Chapter 1

Reaction Systems: A Model of Computation Inspired by the Functioning of the Living Cell

Andrzej Ehrenfeucht
University of Colorado, Department of Computer Science
Boulder CO 80309-0347 USA

Ion Petre
Computational Biomodeling Laboratory
Åbo Akademi University and Turku Centre for Computer Science
Turku 20500, Finland

Grzegorz Rozenberg
Leiden Institute of Advanced Computer Science, Leiden University
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
and
University of Colorado, Department of Computer Science
Boulder CO 80309-0347 USA

Reaction systems are a model of computation inspired by the functioning of the living cell. They formalize the interactions between biochemical reactions that form the basic mechanism (the skeleton) underlying this functioning. This paper is a tutorial-style introduction to reaction systems – it introduces the basic notions, and reviews a number of research directions which are motivated either by biological considerations, or by the need to understand the basic formal processes (computations) underlying the dynamic behavior of reaction systems.

Contents

1.1. Introduction 2
1.2. Preliminaries 5
1.3. Basic Notions 5
1.4. Power Set Functions 9
1.5. Relationship to Transition Systems 12
1.6. Measurement Functions 15

1.7. Reaction Systems with Duration 19
1.8. A Reaction Systems Model for the Self-Assembly of Intermediate Filaments 21
1.9. A Reaction Systems Model for the Heat Shock Response 23
1.10. Discussion 27
References 29

1.1. Introduction

Natural Computing (see, e.g., [27, 37]) is concerned with models of computation, computational techniques, and computing technologies (referred to as *human-designed computing*) inspired by nature, as well as with investigating, in terms of information processing, phenomena taking place in nature (referred to as *computing taking place in nature*).

Well-known examples of the former strand of research include evolutionary computation (inspired by Darwinian evolution of species), neural computation (inspired by the functioning of the central nervous system and the brain), artificial immune systems (inspired by the natural immune system), quantum computing (inspired by quantum mechanics), and molecular computing (inspired by molecular biology). Examples of the latter strand include investigations into the computational nature of self-assembly, the computational nature of brain processes, the computational nature of developmental processes, and the computational nature of biochemical reactions.

A research line which has attracted a lot of attention in natural computing is the functioning of the living cell. It is a central research topic for biology and biochemistry, while at the same time it is very attractive for computer science because, for example, it leads to novel models of computation.

This paper belongs to this research line. The methodology underlying the model presented in this paper is to first propose the basic mechanism underlying the functioning of the living cell, and then to attempt (first) to understand this mechanism only. It consists of the interactions between biochemical reactions taking place in the living cell. These interactions form the bare skeleton of the functioning of the living cell. It will still require many additional levels (on top of the skeleton) to achieve a basic understanding of how the living cell functions.

A key property of interactions between biochemical reactions is that they are based on two mechanisms: *facilitation* and *inhibition*. The product of one reaction may contain reactants of some reactions (and hence facilitate

these reactions) and it may contain inhibitors of some reactions (and hence inhibit these reactions).

The model of *reaction systems* (introduced in [19]; also see, e.g., [6, 14, 15]) formalizes biochemical reactions in such a way that dynamic processes taking place in reaction systems formalize these interactions. Moreover, because this model is concerned with interactions taking place *within the living cell*, the formal notion of a dynamic process in reaction systems also captures the interactions with an environment, reflecting the fact that the living cell is an open system.

This paper is a tutorial-style introduction to reaction systems. It introduces basic notions together with the underlying intuition and motivation, and then it presents a number of representative research directions. The paper is organized as follows.

In Section 1.3 we first introduce the basic (formal) notion of a *reaction* and define its effect on the current state of a (biochemical) system. Each reaction b is of the form $b = (R_b, I_b, P_b)$, where R_b are all *reactants* b needs to take place, I_b is the set of all *inhibitors* of b (if any of them is present in the current state T, then b will not take place in T), and P_b is the *product* set of b (if b takes place in the current state T, then it contributes its product set P_b to the successor state of T). Then we present *reaction systems* as a model of interactions between biochemical reactions in the living cell, where the dynamic processes induced by such interactions (also involving interactions with the environment) are formalized as so-called *interactive processes*.

From a formal, mathematical point of view, a reaction system is an implementation (through the cumulative effect of all reactions of the system) of a function from the set of states (of the system) into itself. This point of view is explored in Section 1.4, where the focus is on the understanding of minimal implementations, i.e., implementations through reaction systems which use reactions with the minimal number of reactants or reactions with the minimal number of inhibitors.

Although the model of reaction systems was inspired by biology, the topics guiding research on reaction systems are motivated by both biological motivations and by the need to understand the underlying computations (the dynamic processes taking place in reaction systems). In fact, reaction systems turns out to be a novel model of computation. From this point of view it is important to understand the relationship of reaction systems to other models of computation. In Section 1.5 we discuss the relationship of

reaction systems to finite transition systems, one of the classic models of computation.

The basic model of reaction systems abstracts from many technical (numerical) properties of biochemical reactions, so that it is a qualitative, rather than a quantitative, model. Consequently, there is no counting in reaction systems (the basic data structure used here are sets rather than multisets). However, it is very common in biology to assign quantitative parameters to states (e.g., the numbers of specific kinds of molecules present in the given state). To account for this, the basic model of reaction systems is equipped with a finite number of the so-called *measurement functions*, where each such function measures a specific (numerical) property, i.e., it assigns numerical values (real numbers) to the states of a reaction system. The so-obtained extensions of reaction systems are called *reaction systems with measurements* and they are discussed in Section 1.6. Of special interest in modeling biological systems (the living cell) is the case when the values of measurements (the values given by the measurement functions) for the given state influence the identity of the successor state. Although this case seems to be very quantitative, we demonstrate that it can be implemented (modeled) by ordinary (qualitative) reaction systems.

One of the basic properties of reaction systems is *non-permanency*: an entity x present in the current state T of the reaction system will *vanish* (i.e., it will *not* be present in the successor state T') unless it is produced by a reaction of the system (which is enabled/active in T) or it is "thrown into" T' by the environment. The non-permanency property reflects the basic bioenergetics of the living cell. This vanishing in reaction systems is instantaneous (x is present in T but not present in the successor state T'). However biochemical entities do not vanish instantly, but rather decay within a certain time interval. To account for this decay, reaction systems are extended by duration functions which assign a decay time (a positive integer) to each entity of a reaction system. The so-obtained *reaction systems with durations* are discussed in Section 1.7. In particular, we demonstrate that the phenomenon of durations/decay may be implemented (modeled) by ordinary reaction systems.

In Sections 1.8 and 1.9 we present two examples of modeling of bioprocesses by reaction systems. More specifically, we discuss reaction systems modeling of the self-assembly of intermediate filaments and of the eukaryotic heat shock response, respectively. In both models we show that dynamic behavior typically explained in standard modeling frameworks through a numerical interplay driven by kinetic rate constants may also

be explained qualitatively through intricate sequences of facilitation and inhibition between biochemical reactions.

Finally, in the last section we review topics covered in the paper and then provide a brief review of some research topics not covered here.

1.2. Preliminaries

Throughout the paper we use standard mathematical terminology and notation. In particular:

(1) We use $\mathbb{Z}^+$ and $\mathbb{R}$ to denote the sets of positive integers and real numbers, respectively.
(2) The empty set is denoted by $\emptyset$. For a set X, its cardinality is denoted by $|X|$ and 2^X denotes the power set of X. For sets X and Y, $X - Y$, $X \cup Y$, and $X \cap Y$ denote set difference, set union, and set intersection, respectively, while $X \subseteq Y$ denotes set inclusion. For a family $\mathcal{L}$ of sets, $\bigcup \mathcal{L}$ denotes the union of sets from $\mathcal{L}$.
(3) For a sequence of sets $\zeta = Z_0, \ldots, Z_n$ and a set Q, the *Q-projection* of ζ, denoted $\mathsf{proj}_Q(\zeta)$, is the sequence of sets $Z_0 \cap Q, Z_1 \cap Q, \ldots, Z_n \cap Q$.

1.3. Basic Notions

The basic intuition behind a biochemical reaction is that its functioning is based on two mechanisms, facilitation and inhibition, with the former represented by the reactants and the latter represented by the inhibitors. Accordingly, the formal notion of a reaction is defined as follows.

Definition 1. *A* reaction *is a triplet $b = (R, I, P)$ such that R, I, P are finite nonempty sets with $R \cap I = \emptyset$.*

The sets R, I, and P are called the *reactant set of* b, the *inhibitor set of* b, and the *product set of* b, respectively. They are also written as R_b, I_b, and P_b. The set $M_b = R_b \cup I_b$ is the *set of resources of* b. If S is a set such that $R_b, I_b, P_b \subseteq S$, then b is a *reaction in* S. The set of all reactions in S is denoted by $\mathsf{rac}(S)$.

A biochemical reaction can take place (in a given biochemical state/ environment) if all of its reactants are present and none of its inhibitors is present; moreover, if it takes place then it produces its product. This intuition underlies the following formal definition.

Definition 2. *Let S be a finite set and $T \subseteq S$. A reaction $b \in \mathsf{rac}(S)$ is* enabled by T, *denoted by* $\mathsf{en}_b(T)$, *if and only if $R_b \subseteq T$ and $I_b \cap T = \emptyset$. The* result of b on T, *denoted by* $\mathsf{res}_b(T)$, *is defined by:* $\mathsf{res}_b(T) = P_b$ *if* $\mathsf{en}_b(T)$, *and* $\mathsf{res}_b(T) = \emptyset$ *otherwise.*

A set T formalizes here a state of a biochemical system which is simply a set of currently present biochemical entities. The first part of the above definition says indeed that b can take place in a state T (b is enabled by T) if *all* reactants from R_b are present in T and *none* of the inhibitors from I_b is present in T; hence T separates R_b from I_b. The second part of the definition says that if b is enabled by T, then the result of b on T is the product P_b of b; however if b is not enabled by T, then it will produce "nothing" ($\emptyset$) on T. We will consider finite sequences of states where each state (except for the final state) has a successor. Then $\mathsf{res}_b(T) = P_b$ means that b will contribute P_b to the successor of T, while $\mathsf{res}_b(T) = \emptyset$ means that b will not contribute to the successor of T.

In considering the functioning of a living cell we are interested in the effects (on the current state) of *sets* of reactions taking place. This is formalized as follows.

Definition 3. *Let S be a finite set and $T \subseteq S$. The* result of *a set of reactions $B \subseteq \mathsf{rac}(S)$* on T, *denoted by* $\mathsf{res}_B(T)$, *is defined by* $\mathsf{res}_B(T) = \bigcup\{\mathsf{res}_b(T) : b \in B\}$.

