Newcastle
University

COMPUTING
SCIENCE

Qualitative and Quantitative Aspects of a Model for Processes Inspired by the Functioning of the Living Cell

Andrzej Ehrenfeucht, Jetty Kleijn, Maciej Koutny and Grzegorz Rozenberg

TECHNICAL REPORT SERIES

No. CS-TR-1305 November 2011

TECHNICAL REPORT SERIES

No. CS-TR-1305 November, 2011

Timed Migration and Interaction with Access Permissions

A. Ehrenfeucht, J. Kleijn, M. Koutny, G. Rozenberg

Abstract

Reaction systems are a formal model for processes inspired by the functioning of the living cell. The underlying idea of this model is that the functioning of the living cell is determined by the interactions of biochemical reactions, and these interactions are based on the mechanisms of facilitation and inhibition. In this paper we first review the main notions of the basic model of reaction systems which is a qualitative model. Then we discuss various ways of taking into account quantitative properties.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

EHRENFEUCHT, A., KLEIJN, J., KOUTNY, M., ROZENBERG, G.

Qualitative and Quantitative Aspects of a Model for Processes Inspired by the Functioning of the Living Cell
[By] A. Ehrenfeucht, J. Kleijn, M. Koutny, G. Rozenberg
Newcastle upon Tyne: Newcastle University: Computing Science, 2011.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1305)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1305

Abstract

Reaction systems are a formal model for processes inspired by the functioning of the living cell. The underlying idea of this model is that the functioning of the living cell is determined by the interactions of biochemical reactions, and these interactions are based on the mechanisms of facilitation and inhibition. In this paper we first review the main notions of the basic model of reaction systems which is a qualitative model. Then we discuss various ways of taking into account quantitative properties.

About the authors

Andrzej Ehrenfeucht is a Professor in the Department of Computer Science, University of Colorado, Boulder, USA.

Jetty Kleijn is a visiting fellow within the School of Computing Science, Newcastle University.

Maciej Koutny obtained his MSc (1982) and PhD (1984) from the Warsaw University of Technology. In 1985 he joined the then Computing Laboratory of the University of Newcastle upon Tyne to work as a Research Associate. In 1986 he became a Lecturer in Computing Science at Newcastle, and in 1994 was promoted to an established Readership at Newcastle. In 2000 he became a Professor of Computing Science.

Grzegorz Rozenberg is a Professor of Computer Science at Leiden University, The Netherlands, and an Adjunct Professor at the Department of Computer Science of University of Colorado at Boulder, U.S.A. He is the head of the Theoretical Computer Science group at Leiden Institute of Advanced Computer Science (LIACS), and the scientific director of Leiden Center for Natural Computing (LCNC). He has published more than 500 papers, 6 books, and is a (co-)editor of about 90 books. He is a member of an editorial board of about 20 journals and book series. He has been a member of the program committees and invited speaker for practically all major conferences in theoretical computer science in Europe. He is a Foreign Member of the Finnish Academy of Sciences and Letters, a member of Academia Europaea, and he is holder of Honorary Doctorates of the University of Turku, Finland, the Technical University of Berlin, Germany, and the University of Bologna, Italy. He has received the Distinguished Achievements Award of the European Association for Theoretical Computer Science "in recognition of his outstanding scientific contributions to theoretical computer science". He is a Highly Cited Researcher by ISI.

Suggested keywords

REACTION SYSTEM
GENERALISED REACTION SYSTEM
LIVING CELL
NATURAL COMPUTING
GENETIC REGULATORY NETWORK
TRANSITION SYSTEM,
MEASUREMENT FUNCTION
APPROXIMATION
SIMULATION

Qualitative and Quantitative Aspects of a Model for Processes Inspired by the Functioning of the Living Cell

Andrzej Ehrenfeucht[1], Jetty Kleijn[2],
Maciej Koutny[3], and Grzegorz Rozenberg[1,2]

[1] Department of Computer Science, University of Colorado at Boulder
430 UCB Boulder, CO 80309-0430, U.S.A.
andrzej@cs.colorado.edu
[2] LIACS, Leiden University, 2300 RA, The Netherlands
{kleijn,rozenber}@liacs.nl
[3] School of Computing Science, Newcastle University, NE1 7RU, UK
maciej.koutny@ncl.ac.uk

Abstract. Reaction systems are a formal model for processes inspired by the functioning of the living cell. The underlying idea of this model is that the functioning of the living cell is determined by the interactions of biochemical reactions, and these interactions are based on the mechanisms of facilitation and inhibition. In this paper we first review the main notions of the basic model of reaction systems which is a qualitative model. Then we discuss various ways of taking into account quantitative properties.

Keywords: reaction system, generalised reaction system, living cell, natural computing, genetic regulatory network, transition system, measurement function, approximation, simulation.

1 Introduction

Natural computing is concerned with human-designed computing inspired by nature and with computing taking place in nature (see, e.g., [1] and [2]). The former strand investigates models and computational techniques inspired by nature, while the latter investigates, in terms of information processing, phenomena taking place in nature. The former strand includes research areas such as evolutionary computation, neural computation, quantum computation, and molecular computation. The latter strand includes investigations into the computational nature of self-assembly, the computational nature of developmental processes, the computational nature of brain processes, the system biology approach to bionetworks, and the computational nature of biochemical reactions. Clearly, the two research strands are not disjoint.

Biomolecular computation is a topic of intense research in natural computing. Within the former strand this research focuses on constructing, either in vitro or in vivo, various building blocks of computing devices (such as switches, gates and biosensors). Within the latter strand this research is more concerned with establishing how biocomputations drive natural processes — the essence of this research is nicely captured by the following statement by Richard Dawkins, a world leading expert in evolutionary biology (see [3]): 'If you want to understand life, don't think about vibrant throbbing gels and oozes, think about information technology'.

This paper falls into this second strand of research. It discusses reaction systems which is a formal model for the investigation of the functioning of the living cell. The functioning is viewed in terms of formal processes resulting from interactions between biochemical reactions taking place in the living cell. Moreover, we assume that these interactions are driven by two mechanisms, facilitation and inhibition: the (products of the) reactions may facilitate or inhibit each other. The basic model of reaction systems abstracts from various (technical) features of biochemical reactions to such extent that it becomes a qualitative rather than quantitative model. However, it takes into account the basic bioenergetics (flow of energy) of the living cell, and it also takes into account that the living cell is an open system and its behaviour is influenced by its environment. The broader *framework of reaction systems* is formed by the central model of reaction systems and its various extensions. The main focus of research is on understanding processes that take place in these models.

The paper can be seen as consisting of two parts. The first part (Sections 2, 3 and 4) reviews the main notions (together with underlying motivation) of reaction systems. As already mentioned, the basic model of reaction systems is qualitative. However, there are various situations in biology/biochemistry where one needs to consider quantities assigned to the states of a biochemical system. To account for this, the broad framework of reaction systems includes the notion of reaction systems with measurements — they are recalled in Section 6. Then, Sections 7, 8 and 9 discuss various ways of dealing with quantitative parameters in reaction systems. These sections present new material (the theorem presented in Section 6 is also new). The discussion in Section 10 concludes this paper.

2 Reactions

The formal notion of a reaction (introduced in [4]) formalises the basic intuition of a biochemical reaction — it will take place if all its reactants are present and none of its inhibitor is present; when a reaction takes place it creates its products.

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

The sets R, I, P are called the *reactant set of* b, the *inhibitor set of* b, and the *product set of* b, respectively — they are also denoted as R_b, I_b and P_b, respectively. If $R, I, P \subseteq Z$ for a finite set Z, then we say that b is a *reaction in* Z. We use $rac(Z)$ to denote the set of all reactions in Z — note that $rac(Z)$ is finite.

To define the effect of a set of reactions on a current state of the living cell we first define the effect of a single reaction.

