Membrane Systems in Algebraic Biology: From a Toy to a Tool

Thomas Hinze

Friedrich Schiller University Jena
School of Biology and Pharmacy
Department of Bioinformatics

thomas.hinze@uni-jena.de

January 29, 2009
Membrane Systems: Inspired by Cells and Tissues

⇒ Capturing specialties of intra- and intercellular processes
(I) Nested Compartments Delimited by Permeable Membranes

Spatial regions wherein chemical reactions can occur
(II) Dynamics in Compartmental Cell Structure

Plasticity initiated by dedicated reaction networks

Membrane Systems in Algebraic Biology

Thomas Hinze
(III) Complex Polymeric Biomolecules in Low Concentrations

Reaction pathways forming specific biomolecules
## Outline

Membrane Systems in Algebraic Biology: From a Toy to a Tool

1. **A Primer: Reaction systems**
   composed of discrete entities

2. **Membrane systems:**
   Some introductory examples

3. **Features and varieties:**
   Classification and properties

4. $\Pi_{CSN}$: A modelling framework
   for cell signalling networks

5. **Bio-Applications:**
   Appetizers for systems biologists

6. **Membrane systems as computing devices**

7. **The P page:** An online repository and more

8. **Quo vadis:** Concluding remarks and outlook
Multiset: Molecular Configuration within a Membrane

Example

\[ \mathcal{L} = \{(A, 3), (B, 2), (C, 0), (D, 1)\} \]
\[ \text{supp}(\mathcal{L}) = \{A, B, D\} \]
\[ \text{card}(\mathcal{L}) = 6 \]

Definitions

**Multiset:** Let \( F \) be a set. A multiset over \( F \) is a mapping \( \mathcal{F} : F \rightarrow \mathbb{N} \cup \{\infty\} \) that specifies the multiplicity of each element \( a \in F \).

**Support:** Let \( \mathcal{F} : F \rightarrow \mathbb{N} \cup \{\infty\} \) be a multiset. A set \( S \subseteq F \) is called \( \text{supp}(\mathcal{F}) \) iff \( S = \{s \in F \mid \mathcal{F}(s) > 0\} \).

**Cardinality:** \( \text{card}(\mathcal{F}) := \sum_{a \in F} \mathcal{F}(a) \)
Term Rewriting: Employ a Reaction by Set Operations

Example

\{(A, 3), (B, 2), (C, 0), (D, 1)\} \ominus \{(A, 2), (B, 1)\} \uplus \{(C, 1)\} = \{(A, 1), (B, 1), (C, 1), (D, 1)\}

Multiset operations

Difference: \( \mathcal{F} \ominus \mathcal{G} := \{(a, \max(\mathcal{F}(a) - \mathcal{G}(a), 0)) \mid a \in F \setminus G\} \)

Sum: \( \mathcal{F} \uplus \mathcal{G} := \{(a, \mathcal{F}(a) + \mathcal{G}(a)) \mid a \in F \cup G\} \)

Union: \( \mathcal{F} \cup \mathcal{G} := \{(a, \max(\mathcal{F}(a), \mathcal{G}(a))) \mid a \in F \cup G\} \)

Intersection: \( \mathcal{F} \cap \mathcal{G} := \{(a, \min(\mathcal{F}(a), \mathcal{G}(a))) \mid a \in F \cap G\} \)
Possible strategies to decide among satisfied reactions

Nondeterminism: Maximal parallel enumeration of all potential scenarios

Prioritisation of reactions: Determinisation by a predefined order for applicability of reactions

Stochasticity: Randomly select a satisfied reaction
Evolution over Time

Time-discrete iteration scheme

- Starting from an initial molecular configuration $\mathcal{L}_0$
- Iterative term rewriting for transition(s) $\mathcal{L}_t \rightarrow \mathcal{L}_{t+1}$. Each iteration corresponds to a discrete period in time $\Delta \tau$
- Within each iteration turn, applicable reactions are figured out and subsequentially employed once or several times (e.g. kinetic function $f : \mathcal{L} \rightarrow \mathbb{N}$ in concert with discretised kinetic laws)
- Obtaining a derivation tree that lists sequences of molecular configurations as nodes

