

# Processes Based on Biochemical Interactions: Natural Computing Point of View\*

Andrzej Ehrenfeucht and Grzegorz Rozenberg

## Introduction

In this paper we investigate the interactions between biochemical reactions from the natural computing point of view. Natural computing (see, e.g., [6, 7]) is concerned with human-designed computing inspired by nature and with computation taking place in nature (i.e., it investigates processes taking place in nature in terms of information processing). The former strand of research is quite well-established: some of the well-known examples are evolutionary computing, neural computing, cellular automata, swarm intelligence, molecular computing, quantum computing, artificial immune systems, and membrane computing. Examples of research themes from the latter strand of research are: computational nature of self-assembly, computational nature of developmental processes, computational nature of bacterial communication, computational nature of brain processes, computational nature of biochemical reactions, and system biology approach to bionetworks. A lot of research from this research strand underscores the fact that computer science is also the fundamental science of information processing, and as such a basic science for other scientific disciplines such as, e.g., biology.

This paper is concerned with the computational nature of processes driven by interactions between biochemical reactions in living cells. It presents a formal framework for investigating such processes, called the framework of *reaction systems* (see, e.g., [1–4]). In particular, it provides basic definitions together with the intuition/motivation behind them. The paper is of a tutorial and rather informal style – the reader is advised to consult the references provided in the paper for a precise formal treatment of reaction systems.

## 1 Reactions

The functioning of a biochemical reaction is based on facilitation and inhibition: a reaction can take place if all of its reactants are present and none of its inhibitors is present. If a reaction takes place, then it creates its product. Therefore to specify a reaction one needs to specify its set of reactants, its set of inhibitors, and its set of products – this leads to the following definition.

---

\* This paper is dedicated to Hans-Joerg Kreowski on the occasion of his 60th birthday.

**Definition 1.** A reaction is a triplet  $a = (R, I, P)$ , where  $R, I, P$  are finite sets. If  $S$  is a set such that  $R, I, P \subseteq S$ , then  $a$  is a reaction in  $S$ .

The sets  $R, I, P$  are also denoted by  $R_a, I_a, P_a$ , and called the *reactant set of  $a$* , the *inhibitor set of  $a$* , and the *product set of  $a$* , respectively. Also,  $\text{rac}(S)$  denotes the set of all reactions in  $S$ .

For a finite set of reactions  $A$ ,  $R_A = \bigcup_{a \in A} R_a$ ,  $I_A = \bigcup_{a \in A} I_a$ , and  $P_A = \bigcup_{a \in A} P_a$  are called the *reactant set of  $A$* , the *inhibitor set of  $A$* , and the *product set of  $A$* , respectively.

The effect of a reaction  $a$  is conditional: if  $R_a$  is present and no element of  $I_a$  is present then  $P_a$  is produced, otherwise reaction does not take place and “nothing” is produced. This is formalised as follows.

**Definition 2.** Let  $a$  be a reaction,  $A$  a finite set of reactions, and  $T$  a finite set.

(1)  $a$  is enabled by  $T$ , denoted by  $a \text{ en } T$ , if  $R_a \subseteq T$  and  $I_a \cap T = \emptyset$ .

(2) The result of  $a$  on  $T$ , denoted by  $\text{res}_a(T)$ , is defined by:  $\text{res}_a(T) = P_a$  if  $a \text{ en } T$ , and  $\text{res}_a(T) = \emptyset$  otherwise.

(3) The result of  $A$  on  $T$ , denoted by  $\text{res}_A(T)$ , is defined by:  
 $\text{res}_A(T) = \bigcup_{a \in A} \text{res}_a(T)$ .

Clearly, if  $R_a \cap I_a \neq \emptyset$ , then  $\text{res}_a(T) = \emptyset$  for every  $T$ . Therefore we assume that, for each reaction  $a$ ,  $R_a \cap I_a = \emptyset$ ; in this paper we will also assume that  $R_a \neq \emptyset$ ,  $I_a \neq \emptyset$ , and  $P_a \neq \emptyset$ .

