Complexity of the dynamics of reaction systems☆

Alberto Dennunzio[a], Enrico Formenti[b], Luca Manzoni[a], Antonio E. Porreca[a,c]

[a]*Dipartimento di Informatica, Sistemistica e Comunicazione*
Università degli Studi di Milano-Bicocca
Viale Sarca 336/14, 20126 Milano, Italy
[b]*Université Côte d'Azur (UCA), CNRS, I3S, France*
[c]*Aix Marseille Université, Université de Toulon, CNRS, LIS, Marseille, France*

Abstract

Reaction systems are discrete dynamical systems inspired by bio-chemical processes, whose dynamical behaviour is expressed by set-theoretic operations on finite sets. Reaction systems thus provide a description of bio-chemical phenomena that complements the more traditional approaches, for instance those based on differential equations. A comprehensive list of decision problems about the dynamical behavior of reaction systems (such as cycles and fixed/periodic points, attractors, and reachability) is provided along with the corresponding computational complexity, which ranges from tractable problems to **PSPACE**-complete problems.

Keywords: reaction systems, computational complexity, natural computing, discrete dynamical systems

1. Introduction

Reaction systems (RS) are a computational model recently introduced by Ehrenfeucht and Rozenberg [15] which is inspired by chemical reactions. Roughly speaking, a RS is made of a finite set of entities (molecules of substances) and a finite set of rules (reactions). Entities are used as reactants, inhibitors, and products of a reaction. If, among a current set of entities (state), the reactants of a given reaction are available and there are no inhibitors, then that reaction takes place, *i.e.*, the products of the considered reaction are generated. The collection of the products of all enabled reactions give the new state of the system. In fact, this framework therefore captures the fundamental mechanisms of facilitation and inhibition on which biochemical reactions are based. Further, the functioning of the model undergoes two important principles, namely, the threshold assumption and the non-permanency assumption. The first one means that if a resource is available then there is always a sufficient amount of it for all the reactions which can take place. In other terms, reactions needing the same resource do not come into conflict. This principle essentially defines RS as a qualitative model (*i.e.*, only the presence or absence of a substance is measured). The latter assumption imposes that an entity will disappear from the current state of the system if no enabled reaction will have produced it.

Thus, RS are very different from the standard frameworks for biomodelling such as, for instance, ordinary differential equations and continuous stochastic process. However, it has been recently shown that they are able to capture the main quantitative characteristics of the ODE-based model for biochemical processes. In [1], for instance, they show how to build a RS reproducing the quantitative behavior of the ODE model for the molecular heat shock response. Another important aspect in the RS framework is that, differently than in the traditional models, one is able to follow the cause-effect relations between reactions, allowing for instance to understand what led a reaction to take place.

☆This paper collects, in a more extended/generalized version, both results presented at CiE 2014 conference [17] and the ones presented at DCFS 2014 conference [16]. It also contains a considerable number of further results.

Email addresses: `dennunzio@disco.unimib.it` (Alberto Dennunzio), `enrico.formenti@unice.fr` (Enrico Formenti), `luca.manzoni@disco.unimib.it` (Luca Manzoni), `antonio.porreca@lis-lab.fr` (Antonio E. Porreca)

As consequence of these facts and the capability of RS to capture the main mechanisms of biochemical reactions, the interest and use of RS to study practical problems concerning biological and/or chemical processes has grown since their introduction. Some biological uses were first detailed in [6]. Successively the study of the cell heat-shock response was carried on in [1]. Recently, RS has been used to simulate a complex real-world biological model [22], namely, the *ErbB* model described in [19]. Finally, a model of the gene regulatory network of the Yeast-Cell cycle was detailed in [3].

The study of the dynamical properties of the model at hand is a central question in the context of modelling. When considering RS as a model, this question might seem trivial at a first glance. Indeed, RS are finite systems and, as such, their dynamics is ultimately periodic. However, in practical applications this information is pointless and a more detailed analysis is expected. For example, in the context of genetic modelling, the number of attractors can be associated with cell differentiation [26], the period of attractors may represent the period of life process (respiration, circadian cycle, *etc.*), the size of the basin of attraction can be associated with the robustness in complex biological systems [21]. The analysis can be pushed even further considering limited resources [14, 11].

In [13], they considered some particular state as a death state (the empty set in this case) and computed the probability of a system reaching the death state. The study of dynamic causality relations was also explored in [4, 5]. Other studies focused on the complexity of deciding if any given RS exhibits a certain dynamical behaviour as, for instance, the appearance of a specific product during the evolution [25, 24].

This paper pursues the seminal results in [15, 25, 24, 18, 10, 8, 2] about the dynamical behavior of RS. The purpose is to cover a broad range of questions involving cycles, attractors and reachability, providing in this way a reference list for them. Several results are stated in a form which is as wide as possible (including constraints as "at least" and "at most") in order to answer the different variants/formulations of a same problem which arise in practical situations. Remark that similar problems are studied also in the context of other models used for applications, as, for instance, Boolean Automata Networks and Cellular Automata [27, 9, 7].

The first group of results concern the complexity of the decision problems about various forms of reachability under the action of a RS. We provide the complexity of establishing if

- a state T leads to a state U in at most k steps (**P**-complete, Theorem 4)
- a state T leads to a state U (**PSPACE**-complete, Theorem 18)
- a state T leads to a cycle of length at most ℓ after at most k steps (**P**-complete, Theorem 6) or after at least k steps (**PSPACE**-complete, Theorem 21)
- there exists a state T leading to a cycle of length at most ℓ after at least k steps (**NP**-complete, Theorem 11),
- a state T leads to a cycle of length at least ℓ after at least k steps (**PSPACE**-complete, Theorem 21) or if such a state exists (**PSPACE**-complete, Theorem 20),
- a state T is reachable from some other state in k steps (**NP**-complete, Theorem 10).
- a state T is periodic of period at most ℓ and reachable from some other ultimately periodic state having a pre-period of length at least k (**NP**-complete, Theorem 9)

As to problems involving cycles, we deal with the complexity of determining if

- a state T belongs to a cycle of length at most ℓ (**P**-complete, Theorem 5)
- there exists a cycle of length at most ℓ (**NP**-complete in both the versions with "for each ℓ" in the statement and with ℓ given as input, Theorems 7 and 8)
- there exists a cycle of length at least ℓ (**PSPACE**-complete, Theorem 19)
- the given RS consists of cycles possibly all of length 1 (**coNP**-complete, Theorems 12 and 13)

Concerning the attractors, we study the complexity of deciding if

- a state T belongs to a local attractor cycle of length at most ℓ with attraction basin of diameter at most d/at least d (**coNP**-complete/**NP**-complete, Theorems 14 and 15)
- there exists a local attractor cycle of length at most ℓ with attraction basin of diameter at least d/at most d (**NP**-complete/$\mathbf{\Sigma_2^P}$-complete, Theorems 16 and 17)
- a state T belongs to a global attractor cycle of length at most ℓ/at least ℓ (**PSPACE**-complete, Theorem 22)
- there exists a global attractor cycle of length at most ℓ/at least ℓ (**PSPACE**-complete, Theorem 22)

Some proofs of our results exploit the idea that RS can be used to evaluate Boolean formulae [15] and are based on reductions from problems over the latter. Other ones consist of reductions from different halting problems over deterministic Turing machines with bounded tape. Indeed, as described in Section 3, the computations of such machines can be simulated by suitable RS. Finally, direct reductions from the problems considered in this paper allow to prove the remaining results.

The paper is structured as follows. Section 2 provides the basic notions on RS and Section 3 illustrates the construction of a RS simulating a Turing machine with bounded tape. The results regarding reachability issues, cycles and attractors appear in the three Sections 4, 5, and 6, depending on the complexity degree of the involved problems. Indeed, this allows a certain uniformity of proof techniques and to factor some parts or ideas. Section 7 provides a recap of the results and discusses possible future developments.

2. Basic notions

This section provides all the basic concepts about RS and those notions about dynamical systems which will be considered in the paper. Notations are taken from [15].

Let S be a finite set and let its elements be called *entities*. Denote by 2^S the power set of S.

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

Definition 2. A *reaction system* $\mathcal{A}$ is a pair (S, A) where S is a finite set, called the *background set*, and $A \subseteq \mathrm{rac}(S)$. Every subset of the background set of $\mathcal{A}$ is said to be a *state*.

Remark that R and I are allowed to be empty as in the original definition of RS from [15]. Another option, not used here, is to assume the non emptiness of these two sets, as done later in [13, 14], for instance. We want to stress that the essence of our results does not change at all in the non emptiness setting, while our choice allows a more elegant formulation of the results.