Considered aspects

- Suitability for small amounts of reacting particles (e.g. cell signalling)
- Compliance with mass conservance for undersatisfied reaction scenarios
First Example: A Single Membrane System

\[ \Pi_{PR} = (V, T, [1]_1, L_0, R) \]

- \( V \) .................. system alphabet
- \( T \subseteq V \) .................. terminal alphabet
- \([1]_1 \) ................ compartmental structure
- \( L_0 \subset V \times (\mathbb{N} \cup \{\infty\}) \) ...... multiset for initial configuration
- \( R = \{r_1, \ldots, r_k\} \) ............. set of reaction rules

Each reaction rule \( r_i \) consists of two multisets (reactants \( \mathcal{E}_i \), products \( \mathcal{P}_i \)) such that

\[ r_i = (\{(A_1, a_1), \ldots, (A_h, a_h)\}, \{(B_1, b_1), \ldots, (B_v, b_v)\}). \]

We write in chemical denotation:

\[ r_i : a_1 A_1 + \ldots + a_h A_h \rightarrow b_1 B_1 + \ldots + b_v B_v \]

\[ \rightarrow \text{Index } i \text{ specifies priority of } r_i: r_1 > r_2 > \ldots > r_k. \]
First Example: A Single Membrane System

\[ \Pi_{PR} = (V, T, [1]_1, \mathcal{L}_0, R) \]

- \( V \) ................................................ system alphabet
- \( T \subseteq V \) ........................................ terminal alphabet
- \([1]_1 \) ......................................... compartmental structure
- \( \mathcal{L}_0 \subset V \times (\mathbb{N} \cup \{\infty\}) \) multiset for initial configuration
- \( R = \{r_1, \ldots, r_k\} \) set of reaction rules

Each reaction rule \( r_i \) consists of two multisets (reactants \( \mathcal{E}_i \), products \( \mathcal{P}_i \)) such that

\[
 r_i = \left( \{ (A_1, a_1), \ldots, (A_h, a_h) \}, \{ (B_1, b_1), \ldots, (B_v, b_v) \} \right).
\]

We write in chemical denotation:

\[
 r_i : a_1 A_1 + \ldots + a_h A_h \rightarrow b_1 B_1 + \ldots + b_v B_v
\]

\( \Longrightarrow \) Index \( i \) specifies priority of \( r_i: r_1 > r_2 > \ldots > r_k \).
Dynamical Behaviour of $\Pi_{PR}$

Iteration scheme for configuration update
incrementing discrete time points $t \in \mathbb{N}$

$$\mathcal{L}_{t+1} = \{(a, \alpha_{a,k}) \mid \forall a \in V$$

$$\wedge \alpha_{a,0} = \text{card}(\mathcal{L}_t \cap \{(a, \infty)\})$$

$$\wedge \beta_{a,i} = \text{card}(\mathcal{E}_i \cap \{(a, \infty)\})$$

$$\wedge \gamma_{a,i} = \text{card}(\mathcal{P}_i \cap \{(a, \infty)\})$$

$$\wedge \alpha_{a,i} = \begin{cases} 
\alpha_{a,i-1} + f_i \cdot \gamma_{a,i} - f_i \cdot \beta_{a,i} \\
\alpha_{a,i-1}
\end{cases} \text{iff } \forall a \in V : \alpha_{a,i-1} \geq f_i \cdot \beta_{a,i}$$

$$\wedge i \in \{1, \ldots, k\}$$

System output (distinction empty/nonempty configuration):
$$\text{supp} \left( \bigoplus_{t=0}^{\infty} (\mathcal{L}_t \cap \{(w, \infty) \mid w \in T\}) \right) \subseteq T$$
Simulation Study of a Concrete Toy System \( \Pi_{PR} \)

