal{RS}(\infty, 0)$ and $\mathcal{RS}(2, 0)$

We begin by describing how inhibitorless RS are able to efficiently simulate Turing machines with bounded tape. A similar simulation was previously published [7], but it required both reactants and inhibitors.

In the following, let M be any single-tape deterministic Turing machine using m tape cells during its computation; let Σ be the tape alphabet, Q the set of states, and $\delta \colon Q \times \Sigma \to Q \times \Sigma \times \{-1, 0, +1\}$ the transition function of M. We are going to define a RS $\mathcal{M} = (S, A) \in \mathcal{RS}(2, 0)$ simulating M.

Entities. The set of entities of $\mathcal{M}$ is

$$S = \{a_j : a \in \Sigma, 1 \leq j \leq m\} \cup \{q_j : q \in Q, 1 \leq j \leq m\} \cup \{\spadesuit_j : 1 \leq j \leq m\}$$

that is, it consists of all symbols of the alphabet, states, and the extra item $\spadesuit$, each of them indexed by every possible tape position.

In this way, the generic configuration where M is in state $q \in Q$, its tape head is located on cell i, and its tape contains the string $x = x_1 \cdots x_m$, is encoded as the following $2m$-entity state:

$$T = \{x_{j,j} : 1 \leq j \leq m\} \cup \{q_i\} \cup \{\spadesuit_j : 1 \leq j \leq m, j \neq i\} \subseteq S$$

In other terms, T contains each symbol x_j of the string x indexed by its position on the tape as element $x_{j,j}$, an entity q_i storing both the current state and the head position, and $m - 1$ entities $\spadesuit_j$, one for each position $j \neq i$ on which the tape head is *not* located.

Example 1. Consider a Turing machine M working in space $m = 4$ and the configuration where M is in state q, its tape head is located on cell 3, and its tape contains the string *abba*. The state of the RS $\mathcal{M}$ encoding such a configuration of M is then $T = \{a_1, b_2, b_3, a_4, \spadesuit_1, \spadesuit_2, q_3, \spadesuit_4\}$.

Reactions. Each transition $\delta(q, a) = (r, b, d)$ of M, with $q, r \in Q$, $a, b \in \Sigma$, and $d \in \{-1, 0, +1\}$, gives rise to the following two sets of reactions:

$$(\{q_i, a_i\}, \varnothing, \{r_{j+d}, b_i\}) \qquad \text{for } 1 \leq i \leq m \tag{1}$$

$$(\{q_i, a_i\}, \varnothing, \{\spadesuit_j : 1 \leq j \leq m, j \neq i + d\}) \qquad \text{for } 1 \leq i \leq m. \tag{2}$$

If the tape head of M is located on cell i, then the i-th reaction from (1) produces the entity encoding both the new state and the new tape head position of M, as well as the symbol written in the position i over the head. The production of one $\spadesuit_j$ for all the tape positions $j \neq i + d$, i.e., those on which the tape head is *not* located after the transition of M, is assured by the i-th reaction from (2).

Finally, the following reactions preserve the encoding of the tape cells j where the head is *not* located, i.e., those indicated by the presence of $\spadesuit_j$:

$$(\{\spadesuit_j, a_j\}, \varnothing, \{a_j\}) \qquad \text{for } 1 \leq j \leq m. \tag{3}$$

The set A of the reactions of $\mathcal{M}$ is defined as the union of the three sets of reactions from (1), (2), and (3).

It is easy to see that if $T \subseteq S$ is an encoding of a configuration of M using space m (i.e., it contains, for all $1 \leq j \leq m$, a single entity a_j for some $a \in \Sigma$, a single entity q_i for some $q \in Q$ and some $1 \leq i \leq m$, and entities $\spadesuit_j$ for all $1 \leq j \leq m$ with $j \neq i$), then the next state $\text{res}_{\mathcal{M}}(T)$ encodes the next configuration of M.