Definition 2. *Let Z be a finite set and let $T \subseteq Z$. Let $b \in rac(Z)$. Then b is* enabled by T*, denoted by $en_b(T)$, if $R_b \subseteq T$ and $I_b \cap T = \varnothing$. The* result of b on T*, denoted by $res_b(T)$, is defined by $res_b(T) = P_b$ if $en_b(T)$, and $res_b(T) = \varnothing$ otherwise.*

Here a finite set T formalises a state of the cell, i.e., the set of biochemical entities currently present in the cell. Then b is enabled by T if T separates R_b from I_b meaning that all reactants from R_b are present in T and none of the inhibitors from I_b is present in T. When b is enabled by T it contributes its product P_b to the successor state; otherwise it does not contribute anything to the successor state.

The effect of a set of reactions on a current state of the cell is cumulative, which is formally defined as follows.

Definition 3. *Let Z be a finite set, let $T \subseteq Z$ and let $B \subseteq rac(Z)$. The* result of B on T*, denoted by $res_{\mathcal{B}}(T)$, is defined by $res_{\mathcal{B}}(T) = \bigcup\{res_b(T) : b \in B\}$.*

Note that if the transition from a current state to its successor is determined only by the reactions (i.e., there is no influence of the environment), then the successor state consists only of entities produced by the reactions enabled in the current state. This implies that in the transition from a current state to its successor state an *entity from T vanishes unless it is sustained/produced by a reaction*. This is the *non-permanency property* and it reflects the basic bioenergetics of the living cell: *without the flow/supply of energy the living cell disintegrates, but the use/absorption of energy by the living cell is realised through biochemical reactions* (see, e.g., [5]).

Although this basic definition implies 'instant non-permanency' (an entity vanishes within *one* state transition unless it is produced by a reaction), we also consider a finite duration of entities (corresponding to their presence in several consecutive states) which takes into account the decay time (see, e.g., [6]).

There is another notable aspect of Definition 3. If a, b are two reactions from B enabled by T, then both of them will take place even if $R_a \cap R_b \neq \varnothing$. Hence we do not have here the notion of conflict between reactions even if they need to share reactants. This is the property of the *threshold nature of resources: either an entity is available and then there is enough of it, or it is not available.* This property reflects the *level of abstraction* we have adopted for the formulation of our basic model: we do not count concentrations of entities/molecules to infer from these which reactions can/will be applied. We operate on a higher level of abstraction: we assume that the cell is running/functioning and we want to understand the ongoing processes.

This level of abstraction can be compared with the level of abstraction of the standard models of computation in computer science, such as Turing machines and finite automata. These standard models turned out to be very successful in understanding computational processes running on electronic computers, and yet nothing in these models takes into account the electronic/quantitative properties of the underlying hardware. It is simply assumed that the underlying electronics/hardware functions 'well' and then the goal is to understand processes running on (implemented by) this hardware. Similarly, we want to understand the processes carried out in the functioning living cell. At this stage we are not interested in the underlying 'hardware properties' of the living cell, but rather in the resulting processes.

Thus: *our basic model is qualitative rather than quantitative — in particular, there is no counting here.*

3 Reaction Systems

Now that the formal notion of a reaction and its effect on states have been established, we can proceed to define reaction systems (introduced in [4]), our abstract model 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 nonempty set and* A *is a finite subset of* $rac(S)$*.*

The set S is called the *background set of* $\mathcal{A}$, and its elements are called the *entities of* $\mathcal{A}$ — they represent molecular entities (e.g., atoms, ions, molecules) that may be present in the states of the biochemical system (e.g., the living cell). The set A is called the *set of reactions of* $\mathcal{A}$; clearly A is finite (as S is finite).

The subsets of S are called the *states of* $\mathcal{A}$. Given a state $T \subseteq S$, the *result of* $\mathcal{A}$ on T, denoted by $res_{\mathcal{A}}(T)$, is defined by $res_{\mathcal{A}}(T) = res_A(T)$.

Thus a reaction system is essentially a set of reactions. We also specify the background set which consists of entities needed for defining the reactions and for reasoning about the system (see the definition of an interactive process below). There are no 'structures' involved in reaction systems (such as, e.g., the tape of a Turing machine). Finally, note that this is a *strictly finite model* — its size is restricted by the size of the background set.

We note here that the non-permanency property is a major difference between reaction systems and the models considered in theory of computation (see, e.g., [7] and [8]). Also, the threshold nature of resources (no conflict) property is a major difference with structural models of concurrency, such as, e.g., Petri nets [9].

The model of reaction systems formalises the 'static structure' of the living cell as the set of all reactions of the cell (together with the set of underlying entities). What we are really interested in are processes instigated by the functioning of the living cell. They are formalised as follows.

Definition 5. *Let* $\mathcal{A} = (S, A)$ *be an rs. An* interactive process *in* $\mathcal{A}$ *is a pair* $\pi = (\gamma, \delta)$ *of finite sequences such that, for some* $n \geq 1$, $\gamma = C_0, \ldots, C_n$ *and* $\delta = D_0, \ldots, D_n$*, where* $C_0, \ldots, C_n, D_0, \ldots, D_n \subseteq S$, $D_0 = \varnothing$*, and* $D_i = res_{\mathcal{A}}(D_{i-1} \cup C_{i-1})$*, 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 $\tau = W_0, \ldots, W_n$, where, for all $i \in \{1, \ldots, n\}$, $W_i = C_i \cup D_i$, is the *state sequence of* π, with $W_0 = C_0$ called the *initial state*. Thus the dynamic process formalised by an interactive process π begins in the initial state W_0. The reactions of $\mathcal{A}$ enabled by W_0 produce then the result set D_1, which together with the context set C_1 forms the successor state $W_1 = res_{\mathcal{A}}(W_0) \cup C_1$. This formation of the successor state is iterated, $W_i = res_{\mathcal{A}}(W_{i-1}) \cup C_i$, resulting in the state sequence $\tau = W_0, \ldots, W_n$.

An interactive process may be visualised by a three-row representation, where the first row represents the context sets and is labelled by 'C', the second row represents result sets and is labelled by 'D', and the third row represents states and is labelled by 'W'. Thus such a representation looks as follows:

$$\begin{tikzcd}
C: & C_0 & C_1 & \cdots & C_{n-1} & C_n \\
D: & \varnothing & D_1 & \cdots & D_{n-1} & D_n \\
W: & C_0 \arrow[ur] & W_1 & \cdots & W_{n-1} \arrow[ur] & W_n
\end{tikzcd}$$

Note that an interactive process π is determined by its context sequence γ (through the result function $res_{\mathcal{A}}$). The context sequence formalises the fact that the *living cell is an open system* in the sense that it is influenced by its environment (the 'rest' of a bigger system).

If, for all $i \in \{1, \ldots, n\}$, $C_i \subseteq D_i$, then we say that π is *context-independent*: whatever C_i adds to the state W_i has already been produced by the system (included in the result D_i) or perhaps C_i adds nothing. If π is context-independent, then (in its analysis) we may as well assume that C_i adds nothing, i.e., for each $1 \leq i \leq n$, $C_i = \varnothing$. Clearly, if π is context-independent, then the initial state $W_0 = C_0$ determines π by the repeated application of $res_{\mathcal{A}}$.

4 Examples

In this section we provide two examples of use of reaction systems. The first one comes from biology — we demonstrate how to model/implement a simple generic genetic regulatory network. The second comes from theory of computation — we demonstrate how to model/implement finite transition systems (finite automata).

Example 1. We will consider genetic regulatory networks (see, e.g., [10]) which are among the most essential ingredients of the living cell. Since we give a formal/abstract model for a very complex component of the living cell, we provide first an extremely simplified (but sufficient for our purpose) description of gene expression — it is this simplified/abstract version that we will model.