As an example consider the reaction  $a$  with  $R_a = \{c, x_1, x_2\}$ ,  $I_a = \{y_1, y_2\}$ , and  $P_a = \{c, z\}$ . We can interpret  $c$  as the catalyzer of  $a$  (it is needed for  $a$  to take place, but is not “consumed” by  $a$ ),  $x_1, x_2$  as “real” reactants,  $y_1, y_2$  as inhibitors (e.g., acids inhibiting the functioning of  $c$  as the catalyzer), and  $z$  as the compound that is produced by this reaction. Then  $a \text{ en } T$  for  $T = \{c, x_1, x_2, z\}$ , and  $a$  is not enabled on neither  $\{c, x_1, x_2, z, y_1\}$  nor on  $\{x_1, x_2, z\}$ .

An important notion is the *activity* of a set of reactions  $A$  on a finite set (state)  $T$  – it is denoted by  $\text{en}_A(T)$ , and defined by:  $\text{en}_A(T) = \{a \in A : a \text{ en } T\}$ . Hence  $\text{en}_A(T)$  is the set of all reactions from  $A$  that are enabled by (active on)  $T$ . Note that  $\text{res}_A(T) = \text{res}_{\text{en}_A(T)}(T)$ : only the reactions from  $A$  which are enabled on  $T$  contribute to the result of  $A$  on  $T$ .

## 2 Basic Assumptions and Intuition

We will discuss now in more detail the basic notions of enabling and application (result) of reactions and sets of reactions, as they reflect our assumptions about biochemical reactions (motivated by organic chemistry of living organisms), which are very different from the underlying assumptions of majority of models (of human-designed systems) in theoretical computer science.

A reaction  $a$  is enabled on a set  $T$  if  $T$  separates  $R_a$  from  $I_a$  (i.e.,  $R_a \subseteq T$  and  $I_a \cap T = \emptyset$ ). We make no assumption about the relationship of  $P_a$  to either  $R_a$

or  $I_a$ . When  $a$  is enabled by a finite set  $T$ , then  $res_a(T) = P_a$ . Thus the result of  $a$  on  $T$  is “*locally determined*” in the sense that it uses only a subset of  $T$ , viz., the set of reactants  $R_a$ . However the result of the transformation is global: in comparing  $T$  with  $P_a$  we note that all elements from  $T - P_a$  “vanished”. This is in great contrast to classical models in theoretical computer science; e.g., in Petri nets (see, e.g., [5]) the firing of a single transition has only a local influence on the global marking which may be changed only on places that are neighbouring the given transition. Our way of defining the result of a reaction on a state  $T$  reflects our assumption that there is *no permanency* of elements: an element (molecule) of a global state vanishes unless it is sustained by a reaction.

The result of applying a set of reactions  $A$  to a state  $T$  is cumulative: it is the union of results of individual reactions from  $A$ . We do not set any conditions on the relationship between reactions in  $A$ . In particular, we do not have the (standard) notion of conflict here: if  $a, b \in A$  with  $a \text{ en } T$  and  $b \text{ en } T$ , then, even if  $R_a \cap R_b \neq \emptyset$ , still both  $a$  and  $b$  contribute to  $res_A(T)$ , i.e.,  $(res_a(T) \cup res_b(T)) \subseteq res_A(T)$ . Such a conflict of resources (standard in classical models such as, e.g., Petri nets) does not exist here. There is no counting in reaction systems, and so we deal with a qualitative rather than a quantitative model. This reflects our assumption about the “threshold supply” of elements (molecules): either an element is present, and then there is “enough” of it, or an element is not present.

There is a notion in reaction systems that reflects an intuition of conflict, viz., the notion of consistency. A set of reactions  $A$  is called *consistent* if  $R_a \cap I_b = \emptyset$ , i.e.,  $R_a \cap I_b = \emptyset$  for any two reactions  $a, b \in A$ ; clearly if  $R_a \cap I_b \neq \emptyset$ , then  $a$  and  $b$  can never be *together* enabled.

### 3 Reaction Systems and Interactive Processes

We are ready now to define reaction systems.

**Definition 3.** A reaction system, abbreviated *rs*, is an ordered pair  $\mathcal{A} = (S, A)$  such that  $S$  is a finite set, and  $A \subseteq \text{rac}(S)$ .

The set  $S$  is called the *background set* of  $\mathcal{A}$ , and  $A$  is called the *set of reactions* of  $\mathcal{A}$ . All the notions and notations introduced for sets of reactions carry over to reaction systems through their underlying sets of reactions. For example, for  $T \subseteq S$ ,  $en_{\mathcal{A}}(T) = en_A(T)$  and  $res_{\mathcal{A}}(T) = res_A(T)$  – also, we say that  $T$  is *active* in  $\mathcal{A}$ , if  $en_{\mathcal{A}}(T) \neq \emptyset$ .

