Electronic Communications of the EASST
Volume 26 (2010)

Manipulation of Graphs, Algebras and Pictures

Essays Dedicated to Hans-Jörg Kreowski on the Occasion of His 60th Birthday

Reaction Systems: a Formal Framework for Processes Based on Biochemical Interactions

Andrzej Ehrenfeucht and Grzegorz Rozenberg

9 pages

Guest Editors: Frank Drewes, Annegret Habel, Berthold Hoffmann, Detlef Plump
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Reaction Systems: a Formal Framework for Processes Based on Biochemical Interactions*

Andrzej Ehrenfeucht[1] **and Grzegorz Rozenberg**[2]

[1] Department of Computer Science
University of Colorado at Boulder, USA

[2] Department of Computer Science
University of Colorado at Boulder, USA
and
Leiden Institute of Advanced Computer Science
Leiden Center for Natural Computing,
Leiden University, The Netherlands

Abstract: This paper presents a formal framework for investigating processes driven by interactions between biochemical reactions in living cells. These interactions are based on the mechanisms of facilitation and inhibition, which underlie the definition of reaction systems – the central construct of our framework. We discuss in this paper the basic setup for reaction systems, and its motivation. We also present an important extension of reaction systems as well as some research topics and results.

Keywords: natural computing; biochemical interactions; reaction systems

1 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 also with computation taking place in nature (i.e., it also 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 approaches 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, 2, 3, 4]). In particular,

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

it provides basic definitions together with the intuition/motivation behind them as well as some research themes, and results concerning the formation of structures (modules) during runs of reaction systems.

2 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.

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, $rac(S)$ 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 formalized 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 *en* T, if $R_a \subseteq T$ and $I_a \cap T = \emptyset$.
(2) The *result of* a *on* T, denoted by $res_a(T)$, is defined by: $res_a(T) = P_a$ if a *en* T, and $res_a(T) = \emptyset$ otherwise.
(3) The *result of* A *on* T, denoted by $res_A(T)$, is defined by:
$res_A(T) = \bigcup_{a\in A} res_a(T)$.

Clearly, if $R_a \cap I_a \neq \emptyset$, then $res_a(T) = \emptyset$ 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 *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 $en_A(T)$, and defined by: $en_A(T) = \{a \in A : a \text{ en } T\}$. Hence $en_A(T)$ is the set of all reactions from A that are enabled by (active on) T. Note that $res_A(T) = res_{en_A(T)}(T)$: only the reactions from A which are enabled on T contribute to the result of A on T.

3 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 a 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 *en* T and b *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.

We would like to mention here that there is a notion in reaction systems, viz., the notion of consistency (see, e.g., [4]), that reflects an intuition of conflict. A set of reactions A is called *consistent* if $R_A \cap I_A = \emptyset$, 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.

4 Reaction Systems and Interactive Processes

We are ready now to define reaction systems.

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

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

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, in this sense 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 $\mathscr{A}_1 = (S_1, A_1)$ and $A_2 = (S_2, A_2)$ their union, denoted $\mathscr{A}_1 + \mathscr{A}_2$, is defined by $\mathscr{A}_1 + \mathscr{A}_2 = (S_1 \cup S_2, A_1 \cup A_2)$. This way of combining reaction systems reflects the bottom-up modularity: local descriptions (reaction systems $\mathscr{A}_1, \mathscr{A}_2$) are combined into the global picture ($\mathscr{A}_1 + \mathscr{A}_2$) 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 $\mathscr{A} = (S, A)$ be a *rs*. An *interactive process* in $\mathscr{A}$ is a pair $\pi = (\gamma, \delta)$ of finite sequences such that $\gamma = C_0, C_1, ...C_n$, $\delta = D_1, ..., D_n, n \geq 1$, where $C_0, ..., C_n, D_1, ..., D_n \subseteq S$, $D_1 = res_{\mathscr{A}}(C_0)$, and $D_i = res_{\mathscr{A}}(D_{i-1} \cup C_{i-1})$ for each $2 \leq i \leq n$.