Hence, for the purpose of this example, a gene g is a segment of a DNA molecule, and it consists of the promoter field followed by the coding region. The promoter plays the role of a 'landing site' for RNA polymerase. If this site is not 'occupied', then RNA polymerase can land there and then move/slide through the coding region producing its transcript in the form of a molecule

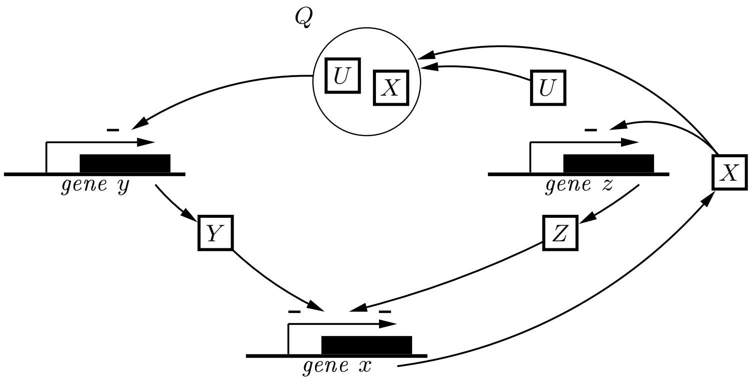


Fig. 1. A genetic regulatory network.

called messenger RNA. This messenger RNA will leave the nucleus (where DNA resides), and it will then be processed outside the nucleus, eventually yielding the protein specified by the coding region of g.

If the cell wants to interrupt the production of this protein, then it 'sends' an inhibitor molecule which lands on the promoter field. Consequently, RNA polymerase cannot land there and thus the transcription phase of the expression process cannot begin, and the protein specified by g cannot be produced anymore.

With this in mind, consider the simple generic regulatory network given in an informal graphical form in Figure 1. The network consists of three genes x, y, z expressing proteins X, Y, Z, respectively. Moreover protein X interacts with protein U (if it is present in a given state of the network) to form a protein complex Q. There are a lot of interactions going on in the network: protein X inhibits (as explained above) the expression of gene z, the presence of either of the proteins Y or Z inhibits the expression of gene x, and the protein complex Q inhibits the expression of gene y.

To implement this network by a reaction system we will need four sets of reactions: A_x, A_y, A_z implementing the expression of genes x, y, z, respectively, and A_Q implementing the formation of Q:

$$\begin{aligned}
A_x &= \big\{(\{x\}, I_x, \{x\}), (\{x\}, \{Y, Z\}, \{x'\}), (\{x, x'\}, I_{ex}, \{X\})\big\}\\
A_y &= \big\{(\{y\}, I_y, \{y\}), (\{y\}, \{Q\}, \{y'\}), (\{y, y'\}, I_{ey}, \{Y\})\big\}\\
A_z &= \big\{(\{z\}, I_z, \{z\}), (\{z\}, \{X\}, \{z'\}), (\{z, z'\}, I_{ez}, \{Z\})\big\}\\
A_Q &= \big\{(\{U, X\}, I_Q, \{Q\})\big\}
\end{aligned}$$

The set of reactions A_x implements/formalises the functioning of gene x as follows:

– $(\{x\}, I_x, \{x\})$ ensures that if x is available/functional in the current state, then it is also available in the successor state unless 'something bad' happens to x as expressed by I_x (we did not specify I_x as it is irrelevant for our considerations here, but 'something bad' may be e.g., a high level of radiation — discrete levels of radiation are easily specifiable by I_x).

- $(\{x\}, \{Y, Z\}, \{x'\})$ formalises the role of the promoter: if x is available/functional in the current state and proteins Y, Z are not present in this state, then RNA polymerase x' will land on the promoter of x.
- $(\{x, x'\}, I_{ex}, \{X\})$ formalises the role of the coding region: if x is available/functional and x' sits on the promoter in the current state, then, unless inhibited by I_{ex}, X will be expressed and hence present in the successor state.

We note here that this reaction formalises the expression of X in a very 'compact way'. However, if needed, it could be expanded to a set of reactions which formalise various details of this process.

An analogous explanation/intuition holds for the reactions in A_y and A_z. The reaction $(\{U, X\}, I_Q, \{Q\})$ ensures that if U and X are present in the current state, then Q will be present in the successor state.

Now, if we combine all these reactions for G forming $A_G = A_x \cup A_y \cup A_z \cup A_Q$, then the rs $\mathcal{A}_G = (S_G, A_G)$, with S_G consisting of all the entities occurring in reactions from A_G, implements/formalises the structure of G. The reasoning about the functioning of G is formalized through the reasoning about the processes of $\mathcal{A}_G$.

It is important to notice that in fact $\mathcal{A}_G$ is the 'union' of the reaction systems $\mathcal{A}_x = (S_x, A_x)$, $\mathcal{A}_y = (S_y, A_y)$, $\mathcal{A}_z = (S_z, A_z)$, and $\mathcal{A}_Q = (S_Q, A_Q)$, where S_x, S_y, S_z, and S_Q are all the entities occurring in reactions from A_x, A_y, A_z, and A_Q, respectively. The operation of union on reaction systems is easily defined (as sets are our basic data structure): for reaction systems $\mathcal{B}_1 = (S_1, B_1)$ and $\mathcal{B}_2 = (S_2, B_2)$, their union is the rs $(S_1 \cup S_2, B_1 \cup B_2)$.

As a matter of fact, the union of reaction systems is the basic mechanism for composing reaction systems. It expresses our assumption about bottom-up combination of local descriptions into a global picture. This combination happens 'automatically': the sheer fact that all 'ingredients' are present in the same biochemical medium (molecular soup) makes interactions possible. *There is no need for providing additional interfaces here.* This is a fundamental difference with models of computation in computer science; see, e.g., [7] and [8].

Example 2. This example relates reaction systems to the classical model of computation, viz., finite transition systems (which become finite automata once the initial and terminal states are chosen) — see, e.g., [7, 8]. In particular, we will demonstrate how transition system behaviour can be implemented by reaction systems.

We briefly recall that a *deterministic transition system* is a triplet $F = (Q, \Sigma, \delta)$, where Q is a nonempty finite set of *states*, Σ is a finite set of characters (the *input alphabet*) and $\delta : Q \times \Sigma \to Q$ is a *transition function.* Then, the *behaviour* of F is given by finite transition sequences of the form $q_0 \overset{x_1}{\to} q_1 \overset{x_2}{\to} q_2 \overset{x_3}{\to} \cdots \overset{x_n}{\to} q_n$, for some $n \geq 0$, such that $\delta(q_i, x_{i+1}) = q_{i+1}$, for each $i \in \{0, 1, \ldots, n-1\}$.

For the explanation of the implementation of F by a reaction system it is convenient to assume that $Q \cap \Sigma = \varnothing$ and $|Q \cup \Sigma| > 2$.

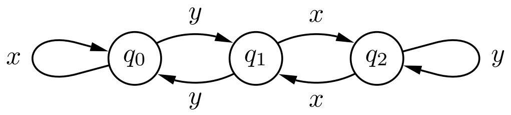
The aim of the implementation is to construct a reaction system $\mathcal{A}_F = (S_F, A_F)$ such that $q_0 \xrightarrow{x_1} q_1 \xrightarrow{x_2} q_2 \xrightarrow{x_3} \cdots \xrightarrow{x_n} q_n$ is a behaviour of F if and only if

$$\begin{tikzcd}
C: & x_1 & x_2 & x_3 & & & x_n & \varnothing \\
D: & q_0 & q_1 & q_2 & {} & \cdots & q_{n-1} & q_n \\
W: & {q_0, x_1} \arrow[ur] & {q_1, x_2} \arrow[ur] & {q_2, x_3} \arrow[ur] & & & {q_{n-1}, x_n} \arrow[ur] &
\end{tikzcd}$$

is an interactive process of the reaction system $\mathcal{A}_F$, i.e., $res_{\mathcal{A}_F}(\{q_i, x_{i+1}\}) = q_{i+1}$, for each $i \in \{0, 1, \ldots, n-1\}$. Note that here $D_0 = \{q_0\}$, while the formal definition of an interactive process requires $D_0 = \varnothing$. This is done to ease explanations; to get $D_0 = \varnothing$ one can set $C_0 = \{q_0, x_1\}$ and $D_0 = \varnothing$.

Let, for all states $p, q \in Q$ and characters $x \in \Sigma$, $a_{p,q,x}$ be the reaction defined by $(\{p, x\}, S_F \setminus \{p, x\}, \{q\})$. Then $\mathcal{A}_F = (S_F, A_F)$, where $S_F = Q \cup \Sigma$ and $A_F = \{a_{p,x,q} : \delta(p, x) = q\}$. Since we require that $I_a \neq \varnothing$, for each reaction a in a reaction system, we assumed that $|Q \cup \Sigma| > 2$ (so $S_F \setminus \{p, x\} \neq \varnothing$ as required).