It is important to note here that, in the setup of reaction systems, reactions are primary while structures are secondary. Since we do not have permanency of elements – elements vanish unless they are sustained by reactions, (sets of) reactions *create* states rather than *transform* states. Thus reaction systems do not work in an environment, but rather they create an environment.

The interactions of reaction systems is given by unions. For reaction systems  $\mathcal{A}_1 = (S_1, A_1)$  and  $\mathcal{A}_2 = (S_2, A_2)$  their union, denoted  $\mathcal{A}_1 + \mathcal{A}_2$ , is defined

by  $\mathcal{A}_1 + \mathcal{A}_2 = (S_1 \cup S_2, A_1 \cup A_2)$ . This way of combining reaction systems reflects the bottom-up modularity: local descriptions (reaction systems  $\mathcal{A}_1, \mathcal{A}_2$ ) are combined into global picture ( $\mathcal{A}_1 + \mathcal{A}_2$ ) in such a way that the interactions of local descriptions is provided automatically. Thus a major difference with standard models in theoretical computer science is that *no interface* is given/needed for combining reaction systems: the sheer fact that the sets of reactions  $A_1, A_2$  operate in the same molecular soup (tube) causes  $A_1, A_2$  to interact (again through facilitation and inhibition). Thus union is the basic mechanism for composing/decomposing reaction systems.

The dynamic behaviour of reaction systems is captured through the notion of an interactive process which is formally defined as follows.

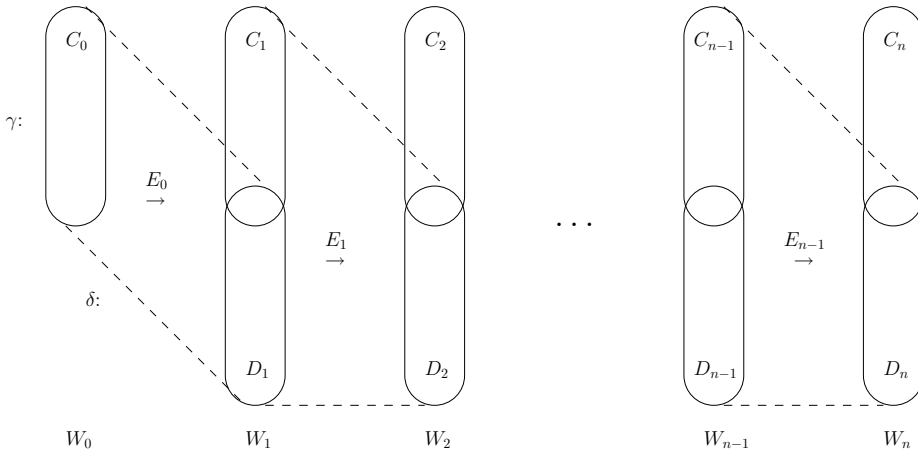
**Definition 4.** Let  $\mathcal{A} = (S, A)$  be a rs. An interactive process in  $\mathcal{A}$  is a pair  $\pi = (\gamma, \delta)$  of finite sequences such that  $\gamma = C_0, C_1, \dots, C_n$ ,  $\delta = D_1, \dots, D_n$ ,  $n \geq 1$ , where  $C_0, \dots, C_n, D_1, \dots, D_n \subseteq S$ ,  $D_1 = \text{res}_{\mathcal{A}}(C_0)$ , and  $D_i = \text{res}_{\mathcal{A}}(D_{i-1} \cup C_{i-1})$  for each  $2 \leq i \leq n$ .

The sequence  $C_0, \dots, C_n$  is the *context sequence* of  $\pi$ , and the sequence  $D_1, \dots, D_n$  is the *result sequence* of  $\pi$ . Let  $W_0 = C_0$ , and  $W_i = D_i \cup C_i$  for all  $1 \leq i \leq n$ . Then the sequence  $W_0, \dots, W_n$  is the *state sequence* of  $\pi$ , denoted  $\text{sts}(\pi)$ , and  $W_0$  is the initial state of  $\pi$ . For each  $0 \leq j \leq n$ ,  $C_j$  is the *context* of  $W_j$ . The sequence  $E_0, \dots, E_{n-1}$  of subsets of  $A$  such that  $E_i = \text{en}_{\mathcal{A}}(W_i)$ , for all  $0 \leq i \leq n-1$ , is the *activity sequence* of  $\pi$ , denoted  $\text{act}(\pi)$ . If  $\text{act}(\pi)$  consists of nonempty sets only, then  $\text{sts}(\pi)$  is *active* – in this case all states  $W_1, \dots, W_{n-1}$  are active. The set of all state sequences of (all interactive processes in)  $\mathcal{A}$  is denoted by  $\text{STS}(\mathcal{A})$ .