\[
\Pi_{PR} = (\{Z_0, Z_1, Z_2, Z_3, Z_4, N, Y, 0, 1, \Phi, C\}, \{\Phi\}, [1], L_0, \{r_1, \ldots, r_{12}\})
\]

\[
L_0 = ((Z_0, 2), (0, 1), (1, 1), (C, 1000))
\]

\[
r_1 : Z_3 + 1 + C \rightarrow Y + \Phi + 1 \quad r_5 : Z_0 + 1 + C \rightarrow Z_1 + 1 + Z_0 \quad r_9 : Z_2 + 1 + C \rightarrow Z_3 + 1
\]

\[
r_2 : Z_4 + 0 + C \rightarrow Y + \Phi + 0 \quad r_6 : Z_0 + 0 + C \rightarrow Z_2 + 0 + Z_0 \quad r_{10} : Z_2 + 0 + C \rightarrow N + 0
\]

\[
r_3 : Y + 1 + C \rightarrow Y + \Phi + 1 \quad r_7 : Z_1 + 1 + C \rightarrow N + 1 \quad r_{11} : Z_3 + 0 + C \rightarrow N + 0
\]

\[
r_4 : Y + 0 + C \rightarrow Y + \Phi + 0 \quad r_8 : Z_1 + 0 + C \rightarrow Z_4 + 0 \quad r_{12} : Z_4 + 1 + C \rightarrow N + 1
\]

Dynamical simulation was carried out using MatLab \((\Delta \tau = 10^{-4})\). Example comes from transformation of a finite automaton.
From a Single Membrane to a Membrane Structure

Hierarchically nested membranes denoted as a tree

Representation as character string (or graph)


Extensions of Rules Beyond Reactions: Transportation

Transportation of molecules through membranes

\[ [4]_4 : \{(A, 2)\} \rightarrow [5]_5 \]

Algebraic settings available for different scenarios like

- Diffusion (unspecific transport through neighboured membranes)
- Receptors (acting as molecular filters)
- Symport/antiport (interactive molecular exchange)
Active Membranes

Rules for dynamical changes in membrane structure

\[
\begin{align*}
[h_1 \, a \, h_1] & \quad \leftrightarrow \quad [h_2 \, b \, h_1] \\
\end{align*}
\]

Specification

- Change in membrane structure $\mu$ effects further system components
- Adjustment of configuration $\mathcal{L}$ and rules $\mathcal{R}$ attached to concerned membranes
- Membrane properties like electric charge (e.g. $[:^+, \, :^-]$)

Structuring the Objects

- Algebraic representation of molecular entities or reactants as **character strings** in terms of **formal languages**
- Placeholders (\(^*\)) for handling **regular expressions** in conjunction with **matching strategies**
Classification of Membrane Systems

- **cell-like**: hierarchically nested membranes
- **tissue-like**: graph-based compartmental topology
- **neural-spiking**: pulses instead of reactions
- **population / conformons**: interacting autonomous agents

Main classes with respect to compartmental topology and principle of operation

A Membrane System for Cell Signalling Processes

- Capture significant aspects of cellular signalling:
  - Components, topology, modularity
  - Protein activation states
  - Dynamical behaviour (kinetics)
  - Signal coding and transduction
  - Coping with incomplete protein information
- Based on tissue-like membrane systems
- Keep formalism tractable
- Balance detailedness with computational needs
- Facilitate system modification, recombination, and construction *ab initio*