The following is a deterministic transition system F (given by the graph of δ) and the list of the reactions of A_F (note that $S_F = \{q_0, q_1, q_2, x, y\}$):



$$A_F = \left\{ \begin{array}{ll} (\{q_0, x\}, \{q_1, q_2, y\}, \{q_0\}) & (\{q_0, y\}, \{q_1, q_2, x\}, \{q_1\}) \\ (\{q_1, x\}, \{q_0, q_2, y\}, \{q_2\}) & (\{q_1, y\}, \{q_0, q_2, x\}, \{q_0\}) \\ (\{q_2, x\}, \{q_0, q_1, y\}, \{q_1\}) & (\{q_2, y\}, \{q_0, q_1, x\}, \{q_2\}) \end{array} \right\}$$

Then, e.g., the transition sequence $q_1 \xrightarrow{x} q_2 \xrightarrow{y} q_2 \xrightarrow{y} q_2 \xrightarrow{x} q_1 \xrightarrow{y} q_0$ in F corresponds to the following interactive process in $\mathcal{A}_F$:

$$\begin{tikzcd}
C: & x & y & y & x & y & \\
D: & q_1 & q_2 & q_2 & q_2 & q_1 & q_0 \\
W: & {q_1, x} \arrow[ur] & {q_2, y} \arrow[ur] & {q_2, y} \arrow[ur] & {q_2, x} \arrow[ur] & {q_1, y} \arrow[ur] &
\end{tikzcd}$$

The implementation of non-deterministic finite transition systems provides an instructive insight into the role of context in interactive processes — it is done as follows. Assume that in our example transition system F the transition from q_0 on y is non-deterministic: $\delta(q_0, y) = \{q_0, q_1\}$. We mark these two transitions by symbols '1' and '2', and accordingly have two reactions: $(\{q_0, y, 1\}, \{q_1, q_2, x, 2\}, \{q_0\})$ and $(\{q_0, y, 2\}, \{q_1, q_2, x, 1\}, \{q_1\})$. Then the implementing reaction system will follow the transition from q_0 by y to q_0 if the context of the current state contains the symbol 1, and it will follow the transition from q_0 by y to q_1 if the context contains the symbol 2.

Thus, e.g., the transition sequence

$$q_0 \xrightarrow{x} q_0 \xrightarrow{y} q_0 \xrightarrow{y} q_1 \xrightarrow{x} q_2$$

in this modified F will correspond in the accordingly modified $\mathcal{A}_{\mathcal{F}}$ to the following interactive process:

$$\begin{tikzcd}
C: & x & y,1 & y,2 & x & \\
D: & q_0 & q_0 & q_0 & q_1 & q_2 \\
W: & q_0,x \arrow[ur] & q_2,y,1 \arrow[ur] & q_0,y,2 \arrow[ur] & q_1,x \arrow[ur] &
\end{tikzcd}$$

Context in interactive processes can be also used to implement stochasticity.

5 Reaction Systems with Measurements

As it was already mentioned, the model of reaction systems is qualitative, e.g., it does not include counting. However, there are many situations in biology where one needs to assign quantitative parameters to states. To account for this, reaction systems are extended to reaction systems with measurements, where numerical values are assigned to the states of a reaction system.

Our main assumption here is that a numerical value can be assigned to a state T of a reaction system if there is a measurement of T yielding this value (which is a real number). Since states of a reaction system are subsets of its background set, the informal notion of a measurement is formalised through the formal notion of a measurement function which assigns reals to the subsets of the background set. Because we deal with abstract sets (in the model of reaction systems we have no knowledge of the nature of entities of the background set), the value of a measurement function for a state must be composed from the values of the measurement function for its elements (here, for simplicity of explanation, we identify a singleton set $\{x\}$ with its element x). Therefore we assume that measurements functions are additive.

This leads to the following definition:

Definition 6.

(1) Let $\mathcal{A} = (S, A)$ be a reaction system. A measurement function for $\mathcal{A}$ *is an additive function $f : 2^S \to \mathbb{R}$.*

(2) A reaction system with measurements, *abbreviated rsm, is a triplet $\mathcal{B} = (S, A, F)$ such that (S, A) is a reaction system and F is a finite set of measurement functions.*

Recall that a function $f : 2^S \to \mathbb{R}$ is *additive* if, for all disjoint $X, Y \in 2^S$, $f(X \cup Y) = f(X) + f(Y)$; this clearly implies that $f(\varnothing) = 0$.

The dynamics of a rsm $\mathcal{B} = (S, A, F)$ is determined by its underlying reaction system $\mathcal{A} = (S, A)$. Hence, in particular, the result function of $\mathcal{B}$, $res_{\mathcal{B}}$, is equal to $res_{\mathcal{A}}$, and the interactive processes of $\mathcal{B}$ are the interactive processes of $\mathcal{A}$. The additional component F of $\mathcal{B}$ provides various global properties (measurements) for the states of $\mathcal{B}$. Since $res_{\mathcal{A}} = res_{\mathcal{B}}$, these measurements do not influence the dynamic behaviour of $\mathcal{B}$ which is identical to the dynamic behaviour of $\mathcal{A}$.

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

We will now prove that each reaction system with measurements can be replaced by an 'equivalent' (in a well-defined sense) reaction system.

Theorem 1. *For every reaction system with measurements $\mathcal{B} = (Z, B, F)$ there exists a reaction system $\mathcal{A} = (S, A)$ such that*

(i) $S = Z \cup K$, *where* $K = \{(f, r) : f \in F \textit{ and } r \in \textit{range}(f)\}$, *and* $Z \cap K = \varnothing$,
(ii) *for each* $a \in A$, $R_a \cup I_a \subseteq Z$,
(iii) *for each* $T \in 2^Z \setminus \{\varnothing, Z\}$,

$$\textit{res}_{\mathcal{A}}(T) = \textit{res}_{\mathcal{B}}(T) \cup \{(f, r) : f \in F \textit{ and } f(\textit{res}_{\mathcal{B}}(T)) = r\} .$$

Proof. Let $\mathcal{B} = (Z, B, F)$ be a reaction system with measurements.
Let $\mathcal{A} = (S, A)$ be a reaction system such that $S = Z \cup K$ and $A = B \cup L$, where:

$$\begin{aligned} K &= \{(f, r) : f \in F \textit{ and } r \in \textit{range}(f)\} \\ L &= \{(T, Z \setminus T, \{(f, r)\}) : T \in 2^Z \setminus \{\varnothing, Z\}, f \in F \textit{ and } f(\textit{res}_{\mathcal{B}}(T)) = r\} \end{aligned}$$

- Clearly, without loss of generality, we may assume that $Z \cap K = \varnothing$. Thus (i) holds.
- It follows directly from the definition of the reactions in A that, for each $a \in A$, $R_a \cup I_a \subseteq Z$. Thus (ii) holds.
- Note that since $B \subseteq A$, for each $T \in 2^Z$, $\textit{res}_{\mathcal{B}}(T) \subseteq \textit{res}_{\mathcal{A}}(T)$. Also, since $A \setminus B = L$, for each $T \in 2^Z \setminus \{\varnothing, Z\}$,

$$\textit{res}_{\mathcal{A}}(T) \setminus \textit{res}_{\mathcal{B}}(T) = \{(f, r) : f \in F \textit{ and } f(\textit{res}_{\mathcal{B}}(T)) = r\} .$$

  Therefore (iii) holds.

Thus the theorem holds. □

Note that condition (iii) from the statement of the theorem says that for each state $T \in 2^Z \setminus \{\varnothing, Z\}$, $\mathcal{A}$ computes the same successor state as $\mathcal{B}$ does, but additionally, $\mathcal{A}$ also computes the values of each measurement function of $\mathcal{B}$ for the successor state (these computed values are now a part of the corresponding successor state of $\mathcal{A}$).