The basic intuition behind the notion of an interactive process is rather straightforward. Context  $C_0$  represents the initial state of  $\pi$ , i.e., the state in which  $\pi$  begins (is initiated), and the contexts  $C_1, \dots, C_n$  represent the influence of (the interaction with) the “rest of the world”. Then  $D_1$  is the result of  $\mathcal{A}$  on  $C_0$ , i.e., the result of applying to  $C_0$  all the reactions from  $\mathcal{A}$  enabled on  $C_0$ . Together with context  $C_1, D_1$  forms the successor state  $W_1$  of the initial state. Then, iteratively, the result of applying  $\mathcal{A}$  to state  $W_{i-1} = D_{i-1} \cup C_{i-1}$  yields the result  $D_i$  which together with the context  $C_i$  forms the successor state  $W_i$ . Note that even if  $D_i = \emptyset$ ,  $W_i$  can still be an active state (if  $\text{en}_{\mathcal{A}}(C_i) \neq \emptyset$ ). The definition of an interactive process is illustrated in Figure 1.

## 4 Extended Reaction Systems

Reaction systems form the basic construct of the broad “framework of reaction systems”. However, within this framework we use an “onion approach” meaning that additional levels/components can be incrementally added (or removed) so that the resulting model is well fitted for the research issue at hand. An example of such an (incremental) approach are extended reaction systems which are suitable for investigating the issue of emergence of modules in biochemical systems,



**Fig. 1.** An interactive process.

investigated in [3] and presented in the next section. We use the notation  $2^S$  to denote the set of subsets of a set  $S$ .

**Definition 5.** An extended reaction system, abbreviated *ers*, is a triplet  $\mathcal{A} = (S, A, R)$  such that  $(S, A)$  is a reaction system, and  $R$  is a binary relation,  $R \subseteq 2^S \times 2^S$ .

We refer to  $(S, A)$  as the *underlying reaction system* of  $\mathcal{A}$  denoted by  $und(\mathcal{A})$ .

The role of the restriction relation is to restrict the set of interactive processes as follows. An interactive process of  $\mathcal{A}$  is an interactive process  $\pi = (\gamma, \delta)$  of  $und(\mathcal{A})$  such that if  $sts(\pi) = W_0, W_1, \dots, W_n$ , then, for each  $0 \leq i \leq n-1$ ,  $(W_i, W_{i+1}) \in R$ . Thus interactive processes of  $\mathcal{A}$  are these interactive processes of  $und(\mathcal{A})$ , where each two consecutive states in the state sequence are related (allowed) by  $R$ . We also require that the restriction relation is not too restrictive, i.e., that for each state sequence  $W_0, W_1, \dots, W_n$  of  $\mathcal{A}$  there exists  $W_{n+1} \subseteq S$  such that  $W_0, W_1, \dots, W_n, W_{n+1}$  is also a state sequence of  $\mathcal{A}$ . In other words, each interactive process of  $\mathcal{A}$  can be extended, as is the case in reaction systems.

A distinct feature of extended reaction systems is the existence of periodic elements – such elements cannot exist in reaction systems. An element  $t$  of an *ers*  $\mathcal{A}$  is *periodic (in  $\mathcal{A}$ )* if there exists a positive integer  $n$  such that for each  $W_0, W_1, \dots, W_n \in STS(\mathcal{A})$ ,  $t \in W_0$  if and only if  $t \in W_n$ ; the smallest such  $n$  is called the *period of  $t$* . Hence, if  $t$  is a periodic element with period  $n$ ,  $W_0, W_1, \dots, W_q \in STS(\mathcal{A})$ , and  $0 \leq i \leq q$ , then if  $t \in W_i$  then also  $t \in W_{i-n}$  (providing that  $i - n \geq 0$ ) and  $t \in W_{i+n}$  (providing that  $i + n \leq q$ ). The set of all periodic elements of  $\mathcal{A}$  is denoted by  $per(\mathcal{A})$ , and for each  $T \subseteq S$ ,  $per_{\mathcal{A}}(T) = T \cap per(\mathcal{A})$  is the set of periodic elements of  $T$ .