We remark that all reactions of $\mathcal{M}$ have exactly two reactants and no inhibitors, that is $\mathcal{M} \in \mathcal{RS}(2, 0)$. We are now able to prove the following:

Lemma 1. *Reachability for $\mathcal{RS}(2, 0)$ is* **PSPACE**-*hard.*

Proof. We reduce reachability of configurations of polynomial-space Turing machines (one of the canonical **PSPACE**-complete problems [12]) to this problem. Given a Turing machine M working in space m and two configurations $\mathcal{C}_1$, $\mathcal{C}_2$ of M, it is possible to build the RS $\mathcal{M} \in \mathcal{RS}(2, 0)$ simulating M as described above; the construction can be done in polynomial time, since the reactions can be built by iterating over all entries of the transition table of M and the range of the m possible tape positions. The question then becomes whether in the RS $\mathcal{M}$ the encoding of the configuration $\mathcal{C}_2$ is reachable from the encoding of the configuration $\mathcal{C}_1$; the construction of $\mathcal{M}$ assures that this happens if and only if $\mathcal{C}_2$ is reachable in M from $\mathcal{C}_1$. Therefore, the reduction holds and reachability for $\mathcal{RS}(2, 0)$ is then **PSPACE**-hard. □

We conclude this section with the complexity result for the inhibitorless classes.

Theorem 1. *Reachability for $\mathcal{RS}(\infty, 0)$ and for $\mathcal{RS}(2, 0)$ is* **PSPACE**-*complete.*

Proof. Recall that reachability for $\mathcal{RS}(\infty, \infty)$ can be decided in polynomial space, by storing the current configuration of the involved RS and applying the reactions one by one at each time step. The thesis follows as a consequence of this fact and Lemma 1. □

4 Reactantless Classes $\mathcal{RS}(0, 1)$ and $\mathcal{RS}(0, \infty)$

It is known [11] that each RS from $\mathcal{RS}(\infty, 0)$ can be simulated with a linear slowdown by a RS from $\mathcal{RS}(0, 1)$. Since the two classes of RS are equivalent from this point of view, it is reasonable to assume that their reachability problems have the same complexity, with reachability for $\mathcal{RS}(0,1)$ being **PSPACE**-complete as well. However, the original simulation does not directly imply this result, since each state of the simulating RS contains a number of auxiliary entities, and it is not obvious which auxiliary entities must appear in the target state.

Therefore, in the next Lemma, we provide a construction, with no auxiliary entities appearing, of an RS from $\mathcal{RS}(0,1)$ simulating a given RS from $\mathcal{RS}(\infty,0)$. In this simulation, the states at even time steps of the former coincide exactly with the states of the simulated RS.

Lemma 2. *Let $\mathcal{A} = (S, A) \in \mathcal{RS}(\infty, 0)$ be a RS such that $\bigcup\{P_a : a \in A\} = S$ and $\mathrm{res}_{\mathcal{A}}(S) = S$. Then, there exists a RS $\mathcal{B} = (S', A') \in \mathcal{RS}(0,1)$ such that, for any $T \subseteq S$, the following condition holds:*

$$\forall t \in \mathbb{N} \qquad \mathrm{res}_{\mathcal{B}}^{2t}(T) = \mathrm{res}_{\mathcal{A}}^{t}(T) \wedge S \subseteq \mathrm{res}_{\mathcal{B}}^{2t+1}(T).$$

Proof. Set $S' = S \cup \{\bar{a} : a \in A\}$, that is, S' is obtained by adding to S one barred entity for each reaction of $\mathcal{A}$. For each reaction $a = (R_a, \varnothing, P_a) \in A$, the set A' contains the reactions

$$(\varnothing, \{s\}, \{\bar{a}\}) \qquad\qquad \text{for } s \in R_a \tag{4}$$

which produce the entity $\bar{a}$ if at least one of the reactants of a is missing in the current state (i.e., if a is not enabled in it). Furthermore, for each $a = (R_a, \varnothing, P_a) \in A$ the set A' also contains the reaction

$$(\varnothing, \{\bar{a}\}, P_a) \tag{5}$$

which gives the same products as a when $\bar{a}$ is missing in the current state, or, equivalently, when a is enabled.