The restriction in condition (iii) that $T \in 2^Z \setminus \{\varnothing, Z\}$ (rather than simply $T \in 2^Z$) is of a technical nature. It assures that, for each reaction $a \in L$, both R_a and I_a are nonempty as required by our definition of a reaction. It is not that essential in the sense that by using simple standard technical tricks one could 'skip' this assumption (adjusting somewhat the statement of the theorem).

Condition (ii) says that the values of measurement functions (hence entities from K) do not influence the applicability (enabling) of reactions in $\mathcal{A}$, which indeed corresponds to the situation in reaction systems with measurement.

We also note that condition (iii) is stated for subsets of Z rather than for subsets of S. This is sufficient, because condition (ii) implies that, for all $T \subseteq S$, $\textit{res}_{\mathcal{A}}(T) = \textit{res}_{\mathcal{A}}(T \cap Z)$.

In a nutshell, the theorem states that adding measurement functions to a reaction system is a mere 'convenience'. For every reaction system with measurements $\mathcal{B}$ one can construct a reaction system $\mathcal{A}$ which from 'inside' (through its reactions) will compute the values of all measurement functions of $\mathcal{B}$ for each state of $\mathcal{B}$ derived by $\mathcal{B}$ during production/construction of its processes.

6 Generalised Reactions

In a reaction system with measurements, the measurements of a current state (determined by the measurement functions) do not influence the successor state in the sense that they neither determine the enabling of reactions in the current state nor influence the products of enabled reactions.

We will consider now the situation where measurement functions influence (drive) the computation of the successor state. As a matter of fact, we will approach this problem by considering first a generalisation of the notion of a reaction — a special case of this generalisation will yield reactions driven by measurement functions.

Definition 7. *Let S be a finite nonempty set.*

(1) A generalised reaction *in S is an ordered pair $d = (\Delta, P)$, where Δ (the* condition *of d) is a unary relation over 2^S and P (the* product *of d) is a subset of S.*
(2) Let $d = (\Delta, P)$ be a generalised reaction in S and $T \subseteq S$. Then d is enabled by *T if $\Delta(T)$.*
(3) The result *of d, denoted res_d, is the function $res_d : 2^S \to 2^S$, for every $T \subseteq S$ defined by:*

$$res_d(T) = \begin{cases} P \text{ if } d \text{ is enabled by } T \\ \varnothing \text{ otherwise .} \end{cases}$$

It is easily seen that a generalised reaction is indeed a generalisation of the notion of reaction as considered in reaction systems. Given a reaction b in a finite set S, the corresponding generalised reaction is (Δ, P_b), where, for each $T \subseteq S$, $\Delta(T)$ if and only if b is enabled by T (i.e., $R_a \subseteq T$ and $I_a \cap T = \varnothing$).

For a finite set B of generalised reactions in S, we define the result function $res_{\mathcal{B}}$ analogously to the way it was defined for sets of ordinary reactions.

Definition 8. *A* generalised reaction system *is an ordered pair $\mathcal{B} = (S, B)$, where S is a finite set and B is a finite nonempty set of generalised reactions in S.*

Then, as was the case with ordinary reaction systems, for a state $T \subseteq S$, the *result of $\mathcal{B}$* on T is defined by $res_{\mathcal{B}}(T) = res_B(T)$.

The goal of the framework of reaction systems is to discover phenomena (described by theorems) that take place within these models (the WHAT? questions) and then provide explanations/mechanisms behind them (the WHY? questions). The explanatory mechanism is given in the form of reactions. According to this methodology, we do not accept/consider *arbitrary* generalised reactions but rather only those that can be explained by reactions (as considered in reaction systems). This leads to the following definition, where 'acceptable' really means 'acceptable in the framework of reaction systems'.

Definition 9. *A generalised reaction d in S is* acceptable *if there exists a reaction system $\mathcal{A} = (S, A)$ such that $res_d = res_{\mathcal{A}}$.*

We now give a characterisation of all acceptable generalised reactions.

Theorem 2. *Let S be a finite nonempty set. A generalised reaction $d = (\Delta, P)$ in S is acceptable if and only if $P \neq \varnothing$, there exists $T \subseteq S$ such that $\Delta(T)$ holds, and neither $\Delta(\varnothing)$ nor $\Delta(S)$ holds.*

Proof. We proceed as follows.

(1) Assume that d is acceptable. Hence, there exists a reaction system $\mathcal{A} = (S, A)$ such that $res_d = res_{\mathcal{A}}$.
   (i) Since $A \neq \varnothing$, there exists $T \subseteq S$ such that $res_{\mathcal{A}}(T) \neq \varnothing$, and so (because $res_d = res_{\mathcal{A}}$) we get $res_d(T) \neq \varnothing$.
   (ii) Since, for each $a \in A$, $P_a \neq \varnothing$, we get $P \neq \varnothing$.
   (iii) Since $res_{\mathcal{A}}(\varnothing) = res_{\mathcal{A}}(S) = \varnothing$, it follows from (ii) that neither $\Delta(\varnothing)$ nor $\Delta(S)$ holds.

(2) Assume that: $P \neq \varnothing$, there exists $T \subseteq S$ such that $\Delta(T)$ holds, and neither $\Delta(\varnothing)$ nor $\Delta(S)$ holds. Let $\mathcal{A} = (S, A)$ be the reaction system such that $A = \{(T, S \setminus T, P) : T \subseteq S \textit{ and } \Delta(T) \textit{ holds}\}$.

Since there exists $T \subseteq S$ such that $\Delta(T)$ holds, we get $A \neq \varnothing$. Moreover, since $P \neq \varnothing$, and neither $\Delta(\varnothing)$ nor $\Delta(S)$ holds, we get, for each $a \in A$, $R_a \neq \varnothing$, $I_a \neq \varnothing$ and $P_a \neq \varnothing$. Thus $\mathcal{A}$ is indeed a reaction system.

It follows directly from the definition of $\mathcal{A}$ that $res_d = res_{\mathcal{A}}$. □

We now can formalise a notion of a reaction driven by a measurement function — it will be a special case of a generalised reaction.

Let $\mathcal{A} = (S, A)$ be a reaction system and let $f : 2^S \to \mathbb{R}$ be a measurement function for $\mathcal{A}$. Then, for each $Y \subseteq range(f)$, let $\Delta_{f,Y}$ be the unary relation over 2^S, for each $T \subseteq S$ defined by: $\Delta_{f,Y}(T)$ if and only if $f(T) \in Y$.

Now, for a nonempty $P \subseteq S$ and $Y \subseteq range(f)$, $d = (\Delta_{f,Y}, P)$ is a generalised reaction in S. Note that for $T \subseteq S$, d is enabled by T if the value of $f(T)$ belongs to a predescribed set Y of 'good' values for f — therefore d is an example of a generalised reaction driven by (the values of) a measurement function.

Again, we are interested in acceptable generalised reactions (hence generalised reactions that are implementable/explainable by reaction systems). It follows directly from Theorem 2 that $d = (\Delta_{f,Y}, P)$ is acceptable if and only if

$$P \neq \varnothing, \quad Y \neq \varnothing, \quad 0 \notin Y \quad \textit{and} \quad f(S) \notin Y .$$

Hence, d is acceptable if and only if

$$P \neq \varnothing \quad \textit{and} \quad \varnothing \subset Y \subseteq range(f) \setminus \{0, f(S)\} ,$$

which says in fact that 'almost all' generalised reactions $(\Delta_{f,Y}, P)$ are acceptable.

Example 3.

(1) Let $\mathcal{A} = (S, A)$ be a reaction system such that $S = \{x, y, z, u, 2\}$, and

$$A = \left\{ \begin{matrix} (\{x,y\},\{z,u\},\{2\}) & (\{x,z\},\{y,u\},\{2\}) \\ (\{x,u\},\{y,z\},\{2\}) & (\{y,z\},\{x,u\},\{2\}) \\ (\{y,u\},\{x,z\},\{2\}) & (\{z,u\},\{x,y\},\{2\}) \end{matrix} \right\}$$

We note that, for each $T \subseteq S$, $res_{\mathcal{A}}(T) = \{2\}$ if and only if $|T \cap \{x, y, z, u\}| = 2$.