Thus the result of a set of reactions B is cumulative: it is the union of results of all individual reactions from B. Since $\mathsf{res}_b(T) = \emptyset$ if b is not enabled by T, $\mathsf{res}_B(T)$ can be obtained by considering only the results of all reactions enabled by T. Consequently, $\mathsf{res}_B(T) = \bigcup\{\mathsf{res}_b(T) : b \in B \text{ and } \mathsf{en}_b(T)\}$.

Note that an entity $x \in \mathsf{res}_B(T)$ if and only if $x \in P_b$ for a reaction b enabled by T. For $x \in T$ we say that x is *sustained by* B in T if $x \in \mathsf{res}_B(T)$. Thus x is sustained by B in T if and only if x is produced by a reaction from B enabled by T. This holds even if $x \notin \bigcup\{R_b : b \in B\}$, i.e., x is not "touched" (not processed) by any reaction in B. This is very different from standard models of computation in computer science, where typically if an element from the current state is not processed, then it also belongs to the successor state. This implies *non-permanency* in reaction systems: in the transition from the current state T to its successor $\mathsf{res}_B(T)$ an entity from T vanishes unless it is sustained by a reaction from B. This non-permanency property reflects the basic bioenergetics of the

living cell: without the supply of energy living cells disintegrate. However, the absorption of energy by the living cell is achieved through biochemical reactions (see [30]).

If b, c are two reactions from B enabled by T, then both $P_b \subseteq \mathsf{res}_B(T)$ and $P_c \subseteq \mathsf{res}_B(T)$ even if $R_b \cap R_c \neq \emptyset$. This means that there is no conflict of resources: both b and c use $R_b \cap R_c$ to produce their products. Thus a nonempty intersection of sets of reactants of enabled reactions does not constitute a conflict. This is an important difference with standard models of concurrent systems such as, e.g., Petri nets (see, e.g., [36]), and it is implied by our *threshold assumption*: either a resource is present and then it is present in a sufficient amount, or it is not present. This reflects the level of abstraction that we have adopted in our basic model of reaction systems – we do not count concentrations/amounts of entities to decide which reactions are enabled in a given state. Our model is formulated on a higher level of abstraction. Thus there is no counting in reaction systems and we deal with sets rather than multisets. Our model is a qualitative (rather than a quantitative) model.

With the notion of a reaction (and its effects on the current state) defined, we are ready to recall the formal notion of a reaction system which was originally proposed as a formal model of (the "skeleton" of) the functioning of the living cell.

Definition 4. *A* reaction system*, abbreviated* rs*, is an ordered pair* $\mathcal{A} = (S, A)$*, where* S *is a finite nonemtpy set and* $A \subseteq \mathsf{rac}(S)$*.*

The set A is called the *set of reactions of* $\mathcal{A}$. Since S is finite, so is A. The set S is called the *background set of* $\mathcal{A}$ and its elements are called *entities*; they represent molecular entities present in the states of biochemical systems.

The subsets of S are called the *states of* $\mathcal{A}$. Thus, all states of $\mathcal{A}$ are finite with the cardinality of each state limited by an a priori fixed number, viz., $|S|$.

For a state T of $\mathcal{A}$, the *result of* $\mathcal{A}$ on T, denoted by $\mathsf{res}_{\mathcal{A}}(T)$, is defined by $\mathsf{res}_{\mathcal{A}}(T) = \mathsf{res}_A(T)$. We will refer to $\mathsf{res}_{\mathcal{A}}$ as the *result function of* $\mathcal{A}$. In fact, $\mathsf{res}_{\mathcal{A}}$ is a function from 2^S into itself.

The dynamic behavior of a reaction system is expressed through interactive processes that take place within it. They are defined as follows.

Definition 5. *Let* $\mathcal{A} = (S, A)$ *be an* rs *and let* $n \geq 1$ *be an integer. An* (n-step) interactive process *in* $\mathcal{A}$ *is a pair* $\pi = (\gamma, \delta)$ *of finite sequences such*

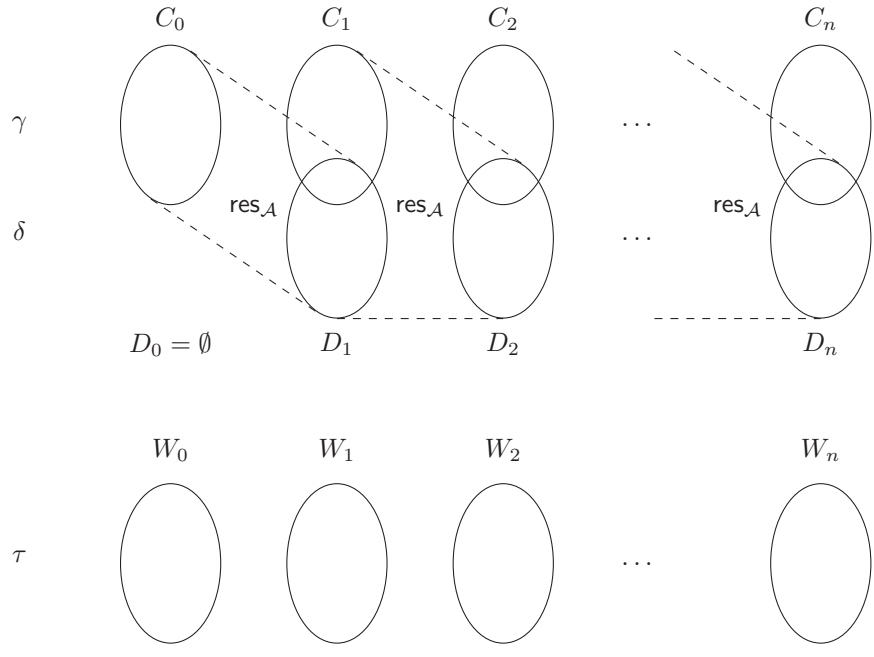

Fig. 1.1. An interactive process.

that $\gamma = C_0, \ldots, C_n$ *and* $\delta = D_0, \ldots, D_n$, *where* $C_0, \ldots, C_n, D_0, \ldots, D_n \subseteq S$, $D_0 = \emptyset$, *and* $D_i = \mathsf{res}_{\mathcal{A}}(D_{i-1} \cup C_{i-1})$ *for all* $i \in \{1, \ldots, n\}$.

The sequence γ is the *context sequence* of π, the sequence δ is the *result sequence* of π, and the sequence $\tau = W_0, \ldots, W_n$ such that $W_0 = C_0$ and $W_i = C_i \cup D_i$ for all $i \in \{1, \ldots, n\}$ is the *state sequence* of π, where W_0 is called the *initial state of* π.

The interactive process π begins in the initial state W_0, and the reactions from A which are enabled by W_0 produce result D_1, which together with context C_1, forms state $W_1 = D_1 \cup C_1$. The state sequence τ is formed by the iteration of this procedure: for each state W_i with $i \in \{0, \ldots, n-1\}$ its *successor* is $W_{i+1} = D_{i+1} \cup C_{i+1}$, where $D_{i+1} = \mathsf{res}_{\mathcal{A}}(W_i)$. This is illustrated in Figure 1.1.

The context sequence γ formalizes the fact that *the living cell is an open system* in the sense that it interacts with its environment (i.e., with the rest of the bigger system).

The first element C_0 of the context sequence is just the initial state and so the rest of the sequence $C_1, \ldots, C_n$ is called the *proper context*

sequence. Note that if the current proper context set C_i is such that $C_i \subseteq D_i$, then this context does not influence the current state W_i (all entities from C_i are already contributed by the reactions of the system, because $D_i = \mathsf{res}_{\mathcal{A}}(D_{i-1})$). Accordingly, the sequence $C_1 - D_1, \ldots, C_n - D_n$ is called the *significant context sequence*. In fact, for computing the result sequence and the state sequence, one can replace the proper context sequence by the significant context sequence.

If the significant context sequence consists of empty sets only, then π is *context-independent*. In this case, whatever the environment adds to the current state W_i is already included in the result D_i, hence already produced by $\mathsf{res}_{\mathcal{A}}$ from the previous state. Hence, if π is context-independent, then it is determined by its initial state $W_0 = C_0$ and the number of steps n.

Note that if π is context-independent and

$$\tau = W_0, \ldots, W_i, W_{i+1}, \ldots, W_n$$

is its state sequence, then during the transition of W_i to W_{i+1} all entities from $W_i - \mathsf{res}_{\mathcal{A}}(W_i)$ vanish (they are not present in W_{i+1}). This reflects the non-permanency property of reaction systems discussed above. For a general (not necessarily context-independent) interactive process, the non-permanency property says that in transition from W_i to W_{i+1} an entity $x \in W_i$ will vanish (will not be present in W_{i+1}) unless it is produced by the system ($x \in \mathsf{res}_{\mathcal{A}}(W_i)$) or it is thrown in by the context ($x \in C_{i+1}$).

We will use $\mathsf{IP}(\mathcal{A})$ to denote the set of all interactive processes of $\mathcal{A}$, $\mathsf{STS}(\mathcal{A})$ to denote the set of all state sequences of all interactive processes of $\mathcal{A}$, and $\mathsf{CISTS}(\mathcal{A})$ to denote the set of all state sequences of all context-independent interactive processes of $\mathcal{A}$.

1.4. Power Set Functions

The state sequence τ of an interactive process $\pi = (\gamma, \delta)$ is formed as a combination of an *internal* effect (given by the result function $\mathsf{res}_{\mathcal{A}}$) and an *external* effect (given by the context sequence γ). In the case of context-independent processes, τ is formed by $\mathsf{res}_{\mathcal{A}}$ only. In either case the understanding of $\mathsf{res}_{\mathcal{A}}$ is crucial for the understanding of the dynamics of reaction systems. Since states of a reaction system are sets, the result function is a power set function which is defined as follows.

Definition 6. *Let S be a finite set. A function $f : 2^S \to 2^S$ is called a* power set function (over S).

In the context of reaction systems, we are interested in power set functions that are implementable by reaction systems. They are defined as follows.

Definition 7. *A power set function f over a set S is a* reaction system power set function*, abbreviated a* rs power set function*, if there exists a reaction system $\mathcal{A} = (S, A)$ such that $f = \mathsf{res}_{\mathcal{A}}$.*

We note that for each reaction system $\mathcal{A} = (S, A)$ we have $\mathsf{res}_{\mathcal{A}}(S) = \emptyset$ (because for each $b \in A$, I_b is a nonempty subset of S) and $\mathsf{res}_{\mathcal{A}}(\emptyset) = \emptyset$ (because for each $b \in A$, $R_b \neq \emptyset$, and so no $b \in A$ is enabled by $\emptyset$). Since we are interested in rs power set functions, we have to restrict ourselves to power set functions satisfying the above two conditions.