The existence of periodic elements motivates the following definition of computing the images of subsets of a given state (of an interactive process) in the successor state. Let  $\mathcal{A} = (S, A, R)$  be an *ers*, let  $\tau = W_0, W_1, \dots, W_n \in STS(\mathcal{A})$ , let  $i \in \{0, \dots, n-1\}$ , and let  $Q \subseteq W_i$ . Then the *image of  $Q$  in  $W_{i+1}$  (within  $\tau$ )*, denoted by  $im_{\mathcal{A}, \tau, i}(Q)$ , is defined by  $im_{\mathcal{A}, \tau, i}(Q) = res_{E_i}(Q \cup per_{\mathcal{A}}(W_i)) - per_{\mathcal{A}}(res_{E_i}(W_i))$ . The intuition behind this definition of an image is as follows: since periodic elements are included in fixed states of state sequences “independently of the applied reactions” (i.e., we can predict/compute the states of a state sequence where a periodic element belongs without knowing reactions that are actually applied to states), they are added to a “real argument” (i.e.,  $Q$ ) of the  $res_{E_i}$  when computing  $im_{\mathcal{A}, \tau, i}$ . For the same reason we abstract the periodic elements of  $res_{E_i}(W_i)$ , because we want in the image of  $Q$  only the “real results” (which excludes elements from  $per_{\mathcal{A}}(res_{E_i}(W_i))$  which will be in  $W_{i+1}$  anyhow because of their periodicity).

## 5 Events and Modules

Among all the subsets of a state of an interactive process we will distinguish “material subsets” – these are subsets that are the result of applying reactions of a system to subsets of the predecessor state. More formally, let  $\tau = W_0, W_1, \dots, W_n$  be a state sequence of an *ers*  $\mathcal{A}$ , and let us consider state  $W_i$  for some  $1 \leq i \leq n$ . A subset  $X \subseteq W_i$  is a “material subset” of  $W_i$  if there exists a subset  $Y \subseteq W_{i-1}$  such that  $X$  is the product of the set of reactions enabled on  $W_{i-1}$  applied to  $Y$ . Such products included in  $W_i$  are “modules” of  $W_i$ . If we now consider the sequence of modules in consecutive states of  $\tau$  initiated by  $Y \subseteq W_{i-1}$ , beginning in  $W_i$  (with  $X$ ) and ending in some  $W_j$  for  $j \geq i$ , then we are tracing the fate of  $Y$  (as a sequence of products) through  $(j-i+1)$  steps of (an interactive process  $\pi$  behind)  $\tau$ . Such sequences of modules are called events which are formally defined below. If we are interested in a module  $Q$  in some  $W_k$ , for  $1 \leq k \leq n$ , and follow backwards an event that produced  $Q$  in  $W_k$ , then we get a possible history of  $Q$ , hence an explanation of why and how  $Q$  was created in  $W_k$ .

**Definition 6.** Let  $\mathcal{A}$  be an *ers*, let  $\tau = W_0, W_1, \dots, W_n \in STS(\mathcal{A})$ , let  $i, j \in \{1, \dots, n\}$  be such that  $i \leq j$ , and let  $\omega = Q_i, \dots, Q_j$  be such that  $Q_i \subseteq W_i, \dots, Q_j \subseteq W_j$ , and all  $Q_i, \dots, Q_{j-1}$  are nonempty. Then  $\omega$  is an event in  $\tau$  if there is a  $Q_{i-1} \subseteq W_{i-1}$  such that, for each  $k \in \{i, \dots, j\}$ ,  $Q_k = im_{\mathcal{A}, \tau, k-1}(Q_{k-1})$ .

We say that  $\omega$  is *passing through* each of  $W_i, \dots, W_j$ ; if  $Q_j = \emptyset$ , then  $\omega$  *dies in*  $W_j$ . The sets  $Q_i, \dots, Q_j$  are called the *modules of  $\omega$  in  $W_i, \dots, W_j$* , respectively. More specifically, each module  $Q_l$ ,  $i \leq l \leq j$ , is called a *l-module*.