(2) Let f be a measurement function for $\mathcal{A}$ such that

$$f(x) = f(y) = f(z) = f(u) = 1 \quad \textit{and} \quad f(2) = 0 \ .$$

We note that, for each $T \subseteq S$, $f(T) = 2$ if and only if $|T \cap \{x, y, z, u\}| = 2$. Hence f globally computes (predicts) the results of $\mathcal{A}$.

(3) Consider now the generalised reaction $b = (\Delta_{f,\{2\}}, \{2\})$, and the generalised reaction system $\mathcal{B}$ with $\{x, y, z, u, 2\}$ as its background set, and b as its only (generalised) reaction.
Hence $\mathcal{B}$ with one (generalised) reaction does the same job as $\mathcal{A}$ does with six reactions. Clearly, if rather than considering the four element set $\{x, y, z, u\}$ we considered a larger set (and modified $\mathcal{A}$ accordingly), the difference would be even more dramatic.

The above example illustrates how the use of measurement functions (generalised reactions) allows for a more efficient/succinct specification of a set of processes.

7 A Generic Quantitative Model

We will now demonstrate the flexibility of reaction systems with measurements in dealing with quantitative parameters assigned to states. Rather than developing a ‘heavy’ general formal framework for demonstrating this flexibility, we will consider a generic quantitative model and then discuss how to deal with it using reaction systems with measurements.

In our considerations we do not discuss various ways of dealing with quantities in reaction systems, but instead we discuss *simulations* of other models by reaction systems. The quantitative model that we will consider is a *generic model* in the sense that we do not discuss one specific mechanism but rather a general scheme of mechanisms. This generic model is a model of DNA expression which uses a quantitative description of expression products.

In this model, we have a set G of *genes*, a set V of their *products*, and a set Q of states. Each state is an ordered pair $q = (H, \phi)$ with $H \subseteq G$ and $\phi : V \to \mathbb{R}_0$ (we use $\mathbb{R}_0$ to denote the set of nonnegative reals); ϕ is called the *quantitative component* of q, and we assume that ϕ is not the zero function (with $\phi(v) = 0$ for each $v \in V$). The intuition behind a current state $q = (H, \phi)$ is that H is the set of genes that are currently expressed, and ϕ is a quantitative description of ‘the amount’ of each product $v \in V$ currently present (the ‘amount’ can be the number of particles, mass, volume, concentration, etc.).

Since we deal with discrete time, we have a transition function $\mathit{\Gamma} : Q \to Q$ which to any given state assigns its successor state: $\mathit{\Gamma}((H, \phi)) = (K, \psi)$. This transition function may be seen as consisting of two components $\mathit{\Gamma} = (\mathit{\Gamma}_G, \mathit{\Gamma}_V)$, where:

$$\mathit{\Gamma}_G((H, \phi)) = K \quad \textit{and} \quad \mathit{\Gamma}_V((H, \phi)) = \psi \ .$$

Moreover, $\mathit{\Gamma}_V$ is given by the family of functions $\{\gamma_v : v \in V\}$, where for each $v \in V$, $\gamma_v : Q \to \mathbb{R}_0$. Then, for each $(H, \phi) \in Q$, $\mathit{\Gamma}_V((H, \phi)) = \psi$ where, for each $v \in V$, $\psi(v) = \gamma_v((H, \phi))$. Thus knowing the current state q, γ_v gives the amount of v present in the successor state of q.

As a matter of fact, for each state $q = (H, \phi)$, the quantitative component ϕ can be seen as the vector $(\phi(v_1), \ldots, \phi(v_k))$, where we assume that $V = \{v_1, \ldots v_k\}$ and V is ordered, yielding the sequence $v_1, \ldots, v_k$. Accordingly, we assume that each state $(H, \phi) \in Q$ is of the form $(H, (\phi(v_1), \ldots \phi(v_k)))$, and the transition function $\mathit{\Gamma}$ transforms a current state $(H, (\phi(v_1), \ldots, \phi(v_k)))$ into the successor state $(K, (\psi(v_1), \ldots, \psi(v_k)))$.

Now we can define a gene expression system as a 4-tuple $\mathcal{E} = (G, V, Q, \mathit{\Gamma})$ with the components G, V, Q, and $\mathit{\Gamma}$ as discussed above.

Finally, a (gene expression) *process in* $\mathcal{E}$ is a finite sequence of states

$$\pi = (H_1, Q_1), (H_2, Q_2), \ldots, (H_t, Q_t)$$

with $t \geq 2$, such that, for each $i \in \{2, \ldots, t\}$,

$$\mathit{\Gamma}((H_{i-1}, Q_{i-1})) = (H_i, Q_i) \ .$$

8 Approximations of Gene Expression Systems

When a gene expression system $\mathcal{E}$ (which is an abstract model of gene expression) is implemented, the basic step of such an implementation is an approximation of nonnegative real numbers. Assume that this implementation is done through binary numbers, where the numbers to be implemented are bounded by 2^{n_1} and their fractional part is determined with precision 2^{-n_2}. For this implementation n-bits binary numbers are used, where $n = n_1 + n_2$.

These numbers have the positional binary representation of the form:

$$2^{n_1-1} \ldots 2^1 2^0 \centerdot 2^{-1} 2^{-2} \ldots 2^{-n_2}$$

and we refer to them as (n_1, n_2)-*binary numbers.* Thus, e.g., for $n_1 = 5$ and $n_2 = 4$, the 9-bit (5,4)-binary number 100101001 represents $18 + \frac{9}{16} = 18.53125$ in the decimal notation.

Let $B_{(n_1,n_2)}$ be the subset of $\mathbb{R}_0$ represented by (n_1, n_2)-binary numbers.

Now, for each real $r \in \mathbb{R}_0$, we consider numbers b from $B_{(n_1,n_2)}$ which yield the minimal difference $|r - b|$. Clearly, either there is one such number b, or there are two such numbers b_1, b_2; in the latter case we choose the smaller of the two.

In this way, for each $r \in \mathbb{R}_0$, we obtain a unique number from $B_{(n_1,n_2)}$ which is the (n_1, n_2)-binary approximation of r, denoted $b(r)$.

Accordingly, for each function $\phi : V \to \mathbb{R}_0$ represented/defined by the vector $(\phi(v_1), \ldots, \phi(v_k))$, we obtain the (n_1, n_2)-binary approximation of ϕ, $b(\phi) : V \to B_{(n_1,n_2)}$ represented/defined by the vector $(b(\phi(v_1)), \ldots, b(\phi(v_k)))$. Finally, for each state $(H, \phi) \in Q$ we define the (n_1, n_2)-binary approximation of (H, ϕ) to be $(H, b(\phi))$.

In order to simplify the terminology we will use the phrase 'approximation' rather than '(n_1, n_2)-binary approximation' assuming that the parameters n_1, n_2 of binary numbers are fixed for our considerations.

By representing nonnegative reals through their approximations, a gene expression system $\mathcal{E} = (G, V, Q, \Gamma)$ can be transformed into a gene expression system $\widehat{\mathcal{E}} = (G, V, \widehat{Q}, \widehat{\Gamma})$ operating on $B_{(n_1,n_2)}$ rather than $\mathbb{R}_0$. Here each state $\widehat{q} \in \widehat{Q}$ is of the form $\widehat{q} = (H, \widehat{\phi})$ where $H \subseteq G$, and $\widehat{\phi} : V \to B_{(n_1,n_2)}$ is represented by the vector $(\widehat{\phi}(v_1), \ldots, \widehat{\phi}(v_k))$.