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

The basic intuition behind the notion of an interactive process is rather straightforward. Context C_0 represents the initial state of π, i.e., the state in which π begins (is initiated), and the contexts $C_1, ..., C_n$ represent the influence of (the interaction with) the "rest of the world". Then D_1 is the result of $\mathscr{A}$ on C_0, i.e., the result of applying to C_0 all the reactions from $\mathscr{A}$ 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 $\mathscr{A}$ 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 $en_{\mathscr{A}}(C_i) \neq \emptyset$). The definition of an interactive process is illustrated in Figure 1.

Note that the background set S provides the elements of all the sets (reactants, inhibitors, products as well as contexts, results, states, ...) used in defining/analyzing a given reaction system.

5 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/structures in bio-

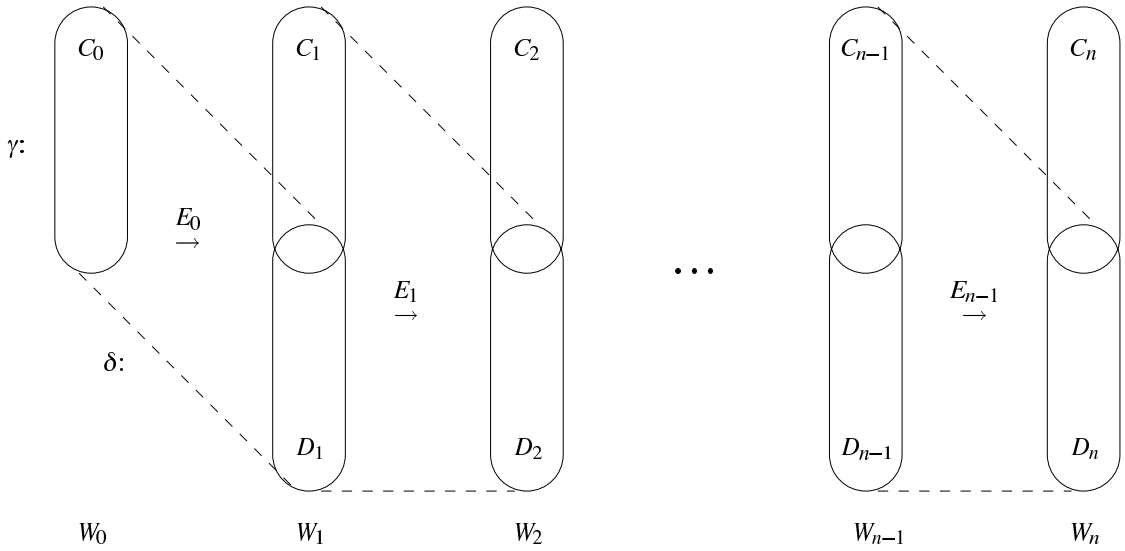

Figure 1: An interactive process.

chemical systems, 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 $\mathscr{A} = (S,A,R)$ 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* $\mathscr{A}$ denoted by $und(\mathscr{A})$.

The role of the restriction relation is to restrict the set of interactive processes as follows. An interactive process of $\mathscr{A}$ is an interactive process $\pi = (\gamma,\delta)$ of $und(\mathscr{A})$ such that if $sts(\pi) = W_0,W_1,\ldots,W_n$, then, for each $0 \leq i \leq n-1, (W_i,W_{i+1}) \in R$. Thus interactive processes of $\mathscr{A}$ are those interactive processes of $und(\mathscr{A})$, 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 $W_0,W_1,...,W_n$ of $\mathscr{A}$ there exists $W_{n+1} \subseteq S$ such that $W_0,W_1,...,W_n,W_{n+1}$ is also a state sequence of $\mathscr{A}$. In other words, each interactive process of $\mathscr{A}$ can be extended, as is the case in reaction systems.

A possible intuition for relation R is observability: the only state transitions that are observable are those specified by R – therefore interactive processes of extended reaction systems are observable processes. However, in general, R may express all kinds of restrictions on state transitions in reaction systems.