Thus, intuitively, an event ( $\omega$ ) is tracing the fate of a subset ( $Q_{i-1}$ ) of a state ( $W_{i-1}$ ) of a state sequence  $\tau$  within a segment ( $W_i, \dots, W_j$ ) of  $\tau$ . More specifically, suppose that we are interested in a state sequence  $\tau$  (or in an interactive

process  $\pi$  with  $sts(\pi) = \tau$ ), and in particular we are interested in the dynamic development of some  $Q_{i-1} \subseteq W_{i-1}$  as  $\tau$  evolves from  $W_i$  on until  $W_j$  is reached. This dynamic development of  $Q_{i-1}$  in the segment  $W_i, \dots, W_j$  is the sequence  $Q_i, \dots, Q_j$  of material subsets (modules) of  $W_i, \dots, W_j$ , respectively. Note that both the notion of the result of transforming  $Q_l$  into  $Q_{l+1}$ ,  $l \in \{i, \dots, j-1\}$ , and the notion of a material subset are modified (w.r.t. reaction systems) to take into account the existence of periodic elements in extended reaction systems.

When an event  $\omega$  is passing through a state  $W_l$  then it leaves a “trace” there, viz., its module  $Q_l$ . The set of all such traces in  $W_l$  left there by all events passing through  $W_l$  is called the *snapshot of  $W_l$  in  $\tau$* , denoted by  $snp_\tau(k)$ . Thus for the given state sequence  $\tau = W_0, \dots, W_n$  we get the corresponding sequence of snapshots  $snp(\tau) = \mathcal{S}_1, \dots, \mathcal{S}_n$ , where  $\mathcal{S}_i = snp_\tau(i)$  for each  $1 \leq i \leq n$ , called the *snapshot sequence of  $\tau$* , and also called the *snapshot sequence of  $\mathcal{A}$* .

Given a snapshot sequence  $\rho = \mathcal{S}_1, \dots, \mathcal{S}_n$  of a state sequence  $\tau = W_0, \dots, W_n$  there exists a natural sequence of partial functions  $next_{\tau,1}, next_{\tau,2}, \dots, next_{\tau,n-1}$  transforming consecutive snapshots of  $\rho$  into their successor snapshots, where, for each  $1 \leq k \leq n-1$ ,  $next_{\tau,k} : \mathcal{S}_k \rightarrow \mathcal{S}_{k+1}$  is defined as follows. For  $Q \in \mathcal{S}_k$  and  $Q' \in \mathcal{S}_{k+1}$ ,  $next_{\tau,k}(Q) = Q'$  if and only if  $Q, Q'$  are nonempty and there exists an event  $\omega$  in  $\tau$  such that  $Q$  is the module of  $\omega$  in  $W_k$  and  $Q'$  is the module of  $\omega$  in  $W_{k+1}$ . If we extend the  $next_{\tau,k}$  function also to pairs  $(Q, Q')$  with  $Q'$  possibly empty, then the resulting function is denoted by  $suc_{\tau,k}$ . Thus, intuitively, the function  $next_{\tau,k}$  connects nonempty modules that are consecutive in an event passing through  $W_k$  and  $W_{k+1}$ . In this way the sequence of functions  $next_{\tau,1}, \dots, next_{\tau,n-1}$  delineate all the events of  $\tau$  as they are passing through the states of  $\tau$ , but it does not explicitly indicate the “moment of death” (if an event dies). The sequence of functions  $suc_{\tau,1}, \dots, suc_{\tau,n-1}$  does indicate also the death moments. As a matter of fact the empty module has really no physical interpretation – it is clearly no material subset, but rather its role is to signal the termination (the death) of an event. It is therefore convenient to consider snapshots with the empty set removed. In this way, for a given snapshot sequence  $\rho = \mathcal{S}_1, \dots, \mathcal{S}_n$  we obtain its  $\emptyset$ -free version  $\bar{\rho} = \bar{\mathcal{S}}_1, \dots, \bar{\mathcal{S}}_n$ , where for each  $1 \leq i \leq n$ ,  $\bar{\mathcal{S}}_i = \mathcal{S}_i - \{\emptyset\}$ . Accordingly, each  $next_{\tau,k}$  function is modified to the  $rnext_{\tau,k}$  function which is  $next_{\tau,k}$  restricted to  $\bar{\mathcal{S}}_k$ .