Thus, for any state $T \subseteq S$ and any state $T' \subseteq \{\bar{a} : a \in A\}$ it holds that $\mathrm{res}_{\mathcal{B}}(T) = \{\bar{a} : a \notin \mathrm{en}_{\mathcal{A}}(T)\} \cup S$ and $\mathrm{res}_{\mathcal{B}}(T') = \bigcup\{P_a : a \in A, \bar{a} \notin T'\}$.

Choose now an arbitrary state $T \subseteq S$. We are going to prove the thesis condition by induction on t. Clearly, $\mathrm{res}_{\mathcal{B}}^{2\cdot 0}(T) = T = \mathrm{res}_{\mathcal{A}}^{0}(T)$. Furthermore, since T contains no entity $\bar{a}$, all reactions of type (5) are enabled, and so $\mathrm{res}_{\mathcal{B}}^{2\cdot 0+1}(T) \supseteq \bigcup\{P_a : a \in A\} = S$.

Assume now that the thesis condition holds for t. Then,

$$\begin{aligned}
\mathrm{res}_{\mathcal{B}}^{2(t+1)}(T) &= \mathrm{res}_{\mathcal{B}}^{2}\left(\mathrm{res}_{\mathcal{B}}^{2t}(T)\right) \\
&= \mathrm{res}_{\mathcal{B}}^{2}\left(\mathrm{res}_{\mathcal{A}}^{t}(T)\right) \\
&= \mathrm{res}_{\mathcal{B}}\left(\{\bar{a} : a \notin \mathrm{en}_{\mathcal{A}}(\mathrm{res}_{\mathcal{A}}^{t}(T))\} \cup S\right)
\end{aligned}$$

Since S disables all reactions of type (4), it follows that

$$\begin{aligned}
\mathrm{res}_{\mathcal{B}}^{2(t+1)}(T) &= \mathrm{res}_{\mathcal{B}}\left(\{\bar{a} : a \notin \mathrm{en}_{\mathcal{A}}(\mathrm{res}_{\mathcal{A}}^{t}(T))\}\right) \\
&= \bigcup\{P_a : a \in \mathrm{en}_{\mathcal{A}}(\mathrm{res}_{\mathcal{A}}^{t}(T))\} \\
&= \mathrm{res}_{\mathcal{A}}^{t+1}(T)
\end{aligned}$$

In particular, due to the reactions of type (5), one obtains that $S \subseteq \mathrm{res}_{\mathcal{B}}^{2t+1}(T)$ for all $t \in \mathbb{N}$. □

By exploiting Lemma 2 we can finally show the complexity of reachability for reactantless RS.

Theorem 2. *Reachability for $\mathcal{RS}(0,1)$, and thus for $\mathcal{RS}(0,\infty)$, is* **PSPACE**-*complete.*

Proof. It is enough to prove the **PSPACE**-hardness of the problem for $\mathcal{RS}(0,1)$, which will be accomplished by reduction from reachability for $\mathcal{RS}(\infty,0)$ as follows. Given a RS $\mathcal{A} = (S, A) \in \mathcal{RS}(\infty, 0)$, let $\mathcal{A}' = (S', A')$ be the RS with $S' = S \cup \{\spadesuit\}$ for some $\spadesuit \notin A$, and $A' = A \cup \{(S', \varnothing, S')\}$. Clearly, $\mathcal{A}' \in \mathcal{RS}(\infty, 0)$ and it has the same behaviour of $\mathcal{A}$ as long as its initial state does not contain $\spadesuit$, i.e., $\mathrm{res}_{\mathcal{A}'}(T) = \mathrm{res}_{\mathcal{A}}(T)$ whenever $\spadesuit \notin T$.

Moreover, $\mathcal{A}'$ satisfies the hypotheses of Lemma 2 as a consequence of the changes made to build it from $\mathcal{A}$. Let $\mathcal{B} \in \mathcal{RS}(0,1)$ be then the RS obtained from $\mathcal{A}'$ using that lemma. Notice that that the mapping $\mathcal{A}' \mapsto \mathcal{B}$ can be computed in polynomial time.