System Definition $\Pi_{\text{CSN}}$

- System $\Pi_{\text{CSN}} = (V, V', E, M, n)$
  - $V$: alphabet of protein identifiers
  - $V'$: alphabet of protein substructure/property identifiers
  - $M$: modules $\rightarrow$ functional reaction units
  - $E$: graph $\rightarrow$ transduction channels between modules
  - $n$: number of modules (degree of the P system)
- Modules $M_i = (R_{i1}, \ldots, R_{ir_i}, f_{i1}, \ldots f_{ir_i}, A_i) \in M$
  - $R_{ij}$: reaction rule $\rightarrow$ multisets of educts and products may contain meta-symbols $\rightarrow$ matching required
  - $f_{ij}$: function corresponding to kinetics of $R_{ij}$, number of educt objects taken from module within one reaction step
  - $A_i$: multiset of axioms $\rightarrow$ initial contents of $M_i$
- Channels $e_{ij} = (i, j, l_{ij}, d_{ij}) \in E$
  - Weighted directed channel from module $i$ to module $j$
  - $l_{ij}$: filter interface (receptor pattern and conc. gradient)
  - $d_{ij}$: time delay (number of system steps) for passage
Matching and Matching Strategies

- String-based representation of proteins
  - **String-object** $s$: representation of a protein, its properties, substructure, binding domains, activation state, ligands
  - **Structure**: $s \in V^+ \otimes (\{\#\} \otimes ((V')^+ \cup \{\neg\} \otimes (V')^+ \cup \{*\}))^*$
  - **Meta-symbols**: placeholder (wild-card) $\ast \longrightarrow$ appropriate or unknown substructure/property; **exclusion** $\neg$
  - Test whether two string-objects identify same molecule

- Matching strategies
  
  **loose:**
  - two string-objects match iff there is at least one common wildcard free representation

  **strict:**
  - participating string-objects interpreted as a pattern and a candidate (concretion of the pattern)
Matching and Matching Strategies

<table>
<thead>
<tr>
<th>Patterns</th>
<th>Cα#GDP#p</th>
<th>Cα#GDP¬p</th>
<th>Cα#GTP#p</th>
<th>Cα#GTP¬p</th>
<th>Cα##p</th>
<th>Cα##*</th>
<th>Dβ:E###*</th>
</tr>
</thead>
<tbody>
<tr>
<td>loose matching</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Cα#GDP#p | | | | | | | |
| Cα#GDP¬p | | | | | | | |
| Cα#GTP#p | | | | | | | |
| Cα#GTP¬p | | | | | | | |
| Cα##p | | | | | | | |
| Cα##* | | | | | | | |
| Dβ:E###* | | | | | | | |

V = \{Cα, Dβ:E\}
V' = \{GDP, GTP, p\}

V = \{Cα, Dβ:E\}
V' = \{GDP, GTP, p\}
System Behaviour and Properties

- Definition of dynamical system behaviour
  - Contents of module $M_i$ at global time $t \in \mathbb{N}$: multiset $L_i(t)$
  - System step by module $M_i$:

    $L_i(0) = A_i$
    $L'_i(t) = L_i(t) \ominus Educts_i(t) \cup Products_i(t)$
    $L_i(t + 1) = L'_i(t) \ominus Outgoing_i(t) \cup Incoming_i(t)$

1. Determine multiset of educts using $L_i(t), R_{i1}, \ldots, R_{ir_i}, f_{i1}, \ldots f_{ir_i}$; involves matching
2. Remove educt objects from module contents
3. Determine and add multiset of reaction products, obtain $L'_i(t)$
4. Determine and separate objects leaving host module evaluate $L'_i(t)$ and $I$ for each outgoing channel, matching
5. Add objects received from incoming channels, consider d

- System properties
  - Modularity – static system topology – ability to identify objects/substructures – flexibility in level of abstraction
  - Determinism – computational tractability – universality
Example: Yeast Pheromone Pathway

