

Views on Membranes and Ciliates

Interdisciplinary Research in Natural Computing



Universiteit Leiden

R. Brijder†, H.J. Hoogeboom, J.N. Kok, G. Rozenberg S. Hille, F. Merkl, M. Muskulus‡, S. M. Verduyn Lunel

Leiden Institute of Advanced Computer Science (LIACS)

Mathematical Institute (Leiden University)



Netherlands Organisation for Scientific Research

Contact:

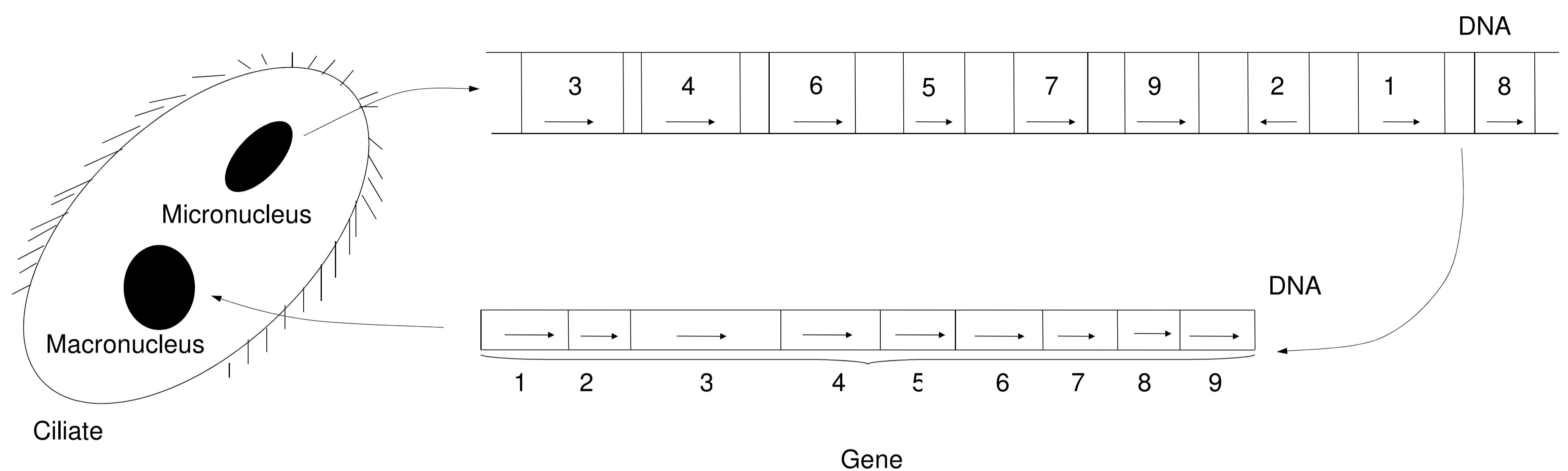
† rbrijder@liacs.nl, ‡ muskulus@math.leidenuniv.nl

H. Spink

Institute for Biology (Leiden University)

Natural Computing

A number of basic life processes can be considered as computations. Natural Computing studies these computations. We can distinguish two types of Natural Computing. One is devoted to understanding the computations in biology and the other is to develop human designed computing inspired by nature. A representative of the first type is Gene Assembly and a representative of the second type is Membrane Computing.