For any state $T \subseteq S$ and any reaction $a \in \mathrm{rac}(S)$, a is said to be *enabled* in T if $R_a \subseteq T$ and $I_a \cap T = \varnothing$. The *result function* $\mathrm{res}_a : 2^S \to 2^S$ of a is defined as

$$\forall T \subseteq S, \quad \mathrm{res}_a(T) = \begin{cases} P_a & \text{if } a \text{ is enabled in } T \\ \varnothing & \text{otherwise.} \end{cases}$$

The definition of res_a naturally extends to sets of reactions. Indeed, given a set of reaction $B \subseteq \mathrm{rac}(S)$, define $\mathrm{res}_B : 2^S \to 2^S$ as $\mathrm{res}_B(T) = \bigcup_{a \in B} \mathrm{res}_a(T)$ for every $T \subseteq S$. The result function $\mathrm{res}_{\mathcal{A}}$ of a RS $\mathcal{A} = (S, A)$ is res_A. In this way, the discrete dynamical system with the set of states 2^S and the next state map $\mathrm{res}_{\mathcal{A}}$ is associated with the RS $\mathcal{A} = (S, A)$.

The following is an example of a simple RS which is able to simulate a NAND gate.

Example 3 (NAND gate). To implement a NAND gate using a RS we use $S = \{0_a, 1_a, 0_b, 1_b, 0_{\text{out}}, 1_{\text{out}}\}$ as background set. The first four elements represent the two inputs (denoted by the subscripts a and b), the last two, on the other hand, denote the two possible outputs. The reactions used to model a NAND gate are the following: $(\{0_a, 0_b\}, \varnothing, \{1_{\text{out}}\})$, $(\{0_a, 1_b\}, \varnothing, \{1_{\text{out}}\})$, $(\{1_a, 0_b\}, \varnothing, \{1_{\text{out}}\})$, and $(\{1_a, 1_b\}, \varnothing, \{0_{\text{out}}\})$. Similarly to NAND gates, others gates can be simulated and it is possible to build circuits with gates of limited fan-in using only a number of entities and reactions that is linear in the size of the modelled circuit.

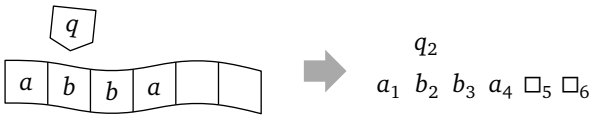

Figure 1: Encoding of a configurations of M as state $T = \{a_1, b_2, b_3, a_4, \square_5, \square_6, q_2\}$ of $\mathcal{M}$ where each tape symbol is subscripted with its own position, while the current state of M is subscripted with the position of the head. The square $\square$ represents the blank symbol.

We now proceed by recalling the necessary definitions of the dynamical properties investigated in this work.

Let $\mathcal{A} = (S, A)$ be a RS. For any given state $T \subseteq S$, the *dynamical evolution* or *dynamics* of $\mathcal{A}$ starting from T is the sequence $(T, \mathrm{res}_{\mathcal{A}}(T), \mathrm{res}^2_{\mathcal{A}}(T), \ldots)$ of states visited by the system starting from T, *i.e.*, the sequence in which for every $i \in \mathbb{N}$ the i-th element is $\mathrm{res}^i_{\mathcal{A}}(T)$. We say that T *leads to a state* U *in* t *steps* if $U = \mathrm{res}^t_{\mathcal{A}}(T)$. In that case, T is said to be a t-*ancestor* of U. We simply say that T leads to U if T leads to a state U in t steps, for some $t \in \mathbb{N}$. Since 2^S is finite, any of the above sequences is ultimately periodic, *i.e.*, there exist two integers $h \in \mathbb{N}$ and $k > 0$ such that $\mathrm{res}^{h+k}_{\mathcal{A}}(T) = \mathrm{res}^h_{\mathcal{A}}(T)$. Notice that, since $\mathrm{res}_{\mathcal{A}}$ is a function, this also implies that for all $i \geq 0$, $\mathrm{res}^{h+k+i}_{\mathcal{A}}(T) = \mathrm{res}^{h+i}_{\mathcal{A}}(T)$. The smallest values of h and k for which this condition holds are called *preperiod* and *period* of T, respectively. If $h = 0$, T is a *periodic point* and the set $\{T, \mathrm{res}_{\mathcal{A}}(T), \ldots, \mathrm{res}^{k-1}_{\mathcal{A}}(T)\}$ is said to be a *cycle* (of *length* k). We say that a periodic point T is a *fixed point* if $k = 1$.

The notion of attractor is a central concept in the study of dynamical systems. Recall that an *invariant set* for $\mathcal{A}$ is a set of states $\mathcal{U}$ with $\mathrm{res}_{\mathcal{A}}(\mathcal{U}) = \{\mathrm{res}_{\mathcal{A}}(U) : U \in \mathcal{U}\} = \mathcal{U}$. Remark that each invariant set of any RS consists of cycles. A *local attractor* in a RS is an invariant set $\mathcal{U}$ such that there exists $T \notin \mathcal{U}$ with $\mathrm{res}_{\mathcal{A}}(T) \in \mathcal{U}$. Intuitively, a local attractor is a set of states $\mathcal{U}$ from which the dynamics never escapes and such that there exists at least one external state whose dynamics ends up in $\mathcal{U}$. For example, if $\{T\}$ is an invariant set (i.e., $\mathrm{res}_{\mathcal{A}}(T) = T$) and there exists $U \neq T$ such that $\mathrm{res}_{\mathcal{A}}(U) = T$, then $\{T\}$ is an attractor. A *global attractor* for an RS $\mathcal{A}$ is an invariant set of states $\mathcal{U}$ such that for all $T \in 2^S$ there exists $t \in \mathbb{N}$ such that $\mathrm{res}^t_{\mathcal{A}}(T) \in \mathcal{U}$. A local/global attractor $\mathcal{U}$ is a *local/global fixed-point attractor* if $\mathcal{U} = \{T\}$ and hence, necessarily, T is a fixed point. Similarly, we call a local/global attractor $\mathcal{U}$ a *local/global attractor cycle* if all the states in $\mathcal{U}$ belong to the same cycle. The *attraction basin* of a local/global attractor $\mathcal{U}$ is the set of all states whose dynamics ends up in $\mathcal{U}$. The maximum value of all the preperiods of such states is called the *diameter* of $\mathcal{U}$.

3. Simulating Turing machines with bounded tape

In this section we illustrate the simulation of Turing machines with bounded tape performed by RSs. For an introduction, basic results, and notions on Turing machines, we refer the reader to [20].

Let M be any single-tape deterministic Turing machine with tape of length m, set of states Q, tape alphabet Σ, and transition function $\delta : Q \times \Sigma \to Q \times \Sigma \times \{-1, 0, +1\}$. The computations of M can be simulated by the reaction system $\mathcal{M} = (S, A)$ defined as follows.

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

$$S = \{a_i : a \in \Sigma, 1 \leq i \leq m\} \cup \{q_i : q \in Q, 1 \leq i \leq m+1\}$$

that is, S consists of all symbols of the alphabet and all states of M, each of them indexed by every possible tape position (with the additional position $m+1$ for the states, too).

The generic configuration where M is in state $q \in Q$, its tape head is located on cell i, and its tape consists of the string $x = x_1 \cdots x_m$, is encoded as the following state

$$T = \{(x_j)_j : 1 \leq j \leq m\} \cup \{q_i\}$$

In other terms, T contains each symbol of x indexed by its position on the tape and the entity q_i (state entity) storing both the current state and the head position of M (see Fig. 1).

Reactions. We are going to illustrate the elements of the set A. Each transition $\delta(q,a) = (q',a',d)$ of M, with $q,q' \in Q$, $a,a' \in \Sigma$, and $d \in \{-1,0,1\}$, gives rise to the following set of reactions:

$$(\{q_i, a_i\}, \emptyset, \{q'_{i+d}, a'_i\}) \qquad \text{for } 1 \le i \le m; \tag{1}$$

If the head of M is located on the cell i of the tape, then the i-th reaction simulates the action of δ, by producing the entity encoding both the new state and the new head position of M, as well as the symbol written on the tape in position i[1].

The following reactions assure that the symbols on the tape where the head is not located stay unchanged:

$$(\{a_i\}, \{q_i : q \in Q\}, \{a_i\}) \qquad \text{for } 1 \le i \le m; \tag{2}$$

Indeed, for every i, the i-th reaction of type (2) is inhibited by the presence of any state with the same subscript i, to allow that only a reaction of type (1) takes place when the head of M is over the tape cell i.