Definition 8. *A power set function over a set S is* boundary *if $f(S) = f(\emptyset) = \emptyset$.*

It turns out that reaction systems are powerful implementations of power set functions: almost all of them are rs power set functions.

Theorem 1 ([19])**.** Let S be a finite set and let f be a power set function over S. Then f is an rs power set function if and only if f is a boundary power set function.

An important research topic concerning reaction systems is the power of resources. In a general definition of an rs $\mathcal{A}$ one does not restrict either the cardinalities of the sets of reactants or the cardinalities of the sets of inhibitors of reactions of $\mathcal{A}$. One way to understand the role/power of resources is to consider reactions with the minimal number of reactants and reactions with the minimal number of inhibitors. This leads to the following definition.

Definition 9. *Let $\mathcal{A} = (S, A)$ be an* rs*.*

(i) $\mathcal{A}$ is reactant-minimal *if $|R_a| = 1$ for each $a \in A$.*
(ii) $\mathcal{A}$ is inhibitor-minimal *if $|I_a| = 1$ for each $a \in A$.*
(iii) $\mathcal{A}$ is resource-minimal *if $|M_a| = 2$ for each $a \in A$.*

The above definition translates naturally into the following definition for rs power set functions.

Definition 10. *Let f be an* rs *power set function.*

(1) f is reactant-minimal *if there exists a reactant-minimal* rs $\mathcal{A}$ *such that* $f = \mathsf{res}_{\mathcal{A}}$.
(2) f is inhibitor-minimal *if there exists an inhibitor-minimal* rs $\mathcal{A}$ *such that* $f = \mathsf{res}_{\mathcal{A}}$.
(3) f is resource-minimal *if there exists a resource-minimal* rs $\mathcal{A}$ *such that* $f = \mathsf{res}_{\mathcal{A}}$.

We will state now a characterization (from [13]) of reactant-minimal, inhibitor-minimal, and resource-minimal rs power set functions. First we need an auxiliary definition.

Definition 11. *Let S be a finite set and let f be a power set function over S.*

(1) f is union-subadditive *if $f(X \cup Y) \subseteq f(X) \cup f(Y)$ for all $X, Y \subseteq S$.*
(2) f is intersection-subadditive *if $f(X \cap Y) \subseteq f(X) \cup f(Y)$ for all $X, Y \subseteq S$.*

The basic intuition behind the notion of a union-subadditive function f is that of a set-theoretical union related constraint in defining f on 2^S. For all subsets X, Y of S, once the values of f on X and on Y (hence $f(X)$ and $f(Y)$) are defined, the value of f on the union $X \cup Y$ (hence $f(X \cup Y)$) must be a subset of $f(X) \cup f(Y)$. In other words, each element of S included in $f(X \cup Y)$ is already included in either $f(X)$ or in $f(Y)$. Similarly, if f is intersection-subadditive, then each element of S included in $f(X \cap Y)$ is already included in either $f(X)$ or in $f(Y)$.

Example 1. Let $S = \{x, y, z, u\}$.

(1) Let $f_1 : 2^S \to 2^S$ be such that $f_1(\{x\}) = \{x, y\}$, $f_1(\{y\}) = \{x\}$, and $f_1(\{x, y\}) = \{z\}$. Then f_1 is *not* union-subadditive, as $z \in f_1(\{x\} \cup \{y\}) = f_1(\{x, y\})$, while $z \notin f_1(\{x\})$ and $z \notin f_1(\{y\})$.
(2) Let $f_2 : 2^S \to 2^S$ be such that $f_2(\{u, z\}) = \{u, z\}$, $f_2(\{u, y\}) = \{u\}$, and $f_2(\{u\}) = \{y\}$. Then f_2 is *not* intersection-subadditive, as $y \in f_2(\{u, z\} \cap \{u, y\}) = f_2(\{u\})$, while $y \notin f_2(\{u, z\})$ and $y \notin f_2(\{u, y\})$.

Theorem 2 ([13])**.** Let f be an rs power set function. Then

(1) f is reactant-minimal if and only if f is union-subadditive.
(2) f is inhibitor-minimal if and only if f is intersection-subadditive.
(3) f is resource-minimal if and only if f is both union-subadditive and intersection-subadditive.

Corollary 3 ([13])**.** Let f be an rs power set function. If f is reactant-minimal and intersection-minimal, then f is resource-minimal.

The research on minimal systems is by now rich and broader than what we could cover in this section. Typical research topics in this area include simulation of state sequences [44]; functions defined by minimal reaction systems [42]; computational complexity of various problems in such systems [41, 25]; normal forms [33]; and connections to Boolean lattices [32].

1.5. Relationship to Transition Systems

A standard (and natural) research problem about models of computation is concerned with their computational power, and this problem is often resolved by relating the given model to already existing, preferably "traditional", models of computation. In this section we demonstrate a close relationship between reaction systems and finite transition systems (see, e.g., [1]) by showing how to simulate finite transition systems by reaction systems and how to simulate reaction systems by finite transition systems.

(I) Let's begin with finite transition systems. For didactic reasons we will first consider *deterministic* finite transition systems.

Recall that a deterministic finite transition system is a 3-tuple $\mathcal{T} = (Q, \Sigma, \delta)$, where Q is a finite nonempty set of *states*, Σ is a finite nonempty set of *symbols* (the *input alphabet*), and $\delta : Q \times \Sigma \to Q$ is the *state transition function* (in general, δ is a partial function).

For $n \geq 1$, an n-step *transition process* in $\mathcal{T}$ is a pair $\mu = (\rho, \alpha)$, where $\rho = q_0, \ldots, q_n$ is a sequence of *states*, and $\alpha = x_1, \ldots, x_n$ is a sequence of *symbols* such that $\delta(q_{i-1}, x_i) = q_i$ for each $i \in \{1, \ldots, n\}$.

We will now construct (see [6]) an rs $\mathcal{A}_\mathcal{T} = (S_\mathcal{T}, A_\mathcal{T})$, which simulates $\mathcal{T}$ in the sense that transition processes of $\mathcal{T}$ will be in one-to-one correspondence with certain kinds of interactive processes of $\mathcal{A}_\mathcal{T}$. To simplify the construction we will assume that $Q \cap \Sigma = \emptyset$ and $|Q \cup \Sigma| > 2$. More specifically, $\mu = (\rho, \alpha)$ with $\rho = q_0, \ldots, q_n$ and $\alpha = x_1, \ldots, x_n$ is a transition process in $\mathcal{T}$ if and only if $\pi_\mu = (\gamma, \delta)$ is an interactive process in $\mathcal{A}_\mathcal{T}$, where $\gamma = (\{q_0, x_1\}, \{x_2\}, \ldots, \{x_n\}, \emptyset)$ and $\delta = (\emptyset, \{q_1\}, \ldots, \{q_n\})$. The state sequence of π is then $W = \{q_0, x_1\}, \{q_1, x_2\}, \ldots, \{q_n\}$. Thus the initial state of π is $W_0 = \{q_0, x_1\}$ corresponding to $\mathcal{T}$ reading symbol x_1 in state q_0, the successor state of W_0 is $W_1 = \{q_1, x_2\}$, which corresponds to $\mathcal{T}$ reading symbol x_2 in state $q_1 = \delta(q_0, x_1), \ldots$, and finally the last state $W_n = \{q_n\}$ is the state of $\mathcal{T}$ after reading the last symbol x_n of α in state q_{n-1}.

The $\mathsf{rs}\ \mathcal{A}_{\mathcal{T}} = (S_{\mathcal{T}}, A_{\mathcal{T}})$, where $S_{\mathcal{T}} = Q \cup \Sigma$ and

$$A_{\mathcal{T}} = \{(\{q,x\}, S_{\mathcal{T}} - \{q,x\}, \{\delta(q,x)\}) : q \in Q, x \in \Sigma, \text{ and } \delta(q,x) \text{ is defined}\}.$$

It is easily seen that indeed the so-constructed $\mathsf{rs}\ \mathcal{A}_{\mathcal{T}}$ simulates the given transition system $\mathcal{T}$ in the sense explained above. Also, this simulation is "natural", as the prime intuition for the dynamic behavior of a transition system is that it is currently in a specific state q (an *internal* parameter of the transition system) and the current symbol x to be read is provided *externally* (by the "outside world"). Then, for the provided symbol x, $\mathcal{T}$ computes internally (by its transition function δ) the successor state $\delta(q,x)$. Hence the behavior of $\mathcal{T}$ results from the interaction of its internal structure (function δ) and the external environment (providing the sequence of input symbols). It is exactly this interaction that is simulated by the $\mathsf{rs}\ \mathcal{A}_{\mathcal{T}}$.

For the above construction we have assumed that $|Q \cup \Sigma| > 2$, as otherwise we would get reactions with empty inhibitor sets. Then, however, we could use an additional entity serving only as a "formal inhibitor" for reactions with the empty inhibitor sets. We also required that $Q \cap \Sigma = \emptyset$. If this is not the case then we could, e.g., use a coding which "primes" the states from Q creating Q' disjoint with Σ. Then the correspondence between $\mathcal{T}$ and (the modified) $\mathcal{A}_{\mathcal{T}}$ is expressed through this coding.

Above, we have considered *deterministic* finite transition systems. For a general (possibly nondeterministic) finite transition system $\mathcal{T} = (Q, \Sigma, \delta)$, $\delta \subseteq Q \times \Sigma \times Q$ is a relation, meaning that for a given state q and a given symbol x, if $\delta(q,x)$ is defined, then one may have several states $p_1, \ldots, p_m$ such that $(q,x,p_1), \ldots, (q,x,p_m) \in \delta$: reading x in state q may nondeterministically lead to any of the states $p_1, \ldots, p_m$.

This nondeterminism in transitions can be simulated by reaction systems through the use of context sequences as follows. First, for each $q \in Q$ and $x \in \Sigma$ such that $\delta(q,x)$ is defined, let $m_{q,x}$ be the number of different transitions in δ of the form (q,x,p) where $p \in Q$. This is the *degree of nondeterminism* for x in q. Then let m_δ be the maximal number among all $m_{q,x}$; this is the *degree of nondeterminism* of δ. Then, for each such $q \in Q$ and $x \in \Sigma$, we fix an (arbitrary) order, $\mathsf{ord}(q,x)$, of all transitions of the form (q,x,p) from δ. Now we modify the construction of $\mathcal{A}_{\mathcal{T}}$ to $\mathcal{A}'_{\mathcal{T}}$ in such a way that each nonempty context set provides not a symbol $x \in \Sigma$, but rather a symbol of the form (x,i), where $i \in \{1, \ldots, m_\delta\}$. Consequently, in an interactive process π'_μ, an intermediate state W_k will be of the form $W_k = \{q_k, (x_{k+1}, i)\}$ (where $C_k = \{(x_{k+1}, i)\}$ and $D_k = \{q_k\}$) and so $D_{k+1} = \{p\}$, where if $i \leq m_{q_k, x_{k+1}}$, then (q_k, x_{k+1}, p) is the i'th

transition of $\mathsf{ord}(q_k, x_{k+1})$ and if $i > m_{q_k,x_{k+1}}$, then (q_k, x_{k+1}, p) is the last transition of $\mathsf{ord}(q_k, x_{k+1})$.