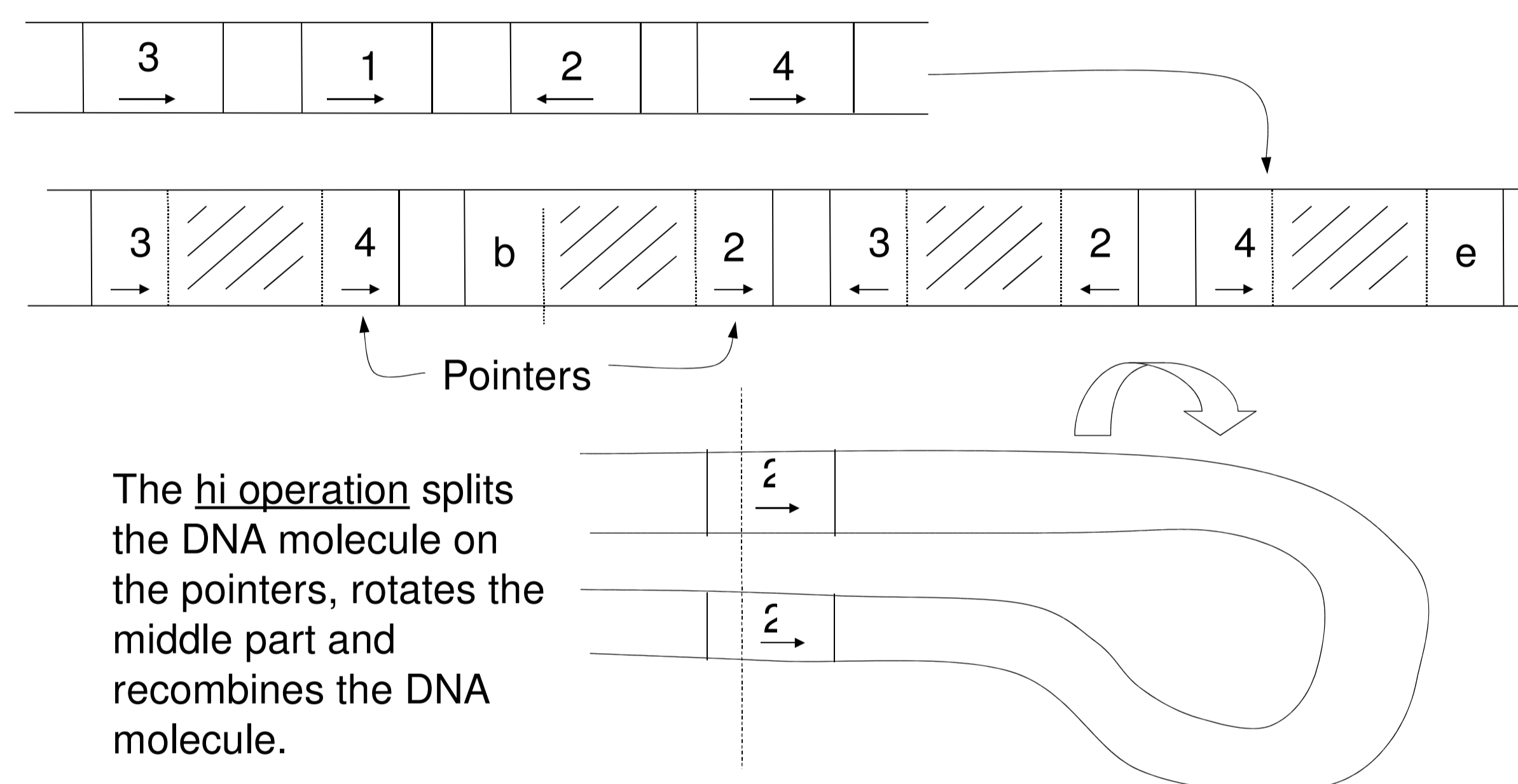


Gene Assembly

Gene Assembly is one of the most complex types of DNA transformation known in organisms. Gene assembly occurs in ciliates, a large group of one cellular organisms. Ciliates have two nuclei (instead of one in other organisms): the macronucleus and the micronucleus. The micronucleus contains chromosomes, but the genes on the chromosomes are silent: segments of the genes (called MDSs) are dispersed on the chromosomes. The macronucleus contains all genes with their segments properly aligned as separate DNA fragments. Gene Assembly is the process which transforms a micronucleus into a macronucleus. The process is non-deterministic and always halts.

How does it work?

The MDSs have pointers. These are double stranded DNA sequences found at the boundaries of MDSs. The pointers indicate how the MDSs need to be aligned to create the transcriptionally competent genes found in the macronucleus. The alignment of the MDSs is done through three molecular operations: *ld*, *hi* and *dlad*.



The *hi operation* splits the DNA molecule on the pointers, rotates the middle part and recombines the DNA molecule.

Formal Models

Two formal models are often used in Gene Assembly. One involves string rewriting. The pointers are described by numbers that form a string and the molecular operations are described by functions which operate on such strings. The occurrence of a bar on the numbers indicate the orientation of the pointers. An example:

$$3\bar{4}2 \quad \bar{3}2\bar{4} \xrightarrow{h_2} 3\bar{4}3\bar{4}$$

The other model involves graph rewriting. The pointers are described by vertices and the edges between the vertices describe the overlap of the pointers. An example:

$$3^+ \begin{matrix} 2^+ \\ 4^- \end{matrix} \xrightarrow{h_2} 3^- \begin{matrix} 2^- \\ 4^- \end{matrix}$$

Open questions and future work

Starting from these models, we are interested in possibilities for the parallel application of assembly rules. There are also interesting combinatorial questions, e.g. with respect to Intersection Graph Theory [8], that we want to address. The biological reality is also not fully understood, and we hope to elucidate some of the mechanisms used in biological Gene Assembly via extended models.