Hence, the reaction system $\mathcal{M}$ maps the state encoding any configuration of M on input x to the state corresponding to the next configuration of M. Notice that some states of $\mathcal{M}$ are "malformed", as they contain zero or more than one state entity, or zero or more than one symbol making reference to a position. In this case, the behaviour of $\mathcal{M}$ is unrelated with the one of M.

We also stress that the reaction system $\mathcal{M}$ can be built from the description of M and the unary encoding of m in logarithmic space, since its reactions can be output by iteration over all tape positions $1 \le i \le m$ in binary notation: this number requires only $O(\log_2(m))$ bits to be represented, and all other quantities involved, like the size of the set of states, the alphabet, and the table representing the transition function, are constant in size.

4. Problems Solvable in Polynomial Time

Some interesting problems related to the dynamics of reaction systems involve their forward dynamics and for a limited amount of time (polynomial). These problems are usually efficiently solvable, although often they are among the hardest problems in **P**. The canonical example is "local" reachability.

Theorem 4. *Given RS $\mathcal{A} = (S,A)$, two states $T,U \subseteq S$, and a unary integer k, it is* **P***-complete to decide if T leads to U in at most k steps.*

Proof. We show this result by reduction from the **P**-complete bounded halting problem [23]

> Given a deterministic Turing machine M, a string x and a unary integer k, does M accept x within k steps?

Without loss of generality, we assume that Turing machine M has a unique accepting configuration, with the tape head in the accepting state located on the first tape cell, and with an empty tape (i.e., each tape cell is blank). We can build the reaction system $\mathcal{M}$ simulating M with $k+1$ cells of tape (the maximum amount of tape exploitable in k steps) as described in Section 3. Let T be the encoding of the initial configuration of M on input x, and let U be the encoding of the unique accepting configuration. Then, it holds that state T leads to U within k steps (the property to be decided) if and only if M accepts x within k steps.

Since the mapping $(M,x,k) \mapsto (\mathcal{M},T,U,k)$ can be carried out in logarithmic space, the problem is **P**-hard. It also belongs to **P**, since the property can be checked by simulating k steps of the reaction system. □

The problem of detecting the presence of "short" cycles is also **P**-complete.

Theorem 5. *Given a RS $\mathcal{A} = (S,A)$, a state $T \subseteq S$, and a unary integer $\ell \ge 1$ as input, it is* **P***-complete to decide if T is a periodic point of period at most ℓ.*

[1] Notice that, for simplicity, the background set S and the reactions of type (1) refer to the entities q_{m+1} for all $q \in Q$; however, in this section the tape head, by hypothesis, does never reach the $(m+1)$-th tape cell. The entities q_{m+1} will play a different role in Section 6.

Proof. We proceed by reduction from the bounded halting problem. Let M be a Turing machine having a unique accepting configuration in which the accepting state is r, the head is over the tape cell 1, and the tape is empty. Consider the reaction system simulating M from Section 3 and change it so that r_1 is added to the inhibitors of all reactions of types (1) and (2):

$$(\{q_i, a_i\}, \{r_1\}, \{q'_{i+d}, a'_i\}) \qquad \text{for } 1 \leq i \leq m$$
$$(\{a_i\}, \{q_i : q \in Q\} \cup \{r_1\}, \{a_i\}) \qquad \text{for } 1 \leq i \leq m$$

This means that the original reactions are now disabled if M reaches its accepting state. The reaction system $\mathcal{M}$ we are building is also equipped of an additional reaction, which will be instead enabled if M accepts. Let U be the state of $\mathcal{M}$ encoding the initial configuration of M on a given input string x. Such a reaction is:

$$(\{r_1\}, \varnothing, U)$$

Then, the state T, encoding the final configuration of M, belongs to a cycle of length at most ℓ if and only if M accepts x within $\ell - 1$ steps.

Since the mapping $(M, x, k = \ell - 1) \mapsto (\mathcal{M}, T, \ell)$ can be computed in logarithmic space, the problem under consideration is **P**-hard due to the complexity of the bounded halting problem. The **P** upper bound also holds. Indeed, establishing whether any state T is periodic of period at most ℓ can be checked by performing a simulation of the reaction system with initial state T and testing whether the state T itself is reached within ℓ steps. □

Even when we search for short cycles with a short preperiod we obtain **P**-completeness.

Theorem 6. *Given a RS $\mathcal{A} = (S, A)$, a state $T \subseteq S$, and two unary integers $\ell \geq 1$ and k as input, it is* **P***-hard to decide if, when $\mathcal{A}$ starts at T, it reaches a cycle of period at most ℓ after a preperiod of length at most k.*

Proof. This problem is **P**-hard by trivial reduction, since it is a generalisation of the problem of Theorem 5, the latter having the fixed value $k = 0$ for the preperiod. A polynomial time algorithm for this problem computes a sequence of $k + \ell$ steps of $\mathcal{A}$ starting from any initial state, and checks that a cycle is reached within the first k steps. □

5. Problems in the Polynomial Hierarchy

A jump in complexity happens when, instead of checking a local dynamical behaviour that only involves the forward dynamics, we search for the existence of a local behaviour somewhere in the system or the property involves the backward dynamics, i.e., the preimages of a state.

While deciding if a state is part of a short cycle can be done in deterministic polynomial time, establishing the existence of such a state is **NP**-complete. We will see that this holds even when the length of the cycle is fixed a priori.

In the sequel, for any Boolean formula φ in conjunctive or disjunctive normal form and with clauses $\varphi_1, \ldots, \varphi_m$, for each clause φ_i we denote by $\text{pos}(\varphi_j)$ (resp., $\text{neg}(\varphi_i)$) the set of variables appearing in φ_i as positive (resp., negative) literals.

Theorem 7. *For each integer $\ell \geq 1$, it is* **NP***-complete, on input a RS $\mathcal{A}$, to decide if $\mathcal{A}$ has a periodic point of period at most ℓ.*

Proof. We prove the **NP**-hardness of this problem by reduction from the Boolean satisfiability problem. For any Boolean formula $\varphi = \varphi_1 \wedge \cdots \wedge \varphi_m$ in conjunctive normal form over the set of variables $X = \{x_1, \ldots, x_n\}$ we build a reaction system $\mathcal{A} = (S, A)$ with set of entities $S = X \cup Z$, where $Z = \{\spadesuit_0, \ldots, \spadesuit_\ell\}$. For any state $T \subseteq S$, the set $T \cap X$ encodes an assignment of φ in which x_i has true value iff $x_i \in T \cap X$. First of all, the set A contains the following group of reactions

$$(\text{neg}(\varphi_j), \text{pos}(\varphi_j) \cup Z, \{\spadesuit_0\}) \qquad \text{for } 1 \leq j \leq m; \tag{3}$$

Thus, when the system is in a state $T \subseteq X$, this group of reactions produces the entity $\spadesuit_0$ (*i.e.*, at least one among them is enabled in T) if there exists a clause φ_j not satisfied by the assignment encoded by T (hence φ itself is not satisfied). Notice that these reactions are disabled in any state T containing some $\spadesuit_t$.

The set A also includes the following reactions

$$(\{x_i\}, Z, \{x_i\}) \qquad \text{for } 1 \leq i \leq n; \tag{4}$$

which preserve the value of the assignment of φ encoded by a current state T in the next state when $T \subseteq X$.

In this way, the reactions of type (3) and (4) ensure that $\mathrm{res}_{\mathcal{A}}(T) = T$ for any $T \subseteq X$ encoding an assignment satisfying φ, and $\mathrm{res}_{\mathcal{A}}(T) = T \cup \{\spadesuit_0\}$ if $T \subseteq X$ but T codifies an assignment which does not satisfy φ.

The following last group of reactions constituting A creates a cycle of length $\ell + 1$ formed exactly by the entities of Z

$$(\{\spadesuit_t\}, \{\spadesuit_s : 0 \leq s < t\}, \{\spadesuit_{(t+1) \bmod (\ell+1)}\}) \qquad \text{for } 0 \leq t \leq \ell; \tag{5}$$

Indeed, it trivially holds that $\mathrm{res}_{\mathcal{A}}(\{\spadesuit_t\}) = \{\spadesuit_{(t+1) \bmod (\ell+1)}\}$. Moreover, if the system is in a state T with $T \cap Z \neq \emptyset$, then its next state is $\{\spadesuit_{t+1 \bmod \ell+1}\}$, where $t = \min\{s : \spadesuit_s \in T\}$. Hence, the reaction system $\mathcal{A}$ has both a cycle of length $\ell + 1$ (namely $\{\spadesuit_0\} \to \cdots \to \{\spadesuit_\ell\} \to \{\spadesuit_0\}$) and at least a fixed point, that is a cycle of length at most ℓ, if and only if the formula φ admits a satisfying assignment.