Thus nondeterministic finite transition systems can also be simulated by reaction systems in a "natural way", meaning that both the choice of the current input symbol and the choice of the transition to be applied are taken care of by the context sequence (while the computation of the consecutive result sets is taken care of by the result sequence).

(II) Now we demonstrate how to simulate reaction systems by finite transition systems. Let $\mathcal{A} = (S, A)$ be an rs. Let $\mathcal{T}_\mathcal{A} = (Q, \Sigma, \delta)$ be a finite transition system such that $Q = 2^S$, $\Sigma = 2^S$, and let $\delta : Q \times \Sigma \to Q$ be defined by: for all $U, X, T \subseteq S$, $\delta(U, X) = T$ if and only if $T = \mathsf{res}_\mathcal{A}(U) \cup X$, where $X \cap \mathsf{res}_\mathcal{A}(U) = \emptyset$.

To see that transition processes in $\mathcal{T}_\mathcal{A}$ simulate interactive processes in $\mathcal{A}$, recall that when $\pi = (\gamma, \delta)$ is an interactive process with $\gamma = C_0, \ldots, C_n$, $\delta = D_0, \ldots, D_n$, and $\tau = W_0, \ldots, W_n$, then, for each $i \in \{0, \ldots, n-1\}$, $D_{i+1} = \mathsf{res}_\mathcal{A}(W_i)$ and $W_{i+1} = \mathsf{res}_\mathcal{A}(W_i) \cup X_{i+1}$, where $X_1, \ldots, X_n$ is the significant context sequence of π. Moreover, any subset of S appears as the context set on position $i + 1$ in some interactive process of $\mathcal{A}$. These two observations explain the above definition of the transition function δ of the transition system $\mathcal{T}_\mathcal{A}$ simulating $\mathcal{A}$.

It follows directly from the definition of $\mathcal{T}_\mathcal{A}$ that the precise relationship between the interactive processes in $\mathcal{A}$ and the transition processes in $\mathcal{T}_\mathcal{A}$ (hence the definition of the way that $\mathcal{T}_\mathcal{A}$ simulates $\mathcal{A}$) is as follows.

(1) Let $\pi = (\gamma, \delta)$ be an interactive process in $\mathcal{A}$ with $\gamma = C_0, C_1, \ldots, C_n$ and $\delta = D_0, \ldots, D_n$ for some $n \geq 1$, let $\eta = X_1, \ldots, X_n$ be the significant context sequence of π, and let $\tau = W_0, \ldots, W_n$ be the state sequence of π. Then $\mu = (\tau, \eta)$ is a transition process in $\mathcal{T}_\mathcal{A}$.
(2) Let $\mu = (\rho, \alpha)$ be a transition process in $\mathcal{T}_\mathcal{A}$ with $\rho = q_0, \ldots, q_n$ and $\alpha = x_1, \ldots, x_n$ for some $n \geq 1$. Then $\pi = (\gamma, \delta)$ such that $\gamma = C_0, C_1, \ldots, C_n$ and $\delta = D_0, D_1, \ldots, D_n$, where $C_0 = q_0, C_1 = x_1, \ldots, C_n = x_n$, $D_0 = \emptyset$, and $D_i = q_i - x_i$ for each $i \in \{1, \ldots, n\}$, is an interactive process in $\mathcal{A}$.

We end this section by observing that one can represent state sequences of $\mathcal{A}$ by finite *unlabeled* transition systems. Such a system is of the form $\mathcal{U} = (Q, \mu)$ where $\mu \subseteq Q \times Q$. Here, for states $q, q' \in Q$, q' is a *successor of* q if and only if $(q, q') \in \mu$. The transition from q to q' is unlabeled, i.e., there is no symbol marking/labeling/causing this transition. As a matter

of fact, there is no set of symbols (Σ) here (which is the case for finite transition systems).

For $n \geq 1$, an *n-step transition process* in $\mathcal{U}$ is a sequence of states $q_0, q_1, \ldots, q_n$ such that, for each $i \in \{0, \ldots, n-1\}$, $(q_i, q_{i+1}) \in \mu$.

For a given rs $\mathcal{A} = (S, A)$, as above, let $\mathcal{U}_\mathcal{A} = (Q, \mu)$ be the finite unlabeled transition system such that $Q = 2^S$ and $\mu \subseteq Q \times Q$ is defined by: for all $T, Z \subseteq S$, $(T, Z) \in \mu$ if and only if $\mathsf{res}_\mathcal{A}(T) \subseteq Z$. We note that

(1) It follows directly from the definition of the state sequence τ of an interactive process in $\mathcal{A}$ that, if W, W' are two consecutive states of τ, then $(W, W') \in \mu$.

(2) Since for each ordered pair (C, C') with $C, C' \subseteq S$ there is an interactive process π for which C, C' are two consecutive context sets of the context sequence of π, it follows that for each $(q, q') \in \mu$ there exists an interactive process of $\mathcal{A}$ such that q, q' are two consecutive states of the state sequence of this process.

It follows from (1) and (2) that $\mathcal{U}_\mathcal{A}$ simulates the state sequences of $\mathcal{A}$. More precisely, for $n \geq 1$, a sequence $\tau = q_0, q_1, \ldots, q_n$ is an n-step transition process in $\mathcal{U}_\mathcal{A}$ if and only if there exists an interactive process π in $\mathcal{A}$ such that τ is the state sequence of π.

Note that the unlabeled transition system $\mathcal{U}_\mathcal{A}$ simulates *only* the state sequences of interactive processes in $\mathcal{A}$, while the transition system $\mathcal{T}_\mathcal{A}$ discussed above simulates (through the use of symbols from Σ as labels of transitions) interactive processes in $\mathcal{A}$ *together* with their state sequences. Clearly, one can obtain the *graph* of $\mathcal{U}_\mathcal{A}$ (where 2^S is the set of nodes and there is a directed edge from a node T to a node Z if and only if $(T, Z) \in \mu$) from the *labeled graph* of $\mathcal{T}_\mathcal{A}$ (where 2^S is the set of nodes and there is a directed edge labeled by X from a node T to a node Z if and only if $\delta(T, X) = Z$) by simply removing the labels of edges.

For an rs $\mathcal{A}$, the (graph of the) unlabeled transition system $\mathcal{U}_\mathcal{A}$ provides the basic structure of transitions between states of $\mathcal{A}$. In particular it illustrates the "inclusion feature" of these transitions very well: if a state Z is a successor of a state T (i.e., (T, Z) is a state sequence of $\mathcal{A}$) and state Z is a subset of a state W, then W is also a successor of T.

1.6. Measurement Functions

As discussed already, the basic model of reaction systems satisfies the threshold assumption, and so it is a qualitative model. On the other hand,

it is a common practice in biology (reflected in various formal models) to assign quantitative/numerical parameters to states. To account for this, the basic model of reaction systems was extended in [20] to reaction systems with measurements.

The basic idea here is that a numerical value assigned to a state reflects a measurement performed on this state, where the intuitive notion of a measurement is formalized through the formal notion of a measurement function, which assigns a real number to each state of a reaction system.

Definition 12. *Let $\mathcal{A} = (S, A)$ be an* rs*. A* measurement function for $\mathcal{A}$ *is a function $f : 2^S \to \mathbb{R}$ such that, for all $X, Y \in 2^S$ with $X \cap Y = \emptyset$, $f(X \cup Y) = f(X) + f(Y)$.*

Note that the *additive property* of f required in the above definition implies that $f(\emptyset) = 0$. The additive property is natural here, as states of an rs are abstract sets (in general, we do not know the nature of the entities from the background set), while the value of a measurement function for a set Z should be obtainable from the values of this function for the singleton sets containing individual elements of Z.

We are ready now to extend the notion of an rs by a finite set of measurement functions (with each of them assigning a numerical parameter to each state of an rs).

Definition 13. *A* reaction system with measurements*, abbreviated* rsm*, is a triplet $\mathcal{A} = (S, A, F)$, where (S, A) is an* rs *and F is a finite set of measurement functions for (S, A).*

The rs (S, A) is the *underlying reaction system of* $\mathcal{A}$, denoted by $\mathsf{und}(\mathcal{A})$. The *result function of* $\mathcal{A}$ is simply the result function of $\mathsf{und}(\mathcal{A})$; hence the dynamics of $\mathcal{A}$ are determined by the dynamics of $\mathsf{und}(\mathcal{A})$. In particular, interactive processes of $\mathcal{A}$ are just the interactive processes of $\mathsf{und}(\mathcal{A})$. The measurement functions from F can be seen as annotations of the state sequences of $\mathcal{A}$. These annotations are global (in general, for a measurement function f and a state T, the value $f(T)$ depends on the whole state T) and they are computed externally (outside of $\mathsf{und}(\mathcal{A})$), i.e., not given as parts of the states of state sequences of interactive processes of $\mathsf{und}(\mathcal{A})$. This is illustrated in Figure 1.2 for a $\mathsf{rsm}\,\mathcal{A} = (S, A, F)$, where $F = \{f_1, \ldots, f_n\}$.

All of the notation and terminology of reaction systems carry over to reaction systems with measurements through their underlying reaction systems.

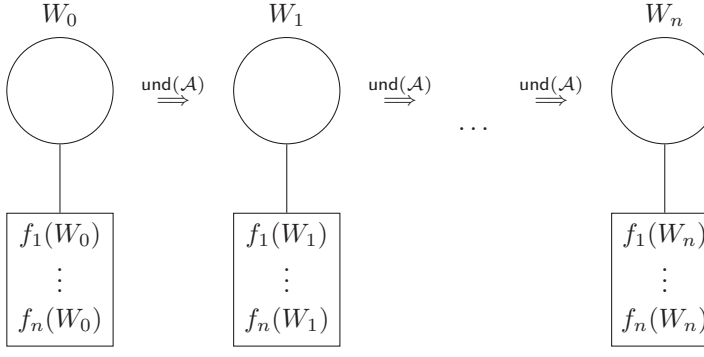

Fig. 1.2. Annotating state sequences by measurement functions.

In quantitative models of biochemical processes, the numerical values assigned to the current state (such as, e.g., the number of molecules or concentrations) do influence the state transitions, i.e., the successor state q' of state q depends on, e.g., the concentrations of molecules in q. This is not the case for reaction systems with measurements; the measurement function just yields the values (of various numerical parameters) for each state, but these values do not influence the successor state.