Signal transduction in *Saccharomyces cerevisiae*

\[
\Pi = (V, V', E, M, 4)
\]

\[
V = \{\text{Ste2, } \alpha, \text{G} \beta \gamma, \text{G} \alpha, \ldots\}
\]

\[
V' = \{a, \text{GDP, GTP, p}\}
\]

\[
M = \{M_1, M_2, M_3, M_4\}
\]

\[
M_1 = (R_{11}, R_{12}, R_{13}, R_{14}, R_{15}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, A_1)
\]

\[
R_{11} = \text{Ste2} \# \neg a + \alpha \rightarrow \text{Ste2} \# a
\]

\[
R_{12} = \text{Ste2} \# a \rightarrow \text{Ste2} \# \neg a
\]

\[
f_{11} = \left[ k_{11} [\text{Ste2} \# \neg a][\alpha]/V_1^2 \right]
\]

### Bio-applications

2. P System Models for Mechanosensitive Channels  
   *Ioan I. Ardelean, Daniela Besozzi, Max H. Garzon,*  
   Giancarlo Mauri, Sujay Roy .................................................. 43

3. P Systems for Biological Dynamics  
   *Luca Bianco, Federico Fontana,*  
   Giuditta Franco, Vincenzo Manca ............................................ 83

4. Modeling Respiration in Bacteria and Respiration/Photosynthesis  
   Interaction in Cyanobacteria Using a P System Simulator  
   *Matteo Cavaliere, Ioan I. Ardelean* ........................................ 129

5. Modeling Cell-Mediated Immunity by Means of P Systems  
   *Gabriel Ciobanu* ................................................................. 159

6. A Membrane Computing Model of Photosynthesis  
   *Taishin Yasunobu Nishida* ..................................................... 181

7. Modeling p53 Signaling Pathways by Using Multiset Processing  
   *Yasuhiko Suzuki, Hiroshi Tanaka* .......................................... 203

---

Membrane Systems as Universal Computing Devices

Models and concepts for biologically inspired computing

- (Bio)molecular computation
- Genetic circuits
- Cell-based computing
- Neural networks
- Gene assembly
- Evolutionary computing
- Amorphous computing

\[ \Rightarrow \text{Covered by variants of P systems} \]

\[ \Rightarrow \text{Computational completeness shown by simulation of Turing machines or equivalents} \]
Random Access Machine (RAM)

Established Turing-Complete Model for Computation

- Syntactical denotation of components
  \[ RAM = (R, L, P, l_1) \]
  - \( R \) = finite set of registers \( R = \{r_1, \ldots, r_m\}, \ r_k \in \mathbb{Z} \)
  - \( L \) = finite set of jump labels \( L = \{l_1, \ldots, l_c\} \)
  - \( P \) = program (finite set of instructions)

- Available instructions
  - \( l_i : \text{INCR}(r_k), l_j \) increment register \( r_k \), jump to \( l_j \)
  - \( l_i : \text{DECR}(r_k), l_j \) decrement register \( r_k \), jump to \( l_j \)
  - \( l_i : r_k > 0, l_j, l_p \) if \( r_k > 0 \) jump to \( l_j \) else jump to \( l_p \)
  - \( l_i : \text{HALT} \) terminate program and output

- Useful assumptions
  - Consecutive indexing of jump labels and registers
  - Determinism
  - Initialisation of registers at start
  - Output of all \( m \) registers when \( \text{HALT} \)
Simulation of Register Machines by $\Pi_{CSN}$

- Consider for simulation and transformation

$$\text{RAM} = (R, L, P, l_1)$$

- $R = \{r_1, r_2\}$
- $L = \{l_1, l_2, l_3, l_4\}$
- $P = \{l_1 : r_1 > 0, l_2, l_4, l_2 : \text{DECR}(r_1), l_3, l_3 : \text{INCR}(r_2), l_1, l_4 : \text{HALT}\}$

- Initialisation: $r_1 = 1$ and $r_2 = 0$ (arbitrarily chosen)
- Program moves contents of $r_1$ cumulatively to $r_2$
- RAM consists of $m = 2$ registers and $c = 4$ instructions
- Each register and each instruction forms separate module of membrane system $\Pi_{CSN} = (V, V', E, M, c + m)$