For any two states $U, V \subseteq S$, it holds that $\mathrm{res}^t_{\mathcal{A}}(U) = V$ for some $t \in \mathbb{N}$ if and only if $\mathrm{res}^{2t}_{\mathcal{B}}(U) = V$. Furthermore, we have $\mathrm{res}^{2s+1}_{\mathcal{B}}(U) \neq V$ for all $s \in \mathbb{N}$, because $\spadesuit \in S' \subseteq \mathrm{res}^{2s+1}_{\mathcal{B}}(U)$, while $\spadesuit \notin S$ and, in particular, $\spadesuit \notin V$. Hence, the state V is reachable from U in the RS $\mathcal{B}$ if and only if the same occurs in $\mathcal{A}$. Therefore, reachability for $\mathcal{RS}(\infty,0)$ is reducible to reachability for $\mathcal{RS}(0,1)$ in polynomial time and the thesis then follows from Theorem 1. □

As a consequence of Theorem 2, we also obtain the **PSPACE**-completeness of the reachability problem for $\mathcal{RS}(1,1)$ and for the class $\mathcal{RS}(\infty,\infty)$ (the latter having already been proved in a different way [7]).

Corollary 1. *Reachability for $\mathcal{RS}(1,1)$, and thus for $\mathcal{RS}(\infty,\infty)$, is* **PSPACE**-*complete.* □

5 Single-Reactant Inhibitorless Class $\mathcal{RS}(1,0)$

We proved that disallowing either reactants or inhibitors does not decrease the complexity of reachability problems. However, reducing the number of reactants to 1 in inhibitorless RS makes the evolution of each single entity "context-free", i.e., not influenced by the presence or absence of other entities. Indeed, the result functions of the RS from $\mathcal{RS}(1,0)$ are always upper-semilattice endormorphisms, that is, $\mathrm{res}_{\mathcal{A}}(U \cup V) = \mathrm{res}_{\mathcal{A}}(U) \cup \mathrm{res}_{\mathcal{A}}(V)$ for all $\mathcal{A} \in \mathcal{RS}(1,0)$ and arbitrary states U, V [11]. It is thus reasonable to conjecture that reachability for $\mathcal{RS}(1,0)$ might be easier than for other variants, as the entities of the target configuration can be traced back to a set of originating entities independently one from another, and the only difficulty is to find a common number of backwards steps. In this section we show that this is actually the case, under the assumption that **NP** $\neq$ **PSPACE**.

We begin by recalling the notion of *influence graph* [1], which describes static causality relations in a RS. Given a RS $\mathcal{A} = (S, A)$, the associated influence graph is the directed graph $G = (S, E)$, where $(x, y) \in E$ if and only if there exists a reaction $(R_a, I_a, P_a) \in A$ such that $x \in R_a \cup I_a$ and $y \in P_a$. In other words, there is an edge (x, y) whenever the presence or absence of x contributes to the appearance of y.

In particular, if $\mathcal{A} \in \mathcal{RS}(1, 0)$ we have $(x, y) \in E$ if and only if $y \in \text{res}_{\mathcal{A}}(\{x\})$. An entity y appearing at time t can thus be recursively traced back to a single entity occurring in the initial state of the RS (or to multiple independent entities, only one of which needs to occur in order that y appears at time t).

Recall that the powers G^t of the Boolean adjacency matrix of any graph G can be computed in polynomial time even for exponential values of t, by repeated squaring; the entry $G^t_{i,j}$ is 1 if and only if a (not necessarily simple) path of length t exists between v_i and v_j.

Let $\mathcal{A} = (S, A) \in \mathcal{RS}(1, 0)$ be a RS with $S = \{s_1, \ldots, s_n\}$ and let G be its influence graph. Any state $U \subseteq S$ can be viewed as a column vector in $\{0, 1\}^n$, where $U_i = 1$ if and only if $s_i \in U$. Then, a state $V \subseteq S$ is reachable from a state $U \subseteq S$ in $\mathcal{A}$ if and only if $G^t U = V$, where $G^t U$ is the product of the matrix G^t and the vector U. This observation allows us to prove the following result:

Theorem 3. *Reachability for $\mathcal{RS}(1, 0)$ is in* **NP**.