A distinct technical feature of extended reaction systems is the existence of periodic elements – such elements cannot exist in reaction systems. An element t of an *ers* $\mathscr{A}$ is *periodic* (*in* $\mathscr{A}$) if there exists a positive integer n such that for each $W_0,W_1,...,W_n \in STS(\mathscr{A}), t \in W_0$ 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 $n, W_0,W_1,...,W_q \in STS(\mathscr{A})$, 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 $\mathscr{A}$ is denoted by $per(\mathscr{A})$, and for each $T \subseteq S$, $per_{\mathscr{A}}(T) = T \cap per(\mathscr{A})$ 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 $\mathscr{A} = (S,A,R)$ be an *ers*, let $\tau = W_0, W_1, ..., W_n \in STS(\mathscr{A})$, let $i \in \{0,...,n-1\}$, and let $Q \subseteq W_i$. Then the *image of Q in W_{i+1} (within τ)*, denoted by $im_{\mathscr{A},\tau,i}(Q)$, is defined by $im_{\mathscr{A},\tau,i}(Q) = res_{E_i}(Q \cup per_{\mathscr{A}}(W_i)) - per_{\mathscr{A}}(res_{E_i}(W_i))$. 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 $im_{\mathscr{A},\tau,i}$. For the same reason we substract 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 $per_{\mathscr{A}}(res_{E_i}(W_i))$ which will be in W_{i+1} anyhow because of their periodicity).

6 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 (in this interactive process). More formally, let $\tau = W_0, W_1, \ldots, W_n$ be a state sequence of an *ers* $\mathscr{A}$, 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 τ initiated by some nonempty $Y \subseteq W_{i-1}$, beginning with X in W_i 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 π behind) τ. 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 $\mathscr{A}$ be an *ers*, let $\tau = W_0, W_1, \ldots, W_n \in STS(\mathscr{A})$, let $i, j \in \{1, \ldots, n\}$ be such that $i \leq j$, and let $\omega = Q_i, \ldots, Q_j$ be such that $Q_i \subseteq W_i, \ldots, Q_j \subseteq W_j$, and all $Q_i, \ldots, Q_{j-1}$ are nonempty. Then ω is *an event in τ* if there is a $Q_{i-1} \subseteq W_{i-1}$ such that, for each $k \in \{i, \ldots, j\}$, $Q_k = im_{\mathscr{A},\tau,k-1}(Q_{k-1})$.

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

Thus, intuitively, an event (ω) is tracing the fate of a subset (Q_{i-1}) of a state (W_{i-1}) in a state sequence τ within a segment ($W_i, \ldots, W_j$) of τ. More specifically, suppose that we are interested in a state sequence τ (or in an interactive process π with $sts(\pi) = \tau$), and in particular we are interested in the dynamic development of some $Q_{i-1} \subseteq W_{i-1}$ as τ evolves from W_i on until W_j is reached. This dynamic development of Q_{i-1} in the segment $W_i, \ldots, W_j$ is the sequence $Q_i, \ldots, Q_j$

of material subsets (modules) of $W_i,\ldots,W_j$, respectively. Note that both the notion of the result of transforming Q_l into Q_{l+1}, $l \in \{i,\ldots,j-1\}$, and the notion of a material subset take into account the existence of periodic elements in extended reaction systems.

When an event ω 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 τ*, denoted by $snp_\tau(l)$. Thus, for a given state sequence $\tau = W_0,\ldots,W_n$ we get the corresponding sequence of snapshots $snp(\tau) = \mathscr{S}_1,\ldots,\mathscr{S}_n$, where $\mathscr{S}_i = snp_\tau(i)$ for each $1 \leq i \leq n$, called the *snapshot sequence of τ*, and also called a *snapshot sequence of $\mathscr{A}$*.