Membrane Computing

1. Introduction

Inspired by computational processes in living cells, Gheorghe Paun invented the Membrane System (or P System) model in 1998 [1]. Its features encompass spatial localization of objects inside a hierarchical membrane structure, rewriting rules on multisets of those objects, and a synchronous, non-deterministic and massively parallel execution of rule application.

So far, many researchers have created variants of P Systems [3], and most research has been centered on the computational aspects of the model, especially as concerns their grammatical complexity (in the sense of Chomsky hierarchy [4]).

Recently, some authors have become interested in modeling biological processes via (deterministic) P Systems and investigating their dynamical aspects [5, 6].

This approach is limited in a number of ways. We are therefore interested in alternative methods to represent P Systems, which make them more accessible to the large body of traditional mathematical knowledge in Dynamical Systems theory, Probability theory and Geometry. This is the reason why we would like to think of this undertaking as trying to find new "views" on P Systems.

2. Formal Definitions

A (standard) P System is a construct

$$\Pi = (V, T, C, \mu, w_1, \dots, w_m, (R_1, \rho_1), \dots, (R_m, \rho_m), \delta_0)$$

where

- (i) V is a (finite) alphabet of objects
- (ii) $T \subseteq V$ is the output alphabet
- (iii) $C \subseteq V, C \cap T = \emptyset$ are catalyst objects
- (iv) μ is a membrane structure consisting of m (labeled) membranes
- (v) $w_i, 1 \leq i \leq m$, are multisets over V associated with the regions (compartments) $1, 2, \dots, m$ of μ .
- (vi) $R_i, 1 \leq i \leq m$, are (finite) sets of evolution rules over V associated with the regions $1, 2, \dots, m$ of μ ; ρ_i is a partial order relation specifying a priority relation among the rules of R_i .
- (vii) i_0 is a number between 1 and m specifying the output membrane.

An evolution rule is a pair (u, v) , which we will write as $u \rightarrow v$, where u, v are multisets of objects.

The system starts in an initial state given by the initial objects $w_i, 1 \leq i \leq m$, and goes through a sequence of configurations. At each time step all membrane regions develop in parallel using their evolution rules. These are applied in a non-deterministic, maximal parallel way: As long as there are still rules applicable (i.e., there are enough objects available as specified on the left-hand side of a rule), one of these is randomly chosen, the objects temporarily removed and this process repeated until no more rules can be applied at this step.

In between (discrete) timesteps, the products (right-hand sides) become available for the next reactions.

Only computations that eventually lead to a sink, i.e. a configuration in which no rules can be applied anymore, are considered as successful (halting).

3. Traditional Approaches from Theoretical Computer Science

So far, many variants of P Systems have been investigated. The main emphasis has been on their grammatical complexity with respect to certain restrictions, e.g. how many membranes/objects/rules/catalysts are needed to result in computational (Turing) universality.