Proof. Consider a RS $\mathcal{A} \in \mathcal{RS}(1, 0)$ and two states U, V. Let G be the influence graph of $\mathcal{A}$. Since G^t can be computed in polynomial time, the validity of the equation $G^t U = V$ can be checked in polynomial time for any fixed t, even when the latter is exponential with respect to the number n of entities. It is enough to use the guessing power of a nondeterministic Turing machine to choose an n-bit integer $0 \le t < 2^n$, since state V is either reached within $2^n - 1$ steps, or it is never reached. □

It is unknown whether this problem is also **NP**-hard. The variant where the target state V consists of a single entity is in **NL**, being the reachability problem (with several possible source vertices) for the influence graph in **NL** too. It is actually **NL**-complete, since every graph is the influence graph of a RS (the one just having the vertex set as background set and reactions $(\{u\}, \varnothing, \{v\})$, one for each edge (u, v) of the graph). On the other hand, the variant where we check if a *superset* of V is reachable is **NP**-complete.

Theorem 4. *It is* **NP**-*complete to decide, given a RS $\mathcal{A} = (S, A) \in \mathcal{RS}(1, 0)$ and two states $U, V \subseteq S$, whether $V \subseteq \text{res}^t_{\mathcal{A}}(U)$ for some $t \in \mathbb{N}$.*

Proof. Membership in **NP** is proved in a similar way to what has been done in the proof of Theorem 3, i.e, for a RS $\mathcal{A} = (S, A) \in \mathcal{RS}(1, 0)$ and two states $U, V \subseteq S$, by guessing $0 \le t < 2^n$ and checking whether $V \le G^t U$, where G is the influence graph of $\mathcal{A}$ and the comparison is made element-wise.

The **NP**-hardness of the problem is proved by reduction from Boolean satisfiability of CNF formulae [12]. Let $\varphi = \varphi_1 \wedge \cdots \wedge \varphi_m$ be a CNF formula with m clauses $C = \{\varphi_1, \ldots, \varphi_m\}$ over the n variables $X = \{x_1, \ldots, x_n\}$.

Denote by p_i the i-th prime number and set $X_i = \{x_{i,j} : 0 \leq j < p_i\}$. Define the RS $\mathcal{A} = (S, A) \in \mathcal{RS}(1, 0)$, where $S = X_1 \cup \cdots \cup X_n \cup C$ and A consists of the reactions described in the following.

For each variable x_i, we build a cycle of prime length p_i iterating across all elements of X_i by means of the reactions

$$(\{x_{i,j}\}, \varnothing, \{x_{i,(j+1) \bmod p_i}\}) \qquad \text{for } 0 \leq j < p_i. \tag{6}$$

We define a "well-formed" state T of $\mathcal{A}$ as a state containing exactly one entity $x_{i,j}$ for each $1 \leq i \leq n$. Such a state T is interpreted as the truth assignment $v \colon X \to \{0, 1\}$ to φ defined as

$$v(x_i) = \begin{cases} 1 & \text{if } x_{i,0} \in T \\ 0 & \text{otherwise} \end{cases}$$

that is, all elements $x_{i,j}$ with $j > 0$ denote a false value of x_i.

Since the lengths of the cycles associated to the variables of φ are pairwise coprime, all 2^n assignments of φ will be eventually reached in $\mathcal{A}$, possibly with several repetitions (since distinct states encode the same truth assignment). Therefore, if the initial state of $\mathcal{A}$ is $U = \{x_{1,0}, \ldots, x_{n,0}\}$, then a state encoding the assignment $v \colon X \to \{0, 1\}$ will be reached at time step $\prod_{i=1}^{n} p_i^{v(x_i)}$.