Since the mapping $\varphi \mapsto \mathcal{A}$ can be computed in polynomial time, the problem is **NP**-hard. The membership in **NP** follows by the existence of a non-deterministic algorithm guessing an initial state T and checking in polynomial time whether T is again reached within ℓ steps. □

The difficulty of the problem does not increase when the length of the cycle is given in input.

Theorem 8. *Given a RS $\mathcal{A}$ and a unary integer $\ell \geq 1$ as input, it is* **NP***-complete to decide if $\mathcal{A}$ has a periodic point of period at most ℓ.*

Proof. The problem is **NP**-hard by reduction from any of the problems of Theorem 7, which are **NP**-hard for every fixed $\ell \geq 1$. The same algorithm mentioned at the end of the proof of Theorem 7 proves the membership in **NP**, since the extra input ℓ is given in unary notation. □

Checking if a state is part of a short cycle can be performed in deterministic polynomial time. However, checking if there exists another state reaching it after a certain amount of steps makes the problem **NP**-complete.

Theorem 9. *Given a RS $\mathcal{A} = (S, A)$, a state $T \subseteq S$, and two unary integers $\ell \geq 1$ and k as input, it is* **NP***-complete to decide if T is a periodic point of period at most ℓ and there exists a state $U \subseteq S$ leading to T with preperiod at least k.*

Proof. We proceed by reduction from the Boolean satisfiability problem. For any Boolean formula $\varphi = \varphi_1 \wedge \cdots \wedge \varphi_m$ in conjunctive normal form over the set of variables $X = \{x_1, \ldots, x_n\}$ we build a reaction system $\mathcal{A} = (S, A)$ with set of entities $S = X \cup C$, where $C = \{\varphi_1, \ldots, \varphi_m\}$. As in the proof of Theorem 7, for any state $T \subseteq S$, the set $T \cap X$ encodes a truth assignment of φ. The set A contains the following two groups of reactions

$$\big(\{x_i\}, \emptyset, \{\varphi_j : x_i \in \mathrm{pos}(\varphi_j)\}\big) \qquad \text{for } 1 \leq i \leq n$$
$$\big(\emptyset, \{x_i\}, \{\varphi_j : x_i \in \mathrm{neg}(\varphi_j)\}\big) \qquad \text{for } 1 \leq i \leq n$$

By evaluating φ, these reactions map a state T to the set of clauses satisfied by the assignment encoded by $T \cap X$, ignoring any element of $T \cap C$.

The further reaction in A

$$(C, X, C)$$

ensures that the state consisting exactly of all clauses is a fixed point, since these latter are preserved when they appear all together without any variable from X.

Then, the resulting reaction system $\mathcal{A}$ has a fixed point $T = C$, that is a cycle of length at most $\ell = 1$, with a state U leading to T and having preperiod at least $k = 1$, if and only if $U \cap X$ encodes a assignment which satisfies φ.

Since the mapping $\varphi \mapsto \mathcal{A}$ is polynomial-time computable, the problem is thus **NP**-hard. A polynomial-time nondeterministic algorithm for the problem under consideration exists. It guesses a state U leading to T in exactly k steps, verifies if T belongs to a cycle of length at most ℓ, and checks that the set $\{U, \mathrm{res}_{\mathcal{A}}(U), \ldots, T\}$ and the cycle only have T as a common state. Therefore, the membership in **NP** follows. □

More in general, the problem of exploring the dynamics backwards is **NP**-complete.

Theorem 10. *Given a RS $\mathcal{A} = (S, A)$, a state $T \subseteq S$, and a unary integer k as input, it is* **NP***-complete to decide if T has a k-ancestor.*

Proof. Perform the same reduction as in the proof of Theorem 9, except adding the entity $\heartsuit$ to S and changing the reaction (C, X, C) to $(C, X, C \cup \{\heartsuit\})$. Now the obtained reaction system has the fixed point $C \cup \{\heartsuit\}$ rather than C, and C leads to $C \cup \{\heartsuit\}$ in one step. Then, state C has T as preimage, *i.e.*, T is a 1-ancestor of C, if and only if $T \cap X$ satisfies φ for some set $T \subseteq S$. Thus, the **NP**-hardness follows. The problem is also in **NP**. Indeed, once a state $U \subseteq S$ is guessed, it can be checked in polynomial time whether $\mathrm{res}^k_{\mathcal{A}}(U) = T$. □

Establishing if there exists a state that leads to a short cycle after a given number of steps, i.e., a state far enough from the cycle, turns out be another **NP**-complete problem.

Theorem 11. *Given a RS $\mathcal{A} = (S, A)$ and two unary integers $\ell \geq 1$ and k as input, it is* **NP***-complete to decide if there exists a state $T \subseteq S$ such that, when $\mathcal{A}$ starts in T, it reaches a cycle of length at most ℓ after a preperiod of length at least k.*

Proof. The problem from Theorem 8 is a special case of the one under consideration and is obtained when $k = 0$. Hence, by reduction from the former problem, the one we are dealing with is **NP**-hard too. Its membership in **NP** follows by the existence of an algorithm similar to that at the end of the proof of Theorem 9, except that the state T must also be guessed, rather than being given as input. □

Checking if every state of a reaction system has exactly one preimage is the same as deciding if the result function is a bijection. This problem can be shown to be **coNP**-complete by a variation of a proof by Ehrenfeucht and Rozenberg [15].

Theorem 12. *Given a RS $\mathcal{A} = (S, A)$, it is* **coNP***-complete to decide if $\mathrm{res}_{\mathcal{A}}$ is a bijection, that is, if every state of $\mathcal{A}$ is periodic.*

Proof. We prove **coNP**-hardness by reduction from the tautology problem for Boolean formulae in disjunctive normal form [23]. Given a formula $\varphi = \varphi_1 \vee \ldots \vee \varphi_m$ over the variables $X = \{x_1, \ldots, x_n\}$, we build the reaction system $\mathcal{A} = (S, A)$ where $S = X \cup \{\heartsuit\}$ and the reactions are

$$(\mathrm{pos}(\varphi_j) \cup \{\heartsuit\}, \mathrm{neg}(\varphi_j), \{\heartsuit\}) \qquad \text{for } 1 \leq j \leq m; \tag{6}$$

$$(\{x_i\}, \varnothing, \{x_i\}) \qquad \text{for } 1 \leq i \leq n; \tag{7}$$

As usual, for any state T, $T \cap X$ encodes a truth assignment of φ. In this way, if $\heartsuit$ occurs in the current state, the reactions of type (6) evaluate each conjunctive clause (which is satisfied iff all positive variables are set to true and all negative ones to false) and preserve the element $\heartsuit$ if the clause (and thus the whole formula) is satisfied. The reactions of type (7) preserve all variables in the current state.

Hence, the behaviour of $\mathcal{A}$ is as follows. If the current state $T \subseteq X$, only reactions of type (7) are enabled, and thus $\mathrm{res}_{\mathcal{A}}(T) = T$. On the other hand, if $\heartsuit \in T$, we have $\mathrm{res}_{\mathcal{A}}(T \cup \{\heartsuit\}) = T \cup \{\heartsuit\}$ if at least one reaction of type (6) is enabled, that is, if φ is satisfied by the assignment encoded by T; otherwise, it holds that $\mathrm{res}_{\mathcal{A}}(T \cup \{\heartsuit\}) = T$. As a consequence, each state of $\mathcal{A}$ is a fixed point, and thus a cycle, iff φ is a tautology; otherwise, there exists a state with two preimages, namely, a state $T \subseteq X$ encoding a non-satisfying assignment and having both T and $T \cup \{\heartsuit\}$ as preimages.

The considered problem belongs to **coNP**, since there exists a non-deterministic algorithm guessing two distinct states $T, U \subseteq S$ and checking whether $\mathrm{res}_{\mathcal{A}}(T) = \mathrm{res}_{\mathcal{A}}(U)$ in polynomial time. □

The previous construction actually proves another similar result:

Theorem 13. *Given a RS $\mathcal{A} = (S, A)$, it is* **coNP***-complete to decide if* $\mathrm{res}_{\mathcal{A}}$ *is the identity function, that is, if each state is a fixed point.* □

Establishing whether a local attractor cycle has a small attraction basin is **coNP**-complete.