We will demonstrate now how to make state transitions dependent on the values of measurement functions (see [14]). This is achieved through the use of generalized reactions where the intuition behind them is as follows.

Let S be a finite set and let $b \in \mathsf{rac}(S)$. Then, for each $T \subseteq S$, $\mathsf{res}_b(T) = P_b$ if $\mathsf{en}_b(T)$ and $\mathsf{res}_b(T) = \emptyset$ otherwise. The crucial point of this definition is the partition of 2^S into the set $\Delta_b = \{X \in 2^S : \mathsf{en}_b(X)\}$ of subsets of S which enable b and the set $2^S - \Delta_b$ of subsets of S which do not enable b. With this in mind we can rewrite the definition of res_b as follows: for each $T \subseteq S$, $\mathsf{res}_b(T) = P_b$ if $T \in \Delta_b$ and $\mathsf{res}_b(T) = \emptyset$ if $T \notin \Delta_b$. In this definition we use explicitly the set Δ_b of all subsets of S enabling b. Here one does not have to know *why* a subset T belongs to Δ_b, while the original definition of res_b tells us explicitly why $\mathsf{en}_b(T)$ for a subset T. This leads us to the notion of a generalized reaction.

Definition 14. *Let S be a finite nonempty set. A* generalized reaction *in S is an ordered pair $d = (\Delta, P)$, where $\Delta \subseteq 2^S$ and $P \in 2^S - \{\emptyset\}$.*

The set Δ is called the *condition* of d and the set P is called the *product* of d. They are also written as Δ_b and P_b, respectively.

Definition 15. *Let S be a finite nonempty set and $T \subseteq S$. A generalized reaction b in S is* enabled *by T, denoted* $\mathsf{en}_b(T)$*, if and only if $T \in \Delta_b$. The* result *of b on T, denoted by* $\mathsf{res}_b(T)$*, is defined by* $\mathsf{res}_b(T) = P_b$ *if* $\mathsf{en}_b(T)$ *and* $\mathsf{res}_b(T) = \emptyset$ *otherwise.*

As follows from the discussion preceding Definition 14, the notion of a generalized reaction generalizes the notion of a reaction. The notion of the result function is extended to sets of generalized reactions in the same (cumulative) way as it was done for sets of ordinary reactions.

Definition 16. *Let S be a finite nonempty set and $T \subseteq S$. The* result *of a set B of generalized reactions in S on T, denoted by* $\mathsf{res}_B(T)$*, is defined by* $\mathsf{res}_B(T) = \bigcup\{\mathsf{res}_b(T) : b \in B\}$.

As it was the case for ordinary reactions, also here $\mathsf{res}_B(T)$ can be restated by $\mathsf{res}_B(T) = \bigcup\{\mathsf{res}_b(T) : b \in B \text{ and } \mathsf{en}_b(T)\}$.

Since we are developing the framework of reaction systems, we are interested in generalized reactions that can be explained/implemented by ordinary reactions. Accordingly, we call a generalized reaction d in S *acceptable* if there exists a finite set of reactions $A \in \mathsf{rac}(S)$ such that $\mathsf{res}_d = \mathsf{res}_A$. It turns out that generalized reactions are acceptable providing that they do not violate the boundary condition.

Theorem 4 ([14])**.** Let S be a finite nonemtpy set. A generalized reaction d in S is acceptable if and only if neither $\emptyset \in \Delta_d$ nor $S \in \Delta_d$.

We are ready now to formulate the notion of "a reaction driven by a measurement function" as a special case of a generalized reaction. For didactical reasons, here we consider a single measurement function, but our reasoning easily carries over to finite sets of measurement functions.

Let $f : 2^S \to \mathbb{R}$ be a measurement function and let $Y \subseteq \mathsf{range}(f)$. The intuition behind the set Y is that we want a generalized reaction d to be enabled on a state T if and only if $f(T)$ equals to one of the values specified by Y, for only then will d contribute its product P_d to the successor state of T. In particular, if Y is a singleton, $Y = \{y\}$, then d is enabled on T only if $f(T) = y$.

Let $\Delta_{f,Y} = \{T \in 2^S : T \neq \emptyset, T \neq S, \text{ and } f(T) \in Y\}$ and let d be the generalized reaction defined by $d = (\Delta_{f,Y}, P)$. Clearly, for each $T \in 2^S - \{\emptyset, S\}$, d is enabled by T if and only if $f(T)$ belongs to a prescribed set Y of "good values" of f. Hence, indeed, d is a generalized reaction driven by f, as illustrated in Figure 1.3.

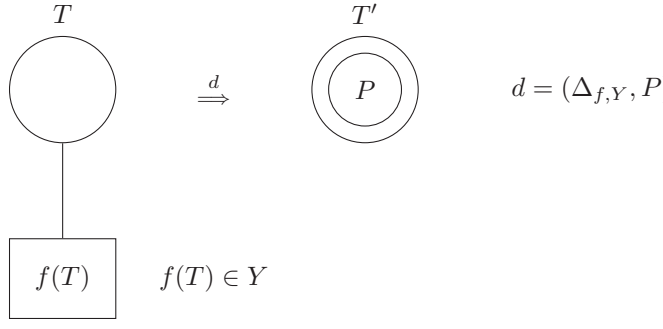

Fig. 1.3. Generalized reaction d is driven by f.

1.7. Reaction Systems with Duration

The non-permanency feature of reaction systems implies that an entity x from the current state will vanish (will not be present in the successor state) unless it is produced by a reaction enabled in the current state or it is introduced by the context of the successor state. This vanishing of x is *instant*: x is present in the current state but it is not present in its successor. However, in organic chemistry entities do not vanish instantly, but rather they have a decay time, meaning that they vanish within a certain time period. To take this into account one considers reaction systems with duration (introduced in [8]).

Definition 17. *A* reaction system with duration*, abbreviated* rsd*, is a triplet* $\mathcal{A} = (S, A, d)$*, where* (S, A) *is a reaction system and* $d : S \to \mathbb{Z}^+$.

The rs (S, A) is called the *underlying* rs *of* $\mathcal{A}$, denoted $\mathsf{und}(\mathcal{A})$, and d is called the *duration function of* $\mathcal{A}$. The *result function of* $\mathcal{A}$, denoted $\mathsf{res}_{\mathcal{A}}$, is the result function of $\mathsf{und}(\mathcal{A})$.

For $x \in S$, $d(x)$ is the *duration of* x (in $\mathcal{A}$). The intuition behind $d(x)$ is that when x is produced by $\mathsf{res}_{\mathcal{A}}$ in an interactive process, its lifetime is $d(x)$ consecutive states (where the count begins in the state where d is produced by $\mathsf{res}_{\mathcal{A}}$).

The notion of an interactive process for a reaction system with duration $\mathcal{A}$ is then modified in order to take into account the durations of entities of $\mathcal{A}$.

Definition 18. *Let* $\mathcal{A} = (S, A, d)$ *be a* rsd *and let* $n \geq 1$ *be an integer. An* (n-step) interactive process in $\mathcal{A}$ *is a triplet* $\pi = (\gamma, \delta, \rho)$ *of finite sequences*

of sets such that $\gamma = C_0, \ldots, C_n$, $\delta = D_0, \ldots, D_n$, *and* $\rho = G_0, \ldots, G_n$, *where* $C_0, \ldots, C_n, D_0, \ldots, D_n, G_0, \ldots, G_n \subseteq S$, $D_0 = \emptyset$, $G_0, G_1 = \emptyset$, $D_i = \mathsf{res}_{\mathcal{A}}(D_{i-1} \cup C_{i-1} \cup G_{i-1})$ *for all* $i \in \{2, \ldots, n\}$, *and* $G_i = \{x \in S : d(x) \geq 2$ *and* $x \in D_j$, *for some* $j \in \{i - (d(x) - 1), \ldots, i - 1\}\}$.

The terminology and notation of reaction systems carry over to reaction systems with duration (through their underlying reaction systems). The sequence ρ is called the *duration sequence of* $\mathcal{A}$. The *state sequence of* π is now defined by $\tau = W_0, W_1, \ldots, W_n$ where, for each $i \in \{0, \ldots, n\}$, $W_i = C_i \cup D_i \cup G_i$. Note that if $d(x) = 1$ for all $x \in S$, then $G_i = \emptyset$ for all $i \in \{0, \ldots, n\}$ and in fact $\mathcal{A}$ behaves just as an ordinary rs.

To incorporate the notion of decay into reaction systems we have extended the notion of rs by adding a duration function to it, which is a "numerical function" assigning to each entity its duration, which is a positive integer. It turns out that one can incorporate the notion of duration in a "non-numerical" way, by extending a given rs $\mathcal{A}$ to a "bigger" rs $\mathcal{A}'$ which will ensure that the entities from $\mathcal{A}$ will have the durations required by a specific duration function. This is formalized as follows.

Definition 19. *Let* $\mathcal{A} = (S, A)$ *and* $\mathcal{A}' = (S', A')$ *be reaction systems. We say that* $\mathcal{A}'$ *is an* extension of $\mathcal{A}$ *if* $S \subseteq S'$ *and* $A \subseteq A'$*; we also say then that* $\mathcal{A}$ *is* embedded in $\mathcal{A}'$.

The notion of extension provides a convenient formalization of an environment of an rs: if $\mathcal{A}'$ is an extension of $\mathcal{A}$, then $\mathcal{A}'$ provides an environment for $\mathcal{A}$. Since $\mathcal{A}$ "knows" only the entities from its own background set S, its basic behavior *within* (the environment) $\mathcal{A}'$ is expressed by the projection $\mathsf{proj}_S(\mathsf{CISTS}(\mathcal{A}'))$. In this way we can investigate the behavior of $\mathcal{A}$ within different environments.

As announced already, the notion of an extension can also account for the phenomenon of duration/decay, as expressed by the following result.

Theorem 5 ([8])**.** For every rsd $\mathcal{A} = (S, A, d)$ there exists a reaction system $\mathcal{A}' = (S', A')$ such that $\mathcal{A}'$ is an extension of (S, A) and $\mathsf{proj}_S(\mathsf{CISTS}(\mathcal{A}')) = \mathsf{CISTS}(\mathcal{A})$.

The construction of $\mathcal{A}'$ is dependent on d, and so this result interprets the phenomenon of duration as an interaction with an environment.

Another area of research on reaction systems with duration is applications of the Chinese remainder theorem; see [43].