Here is one possible straightforward way of defining the transition function $\widehat{\Gamma}$ — it is given by the following commuting diagram

$$\begin{tikzcd}
(H, \widehat{\phi}) \arrow[rr, "\widehat{\Gamma}"] \arrow[dr, "\Gamma"'] & & (K, b(\psi)) \\
& (K, \psi) \arrow[ur, "b"'] &
\end{tikzcd}$$

The transition function Γ applied to a state $(H, \widehat{\phi})$ from $\widehat{Q}$ yields the intermediate successor state (K, ψ) which does not have to be in $\widehat{Q}$. However, taking the approximation $b(\psi)$ yields the state $(K, b(\psi))$ which is then the successor state of $(H, \widehat{\phi})$ in $\widehat{\mathcal{E}}$. Hence for this way of approximating $\mathcal{E}$, the transition function $\widehat{\Gamma} = (\widehat{\Gamma}_G, \widehat{\Gamma}_V)$ is defined by:

$$\widehat{\Gamma}_G((H, \widehat{\phi})) = \Gamma_G((H, \widehat{\phi})) \quad \textit{and} \quad \widehat{\Gamma}_V((H, \widehat{\phi})) = b(\Gamma_V((H, \widehat{\phi}))) \ .$$

Now, each process

$$\pi = (H_1, \phi_1), (H_2, \phi_2), \ldots, (H_t, \phi_t)$$

in $\mathcal{E}$ is approximated in $\widehat{\mathcal{E}}$ by the process

$$\widehat{\pi} = (H_1, \widehat{\delta}_1), (H_2', \widehat{\delta}_2), \ldots, (H_t', \widehat{\delta}_t)$$

such that $\widehat{\delta}_1 = b(\phi_1)$, i.e., we begin in $\widehat{\mathcal{E}}$ with the approximation $(H_1, \widehat{\delta}_1)$ of (H_1, ϕ_1) and then proceed in $\widehat{\mathcal{E}}$ through its transition function $\widehat{\Gamma}$.

Clearly there may be many ways of evaluating the quality of such an approximation $\widehat{\pi}$ of π. For example, $\widehat{\pi}$ could be classified as a good approximation if $H_2 = H_2', \ldots, H_t = H_t'$. Consequently, the quality of the approximation $\widehat{\mathcal{E}}$ could be determined by the overall quality of approximations by $\widehat{\pi}$ of π in the class of all processes π of $\mathcal{E}$.

If the quality of approximation turns out to be 'not good enough' one may either 'adjust' or 'totally redefine' the transition function $\widehat{\Gamma}$. Such modifications will in general depend on the knowledge of the nature of the actual functions ϕ involved in the states of c and on the nature of the transition function Γ.

Judging specific approximation strategies is not *our concern in this paper. Our goal is to demonstrate that, given an approximation $\widehat{\mathcal{E}}$, it can be* simulated *by a reaction system (with measurements).*

Anyhow, whatever is the exact procedure for obtaining right approximations, we end up with a system $\widehat{\mathcal{E}} = (G, V, \widehat{Q}, \widehat{\Gamma})$ which becomes an approximation of $\mathcal{E}$. In this way we move from a system $(\mathcal{E})$ with an infinite state space to a system with a finite state space $(\widehat{\mathcal{E}})$.

9 Simulating Approximations by Reaction Systems

Once an approximation $\widehat{\mathcal{E}}$ of a gene expression system $\mathcal{E}$ has been established, we will *simulate* (processes in) $\widehat{\mathcal{E}}$ by (processes in) a reaction system $\mathcal{A}(\widehat{\mathcal{E}})$. Before defining $\mathcal{A}(\widehat{\mathcal{E}})$, we introduce additional notations.

First, we establish a set representation for all numbers in $B_{(n_1,n_2)}$ as follows. For each $x \in B_{(n_1,n_2)}$, $set(x)$ is the set of all numbers

$$\ell \in \{n_1 - 1, \ldots, 0, -1, \ldots, -n_2\}$$

such that the (n_1, n_2)-binary number representing x contains 1 in the position 2^ℓ. For example, for the x represented by the $(5,4)$-binary number 100101001 (that we considered before), $set(x) = \{4, 1, -1, -4\}$, while for the y represented by the $(5,4)$-binary number 101000110, $set(y) = \{4, 2, -2, -3\}$.

Then, for each product $v \in V$ and each state $\widehat{q} = (H, \widehat{\phi}) \in \widehat{Q}$,

$$bits(\widehat{q}, v) = \{\langle v, \ell\rangle : \ell \in set(\widehat{\phi}(v))\} .$$

Intuitively, $bits(\widehat{q}, v)$ gives all the bits used in the set representation of $\widehat{\phi}(v)$, which is the amount of v in the state $\widehat{q}$. Then, for each state $\widehat{q} = (H, \widehat{\phi}) \in \widehat{Q}$,

$$bits(\widehat{q}) = \bigcup\{bits(\widehat{q}, v) : v \in V\} .$$

Intuitively, $bits(\widehat{q})$ gives all the bits used in the set representations of the amounts of v in the state $\widehat{q}$, for all $v \in V$.

Finally, for each state $\widehat{q} = (H, \widehat{\phi})$ of $\widehat{\mathcal{E}}$, we define the *simulation state of* $\widehat{q}$, denoted $sim(\widehat{q})$, by $sim(\widehat{q}) = H \cup bits(\widehat{q})$.

We also make a technical (and easy to implement) assumption about the states of $\widehat{\mathcal{E}}$:

$$\textit{For each } \widehat{q} = (H, \widehat{\phi}) \in \widehat{Q}, \quad \{\widehat{\phi}(v) : v \in V\} \subset B_{(n_1,n_2)} . \qquad (\dagger)$$

We are ready now to define the reaction system $\mathcal{A}(\widehat{\mathcal{E}})$ simulating $\widehat{\mathcal{E}}$.

Let $\mathcal{A}(\widehat{\mathcal{E}}) = (S, A)$ be a reaction system, where:

$$S = G \cup \{\langle v, \ell\rangle : v \in V \textit{ and } \ell \in \{n_1 - 1, \ldots, 1, 0, -1, \ldots, -n_2\}\} ,$$

and A consists of all reactions $a = (R, I, P) \in rac(S)$ such that there exists a state $\widehat{q} = (H, \widehat{\phi}) \in \widehat{Q}$ satisfying:

(i) $R = \mathit{sim}(\widehat{q})$,
(ii) $I = S \setminus R$, and
(iii) $P = \mathit{sim}(\widehat{\Gamma}(\widehat{q}))$.

Note that since we assumed that for each state of $\mathcal{E}$ its quantitative component is not the zero function, both $R \neq \varnothing$ and $P \neq \varnothing$. Also, because of the assumption (†) above, we have $I \neq \varnothing$.

It follows directly from the definition of $\mathcal{A}(\widehat{\mathcal{E}})$ that $\mathit{res}_{\mathcal{A}(\widehat{\mathcal{E}})}(T_1) = T_2$ for nonempty $T_1, T_2 \subseteq S$ if and only if there exist states $\widehat{q}_1, \widehat{q}_2$ of $\widehat{\mathcal{E}}$ such that

$$\widehat{\Gamma}(\widehat{q}_1) = \widehat{q}_2, \quad T_1 = \mathit{sim}(\widehat{q}_1) \quad \textit{and} \quad T_2 = \mathit{sim}(\widehat{q}_2) \ .$$

This implies that state sequences of $\mathcal{A}(\widehat{\mathcal{E}})$ (consisting of nonempty states) indeed simulate gene expression processes of $\widehat{\mathcal{E}}$, meaning that

(1) If $\widehat{\pi} = \widehat{q}_1, \widehat{q}_2, \ldots, \widehat{q}_n$ is a process in $\widehat{\mathcal{E}}$ and $\tau = T_1, T_2, \ldots, T_n$ is the state sequence of a context-independent interactive process in $\mathcal{A}(\widehat{\mathcal{E}})$ such that $T_1 = \mathit{sim}(\widehat{q}_1)$, then, for all $i \in \{2, \ldots, n\}$, $T_i = \mathit{sim}(\widehat{q}_i)$, and
(2) If $\tau = T_1, T_2, \ldots, T_n$ is the state sequence of a context-independent interactive process in $\mathcal{A}(\widehat{\mathcal{E}})$, then, for each $i \in \{1, \ldots, n\}$, there exists a state $\widehat{q}_i$ in $\widehat{\mathcal{E}}$ such that $T_i = \mathit{sim}(\widehat{q}_i)$, and $\widehat{q}_1, \widehat{q}_2, \ldots, \widehat{q}_n$ is a process in $\widehat{\mathcal{E}}$.