We move now to present the structure of snapshots. First we need a couple of set-theoretical notions.

**Definition 7.** Let  $\mathcal{L}$  be a family of sets and let  $\mathcal{F}_1, \mathcal{F}_2 \subseteq \mathcal{L}$  be nonempty.

- (1) We say that  $\mathcal{F}_1$  is embedded in  $\mathcal{F}_2$  if  $\bigcup \mathcal{F}_1 \subseteq \bigcap \mathcal{F}_2$ .
- (2) We say that  $\mathcal{F}_1$  is separated from  $\mathcal{F}_2$  in  $\mathcal{L}$  if there exists  $U \in \mathcal{L}$  such that  $\bigcup \mathcal{F}_1 \subseteq U \subseteq \bigcap \mathcal{F}_2$ .

**Theorem 1.** Let  $\mathcal{A}$  be an ers, let  $\tau = W_0, W_1, \dots, W_n \in STS(\mathcal{A})$  where  $n \geq 2$ , let  $snp(\tau) = \mathcal{S}_1, \dots, \mathcal{S}_n$ , and let  $1 \leq k \leq n-1$ . If  $\mathcal{F}_1, \mathcal{F}_2 \subseteq \bar{\mathcal{S}}_k$  are nonempty families of sets such that  $\mathcal{F}_1$  is embedded in  $\mathcal{F}_2$  and  $next_{\tau,k}$  is defined on all modules in  $\mathcal{F}_1 \cup \mathcal{F}_2$ , then  $next_{\tau,k}(\mathcal{F}_1)$  is separated from  $next_{\tau,k}(\mathcal{F}_2)$  in  $\bar{\mathcal{S}}_{k+1}$ .

This is a remarkable result as it allows us to view (extended) reaction systems as self-organizing systems, where a possible goal of interactive processes is to ensure (improve on) separability!

An interactive process (hence a run) of an *ers*  $\mathcal{A}$  produces a sequence  $\rho$  of snapshots  $\mathcal{S}_1, \dots, \mathcal{S}_k, \dots, \mathcal{S}_n$ . In general such a sequence may be very “unstable” because there may be no “mathematical similarity” between  $\mathcal{S}_k$  and  $\mathcal{S}_{k+1}$ : remember that the context of the state  $W_{k+1}$  (in the state sequence  $\tau = W_0, \dots, W_n$  for which  $\rho = \text{snp}(\tau)$ ) can “throw anything” into  $W_{k+1}$ . So we can talk about local stability (at  $W_k$ ) only if there is a strong mathematical similarity between  $\mathcal{S}_k$  and  $\mathcal{S}_{k+1}$ . Perhaps the most natural choice for such a strong similarity is to require that  $\text{rnext}_{\tau,k}$  is an isomorphism between partial orders  $(\bar{\mathcal{S}}_k, \subseteq)$  and  $(\bar{\mathcal{S}}_{k+1}, \subseteq)$ . When this happens, we get a local stability – it is local because, again, “anything can happen” to  $\mathcal{S}_{k+2}$  (through the context of  $W_{k+2}$ ). Hence we say that  $(\mathcal{S}_k, \mathcal{S}_{k+1})$  is a *locally stable situation* if  $\text{rnext}_k$  is an isomorphism between  $(\bar{\mathcal{S}}_k, \subseteq)$  and  $(\bar{\mathcal{S}}_{k+1}, \subseteq)$ . We want to point out that the situation is quite subtle here, e.g., the fact that  $\bar{\mathcal{S}}_k = \bar{\mathcal{S}}_{k+1}$  does not necessarily imply that  $\text{rnext}_{\tau,k}$  is an isomorphism of  $\bar{\mathcal{S}}_k$  onto  $\bar{\mathcal{S}}_{k+1}$ .

It turns out that local stability is reflected in the structure of the corresponding snapshots.