We are going to introduce the remaining reactions in A. They have the role of evaluating formula φ under the assignment encoded by the $x_{i,j}$'s. We map each entity $x_{i,j}$ to the set of clauses satisfied by $v(x_i) = 1$ (if $j = 0$) or by $v(x_i) = 0$ (if $j > 0$) by the following reactions:

$$(\{x_{i,0}\}, \varnothing, \{\varphi_k\}) \qquad \text{if } x_i \text{ implies } \varphi_k \tag{7}$$

$$(\{x_{i,j}\}, \varnothing, \{\varphi_k\}) \qquad \text{for } j > 0 \text{ if } \neg x_i \text{ implies } \varphi_k \tag{8}$$

The influence graph of the resulting RS for a sample Boolean formula is shown in Figure 2.

As a consequence of the above construction, given a well-formed assignment $Y \subseteq X_1 \cup \cdots \cup X_n$, it follows that $\mathrm{res}_{\mathcal{A}}(Y) = D \cup Y'$ where $D \subseteq C$ is exactly the set of clauses satisfied by Y, and Y' is the next truth assignment in the order given by the reactions of type (6). Since they do not appear as reactants in any reaction, the entities representing the clauses of φ appear only if the previous state encodes an assignment satisfying them.

Consider now the states $U = \{x_{1,0}, \ldots, x_{n,0}\}$ and $V = C$. According to the above reasoning, a superset of V is reachable from U, or, equivalently, $V \subseteq \mathrm{res}_{\mathcal{A}}^{t}(U)$ for some $t \in \mathbb{N}$, if and only if there exists a truth assignment for X satisfying all clauses, i.e., the entire formula φ.

It remains to be proved that the mapping $\varphi \mapsto (\mathcal{A}, U, V)$ can be computed in polynomial time. In particular, we need to show that we can find in polynomial time n primes of polynomially bounded value.

First of all, the Prime Number Theorem [4] implies that p_n is asymptotically $n \ln n$; thus, we only need to check the first $O(n \ln n)$ integers. These can

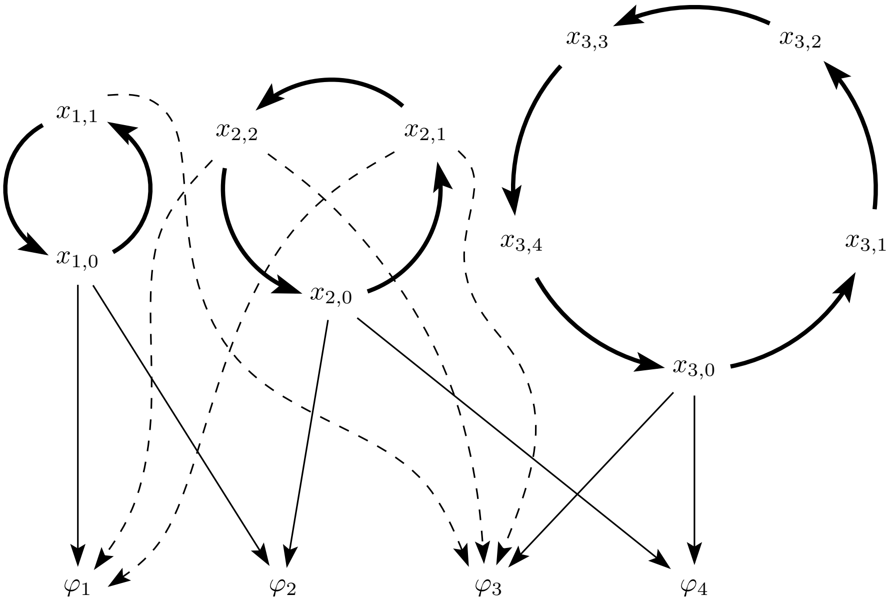

Fig. 2. The influence graph for the RS encoding the formula $\varphi \equiv \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4$, where $\varphi_1 \equiv x_1 \vee \neg x_2$, $\varphi_2 \equiv x_1 \vee x_2$, $\varphi_3 \equiv \neg x_1 \vee \neg x_2 \vee x_3$, and $\varphi_4 \equiv x_2 \vee x_3$. The thick edges represent the reactions of type (6), the continuous thin ones those of type (7), and the dashed thin ones those of type (8).

be checked for primality in polynomial time via a brute force algorithm, since they are polynomial in value with respect to n. The reactions of type (6) are then simple to compute, while those of types (7) and (8) only require to check whether variable x_i appears as a positive or negated literal in φ_k.