Now, for each $v \in V$, we define a measurement function $f_v : 2^S \to \mathbb{R}_0$ for $\mathcal{A}(\widehat{\mathcal{E}})$ by defining it on the background set S as follows:

$$f_v(x) = \begin{cases} 0 & \text{if } x \in G \\ 0 & \text{if } x = \langle u, j \rangle \text{ and } u \neq v \\ 2^j & \text{if } x = \langle v, j \rangle \ . \end{cases}$$

Thus each f_v gives the amount of v present in states of $\widehat{\mathcal{E}}$, meaning that if $\widehat{q} = (H, \widehat{\phi}) \in \widehat{Q}$, then $f_v(\mathit{sim}(\widehat{q})) = \widehat{\phi}(v)$.

Let then $\mathcal{B}(\widehat{\mathcal{E}}) = (S, A, F)$ be the reaction system with measurements such that $F = \{f_v : v \in V\}$.

Hence $\mathcal{A}(\widehat{\mathcal{E}})$ is a reaction system simulating processes in $\widehat{\mathcal{E}}$ in such a way that when it produces the state $\mathit{sim}(\widehat{q})$, which simulates/represents a state $\widehat{q} = (H, \widehat{\phi})$ in $\mathcal{A}(\widehat{\mathcal{E}})$, then it provides the (set) representation of $\widehat{\phi}$ as a part of $\mathit{sim}(\widehat{q})$ — this representation is computed ‘from inside’ of $\mathcal{A}(\widehat{\mathcal{E}})$ by its reactions. Then the reaction system with measurements $\mathcal{B}(\widehat{\mathcal{E}})$ is equipped with measurement functions f_v for each $v \in V$. For each state $\mathit{sim}(\widehat{q})$, each function f_v gives explicitly the amount of v present in $\widehat{q}$ (hence the value $\widehat{\phi}(v)$, where $\widehat{q} = (H, \widehat{\phi})$).

10 Discussion

We begin with a summary of the material presented in this paper.

We have considered (reaction systems which are) a formal framework for investigating processes inspired by the functioning of the living cell. The basic construct of this framework, reaction systems, are a qualitative model — there is no counting here. However, this framework contains various extensions of the basic model which equip reaction systems with additional components often motivated by specific research themes. Hence, e.g., it is clear that there are many situations in biology where one needs to assign quantitative parameters to states, and to account for this one considers reaction systems with measurements.

In Sections 2 and 3 we recalled the basic concepts of reaction systems, and illustrated them in Section 4, where we gave two examples, one from biology and one from theory of computation.

Then in Section 5 we considered reaction systems with measurements, where we proved that adding measurement functions is just a convenience (a useful specification macro), as for each reaction system with measurements there exists an equivalent (in a well defined sense) ordinary reaction system. Since measurement functions do not influence transitions of interactive processes (they merely state the global numerical properties of states), it is natural to consider reactions dependent on the values of measurements. We do this in Section 6 by introducing generalised reactions which allow to define the notion of a 'measurement driven reaction'. It turns our that 'almost always' such a reaction can be simulated by a set of ordinary reactions; however, again, using generalised reactions provides a convenient, often succinct, specification tool.

In Sections 7, 8 and 9 we argued that if a quantitative model is implemented (in a finite precision arithmetics), then such an implementation can be naturally simulated by reaction systems (with measurements).

Based on the material presented in Sections 5–9 we can then claim that reaction systems are quite flexible in dealing with numerical parameters assigned to states.

The framework of reaction systems is quite rich and varied. We now move to discussing a number of research topics from this framework — they are motivated either by biological considerations or by the need to understand the underlying computations.

One of the key features of reaction systems is *no permanency*: an entity is not retained in a transition of an interactive process unless it is either sustained/produced by some reaction or introduced through context. This no permanency is quite immediate — an entity that is not sustained disappears within one transition step. However, a decay of entities in a biochemical environment requires some time (decay time) to be realised. In order to account for decay time one considers in [6] *reaction systems with durations.*

Reaction systems with measurements were introduced in [11] where they were used for assigning time moments to states. In fact [11] deals with fundamental questions such as 'What is time in (models of) biochemical systems?'. 'How can one capture/formalise time in the framework of reaction systems?', and 'Which measurement functions can be used to measure time?'

Formation of modules is an important research area in biology and biochemistry, see, e.g., [12]. The formal notion of a module and its formation (by dynamic events) is discussed in [13], where it is demonstrated that interactive processes lead to the formation of modules, and that the family of all modules in a given 'stable' state of a reaction system forms a lattice. Hence the development of the living cell leads to formation of structures. It is also shown in [13] that reaction systems can be viewed as self-organising systems.

To understand how entities of a reaction system influence each other is important from both biological and computational points of view. Such *causalities* may be of a *static* nature (deducible 'directly' from the set of reactions) or of a *dynamic* nature (deducible from the set of interactive processes) — both kinds of causalities are investigated in [14].

An important line of research concerns the understanding of the result functions ($res_{\mathcal{A}}$) of reaction systems. This corresponds to the investigation of context-independent interactive processes, hence to the investigation of reaction systems as closed systems (without the influence of the environment). Here one considers reaction systems as specifications of finite functions on power sets ($res_{\mathcal{A}} : 2^S \to 2^S$), and the research focuses on the understanding (characterisation) of such functions; see, e.g., [15] and [16].

Acknowledgements

The authors are indebted to Robert Brijder and Michael Main for useful comments. This research was supported by the Pascal Chair award from the Leiden Institute of Advanced Computer Science (LIACS) of Leiden University.

References

1. Kari, L. and Rozenberg, G. (2008) The many facets of natural computing. *Commun. ACM*, **51** (10), 72–83.
2. Rozenberg, G., Bäck, T., and Kok, J. (eds) (2012) *Handbook of Natural Computing*, Springer Verlag.
3. Dawkins, R. (1986) *The Blind Watchmaker*, Penguin, Harmondsworth.
4. Ehrenfeucht, A. and Rozenberg, G. (2007) Reaction systems. *Fundam. Inform.*, **75** (1-4), 263–280.
5. Lehninger, A. (1965) *Bioenergetics: The Molecular Basis of Biological Energy Transformations*, W.A. Benjamin, Inc., New York, Amsterdam.
6. Brijder, R., Ehrenfeucht, A., and G.Rozenberg (2011) Reaction systems with duration, no. 6610 in Lecture Notes in Computer Science, pp. 191–202.
7. Arnold, A. (1994) *Finite Transition Systems: Semantics of Communicating Systems*, Prentice Hall.
8. Hopcroft, J., Motwani, R., and Ullman, J. (2006) *Introduction to Automata Theory, Languages, and Computation*, Prentice Hall.
9. Reisig, W. (1985) *Petri Nets (an Introduction)*, no. 4 in EATCS Monographs on Theoretical Computer Science, Springer, Heidelberg.
10. Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (1998) *Essential Cell Biology*, Garland Publishing, Inc.
11. Ehrenfeucht, A. and Rozenberg, G. (2009) Introducing time in reaction systems. *Theor. Comput. Sci.*, **410** (4-5), 310–322.

12. Schlosser, G. and Wagner, G. (eds) (2004) *Modularity in Development and Evolution*, The University of Chicago Press.
13. Ehrenfeucht, A. and Rozenberg, G. (2007) Events and modules in reaction systems. *Theor. Comput. Sci.*, **376** (1-2), 3–16.
14. Brijder, R., Ehrenfeucht, A., and Rozenberg, G. (2010) A note on causalities in reaction systems. *ECEASST*, **30**.
15. Ehrenfeucht, A., Main, M.G., and Rozenberg, G. (2011) Functions defined by reaction systems. *Int. J. Found. Comput. Sci.*, **22** (1), 167–178.
16. Ehrenfeucht, A., Main, M.G., and Rozenberg, G. (2010) Combinatorics of life and death for reaction systems. *Int. J. Found. Comput. Sci.*, **21** (3), 345–356.