**Theorem 2.** *Let  $\mathcal{A}$  be an *ers*, let  $\tau \in STS(\mathcal{A})$ , and let  $\mathcal{S}, \mathcal{S}'$  be two consecutive elements of  $\text{snp}(\tau)$ . If  $(\mathcal{S}, \mathcal{S}')$  is a locally stable situation, then both  $(\mathcal{S}, \subseteq)$  and  $(\mathcal{S}', \subseteq)$  are complete lattices.*

## 6 Discussion

We have presented in this paper an informal introduction to the framework of reaction systems. It is motivated by organic chemistry of living cells, and more specifically by interactions between biochemical reactions. The basic notions here are reactions and their results, i.e., the way they process states – this way of processing the states of a system is very different from the manner that state processing happens in common models in theoretical computer science. The differences (and motivation between them) are discussed in detail in this paper. The basic model of our framework are reaction systems and the basic notion/tool to investigate their dynamics is an interactive process. Although reaction systems form the core of our framework, the framework is constructed in an “incremental” way: depending on a research issue the notion of reaction system can be modified so that the resulting model is well-suited for the investigation of the given research issue. For example, reaction systems form a qualitative model where we do not have counting (of elements), as is the case for models based on multisets rather than on sets. However there are many situations where one needs to assign quantitative parameters to states (e.g., when dealing with time issues). Our point of view is that a numerical value can be assigned to a state  $T$  if there is a measurement of  $T$  yielding this value. This leads to the notion of



*reaction systems with measurements*, where a finite set of measurement functions is added as a third component to reaction systems (see [4]).

Another example of research leading to an incremental modification of the notion of a reaction system, is the investigation of the way that the products are formed and evolve within the runs of biochemical systems. The resulting extended reaction systems and the formation of products (the topics of [3]) are discussed in detail in this paper. The basic dynamic notion here is the notion of an event which traces the formation of modules (products) within interactive processes of a system. The rather surprising results that (extended) reaction systems can be seen as self-organising systems which in stable situations produce well-structured sets of molecules are also presented.

Altogether this paper presents both the basic setup, its motivation, and some typical research themes and results of the framework of reaction systems.

**Acknowledgements** The authors are indebted to Robert Brijder and Hendrik Jan Hooeboom for comments on this paper, and to Robert Brijder and Marloes van der Nat for their help in producing this paper.

## References

1. Ehrenfeucht, A., Rozenberg, G., Basic notions of reaction systems, Lecture Notes in Computer Science, v. 3340, 27-29, Springer, 2004.
2. Ehrenfeucht, A., Rozenberg, G., Reaction systems, *Fundamenta Informaticae*, v. 75, 263-280, 2007.
3. Ehrenfeucht, A., Rozenberg, G., Events and modules in reaction systems, *Theoretical Computer Science*, v. 376, 3-16, 2007.
4. Ehrenfeucht, A., Rozenberg, G., Introducing time in reaction systems, *Theoretical Computer Science*, v. 410, 310-322, 2009.
5. Engelfriet, J., Rozenberg, G., Elementary net systems, *Lecture Notes in Computer Science*, v. 1491, 12-121, Springer, 1998.
6. Kari, L., Rozenberg, G., The many facets of natural computing, *Communications of the ACM*, v. 51, 72-83, 2008.
7. Rozenberg, G., Computer science, informatics, and natural computing – personal reflections, in S.B. Cooper, B. Löwe, A. Sorbi, eds., *New Computational Paradigms – Changing Conceptions of What is Computable*, Springer, Berlin, Heidelberg, 2007.



**Prof. Dr. Andrzej Ehrenfeucht**

Department of Computer Science  
University of Colorado at Boulder  
Boulder, CO 80309-0430 (U.S.A.)  
andrzej@cs.colorado.edu  
<http://www.cs.colorado.edu/~andrzej>

---

**Prof. Dr. Grzegorz Rozenberg**

Department of Computer Science  
University of Colorado at Boulder  
Boulder, CO 80309-0430 (U.S.A.)

*and*

Leiden Institute of Advanced Computer Science,  
Leiden Center for Natural Computing,  
Leiden University, NL-2333 CA Leiden (The Netherlands)  
rozenber@liacs.nl  
<http://www.liacs.nl/~rozenber>

Grzegorz Rozenberg has been knowing Hans-Jörg Kreowski for about 30 years now. For the first time, they met during one of Grzegorz' visits to Hartmut Ehrig in Berlin. Since then they wrote a number of joint papers, the first one already in 1981. They also participated in a number of joint European research projects.

Hans-Jörg visited Grzegorz many times in Bilthoven, and Grzegorz visited him in Berlin and Bremen. They have become good family friends, with active mutual interest in the artistic careers of their sons Daniel and Kai, and look forward to many more meetings to come.

---