Membrane Systems in Algebraic Biology

Thomas Hinze
Simulation of Register Machines by $\Pi_{CSN}$

- Motivation
- Prerequisites
- Membrane Systems
- Bio-Applications
- Computing Devices
- Concluding Remarks

**PC#1**

- $l_1: r_1 > 0, l_2, l_4$

**M_1**

$PC#1 \rightarrow PC#X + D_1$

$PC#X \rightarrow PC#W + D_1$

$PC#W + 2E_1 \rightarrow PC#2$

$PC#W + H_1 \rightarrow PC#4$

**D_1**

$B + A \rightarrow \emptyset$

$D_1 + A \rightarrow A + E_1$

$D_2 + A \rightarrow A + E_2$

$D_3 + A \rightarrow A + E_3$

$D_4 + A \rightarrow A + E_4$

**M_2**

$PC#2 \rightarrow PC#3 + B$

$PC#3 \rightarrow PC#1 + A$

**M_3**

$M_5$

Register $r_1$

$B + A \rightarrow \emptyset$

$D_1 + A \rightarrow A + E_1$

$D_2 + A \rightarrow A + E_2$

$D_3 + A \rightarrow A + E_3$

$D_4 + A \rightarrow A + E_4$

**M_6**

Register $r_2$

$B + A \rightarrow \emptyset$

$D_1 + A \rightarrow A + E_1$

$D_2 + A \rightarrow A + E_2$

$D_3 + A \rightarrow A + E_3$

$D_4 + A \rightarrow A + E_4$

**M_4**

$l_4: \text{HALT}$

**M_2**

$l_2: \text{DECR}(r_1), l_3$

**M_3**

$l_3: \text{INCR}(r_2), l_1$

**PC#1**

**PC#4**

**PC#2**

**PC#3**

All functions $f = 1, g = 1, d = 0$
Simulation of Register Machines by Π_{CSN}

all functions
\[ f = 1, \quad g = 1, \quad d = 0 \]
Motivation

Prerequisites

Membrane Systems in Algebraic Biology

Bio-Applications

Computing Devices

Concluding Remarks

Simulation of Register Machines by Π_{CSN}

all functions

\[ f = 1, \quad g = 1, \quad d = 0 \]
Simulation of Register Machines by $\Pi_{CSN}$

Motivation
Prerequisites
Membrane Systems
Bio-Applications
Computing Devices
Concluding Remarks

**Simulation of Register Machines by $\Pi_{CSN}$**

**Register $r_1$:**
- $B + A \rightarrow \emptyset$
- $D_1 + A \rightarrow A + E_1$
- $D_2 + A \rightarrow A + E_2$
- $D_3 + A \rightarrow A + E_3$
- $D_4 + A \rightarrow A + E_4$

**Register $r_2$:**
- $B + A \rightarrow \emptyset$
- $D_1 + A \rightarrow A + E_1$
- $D_2 + A \rightarrow A + E_2$
- $D_3 + A \rightarrow A + E_3$
- $D_4 + A \rightarrow A + E_4$

**All functions:** $f = 1$, $g = 1$, $d = 0$
Simulation of Register Machines by $\Pi_{CSN}$

all functions $f = 1$, $g = 1$, $d = 0$

$M_1$

$l_1: r_1 > 0, l_2, l_4$

$PC\#1 \rightarrow PC\#X + D_1$
$PC\#X \rightarrow PC\#W + D_1$
$PC\#W + 2E_1 \rightarrow PC\#2$
$PC\#W + H_1 \rightarrow PC\#4$

$D_1 \rightarrow E_1, H_1$

$B + A \rightarrow \emptyset$
$D_1 + A \rightarrow A + E_1$
$D_2 + A \rightarrow A + E_2$
$D_3 + A \rightarrow A + E_3$
$D_4 + A \rightarrow A + E_4$