Theorem 14. *Given a RS $\mathcal{A} = (S, A)$, a state $T \subseteq S$, and two unary integers $\ell \geq 1$ and d as input, it is* **coNP***-complete to decide if T belongs to a local attractor cycle of length at most ℓ with attraction basin of diameter at most d.*

Proof. We use a reduction from tautology similar to that introduced in the proof of Theorem 12. For any formula $\varphi = \varphi_1 \vee \ldots \vee \varphi_m$ over the variables $X = \{x_1, \ldots, x_n\}$, consider the reaction system from the aforementioned reduction. We add an extra entity ♠ to its background set and slightly change the original reactions as follows:

$$(\mathrm{pos}(\varphi_j) \cup \{\heartsuit\}, \mathrm{neg}(\varphi_j) \cup \{\spadesuit\}, \{\heartsuit\}) \qquad \text{for } 1 \leq j \leq m; \tag{8}$$

$$(\{x_i, \heartsuit\}, \{\spadesuit\}, \{x_i\}) \qquad \text{for } 1 \leq i \leq n; \tag{9}$$

Furthermore, we add the two extra reactions

$$(\varnothing, \{\heartsuit\}, \{\spadesuit\})$$
$$(\{\spadesuit\}, \varnothing, \{\spadesuit\})$$

producing ♠ whenever ♡ is missing, and preserving ♠, respectively.

The resulting reaction system $\mathcal{A}$ behaves as follows. Let $U \subseteq X$. By evaluating φ on the basis the assignment encoded by U, it follows that $\mathrm{res}_{\mathcal{A}}(U \cup \{\heartsuit\}) = U \cup \{\heartsuit\}$ if U encodes an assignment satisfying φ, and $\mathrm{res}_{\mathcal{A}}(U \cup \{\heartsuit\}) = U$ otherwise. Furthermore, either when ♠ belongs to the current state (which is thus either $U \cup \{\spadesuit\}$ or $U \cup \{\spadesuit, \heartsuit\}$), or when neither ♠ nor ♡ do (*i.e.*, the current state is U), it holds that $\mathrm{res}_{\mathcal{A}}(U) = \mathrm{res}_{\mathcal{A}}(U \cup \{\spadesuit\}) = \mathrm{res}_{\mathcal{A}}(U \cup \{\spadesuit, \heartsuit\}) = \{\spadesuit\}$. Hence, state $\{\spadesuit\}$ has a 2-ancestor if and only if the formula has a *non*-satisfying assignment U. In that case, we have $\mathrm{res}^2_{\mathcal{A}}(U \cup \{\heartsuit\}) = \mathrm{res}_{\mathcal{A}}(U) = \{\spadesuit\}$. By letting $T = \{\spadesuit\}$, $\ell = 1$, and $d = 1$, and since the mapping $\varphi \mapsto (\mathcal{A}, T, \ell, d)$ is polynomial-time computable, we obtain the **coNP**-hardness of the problem.

The problem is in **coNP** since once guessed a state U it is possible to check in polynomial time if it falsifies the required property. □

On the other hand, deciding if a small attractor cycle has a large attraction basin is an **NP**-complete problem.

Theorem 15. *Given a RS $\mathcal{A} = (S, A)$, a state $T \subseteq S$, and two unary integers $\ell \geq 1$ and d as input, it is* **NP***-complete to decide if T belongs to a local attractor cycle of length at most ℓ with attraction basin of diameter at least d.*

Proof. The state $\{\spadesuit\}$ in the proof of Theorem 14 has an attraction basin of diameter at least 2 iff the Boolean formula φ is *not* a tautology, and the latter is the statement of a **NP**-complete problem (since its negation is a **coNP**-complete one). Indeed, the same algorithm as in the proof of Theorem 14, but with reversed acceptance and rejection, proves the membership in **NP**. □

Establishing the existence of a small attractor cycle with a large attraction basin remains **NP**-complete.

Theorem 16. *Given a RS $\mathcal{A} = (S, A)$ and two unary integers $\ell \geq 1$ and d as input, it is* **NP***-complete to decide if there exists a local attractor cycle of length at most ℓ with attraction basin of diameter at least d.*

Proof. Consider the reactions system $\mathcal{A}$ from the proof of Theorem 15; all of its states are either fixed points or eventually reach state $\{\spadesuit\}$. Hence, $\{\spadesuit\}$ is the only state with attraction basin of diameter at least 1. Hence, asking whether there exists a fixed point (i.e., a cycle of length $\ell = 1$) with an attraction basin of diameter at least $d = 2$ is the same as asking whether $T = \{\spadesuit\}$ has this property, which is **NP**-complete. The problem under consideration is in **NP**, since a state T can be guessed before the same algorithm in the proof of Theorem 15 is used.

Differently from the previous case, the problem of establishing the existence of a small attractor cycle with a small attraction basin is higher in the polynomial hierarchy.

Theorem 17. *Given a RS $\mathcal{A} = (S, A)$ and two unary integers $\ell \geq 1$ and d as input, it is Σ_2^{P}-complete to decide if there exists a local attractor cycle of length at most ℓ with attraction basin of diameter at most d.*

Proof. First of all, we show the membership in Σ_2^{P} of the problem. There exists an alternating Turing machine able to guess a state $T \subseteq S$ and then, by iterating the $\mathrm{res}_{\mathcal{A}}$ function, check that it belongs to a cycle of length at most ℓ. Subsequently, the machine also guesses a state $U \subseteq S$ and checks if it reaches that cycle in $d+1$ steps (this would prove that the attraction basin has diameter larger than d; if no such guess is possible, then the attraction basin has diameter at most d). If this is the case, the machine rejects, while it accepts, otherwise; the overall result of the machine is acceptance if and only if the answer to the considered problem is true. Therefore, the membership in Σ_2^{P} follows.

We now prove the Σ_2^{P}-hardness of the problem by reduction from $\exists\forall$SAT, i.e., the problem of deciding if $\exists X \forall Y \varphi(X, Y)$, where φ is a Boolean formula in disjunctive normal form with conjunctive clauses $\varphi_1, \ldots, \varphi_k$ over the variables $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$.

For any of such formulae φ, define the reaction system $\mathcal{A} = (S, A)$ where $S = X \cup Y \cup \{\heartsuit, \spadesuit\}$ and A consists of the two group of reactions

$$(\mathrm{pos}(\varphi_j), \mathrm{neg}(\varphi_j), \{\heartsuit\}) \qquad \text{for each clause } \varphi_j; \tag{10}$$

$$(\{x_i\}, \varnothing, \{x_i\}) \qquad \text{for each variable } x_i \in X; \tag{11}$$

which depend on the Boolean formula, and the following further reactions

$$(\varnothing, \varnothing, \{\spadesuit\}) \tag{12}$$

$$(\{\heartsuit\}, \varnothing, \{\heartsuit\}) \tag{13}$$

$$(\{\spadesuit\}, \varnothing, \{\heartsuit\}) \tag{14}$$

We are going to describe the behaviour of $\mathcal{A}$. First of all, consider a state of the form $X_1 \cup Y_1$ for some $X_1 \subseteq X$ and $Y_1 \subseteq Y$. As usual, such a state represents an assignment of φ, where the variables appearing in $X_1 \cup Y_1$ have true value, and the missing ones have false value. A reaction of type (10) is then enabled if and only if the corresponding conjunctive clause φ_j is satisfied. Furthermore, a reaction of type (11) is enabled for each element of X_1, while the reaction (12) is always enabled. As a consequence, we have $\mathrm{res}_{\mathcal{A}}(X_1 \cup Y_1) = X_1 \cup \{\heartsuit, \spadesuit\}$ if at least one clause (and thus the disjunction φ) is satisfied by the assignment represented by $X_1 \cup Y_1$; otherwise, we have $\mathrm{res}_{\mathcal{A}}(X_1 \cup Y_1) = X_1 \cup \{\spadesuit\}$.

Furthermore, we have

$$\mathrm{res}_{\mathcal{A}}(X_1 \cup Y_1 \cup \{\heartsuit\}) = \mathrm{res}_{\mathcal{A}}(X_1 \cup Y_1 \cup \{\spadesuit\}) = \mathrm{res}_{\mathcal{A}}(X_1 \cup Y_1 \cup \{\spadesuit, \heartsuit\}) = X_1 \cup \{\spadesuit, \heartsuit\}$$

irrespective of whether the assignment encoded by $X_1 \cup Y_1$ satisfies φ.

This means that all $2^{|X|}$ states of the form $X_1 \cup \{\spadesuit, \heartsuit\}$ for any $X_1 \subseteq X$ are fixed points and $\mathcal{A}$ admits no further fixed points or cycles. Moreover, each fixed point $X_1 \cup \{\spadesuit, \heartsuit\}$ attracts in one step all states of the forms $X_1 \cup Y_1 \cup \{\heartsuit\}$, $X_1 \cup Y_1 \cup \{\spadesuit\}$, and $X_1 \cup Y_1 \cup \{\spadesuit, \heartsuit\}$ for any $Y_1 \subseteq Y$. On the other hand, the states of the form $X_1 \cup Y_1$ reach $X_1 \cup \{\spadesuit, \heartsuit\}$ in one step iff $X_1 \cup Y_1$ satisfies φ, and in two steps, otherwise.