Therefore, Boolean satisfiability of CNF formulae is reducible to the considered problem in polynomial time and then the thesis follows. □

6 Conclusions

We proved that the reachability problem for RS remains **PSPACE**-complete, as in the general case, even when inhibitors or reactants are disallowed in each reaction. The problem only becomes easier (assuming **NP** $\neq$ **PSPACE**) for inhibitorless RS using only one reactant per reaction: this variant has been proved to be in **NP**. It is left as an open problem to establish whether this problem is also **NP**-hard, as it is in the case where we check the reachability a *superset* of a given state.

It would also be interesting to examine further problems related to the dynamics of RS, such as the detection of fixed points, global and local attractors, and “Gardens of Eden”, in order to check whether these become easier for resource-bounded RS. As a simple example, establishing the existence of fixed points, which is **NP**-complete for $\mathcal{RS}(\infty, \infty)$ [8], becomes entirely trivial for $\mathcal{RS}(\infty, 0)$, since monotonic functions over complete lattices always admit a fixed point by the Knaster-Tarski theorem [10].

References

1. Brijder, R., Ehrenfeucht, A., Main, M., Rozenberg, G.: A tour of reaction systems. International Journal of Foundations of Computer Science 22(7), 1499–1517 (2011)
2. Brijder, R., Ehrenfeucht, A., Rozenberg, G.: Reaction systems with duration. In: Kelemen, J., Kelemenová, A. (eds.) Computation, Cooperation, and Life, Lecture Notes in Computer Science, vol. 6610, pp. 191–202. Springer (2011)
3. Dennunzio, A., Formenti, E., Manzoni, L., Porreca, A.E.: Ancestors, descendants, and gardens of eden in reaction systems. Theoretical Computer Science (2015), in press
4. Dudley, U.: Elementary Number Theory. W.H. Freeman and Company (1969)
5. Ehrenfeucht, A., Rozenberg, G.: Basic notions of reaction systems. In: Calude, C.S., Calude, E., Dinneen, M.J. (eds.) Developments in Language Theory, 8th International Conference, DLT 2004, Lecture Notes in Computer Science, vol. 3340, pp. 27–29. Springer (2005)
6. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. Fundamenta Informaticae 75, 263–280 (2007)
7. Formenti, E., Manzoni, L., Porreca, A.E.: Cycles and global attractors of reaction systems. In: Jürgensen, H., Karhumäki, J., Okhotin, A. (eds.) Descriptional Complexity of Formal Systems, 16th International Workshop, DCFS 2014, Lecture Notes in Computer Science, vol. 8614, pp. 114–125. Springer (2014)
8. Formenti, E., Manzoni, L., Porreca, A.E.: Fixed points and attractors of reaction systems. In: Beckmann, A., Csuhaj-Varjú, E., Meer, K. (eds.) Language, Life, Limits, 10th Conference on Computability in Europe, CiE 2014, Lecture Notes in Computer Science, vol. 8493, pp. 194–203. Springer (2014)
9. Formenti, E., Manzoni, L., Porreca, A.E.: On the complexity of occurrence and convergence problems in reaction systems. Natural Computing 14(1), 185–191 (2015)
10. Granas, A., Dugundji, J.: Fixed Point Theory. Springer Monographs on Mathematics, Springer (2003)
11. Manzoni, L., Poças, D., Porreca, A.E.: Simple reaction systems and their classification. International Journal of Foundations of Computer Science 25(4), 441–457 (2014)
12. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1993)
13. Salomaa, A.: Minimal and almost minimal reaction systems. Natural Computing 12(3), 369–376 (2013)
14. Tošić, P.T.: On the complexity of enumerating possible dynamics of sparsely connected boolean network automata with simple update rules. In: Fatès, N., Kari, J., Worsch, T. (eds.) Automata 2010 - 16th Intl. Workshop on CA and DCS, Nancy, France, June 14-16, 2010. pp. 125–144. Discrete Mathematics and Theoretical Computer Science Proceedings, DMTCS (2010)