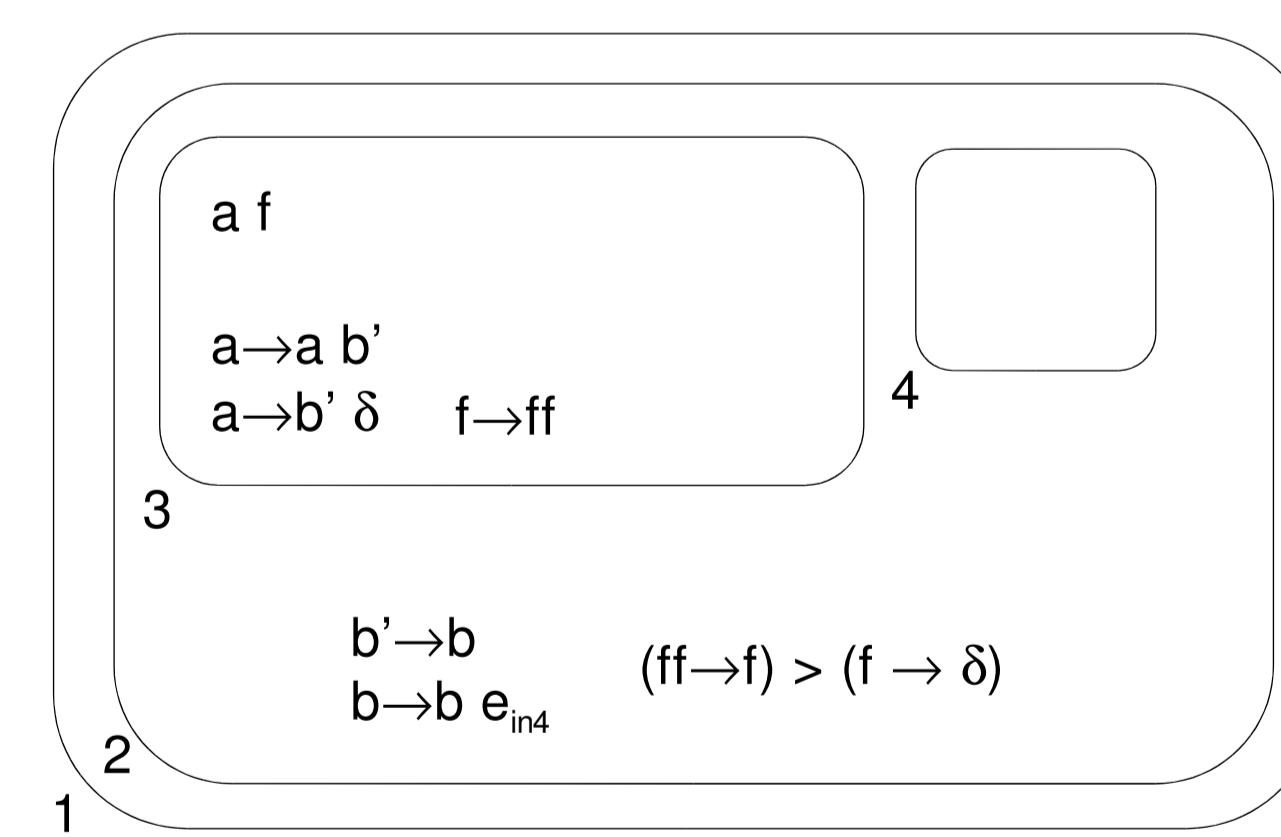


Fig. 1: Graphical representation of an example P System. This particular system generates the language $\{n^2 \mid n \geq 1\}$. Construction from Paun (2001). Note the difference between b and b' , as well as the membrane dissolution symbol δ .

4. Dynamical P Systems

Recently, P Systems have been used to model and analyze simple dynamical systems. The main approach [5, 6] starts from rewriting rules, transforms these via a so-called metabolic algorithm (in approximation of the law of mass-action) to deterministic differential equations and then solves these numerically.

The main shortcoming of this approach is that almost all original features of the P System model are lost, and one ends up with the traditional analysis of systems of coupled ODE, as in Chemical Reaction Network theory [7].

$$\begin{aligned} r1: AC &\rightarrow AB \\ r2: BC &\rightarrow A \\ r3: BBB &\rightarrow BC \end{aligned}$$

$$\begin{aligned} \Delta[A] &= -0 \cdot k_{r1}[AC] + 1 \cdot k_{r2}[BC] - 0 \cdot k_{r3}[BBB] \\ \Delta[B] &= +1 \cdot k_{r1}[AC] - 1 \cdot k_{r2}[BC] - 2 \cdot k_{r3}[BBB] \\ \Delta[C] &= -1 \cdot k_{r1}[AC] - 1 \cdot k_{r2}[BC] + 1 \cdot k_{r3}[BBB] \end{aligned}$$

Fig.2: A simple example consisting of three rewriting rules (above) and the corresponding system of metabolic equations (below). From Manca (2004).

5. Different Views & New Ideas

Our research focusses on new representations of P Systems as points in abstract configuration space. This space can be thought of as either a graph (with edges representing allowed, maximal parallel transitions) or a subset of the regular lattice \mathbb{N}^n .

In a first step we try to understand the power of maximal parallelism, which is a biologically unnatural assumption, and we investigate alternatives as already proposed in the literature: Sequential, Clock-Free and Asynchronous P Systems.

In a second step we are interested in geometrical and probabilistic properties of these representations, e.g. by considering the non-deterministic state transitions as a random walk in configuration space.

References:

- [1] Paun, G.: Computing with Membranes. TUCS Research Report No. 208, Turku, 1998.
- [2] Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D.M., Rozenberg, G.: Computation in Living Cells: Gene Assembly in Ciliates, Springer, 2004.
- [3] For an overview see: Paun, G.: Membrane Computing. An Introduction. Springer, 2002. The P Systems webpage <http://psystems.disco.unimib.it/> contains recent developments.
- [4] Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 2001.
- [5] Bernardini, F., Manca, V.: Dynamical aspects of P systems. BioSystems 70, 2003, 85-93.
- [6] Besozzi, D.: Computational and modelling power of P systems. PhD thesis, University of Milano, 2003.
- [7] see e.g.: Gunawardena, J.: Chemical reaction network theory for in-silico biologists. Lecture notes, 2003. Available from: <http://www.jeremy-gunawardena.com/papers.html>
- [8] McKee, T.A.: Topics in Intersection Graph Theory. SIAM, 1999.

Acknowledgements:

This research has been funded by NWO grant no. 635.100.006