In conclusion, there exists a fixed point (which is a cycle of length at most $\ell = 1$), necessarily of the form $X_1 \cup \{\spadesuit, \heartsuit\}$ for some $X_1 \subseteq X$, with an attraction basin of diameter at most $d = 1$ if and only if $X_1 \cup Y_1$ satisfies φ for all $Y_1 \subseteq Y$. This is equivalent to $\exists X \forall Y \varphi(X, Y)$ being true. Since the mapping $\varphi \mapsto (\mathcal{A}, \ell, d)$ can be computed in polynomial time, the problem is Σ_2^{P}-hard. □

6. Problems Solvable in Polynomial Space

Removing the restrictions on the length of cycles or preperiods leads to properties involving a potentially exponential number of states, and algorithms checking these properties might need to explore all of them. Such

algorithms, however, can be designed to use a polynomial amount of space, since every state contains only a polynomial amount of entities.

The canonical **PSPACE**-complete problem for reactions systems is reachability.

Theorem 18. *Given RS $\mathcal{A} = (S, A)$ and two states $T, U \subseteq S$, it is* **PSPACE***-complete to decide if T leads to U.*

Proof. We prove the hardness of the problem by reduction from the following **PSPACE**-complete problem [23]:

> Given a deterministic Turing machine M, a string x and a unary integer m, does M accept x without using more than m tape cells?

As in the proof of Theorem 4, for any (M, x, m) we build a reaction system $\mathcal{A}$ by exploiting the construction described in Section 3 with $k = m$. By encoding again the initial configuration of M on input x as a state T of $\mathcal{A}$ and the unique accepting configuration of M as state U, we obtain the desired reduction. Indeed, if M accepts x without exceeding m tape cells, then T leads to U. Furthermore, this never happens if M rejects, or fails to halt, or moves the tape head on cell $m+1$; remark that in the latter case, T does not lead to U since the reactions of type (1) are not defined for $i = m+1$, and the element q_{m+1} disappears.

The problem is in **PSPACE** because there exists an algorithm able to check in polynomial space whether T leads to U (such an algorithm rejects if this does not happen within $2^{|S|}$ steps, that is the number of configurations of $\mathcal{A}$). □

To prove the **PSPACE**-hardness of other problems, in this section we are going to introduce a variant of the polynomial space simulation of the Turing machine from Section 3 used in the proof of Theorem 18. Since most of the statements in the next theorems involve a quantification across *every* state of the reaction system, the dynamics of all the possible states will have to be considered and governed. To this extent, for any Turing machine M with state set Q where $f \in Q$ is the final accepting state, tape alphabet Σ, transition function δ, any input x, and any space bound m given in unary, the new construction of the reaction system and the encoding of the configurations of M into related states is based on what follows:

1. Let T be the state of the reaction system encoding the initial configuration of M. As in the proof of Theorem 18, M accepts by ending up in a specific accepting configuration with the tape head on the first tape cell and encoded by the state U in the reaction system which will be now a fixed point; otherwise, M rejects by diverging.
2. All states of the reaction system encoding a configuration of M either lead to the fixed point U (which corresponds to the accepting configuration reached by M) or enter a cycle. We stress that this set of states also includes the ones encoding configurations of M that are not reachable from the initial configuration. A timer is now coupled to the encoding of configurations of M, in order to force all states of the reaction systems belonging to this set to reach T once the timer reaches an appropriate value.
3. If a state of the reaction system does not encode a valid Turing machine configuration and a valid timer, it is detected and forced to reach T. Then, its dynamics will evolve as described in statement 1.

In this way, *all states* lead to the fixed point U only if T itself leads to U (*i.e.*, if M reaches the accepting configuration within the space bound m). This is accomplished by building the new reaction system $\mathcal{M} = (S, A)$ as follows.

Consider the background set of the reaction system from Section 3 and keep unchanged the way to encode any configuration of M into a state of reaction system state. Now, an entity is added to that background set, namely the entity $\triangleleft$ which represents the "reset" to be executed in some cases from a state of $\mathcal{M}$ to the state T, which has well defined dynamics and encoding the initial configuration of M.

Reminding that f_1 is the entity corresponding to the accepting state f in the accepting configuration of M (with the tape head located on the first cell), we introduce the following reaction in order to force the element f_1 to constitute the fixed point $U = \{f_1\}$ when it is the only entity of the state[2]:

$$(\{f_1\}, \varnothing, \{f_1\})$$

[2]Actually, U encodes the state of $\mathcal{M}$ coming immediately after the one encoding the accepting configuration of M. However, the two states coincide if one ignores the content of the tape and the direction.

Since we are interested only in accepting computations within a given space bound m, the set of reactions A also contains the following reactions, which force a "reset" by generating the entity $\triangleleft$ if the tape head reaches the $(m+1)$-th cell:

$$(\{q_{m+1}\}, \{f_1, \triangleleft\}, \{\triangleleft\}) \qquad \text{for } q \in Q$$

In order to deal with the simulation of M by $\mathcal{M}$ upon reaching an accepting configuration or the situation when a reset is needed, reactions 1 and 2 from Section 3 implementing every transition $\delta(q,a) = (q', a', d)$ appear now in A with f_1 and $\triangleleft$ as additional inhibitors:

$$(\{q_i, a_i\}, \{f_1, \triangleleft\}, \{q'_{i+d}, a'_i\}) \qquad \text{for } 1 \le i \le m$$
$$(\{a_i\}, \{q_i : q \in Q\} \cup \{f_1, \triangleleft\}, \{a_i\}) \qquad \text{for } 1 \le i \le m$$

In this way, the transition of M is not executed and the tape is not preserved, thus deleting the current configuration of the simulated Turing machine.

We now describe how $\triangleleft$ is produced by a timer in order to comply with property 2 on page 11. Denote by p_k the k-th prime number and define K as the smallest number such that $\prod_{k=1}^{K} p_k - 1$ is greater than the number of possible distinct configuration of M when working in space m. For each $k = 1, \dots, K$, let $c_{k,0}, \dots, c_{k,p_k-1}$ be further entities in the background set S of $\mathcal{M}$. For each k, the following group of reactions is included in A:

$$(\{c_{k,t}\}, \{f_1, \triangleleft\}, \{c_{k,(t+1) \bmod p_k}\}) \qquad \text{for } 0 \le t < p_k$$

Each group of reactions defines a cycle of length p_k. When all the reactions of all groups act together every time step of the evolution of the reaction system in which f_1 and $\triangleleft$ do not appear, they give rise to a cycle of length $L = \prod_{k=1}^{K} p_k$, since the p_k's are pairwise coprime. In this way, the dynamical evolution of $\mathcal{M}$ starting from any state including $Y_0 = \{c_{1,0}, \dots, c_{K,0}\}$ and containing neither f_1 nor $\triangleleft$ is associated with a timer whose value at time t can be identified by $Y_t = \{c_{1,t_1}, \dots, c_{K,t_K}\}$ where $t_k = t \bmod p_k$. The timer reaches its maximum value once the state of $\mathcal{M}$ contains $Y_t = \{c_{1,p_1-1}, \dots, c_{K,p_K-1}\}$, and the first time it happens is at time $t = L - 1$.

In the following two steps, the entity $\triangleleft$ is produced and the timer is reset to 0, the state of the reactions system $\mathcal{M}$ is set to $C_0 = T \cup Y_0$ as desired, and the simulation of M by $\mathcal{M}$ restarts from the initial configuration encoded by T. This is accomplished by the following two reactions:

$$(\{c_{k,p_k-1} : 1 \le k \le K\}, \{f_1, \triangleleft\}, \{\triangleleft\})$$
$$(\{\triangleleft\}, \{f_1\}, C_0)$$

Remark that, if M works in space m, then it has $|\Sigma|^m \times m \times |Q|$ possible distinct configurations. By letting $K = \lceil \log_2 (|\Sigma|^m \times m \times |Q|) \rceil + 1$, it holds that $\prod_{k=1}^{K} p_k - 1 = L - 1$ is larger that the number of possible configurations of M, since $p_k \ge 2$ for all k. By the Prime Number Theorem [12], the K-th prime is asymptotically $K \ln K$. To find K primes, it is therefore sufficient to test the primality of the first $O(K \ln K)$ natural numbers, where each number can be tested even using a brute force algorithm. Since this value is polynomial with respect to length of the description of M and with respect to m (which is given in unary), the construction of $\mathcal{M}$ we are describing can be carried out in polynomial time.