Given a snapshot sequence $\rho = \mathscr{S}_1,\ldots,\mathscr{S}_n$ of a state sequence $\tau = W_0,\ldots,W_n$ there exists a natural sequence of partial functions $next_{\tau,1}, next_{\tau,2},\ldots,next_{\tau,n-1}$ transforming consecutive snapshots of ρ into their successor snapshots, where, for each $1 \leq k \leq n-1$, $next_{\tau,k} : \mathscr{S}_k \to \mathscr{S}_{k+1}$ is defined as follows. For $Q \in \mathscr{S}_k$ and $Q' \in \mathscr{S}_{k+1}$, $next_{\tau,k}(Q) = Q'$ if and only if Q, Q' are nonempty and there exists an event ω in τ such that Q is the module of ω in W_k and Q' is the module of ω 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 $next_{\tau,1},\ldots,next_{\tau,n-1}$ delineate all the events of τ as they are passing through the states of τ, but it does not explicitly indicate the "moment of death" (if an event dies). The sequence of functions $suc_{\tau,1},\ldots,suc_{\tau,n-1}$ 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 $\rho = \mathscr{S}_1,\ldots,\mathscr{S}_n$ we obtain its $\emptyset$-free version $\bar{\rho} = \bar{\mathscr{S}}_1,\ldots,\bar{\mathscr{S}}_n$, where for each $1 \leq i \leq n$, $\bar{\mathscr{S}}_i = \mathscr{S}_i - \{\emptyset\}$. Accordingly, each $next_{\tau,k}$ function is modified to the $rnext_{\tau,k}$ function which is $next_{\tau,k}$ restricted to $\bar{\mathscr{S}}_k$.

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

Definition 7 Let $\mathscr{L}$ be a family of sets, and let $\mathscr{F}_1,\mathscr{F}_2 \subseteq \mathscr{L}$ be nonempty.
(1) We say that $\mathscr{F}_1$ is *embedded in* $\mathscr{F}_2$ if $\bigcup\mathscr{F}_1 \subseteq \bigcap\mathscr{F}_2$.
(2) We say that $\mathscr{F}_1$ is *separated from* $\mathscr{F}_2$ *in* $\mathscr{L}$ if there exists $U \in \mathscr{L}$ such that $\bigcup\mathscr{F}_1 \subseteq U \subseteq \bigcap\mathscr{F}_2$.

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

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

An interactive process (hence a run) of an *ers* $\mathscr{A}$ produces a sequence ρ of snapshots $\mathscr{S}_1,\ldots,\mathscr{S}_k,\ldots,\mathscr{S}_n$. In general such a sequence may be very "unstable" because there may be no "mathematical similarity" between $\mathscr{S}_k$ and $\mathscr{S}_{k+1}$: remember that the context of the state W_k (in the state sequence $\tau = W_0,\ldots,W_n$ for which $\rho = snp(\tau)$) can "throw anything" into W_k and thus make

W_{k+1} and $\mathscr{S}_{k+1}$ "arbitrarily different" from W_k and $\mathscr{S}_k$ respectively. So we can talk about local stability (at W_k) only if there is a strong mathematical similarity between $\mathscr{S}_k$ and $\mathscr{S}_{k+1}$. Perhaps the most natural choice for such a strong similarity is to require that $rnext_{\tau,k}$ is an isomorphism between partial orders $(\bar{\mathscr{S}}_k, \subseteq)$ and $(\bar{\mathscr{S}}_{k+1}, \subseteq)$. When this happens, we get a local stability – it is local because, again, "anything can happen" to $\mathscr{S}_{k+2}$ (through the context of W_{k+1}). Hence we say that $(\mathscr{S}_k, \mathscr{S}_{k+1})$ is a *locally stable situation* if $rnext_k$ is an isomorphism between $(\bar{\mathscr{S}}_k, \subseteq)$ and $(\bar{\mathscr{S}}_{k+1}, \subseteq)$. We want to point out that the situation is quite subtle here, e.g., the fact that $\bar{\mathscr{S}}_k = \bar{\mathscr{S}}_{k+1}$ does not necessarily imply that $rnext_{\tau,k}$ is an isomorphism of $\bar{\mathscr{S}}_k$ onto $\bar{\mathscr{S}}_{k+1}$.

It turns out that under the local stability assumption snapshots posses an elegant mathematical structure.

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

7 Discussion

We have presented in this paper an 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 between (and motivation behind) 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-organizing systems which in stable situations produce well-structured families of products/modules are also presented.

Altogether this paper presents the basic setup of the framework of reaction systems and its

motivation as well as some research themes and results.

Acknowledgements: The authors are indebted to the referees for useful comments, to Robert Brijder and Hendrik Jan Hoogeboom for comments on this paper, and to Robert Brijder and Marloes van der Nat for their help in producing this paper.

Bibliography

[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.