1.8. A Reaction Systems Model for the Self-Assembly of Intermediate Filaments

Intermediate filaments (IFs) form, alongside microtubules and actin filaments, the cytoskeleton of eukaryotic cells, found in the vast majority of cells of multicellular organisms; see [45]. The main function of intermediate filaments is structural: they form the cellular network of "girders" and "ropes" that give cells mechanical strength and helps maintain their shapes; see [31]. They have an essential contribution to durable structures such as hair, claws, and fingernails. Intermediate filaments form through the self-assembly of specific proteins. We focus in this section on the self-assembly of vimentin proteins (the most widely distributed of all IF proteins), following the model proposed in [28] and further investigated in [9]. Because of restrictions on the size of the paper we only present very briefly some of the main biological aspects of the self-assembly of intermediate filaments. For details we refer the reader to [26] for a review on this topic.

Vimentin proteins are α-helical rods and they rapidly assemble parallelly into *dimers* (complexes consisting of two vimentins). Then two dimers form (antiparallel, half-staggered) vimentin *tetramers* (complexes consisting of four vimentins). Eight tetramers undergo a fast lateral aggregation forming a thick block of tetramers, so-called *unit-length filament* (ULF). These filaments get then *elongated* through longitudinal annealing with other forming filaments. The model proposed in [28] explains the self-assembly of vimentin IFs through the progressive assembly of tetramers (T) into octamers (O), then into hexadecamers (H) and ULFs (U). It consists of the biochemical reactions in Table 1.1; the emerging filaments are denoted by F in this model.

Table 1.1. The molecular model of [28] for the self-assembly of vimentin IFs.

(1) $2T \to O$;	(3) $2H \to U$;	(5) $F + U \to F$;
(2) $2O \to H$;	(4) $2U \to F$;	(6) $2F \to F$.

The numerical properties of this model were analyzed in [28] and [29], by associating with it a mass-action-based mathematical model expressed as a set of ordinary differential equations. It turns out that the model is versatile — depending on its kinetic constants, the model may show both the formation of stable short filaments, as well as the formation of longer and longer filaments. This is in line with experimental observations of [29]

that the chemical properties of the environment (such as ionic strength and pH) may change the dynamics of the IF self-assembly.

The dynamic, context-dependent self-assembly of intermediate filaments can also be followed through reaction systems models. Several such models were introduced in [5]; we discuss one of them here.

To distinguish between (the formation of) two types of filaments, we introduce two species/entities, F_s and F_l, denoting short and long filaments, respectively. Tetramers, octamers, hexadecamers and unit-length filaments are modeled through species/entities T, O, H, U, respectively; for them we use the same names as in the molecular model. We also add to the background set species/entities short and long to be used as a signal from the environment regarding the type of filaments to be formed in the model; we assume that the environment adds, in each step of an interactive process, either short or long, possibly in addition to other species/entities. Finally, we add f to the background set, to be used as a "formal inhibitor" for reactions that need no explicit inhibitor.

Table 1.2. A reaction systems model for the self-assembly of vimentin IFs.

(i)	$(\{T\},\{f\},\{O\})$;	(v.1)	$(\{U,F_s,\mathsf{long}\},\{\mathsf{short}\},\{F_l\})$;
(ii)	$(\{O\},\{f\},\{H\})$;	(v.2)	$(\{U,F_l,\mathsf{long}\},\{\mathsf{short}\},\{F_l\})$;
(iii)	$(\{H\},\{f\},\{U\})$;	(vi.1)	$(\{F_s,\mathsf{long}\},\{\mathsf{short}\},\{F_l\})$;
(iv)	$(\{U\},\{F_l\},\{F_s\})$;	(vi.2)	$(\{F_s,F_l,\mathsf{long}\},\{\mathsf{short}\},\{F_l\})$;
		(vi.3)	$(\{F_l,\mathsf{long}\},\{\mathsf{short}\},\{F_l\})$.

Reactions (i)–(iii) in the reaction system in Table 1.2 correspond to the reactions (1)–(3) in the molecular model in Table 1.1; they stand for the sequence of self-assemblies of tetramers to octomers to hexadecamers to unit-length filaments. This part of the reaction systems model is obtained from the molecular model in a straightforward way: for each reaction in the molecular model we create a reaction in the reaction systems model that uses the same reactants and products as its molecular correspondent. Note that even though each of the molecular reactions (1)–(3) has a reactant with multiplicity 2, we ignore the multiplicity in reactions (i)–(iii) based on the *threshold assumption* discussed in Section 1.3.

The molecular reaction (4) explaining the longitudinal annealing of two unit-length filaments to form the shortest proper filament is translated to reaction (iv) in the reaction systems model. Its product set is $\{F_s\}$, i.e., a *short* filament; we set F_l as the inhibitor of this reaction to indicate that in the case when the formation of long filaments is favored over the formation

of short ones, the unit-length filaments will have a preference to extend existing long filaments rather than start new short ones.

The molecular reaction (5) explaining the elongation of existing filaments with unit-length filaments is captured in our reaction systems model through reactions (v.1) and (v.2). To ensure that this reaction will take place only if the context favors the formation of long filaments, we add long to the set of reactants, in addition to U, F_s in (v.1) and U, F_l in (v.2). The product set in each case is $\{F_l\}$.

The molecular reaction (6) explaining the longitudinal annealing of two existing filaments and the formation of a longer filament is captured in the reaction systems model through reactions (vi.1)-(vi.3). The three reactions differ in the type of filaments (short or long) involved in the elongation, and they all take place only if the context indicates that the formation of long filaments is favored by throwing in the species long.

The reaction systems model successfully captures the dynamic context-dependent formation of short and long filaments in the following sense. For a constant context $\{T, \mathsf{short}\}$, providing in each step of the interactive process more tetramers to support the formation of more filaments and short, the interactive process quickly arrives at a self-loop on state $\{O, H, U, F_s\}$, which corresponds to the situation when only short filaments are being self-assembled, along with the intermediary steps of the assembly from tetramers to filaments. On the other hand, for a constant context $\{T, \mathsf{long}\}$, the interactive process arrives at a self-loop on state $\{U, H, U, F_l\}$, which corresponds to the situation where only existing filaments are being elongated, and new ones are not started. Moreover, the interactive process may cycle between these states by changing the context from $\{T, \mathsf{short}\}$ to $\{T, \mathsf{long}\}$ and the other way around. Generating all these interactive processes may be done through the online reaction system simulator at [48].

1.9. A Reaction Systems Model for the Heat Shock Response

The heat shock response is a cellular defense mechanism against stress-induced protein misfolding. The mechanisms is present in roughly the same form throughout all eukaryotes; even prokaryotes have a similar heat shock response. Because of restrictions on the size of the paper we only present some of the main biological aspects of the heat shock response very briefly. For more details, we refer the reader to [47] for a review of the heat shock response.

Exposure to elevated temperatures (and other types of stress such as oxidative stress or exposure to heavy metals) leads, among others, to highly accelerated protein misfolding. Misfolded proteins tend to form aggregates that may eventually lead to cell death. The cellular response to accumulating misfolded proteins is to accelerate the synthesis of protein chaperones (in our case, the heat shock proteins) which help misfolded proteins regain their native fold and block them from binding to other misfolded proteins. If the cell returns to physiological temperatures, then eventually the synthesis of heat shock proteins is decreased, as a consequence of a lower level of misfolded proteins.

The heat shock response is about the genetic regulation of the heat shock proteins in the presence as well as in the absence of heat stress. Here we follow a simplified version of the molecular model proposed in [35]; its molecular reactions are listed in Table 1.3. The main focus of this model is on the transcription regulation of the heat shock protein-encoding gene. This gene is transcribed with the help of a transcription factor called the *heat shock factor* (hsf); the gene's promoter is called the *heat shock element* (hse). The heat shock factors form homologous trimers ($\mathsf{hsf_3}$) through reaction (1.3), which are able to bind to the heat shock element forming complex $\mathsf{hsf_3{:}\,hse}$ (in reaction (1.3)), and thus promote the transcription of the gene and subsequent synthesis of additional heat shock proteins, through reaction (1.3).

The heat shock proteins have an affinity for binding to heat shock factors in all of their forms (hsf, $\mathsf{hsf_3}$, and $\mathsf{hsf_3{:}\,hse}$); this is modeled through reactions (1.3)–(1.3). They have a much higher affinity to bind to misfolded proteins, through reaction (1.3), and help them get refolded through reaction (1.3). The model also includes the degradation of heat shock proteins, reaction (1.3) and the protein misfolding reaction (1.3); this latter reaction has a flux that depends exponentially on the temperature (see [35]).

Table 1.3. A simplified version of the molecular model of [35] for the heat shock response.

(7) $3\,\mathsf{hsf} \to \mathsf{hsf_3}$;	(12) $\mathsf{hsp} + \mathsf{hsf_3{:}\,hse} \to \mathsf{hsp{:}\,hsf} + 2\,\mathsf{hsf} + \mathsf{hse}$;
(8) $\mathsf{hsf_3} + \mathsf{hse} \to \mathsf{hsf_3{:}\,hse}$;	(13) $\mathsf{hsp} \to \emptyset$;
(9) $\mathsf{hsf_3{:}\,hse} \to \mathsf{hsf_3{:}\,hse} + \mathsf{hsp}$;	(14) $\mathsf{prot} \to \mathsf{mfp}$;
(10) $\mathsf{hsp} + \mathsf{hsf} \rightleftarrows \mathsf{hsp{:}\,hsf}$;	(15) $\mathsf{hsp} + \mathsf{mfp} \to \mathsf{hsp{:}\,mfp}$;
(11) $\mathsf{hsp} + \mathsf{hsf_3} \to \mathsf{hsp{:}\,hsf} + 2\,\mathsf{hsf}$;	(16) $\mathsf{hsp{:}\,mfp} \to \mathsf{hsp} + \mathsf{prot}$.

The numerical properties of the heat shock response model were analyzed in [35] by associating with it a mass-action model based on ordi-

nary differential equations. The model was able to explain all the available quantitative and qualitative data, including the quick temperature-induced transactivation of the heat shock gene and its silencing in the absence of heat stress. These properties were explained through the numerical contributions of the kinetic rate constants of the various reactions in the model. The same basic interplay between temperature and the transactivation of the gene can, however, also be demonstrated through a reaction systems model, i.e., through a *qualitative* interplay between the reactions of the model. We discuss this in the following reaction systems model for the heat shock response proposed in [4] and shown in Table 1.4.

Table 1.4. A reaction systems model for the heat shock response.