In order to comply with statement 3 on page 11, the set of reactions A must also contain reactions generating $\triangleleft$ each time the current state of the reaction systems fails to encode either a valid pair (*timer, configuration of* M) nor it is the fixed point $\{f_1\}$. First of all, the following reactions generate $\triangleleft$ when the current state of $\mathcal{M}$ contains zero or more than one entity encoding the pair (*state, head position of* M):

$$(\varnothing, \{q_i : q \in Q, 1 \le i \le m\} \cup \{f_1, \triangleleft\}, \{\triangleleft\})$$
$$(\{q_i, r_j\}, \{f_1, \triangleleft\}, \{\triangleleft\}) \qquad \text{for } q, r \in Q \text{ and } 1 \le i, j \le m$$

The entity $\triangleleft$ must also be produced each time the state of $\mathcal{M}$ does not encode exactly one symbol per tape cell:

$$(\varnothing, \{a_i : a \in \Sigma\} \cup \{f_1, \triangleleft\}, \{\triangleleft\}) \qquad \text{for } 1 \le i \le m$$
$$(\{a_i, b_i\}, \{f_1, \triangleleft\}, \{\triangleleft\}) \qquad \text{for } a, b \in \Sigma \text{ and } 1 \le i \le m$$

Finally, there must also be reactions in A producing $\triangleleft$ when the state does not include a well-formed timer. That is, when for a single prime p_k there does not appear exactly one entity of the form $c_{k,i}$ in the current state of $\mathcal{M}$:

$$(\emptyset, \{c_{k,t} : 0 \leq t < p_k\} \cup \{f_1, \triangleleft\}, \{\triangleleft\}) \qquad \text{for } 1 \leq k \leq K$$
$$(\{c_{k,t}, c_{k,s}\}, \{f_1, \triangleleft\}, \{\triangleleft\}) \qquad \text{for } 1 \leq k \leq K \text{ and } 0 \leq t, s < p_k$$

In this way, we have obtained a construction complying with properties 1–3 on page 11 and that can be carried out in polynomial time.

We are now able to prove that checking the existence of long cycles is **PSPACE**-complete.

Theorem 19. *Given a RS $\mathcal{A}$ over a background set S and a unary integer $\ell \geq 1$ as input, it is* **PSPACE***-complete to decide if $\mathcal{A}$ has a periodic point of period at least ℓ.*

Proof. The problem is in **PSPACE** since there exists an algorithm that iterates over all the possible states of $\mathcal{A}$ and checks if each of them leads back to itself after at least ℓ steps (but which is necessarily not greater than $2^{|S|}$). This can be performed in polynomial space.

We prove that the problem under consideration is **PSPACE**-hard by a reduction from the same problem used in the proof of Theorem 18. For any Turing machine M, any input string x, and any space bound m, consider the reaction system $\mathcal{M}$ from the above construction. Let $\ell = 2$ and let $k \geq \ell$ be the number of possible configurations for M when working in space m. By construction, it holds that if M accepts on input x, then each state of $\mathcal{M}$ leads to the fixed point $\{f_1\}$, and, in particular, no cycle of length at least ℓ can be reached. Otherwise, if M rejects on input x by diverging, there is a cycle containing the state C_0, which includes the encoding of the initial configuration of M. The length of this cycle is dictated by the time needed for the timer to generate the entity $\triangleleft$, which, by hypothesis, is greater than the number of configuration of M when working in space m, and, therefore, of k and, as a consequence, of ℓ. Hence, there is cycle of length at least ℓ if and only if M does not accept on input x working in space m. This concludes the reduction and the **PSPACE**-hardness of the problem under consideration follows. □

Also checking whether there exists a state that is far enough from a long cycle is **PSPACE**-complete.

Theorem 20. *Given a RS $\mathcal{A} = (S, A)$ and two unary integers $\ell \geq 1$ and k as input, it is* **PSPACE***-complete to decide if there exists a state $T \subseteq S$ such that, when $\mathcal{A}$ starts in T, it reaches a cycle of length at least ℓ after a preperiod of length at least k.*

Proof. The problem is in **PSPACE** since there exists a non-deterministic algorithm able to guess two states T and U and check in polynomial space that T leads to U in at least k steps, that U is in a cycle of length at least ℓ, and that U is the first state of that cycle visited by $\mathcal{A}$ starting from T. Since **PSPACE** is closed under non-determinism, it contains the problem under consideration.

Such a problem is **PSPACE**-hard since in the case $k = 0$ it is equivalent to the one from Theorem 19, which is **PSPACE**-complete. □

The same problem remains **PSPACE**-complete even when the starting state is given in input.

Theorem 21. *Given a RS $\mathcal{A} = (S, A)$, a state $T \subseteq S$ and two unary integers $\ell \geq 1$ and k as input, it is* **PSPACE***-complete to decide if, when $\mathcal{A}$ starts at T, it reaches a cycle of length at least ℓ (resp., at most ℓ) after a preperiod of length at least k.*

Proof. The problem is in **PSPACE** since it is possible to check in polynomial space if each state in the dynamics of $\mathcal{A}$ starting from T is part of a cycle or not, thus obtaining the preperiod and period of T.

The **PSPACE**-hardness follows by a reduction from the same problem used in the proof of Theorem 18. For any Turing machine M, input string x, and unary space bound m, let $\mathcal{M}$ be the reaction system obtained by the construction presented in this section. Take $T = C_0$, $k = 1$, and $\ell = 1$. It holds that T has preperiod at least one if and only if M accepts on input x. In that case, the dynamics reaches, after at least one step, the fixed point $\{f_1\}$, and thus T has period 1, otherwise T is part of a cycle and thus has preperiod 0. Then, the reduction is accomplished. □

All problems related to the existence of global attractors also turn out to be **PSPACE**-complete.

Theorem 22. *Given a RS $\mathcal{A} = (S, A)$ and a unary integer $\ell \geq 1$ as input, it is* **PSPACE***-complete to decide:*

1. *If a state $T \subseteq S$, also given as input, belongs to a global attractor cycle of length at most ℓ (resp., at least ℓ).*
2. *If there exists a global attractor cycle of length at most ℓ (resp., at least ℓ).*

Proof. All these problems are in **PSPACE**, since it is possible to check in polynomial space if any given state T of $\mathcal{A}$ belongs to a cycle of length at most ℓ (resp., at least ℓ) and it is reachable from every other state. The existential version of the problems can be solved by iterating across all the states of $\mathcal{A}$, which can also be performed in polynomial space.

The **PSPACE**-hardness again follows from the reduction from the same problem used in the proof of Theorem 18. For any Turing Machine M, input string x, and unary space bound m, according to the construction presented in this section, the reaction system $\mathcal{M}$ has a global attractor cycle, namely the fixed point $\{f_1\}$, if and only if M accepts on input x within space m. The reduction is completed by setting $\ell = 1$ and, for the first problem, $T = \{f_1\}$. □

7. Conclusions

This paper has tackled the study of the computational complexity of problems associated with the dynamics of RS. Table 1 provides a reference list of such problems. While these problems are mainly theoretical, their formulation takes inspiration from properties that usually have real-world significance in bio-chemical systems and can thus be useful in the study of RS-based models of real-world systems.

From a theoretical point of view, we stress that the interest of such studies is twofold. On one hand, results about RS are lower bounds for similar results of other finite discrete dynamics systems, like Boolean automata networks (BANs). In fact, RS can be considered as a constrained form of BANs, where each node in the network represents an entity and the update function encodes that at least one reaction having that entity as a product was active. In this sense, problems studied here in the context of RS cannot be simpler in the context of BANs, thus providing a lower bound on the complexity in another computational model.

On the other hand, when suitably constrained, RS are a simple model of parsimonious systems (*i.e.*, systems which have a description exponentially shorter than the complete description of their dynamics). A natural research direction consists in trying to understand what exactly causes the complexity jumps between similar problems (cf. the last two entries of Table 1, for example).

Another aspect to study is how to constrain RS in order to be more similar to systems that are actually found in nature. While all the constructions presented here are valid in the framework of reaction systems, it is important to remember that not all of them might be found in a "natural" system. Of particular interest is to understand if the complexity results found here also hold for real-world systems or if there are some differences, and what those differences are.