$M_2$

$l_2: DECR(r_1), l_3$

$PC\#2 \rightarrow PC\#3 + B$

$B + A \rightarrow \emptyset$
$D_1 + A \rightarrow A + E_1$
$D_2 + A \rightarrow A + E_2$
$D_3 + A \rightarrow A + E_3$
$D_4 + A \rightarrow A + E_4$

$M_3$

$l_3: INCR(r_2), l_1$

$PC\#3 \rightarrow PC\#1 + A$

$B + A \rightarrow \emptyset$
$D_1 + A \rightarrow A + E_1$
$D_2 + A \rightarrow A + E_2$
$D_3 + A \rightarrow A + E_3$
$D_4 + A \rightarrow A + E_4$

$M_4$

$l_4: HALT$

$PC\#4$

$B + A \rightarrow \emptyset$
$D_1 \rightarrow H_1$
$D_2 \rightarrow H_2$
$D_3 \rightarrow H_3$
$D_4 \rightarrow H_4$

$M_5$

register $r_1$

$M_6$

register $r_2$
Simulation of Register Machines by $\Pi_{CSN}$

all functions $f = 1, \ g = 1, \ d = 0$

$M_1$

$l_1 : r_1 > 0, l_2, l_4$

$PC\#1 \rightarrow PC\#X + D_1$
$PC\#X \rightarrow PC\#W + D_1$
$PC\#W + 2E_1 \rightarrow PC\#2$
$PC\#W + H_1 \rightarrow PC\#4$

$D_1 \rightarrow E_1, H_1$
$B \rightarrow PC\#2$

$M_2$

$l_2 : \text{DECR}(r_1), l_3$

$PC\#2 \rightarrow PC\#3 + B$

$PC\#3 \rightarrow PC\#1 + A$

$M_3$

$l_3 : \text{INCR}(r_2), l_1$

$A \rightarrow PC\#3$

$M_4$

$l_4 : \text{HALT}$

$PC\#1$

$PC\#2$

$M_5$

register $r_1$

$B + A \rightarrow \emptyset$

$D_1 + A \rightarrow A + E_1$
$D_2 + A \rightarrow A + E_2$
$D_3 + A \rightarrow A + E_3$
$D_4 + A \rightarrow A + E_4$

$2D_1 \rightarrow H_1$
$2D_2 \rightarrow H_2$
$2D_3 \rightarrow H_3$
$2D_4 \rightarrow H_4$

$M_6$

register $r_2$

$B + A \rightarrow \emptyset$

$D_1 + A \rightarrow A + E_1$
$D_2 + A \rightarrow A + E_2$
$D_3 + A \rightarrow A + E_3$
$D_4 + A \rightarrow A + E_4$

$2D_1 \rightarrow H_1$
$2D_2 \rightarrow H_2$
$2D_3 \rightarrow H_3$
$2D_4 \rightarrow H_4$
Simulation of Register Machines by $\Pi_{CSN}$

**Motivation**

**Prerequisites**

**Membrane Systems**

**Bio-Applications**

**Computing Devices**

**Concluding Remarks**

**Simulation of Register Machines by $\Pi_{CSN}$**

- **PC#1**: $l_1: r_1 > 0, l_2, l_4$
  - $PC#1 \rightarrow PC#X + D_1$
  - $PC#X \rightarrow PC#W + D_1$
  - $PC#W + 2E_1 \rightarrow PC#2$
  - $PC#W + H_1 \rightarrow PC#4$

- **M1**: $l_1: r_1 > 0, l_2, l_4$

- **PC#2**: $l_2: \text{DECR}(r_1), l_3$
  - $PC#2 \rightarrow PC#3 + B$
  - $PC#3 \rightarrow PC#1 + A$