(vii.1)	$(\{\mathsf{hsf}\}, \{\mathsf{hsp}\}, \{\mathsf{hsf_3}\})$;
(vii.2)	$(\{\mathsf{hsf}, \mathsf{hsp}, \mathsf{mfp}\}, \{f\}, \{\mathsf{hsf_3}\})$;
(viii.1)	$(\{\mathsf{hsf_3}, \mathsf{hse}\}, \{\mathsf{hsp}\}, \{\mathsf{hsf_3}\colon \mathsf{hse}\})$;
(viii.2)	$(\{\mathsf{hsf_3}, \mathsf{hse}, \mathsf{hsp}, \mathsf{mfp}\}, \{f\}, \{\mathsf{hsf_3}\colon \mathsf{hse}\})$;
(ix.1)	$(\{\mathsf{hsf_3}\colon \mathsf{hse}\}, \{\mathsf{hsp}\}, \{\mathsf{hsf_3}\colon \mathsf{hse}, \mathsf{hsp}\})$;
(ix.2)	$(\{\mathsf{hsf_3}\colon \mathsf{hse}, \mathsf{hsp}, \mathsf{mfp}\}, \{f\}, \{\mathsf{hsf_3}\colon \mathsf{hse}, \mathsf{hsp}\})$;
(x.1)	$(\{\mathsf{hsp}, \mathsf{hsf}\}, \{\mathsf{mfp}\}, \{\mathsf{hsp}\colon \mathsf{hsf}\})$;
(x.2)	$(\{\mathsf{hsp}\colon \mathsf{hsf}, \mathsf{stress}\}, \{\mathsf{nostress}\}, \{\mathsf{hsp}, \mathsf{hsf}\})$;
(x.3)	$(\{\mathsf{hsp}\colon \mathsf{hsf}, \mathsf{nostress}\}, \{\mathsf{stress}\}, \{\mathsf{hsp}\colon \mathsf{hsf}\})$;
(xi)	$(\{\mathsf{hsp}, \mathsf{hsf_3}\}, \{\mathsf{mfp}\}, \{\mathsf{hsp}\colon \mathsf{hsf}\})$;
(xii)	$(\{\mathsf{hsp}, \mathsf{hsf_3}\colon \mathsf{hse}\}, \{\mathsf{mfp}\}, \{\mathsf{hsp}\colon \mathsf{hsf}, \mathsf{hse}\})$;
(xiv.1)	$(\{\mathsf{prot}, \mathsf{stress}\}, \{\mathsf{nostress}\}, \{\mathsf{prot}, \mathsf{mfp}\})$;
(xiv.2)	$(\{\mathsf{prot}, \mathsf{nostress}\}, \{\mathsf{stress}\}, \{\mathsf{prot}\})$;
(xv)	$(\{\mathsf{hsp}, \mathsf{mfp}\}, \{f\}, \{\mathsf{hsp}\colon \mathsf{mfp}\})$;
(xvi)	$(\{\mathsf{hsp}\colon \mathsf{mfp}\}, \{f\}, \{\mathsf{hsp}, \mathsf{prot}\})$;
(hse.1)	$(\{\mathsf{hse}\}, \{\mathsf{hsf_3}\}, \{\mathsf{hse}\})$;
(hse.2)	$(\{\mathsf{hse}, \mathsf{hsf_3}, \mathsf{hsp}\}, \{\mathsf{mfp}\}, \{\mathsf{hse}\})$;
(mfp)	$(\{\mathsf{mfp}\}, \{\mathsf{hsp}\}, \{\mathsf{mfp}\})$.

In the reaction systems model we use the same species names as in the molecular model. We add to the background set the species stress and nostress, and assume that in each step of an interactive process, the context adds either one of them, indicating whether the model is under heat stress or not. We also add species f serving as a "formal inhibitor" for reactions that need no explicit inhibitor.

Building the reaction system corresponding to the molecular reactions in Table 1.3 takes into account the implicit, kinetic rate-driven competition on resources between molecular reactions, and makes it explicit through the inhibition mechanism. Whereas in the molecular model a reaction b could block another reaction a simply through a faster access to the common

reactants driven by its higher kinetic rate constant, in the reaction systems model we explicitly introduce inhibitors such that in states where both reactions would have their reactants available, the reaction corresponding to a would not be enabled.

An example of this situation is in the modeling of the trimer formation reaction (1.3). We introduce two reactions in our reaction systems model, (vii.1)–(vii.2) corresponding to this situation. In (vii.1) we indicate through the set of reactants and products that hsf trimerizes into hsf_3. We indicate through the inhibitor hsp that reaction (1.3) has a higher kinetic rate constant and competes for the same resource hsf and so, if hsp is present in the current state, then (vii.1) is not enabled. On the other hand, even if hsp were present in the current state, hsf could still be left free to trimerize, if hsp is 'hijacked' by reaction (1.3) that has a higher kinetic rate constant than (1.3); this is the case if both hsp and mfp are present in the current state. Hence, we also introduce reaction (vii.2) showing that hsf may get trimerized into hsf_3 in the presence of both hsp and mfp.

Using similar reasoning, the molecular reaction (1.3) is translated into reactions (viii.1) and (viii.2). In the former one, the reactant and the product sets are $\{\mathsf{hsf}_3, \mathsf{hse}\}$ and $\{\mathsf{hsf}_3\colon \mathsf{hse}\}$, respectively. We set hsp as an inhibitor to indicate that it may block this reaction by starving it of hsf_3 through the molecular reaction (1.3) that has a higher kinetic rate constant than (1.3). However, hsp may be itself 'hijacked' by reaction (1.3) that has an even higher kinetic rate constant than (1.3), and thus it will leave hsf_3 and hse able to produce $\mathsf{hsf}_3\colon \mathsf{hse}$ if both hsp and mfp are present in the current state. This situation is captured in reaction (viii.2).

The other reactions of our reaction systems model are deduced in a similar way. Note the additional reactions (hse.1), (hse.2), and (mfp): their role is to make sure that the species hse and mfp do not disappear from the current state if they are not handled by any reaction, as would otherwise be the case due to the *non-permanency principle* discussed in Section 1.1. Note also that the degradation of hsp (reaction (1.3)) need not be explicitly modeled because of the non-permanency principle. We refer to [4] for a detailed discussion about building this model.

The reaction system model obtained in this way successfully captures the temperature-induced dynamics of the transactivation of the heat shock gene in the following sense. Starting from a context $\{\mathsf{hsp}, \mathsf{prot}, \mathsf{hse}, \mathsf{nostress}\}$ introducing the main species into the interactive process, and continuing with constant context $\{\mathsf{nostress}\}$, the interactive process eventually enters into a self-loop on state $\{\mathsf{hse}, \mathsf{hsp}\colon \mathsf{hsf}, \mathsf{prot}\}$. As discussed above, this is a

good qualitative description of the state of the molecular model in the absence of stress: the heat shock factors are bound in complexes with the heat shock proteins, the gene is not transcribed (i.e., the heat shock element is free), and there are 'no' (or very few) misfolded proteins. On the other hand, if one continues after the first context set with a constant context $\{\mathsf{stress}\}$, the interactive process eventually enters into a self-loop on state $\{\mathsf{hsf_3}\colon \mathsf{hse}, \mathsf{hsp}, \mathsf{hsp}\colon \mathsf{mfp}, \mathsf{mfp}, \mathsf{prot}\}$. This is indeed a good qualitative description of the state of the molecular model in the presence of stress: the gene is transcribed (i.e., the heat shock element is occupied by heat shock factor trimers), there are misfolded proteins both free and bound to heat shock proteins, and also free heat shock proteins. Moreover, the interactive process may eventually switch between these states if the (constant) context switches between $\{\mathsf{stress}\}$ and $\{\mathsf{nostress}\}$. These interactive processes may be automatically generated online through the simulator at [48].

1.10. Discussion

In this paper we introduced (in a tutorial fashion) the framework of reaction systems. We first presented (along with the underlying intuition and motivation) the formal notions of a reaction and its effect (as well as the effect of a set of reactions) on the current state of a system and then introduced the formal notion of a reaction system which is the central technical notion of the framework. We discussed two lines of research concerning reaction systems: reaction systems as definitions ("implementations") of power set functions and the relationship between reaction systems and finite transition systems.

Sometimes a research theme leads to a need to add additional components to the basic construct of reaction systems. Reaction systems, together with such extensions, form a broad framework of reaction systems. We have presented two such extensions: reaction systems with measurements and reaction systems with durations. Reaction systems are a qualitative model — there are no numerical parameters (there is no counting) in reaction systems. Extending reaction systems by measurement functions allows to assign numerical parameters to states, something that is often done in biology. We have demonstrated that the dynamic behavior of reaction systems with measurements can be also implemented by ordinary reaction systems using so-called generalized reactions (which are simply macros for finite sets of ordinary reactions). Non-permanence of entities is a basic feature of reaction systems (motivated by bioenergetics of the living cell). It causes

an *instant* vanish of an entity from the current state unless it is sustained either by reactions or by a context. However, biochemical entities decay in nature within a certain period of time. To take this into account reaction systems were extended with duration functions (defining the decay time for each entity). Also in this case it turned out that the duration function can be implemented by ordinary reaction systems (in fact, the phenomenon of duration/delay can be explained as an interaction of the reaction system with its environment).

We also presented two reaction system models, for self-assembly of intermediate filaments and for eukaryotic heat shock response. In both cases, the models were constructed starting from molecular reaction models in such a way that their qualitative dynamic behavior (in terms of interactive processes) captures key aspects of the quantitative dynamic behavior of the corresponding mass-action-based ordinary differential equations models. This demonstrates that reaction systems are a versatile framework capable of capturing sophisticated dynamic behavior and explaining it through cause-effect relationships in terms of facilitation and inhibition between biochemical reactions.

In this paper we have presented only few research lines from the framework of reaction systems. To provide a better perspective we will give now a brief overview of some other research topics.

State sequences are the main (and traditional) "manifestation" of dynamic processes, and they are a central topic of research in the framework of reaction systems. Representative topics for this line of research are 'life and death' properties, stability and chaos, and estimates of the lengths of state sequences; see, e.g., [16, 17, 39, 42, 40, 38, 25, 24, 23].

Structural properties of consecutive states of state sequences are important in understanding the dynamics of reaction systems. In particular, formalization of the so-called *modules* (motivated by research in biochemistry and biology, see [46]) has been investigated in [18].

Understanding causalities in dynamic processes is a classic research topic in computer science. It is also well-motivated, from both the biological and computational points of view, in the framework of reaction systems. Static and dynamic causalities (influences) between entities in reaction systems were investigated in [7].

The topic of time in models of biological processes ("What is time?", "How to assign time to states of biochemical systems?") is both important and fascinating. It turns out that one can introduce time in reaction sys-

tems using measurement functions satisfying certain properties (see [20]). In this way one can introduce and investigate important concepts such as reaction times and time distances between consecutive states.

A recent popular topic of research is that of model checking for reaction systems. Results along this line include computational complexity of checking various properties [24, 23, 2, 12, 10, 11], importing standard modeling concepts (such as steady states and mass conservations) to reaction systems [2, 3], and temporal logic for reaction systems [34].