Acknowledgements

Enrico Formenti acknowledges the partial support from the project PACA APEX FRI. Alberto Dennunzio and Luca Manzoni were partially supported by Fondo d'Ateneo (FA) 2016 of Università degli Studi di Milano Bicocca: "Sistemi Complessi e Incerti: teoria ed applicazioni". Antonio E. Porreca was partially supported by Fondo d'Ateneo (FA) 2015 of Università degli Studi di Milano Bicocca: "Complessità computazionale e applicazioni crittografiche di modelli di calcolo bioispirati".

[1] S. Azimi, I. Bogdan, I. Petre, Reaction system models for the heat shock response, Fundamenta Informaticae 131 (2014) 299–312. URL: https://doi.org/10.3233/FI-2014-1016.

[2] S. Azimi, C. Gratie, S. Ivanov, L. Manzoni, I. Petre, A.E. Porreca, Complexity of model checking for reaction systems, Theoretical Computer Science 623 (2016) 103–113. URL: https://doi.org/10.1016/j.tcs.2015.11.040.

[3] R. Barbuti, P. Bove, R. Gori, F. Levi, P. Milazzo, Simulating gene regulatory networks using reaction systems, in: Proceedings of the 27th International Workshop on Concurrency, Specification and Programming, CS&P, pp. 119–132.

[4] R. Barbuti, R. Gori, F.L. Levi, P. Milazzo, Investigating dynamic causalities in reaction systems, Theoretical Computer Science 623 (2016) 114–145. URL: https://doi.org/10.1016/j.tcs.2015.11.041. doi:10.1016/j.tcs.2015.11.041.

Problem dealt with in this paper	Complexity
$\exists$ periodic point of period at most ℓ (*)	**NP**-complete
T has period $\leq \ell$ & $\exists U$ reaching T in $\geq k$ steps (*)	**NP**-complete
$\exists$ periodic point with period $\leq \ell$ & preperiod len. $\geq k$ (*)	**NP**-complete
T is global attractor of period $\leq \ell$ ($\geq \ell$) (*)	**PSPACE**-complete
Reachability (*)	**PSPACE**-complete
$\exists$ global fixed point attractor of period $\leq \ell$ ($\geq \ell$) (*)	**PSPACE**-complete
T belongs to a cycle of period (*) $\geq \ell$	**PSPACE**-complete
res is bijective (=)	**coNP**-complete
$\exists$ a periodic point with period $\leq \ell$ & preperiod len. $\geq k$ (*)	**NP**-complete
T has period $\leq \ell$	**P**-complete
T has period $\leq \ell$ & preperiod $\geq k$	**P**-complete
T has a k-predecessor	**P**-complete
res is the identity	**coNP**-complete
T is an attractor cycle of period $\leq \ell$ & diameter $\leq d$	**coNP**-complete
T is an attractor cycle of period $\leq \ell$ & diameter $\geq d$	**NP**-complete
$\exists$ an attractor cycle of period $\leq \ell$ & diameter $\geq d$	**NP**-complete
$\exists$ an attractor cycle of period $\leq \ell$ & diameter $\leq d$	$\Sigma_2^{\mathbf{P}}$-complete

Table 1: Comprehensive list of problems on the dynamics of RA and their complexity. (*)/(=): extended/same version of a result presented at either CiE 2014 [17] or DCFS 2014 [16].

[5] R. Barbuti, R. Gori, F.L. Levi, P. Milazzo, Specialized predictor for reaction systems with context properties, Fundamenta Informaticae 147 (2016) 173–191. URL: https://doi.org/10.3233/FI-2016-1403. doi:10.3233/FI-2016-1403.

[6] L. Corolli, C. Maj, F. Marini, D. Besozzi, G. Mauri, An excursion in reaction systems: From computer science to biology, Theoretical Computer Science 454 (2012) 95–108. URL: https://doi.org/10.1016/j.tcs.2012.04.003.

[7] A. Dennunzio, E. Formenti, L. Manzoni, Computing issues of asynchronous CA, Fundamenta Informaticae 120 (2012) 165–180. URL: https://doi.org/10.3233/FI-2012-755. doi:10.3233/FI-2012-755.

[8] A. Dennunzio, E. Formenti, L. Manzoni, Reaction systems and extremal combinatorics properties, Theoretical Computer Science 598 (2015) 138–149. URL: https://doi.org/10.1016/j.tcs.2015.06.001. doi:10.1016/j.tcs.2015.06.001.

[9] A. Dennunzio, E. Formenti, L. Manzoni, G. Mauri, A.E. Porreca, Computational complexity of finite asynchronous cellular automata, Theoretical Computer Science 664 (2017) 131–143. URL: https://doi.org/10.1016/j.tcs.2015.12.003.

[10] A. Dennunzio, E. Formenti, L. Manzoni, A.E. Porreca, Ancestors, descendants, and gardens of eden in reaction systems, Theoretical Computer Science 608 (2015) 16–26. URL: https://doi.org/10.1016/j.tcs.2015.05.046.

[11] A. Dennunzio, E. Formenti, L. Manzoni, A.E. Porreca, Reachability in resource-bounded reaction systems, in: A.H. Dediu, J. Janoušek, C. Martń-Vide, B. Truthe (Eds.), Language and Automata Theory and Applications: 10th International Conference, LATA 2016, volume 9618 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 592–602. URL: https://doi.org/10.1007/978-3-319-30000-9_45.

[12] U. Dudley, Elementary Number Theory, W.H. Freeman and Company, 1969.

[13] A. Ehrenfeucht, M. Main, G. Rozenberg, Combinatorics of life and death for reaction systems, International Journal of Foundations of Computer Science 21 (2010) 345–356. doi:https://doi.org/10.1142/S0129054110007295.

[14] A. Ehrenfeucht, M. Main, G. Rozenberg, Functions defined by reaction systems, International Journal of Foundations of Computer Science 22 (2011) 167–168. URL: https://doi.org/10.1142/S0129054111007927.

[15] A. Ehrenfeucht, G. Rozenberg, Reaction systems, Fundamenta Informaticae 75 (2007) 263–280. URL: http://iospress.metapress.com/content/b86t11hryvwq69l0/.

[16] E. Formenti, L. Manzoni, A.E. Porreca, Cycles and global attractors of reaction systems, in: H. Jürgensen, J. Karhumäki, A. Okhotin (Eds.), Descriptional Complexity of Formal Systems, 16th International Workshop, DCFS 2014, volume 8614 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 114–125. URL: https://doi.org/10.1007/978-3-319-09704-6_11.

[17] E. Formenti, L. Manzoni, A.E. Porreca, Fixed points and attractors of reaction systems, in: A. Beckmann, E. Csuhaj-Varjú, K. Meer (Eds.), Language, Life, Limits, 10th Conference on Computability in Europe, CiE 2014, volume 8493 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 194–203. URL: https://doi.org/10.1007/978-3-319-08019-2_20.

[18] E. Formenti, L. Manzoni, A.E. Porreca, On the complexity of occurrence and convergence problems in reaction systems, Natural Computing 14 (2015) 185–191. URL: https://doi.org/10.1007/s11047-014-9456-3.

[19] T. Helikar, N. Kochi, B. Kowal, M. Dimri, M. Naramura, S.M. Raja, V. Band, H. Band, J.A. Rogers, A comprehensive, multi-scale dynamical model of ErbB receptor signal transduction in human mammary epithelial cells, PLoS ONE 8 (2013) e61757. doi:https://doi.org/10.1371/journal.pone.0061757.
[20] J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.
[21] H. Kitano, Biological robustness, Nature Reviews Genetics 5 (2004) 826–837. doi:10.1038/nrg1471.
[22] M.S. Nobile, A.E. Porreca, S. Spolaor, L. Manzoni, P. Cazzaniga, G. Mauri, D. Besozzi, Efficient simulation of reaction systems on Graphics Processing Units, Fundamenta Informaticae 154 (2017) 307–321. URL: https://doi.org/10.3233/FI-2017-1568.
[23] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, 1993.
[24] A. Salomaa, Functional constructions between reaction systems and propositional logic, International Journal of Foundations of Computer Science 24 (2013) 147–159. URL: https://doi.org/10.1142/S0129054113500044.
[25] A. Salomaa, Minimal and almost minimal reaction systems, Natural Computing 12 (2013) 369–376. URL: https://doi.org/10.1007/s11047-013-9372-y.
[26] I. Shmulevich, E.R. Dougherty, W. Zhang, From Boolean to probabilistic Boolean networks as models of genetic regulatory networks, Proceedings of the IEEE 90 (2002) 1778–1792. doi:10.1109/JPROC.2002.804686.
[27] K. Sutner, On the computational complexity of finite cellular automata, Journal of Computer and System Sciences 50 (1995) 87–97. URL: https://doi.org/10.1006/jcss.1995.1009.