- **M2**: $l_2: \text{DECR}(r_1), l_3$

- **PC#3**: $l_3: \text{INCR}(r_2), l_1$

- **M3**: $l_3: \text{INCR}(r_2), l_1$

- **M4**: $l_4: \text{HALT}$

- **M5**: Register $r_1$
  - $B + A \rightarrow \emptyset$
  - $D_1 + A \rightarrow A + E_1$
  - $2D_1 \rightarrow H_1$
  - $D_2 + A \rightarrow A + E_2$
  - $2D_2 \rightarrow H_2$
  - $D_3 + A \rightarrow A + E_3$
  - $2D_3 \rightarrow H_3$
  - $D_4 + A \rightarrow A + E_4$
  - $2D_4 \rightarrow H_4$

- **M6**: Register $r_2$
  - $B + A \rightarrow \emptyset$
  - $D_1 + A \rightarrow A + E_1$
  - $2D_1 \rightarrow H_1$
  - $D_2 + A \rightarrow A + E_2$
  - $2D_2 \rightarrow H_2$
  - $D_3 + A \rightarrow A + E_3$
  - $2D_3 \rightarrow H_3$
  - $D_4 + A \rightarrow A + E_4$
  - $2D_4 \rightarrow H_4$

- **All functions**: $f = 1$, $g = 1$, $d = 0$
Simulation of Register Machines by $\Pi_{CSN}$

all functions $f = 1$, $g = 1$, $d = 0$

Membrane Systems in Algebraic Biology

Thomas Hinze
Simulation of Register Machines by $\Pi_{CSN}$

all functions $f = 1$, $g = 1$, $d = 0$
Simulation of Register Machines by $\Pi_{CSN}$

all functions $f = 1$, $g = 1$, $d = 0$
Simulation of Register Machines by $\Pi_{CSN}$

all functions $f = 1$, $g = 1$, $d = 0$
Simulation of Register Machines by $\Pi_{CSN}$

**Motivation**

**Prerequisites**

**Membrane Systems**

**Bio-Applications**

**Computing Devices**

**Concluding Remarks**

---

**Simulation of Register Machines by $\Pi_{CSN}$**

**all functions**

$f = 1, \ g = 1, \ d = 0$

---

**Membrane Systems in Algebraic Biology**

Thomas Hinze
In this page you can find computer programs related to P systems, with the instructions to install the applications on your own computer. The requested operating systems and other minimal requirements are specified, too.

If you have any P Systems related software and if you want to put it in this page, please contact the webmaster.

- **April 2007**: Spiking Neural P Systems Simulator, by M.A. Gutierrez Naranjo and D. Ramirez Martinez, University of Sevilla, Spain
  - Click here to download the self-contained zip file of the simulator

- **November 2006**: Simulators for biological processes available at the University of Sheffield, UK
  A framework that accepts P system specifications and creates simulations for them is available at http://www.dcs.shef.ac.uk/~marian/PSimulatorWeb/PSystemMF.htm
  as well as some examples, like apoptosis, lac operon, quorum sensing etc.
  The full text of the systems specified and simulation results are also available.
  The framework may be downloaded from the address above. If you have any queries please do not hesitate to contact either
  Fran Romero-Qampero at fran@us.es or Marian Gheorghe at m_cheorghe@dcs.shef.ac.uk

- **August 2006**: Simulators for conform P systems, by Pierluigi Frisco
  - Details at the web address: http://www.mac.hac.uk/~pier/download.html

http://ppage.psystems.eu
Membrane Systems: International Dimension

- Pioneered in 1998 by Gheorghe Paun
- ≈ 10 years scientific evolution
- ≈ 100 active researchers in community
- ≈ 1,500 publications up to now

Gheorghe Păun
www.imar.ro/~paun
Quo Vadis Membrane Systems? Trends and Visions

State-of-the-art

- Establish modelling technique in Systems Biology
- Capturing aspects of inter/intracellular information proc.
- Approximation towards further bio-applications

Upcoming contributions

- Framework for reverse engineering (e.g. artificial evolution)
- Backtracking P systems
- Molecular tracing

Envisioned: “Membrane Theory"

Figures are parts of recently submitted material.