Finally we want to mention the use of reaction systems within so-called *exploration systems*: a formal framework for exploring a discipline of science. Reaction systems captured two important features of the functioning of the living cell: the fact that the living cell is an open system and the non-permanency of its entities. Another important feature of biological systems is the presence of hierarchical structures on both the physical and the methodological levels. This feature is not addressed by reaction systems but it is addressed by zoom structures (see [21, 22]). Reaction systems together with zoom structures form exploration systems ([21]). An exploration system consists of a depository of knowledge provided by a zoom structure and of dynamic processes (exploring this depository of knowledge) provided by a finite set of reaction systems.

Dedication and Acknowledgments

This paper is dedicated to Janusz Brzozowski, in recognition of his outstanding contributions to computer science. The authors are indebted to Robert Brijder and Sepinoud Azimi for valuable comments concerning the working version of this paper.

References

1. A. Arnold, *Finite Transition Systems: Semantics of Communicating Systems*, Prentice Hall International (UK) Ltd., Hertfordshire, UK, 1994.
2. S. Azimi, C. Gratie, S. Ivanov, L. Manzoni, I. Petre and A. E. Porreca, Complexity of model checking for reaction systems, *Theoret. Comput. Sci.* **623** (2016), 103–113.
3. S. Azimi, C. Gratie, S. Ivanov and I. Petre, Dependency graphs and mass conservation in reaction systems, *Theoret. Comput. Sci.* **598** (2015), 23–39.
4. S. Azimi, B. Iancu and I. Petre, Reaction system models for the heat shock response, *Fund. Inform.* **131** (2014), 1–14.

5. S. Azimi, C. Panchal, E. Czeizler and I. Petre, Reaction systems models for the self-assembly of intermediate filaments, *Annals of University of Bucharest* **LXII**,2 (2015), 9–24.
6. R. Brijder, A. Ehrenfeucht, M. Main and G. Rozenberg, A tour of reaction systems, *Int. J. Found. Comput. Sci.* **22**,07 (Nov. 2011), 1499–1517.
7. R. Brijder, A. Ehrenfeucht and G. Rozenberg, A note on causalities in reaction systems, *Electronic Communications of the EASST* **30** (2010), 1–9.
8. R. Brijder, A. Ehrenfeucht and G. Rozenberg, Reaction Systems with Duration, in *Computation, Cooperation, and Life*, J. Kelemen and A. Kelemenová (eds.), pp. 191–202, *LNCS* vol. 6610, Springer Berlin Heidelberg, 2011.
9. E. Czeizler, A. Mizera, E. Czeizler, R. J. Back, J. E. Eriksson and I. Petre, Quantitative analysis of the self-assembly strategies of intermediate filaments from tetrameric vimentin, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **9**,3 (May 2012), 885–898.
10. A. Dennunzio, E. Formenti, L. Manzoni and A. E. Porreca, Ancestors, descendants, and gardens of eden in reaction systems, *Theor. Comput. Sci.* **608** (2015), 16–26.
11. A. Dennunzio, E. Formenti, L. Manzoni and A. E. Porreca, Preimage problems for reaction systems, in *9th LATA 2015*, A. H. Dediu, E. Formenti, C. Martín-Vide and B. Truthe (eds.), pp. 537–548, *LNCS* vol. 8977, Springer, 2015.
12. A. Dennunzio, E. Formenti, L. Manzoni and A. E. Porreca, Reachability in resource-bounded reaction systems, in *10th LATA 2016*, A. Dediu, J. Janousek, C. Martín-Vide and B. Truthe (eds.), pp. 592–602, *LNCS* vol. 9618, Springer, 2016.
13. A. Ehrenfeucht, J. Kleijn, M. Koutny and G. Rozenberg, Minimal reaction systems, in *Transactions on Computational Systems Biology XIV SE-5*, C. Priami, I. Petre and E. de Vink (eds.), pp. 102–122, *LNCS* vol. 7625, Springer Berlin Heidelberg, 2012.
14. A. Ehrenfeucht, J. Kleijn, M. Koutny and G. Rozenberg, Qualitative and quantitative aspects of a model for processes inspired by the functioning of the living cell, in *Biomolecular Information Processing: From Logic Systems to Smart Sensors and Actuators*, E. Katz (ed.), ch. 16, pp. 303–322, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, July 2012.
15. A. Ehrenfeucht, J. Kleijn, M. Koutny and G. Rozenberg, Reaction systems: A natural computing approach to the functioning of living cells, in *A Computable Universe: Understanding and Exploring Nature as Computation*, Z. Hector (ed.), ch. 10, pp. 189–208, World Scientific, Singapore, 2012.
16. A. Ehrenfeucht, M. Main and G. Rozenberg, Combinatorics of life and death for reaction systems, *Int. J. Found. Comput. Sci.* **21**,03 (June 2010), 345–356.
17. A. Ehrenfeucht, M. Main, G. Rozenberg and A. T. Brown, Stability and chaos in reaction systems, *Int. J. Found. Comput. Sci.* **23**,05 (2012), 1173–1184.

18. A. Ehrenfeucht and G. Rozenberg, Events and modules in reaction systems, *Theoret. Comput. Sci.* **376**,1-2 (May 2007), 3–16.
19. A. Ehrenfeucht and G. Rozenberg, Reaction systems, *Fund. Inform.* **75**,1 (Jan. 2007), 263–280.
20. A. Ehrenfeucht and G. Rozenberg, Introducing time in reaction systems, *Theoret. Comput. Sci.* **410**,4-5 (Feb. 2009), 310–322.
21. A. Ehrenfeucht and G. Rozenberg, Zoom structures and reaction systems yield exploration systems, *Int. J. Found. Comput. Sci.* **25** (2014), 275–305.
22. A. Ehrenfeucht and G. Rozenberg, Standard and ordered zoom structures, *Theoret. Comput. Sci.* **608** (2015), 4–15.
23. E. Formenti, L. Manzoni and A. E. Porreca, Cycles and global attractors of reaction systems, in *16th DCFS 2014*, H. Jürgensen, J. Karhumäki and A. Okhotin (eds.), pp. 114–125, *LNCS* vol. 8614, Springer International Publishing, 2014.
24. E. Formenti, L. Manzoni and A. E. Porreca, Fixed points and attractors of reaction systems, in *Language, Life, Limits*, A. Beckmann, E. Csuhaj-Varjú and K. Meer (eds.), pp. 194–203, *LNCS* vol. 8493, Springer International Publishing, 2014.
25. E. Formenti, L. Manzoni and A. E. Porreca, On the complexity of occurrence and convergence problems in reaction systems, *Natural Computing* **14**,1 (2015), 185–191.
26. H. Herrmann, H. Bär, L. Kreplak, S. V. Strelkov and U. Aebi, Intermediate filaments: from cell architecture to nanomechanics, *Nature Reviews Mol Cell Biol* **8** (2007), 562–573.
27. L. Kari and G. Rozenberg, The many facets of natural computing, *Communications of the ACM* **51**,10 (Oct. 2008), 72–83.
28. R. Kirmse, S. Portet, N. Mücke, U. Aebi, H. Herrmann and J. Langowski, A quantitative kinetic model for the in vitro assembly of intermediate filaments from tetrameric vimentin, *J. Biol. Chem.* **282**,52 (2007), 18563–18572.
29. L. Kreplak, U. Aebi and H. Herrmann, Molecular mechanisms underlying the assembly of intermediate filaments, *Experimental Cell Research*, 2004.
30. A. Lehninger, *Bioenergetics: The Molecular Basis of Biological Energy Transformations*, Addison-Wesley Pub (Sd), 1971.
31. H. Lodish, A. Berk, L. S. Zipursky, P. Matsudaira, D. Baltimore and J. Darnell, *Molecular Cell Biology (4th edition)*, W.H.Freeman, 2000.
32. L. Manzoni, D. Pocas and A. E. Porreca, Simple reaction systems and their classification, *Int. J. Found. Comput. Sci.* **25**,04 (2014), 441–457.
33. L. Manzoni and A. E. Porreca, Reaction systems made simple, in *Unconventional Computation and Natural Computation*, G. Mauri, A. Dennunzio, L. Manzoni and A. E. Porreca (eds.), Berlin, Heidelberg, 2013, pp. 150–161, *LNCS* vol. 7956, Springer Berlin Heidelberg.
34. A. Męski, W. Penczek and G. Rozenberg, Model checking temporal properties of reaction systems, *Inform. Sciences*, 2015.

35. I. Petre, A. Mizera, C. L. Hyder, A. Meinander, A. Mikhailov, R. I. Morimoto, L. Sistonen, J. E. Eriksson and R.-J. Back, A simple mass-action model for the eukaryotic heat shock response and its mathematical validation, *Natural Computing* **10**,1 (2011), 595–612.
36. W. Reisig, *Understanding Petri Nets*, Springer Berlin Heidelberg, 2013.
37. G. Rozenberg, T. H. W. Bäck and J. N. Kok (eds.), *Handbook of Natural Computing*, Springer Heidelberg, 2010.
38. A. Salomaa, Functions and sequences generated by reaction systems, *Theor. Comput. Sci.* **466** (2012), 87–96.
39. A. Salomaa, On state sequences defined by reaction systems, in *Logic and Program Semantics*, R. L. Constable and A. Silva (eds.), pp. 271–282, *LNCS* vol. 7230, Springer-Verlag, 2012.
40. A. Salomaa, Functional constructions between reaction systems and propositional logic, *Int. J. Found. Comput. Sci.* **24**,1 (2013), 147–160.
41. A. Salomaa, Minimal and almost minimal reaction systems, *Natural Computing* **12**,3 (2013), 369–376.
42. A. Salomaa, Minimal reaction systems defining subset functions, in *Computing with New Resources*, C. Calude, R. Freivalds and K. Iwama (eds.), pp. 436–446, *LNCS* vol. 8808, Springer, 2014.
43. A. Salomaa, Applications of the chinese remainder theorem to reaction systems with duration, *Theoret. Comput. Sci.* **598** (2015), 15–22.
44. A. Salomaa, Two-step simulations of reaction systems by minimal ones, *Acta Cybernetica* **22**,2 (2015), 393–311.
45. M. Schliwa, *The Cytoskeleton*, *Cell Biology Monographs* vol. 13, Springer-Verlag, Vienna, Austria, 1986.
46. G. Schlosser and G. P. Wagner (eds.), *Modularity in Development and Evolution*, University of Chicago Press Books, 2004.
47. R. Voellmy, Transduction of the stress signal and mechanisms of transcriptional regulation of heat shock/stress protein gene expression in higher eukaryotes, *Critical reviews in eukaryotic gene expression* **4**,4 (1994), 357—401.
48. The web interface of the reaction system simulator. `http://combio.abo.fi/research/reaction-systems/reaction-system-simulator`, Accessed 01/06/2016.