Efficient Chemical Computing using Deterministic Reaction Systems with Prioritisation of Rules

T. Hinze R. Faßler T. Lenser N. Matsumaru P. Dittrich

Bio Systems Analysis Group
Friedrich Schiller University Jena

{hinze,raf,thlenser,naoki,dittrich}
@minet.uni-jena.de
www.minet.uni-jena.de/csb

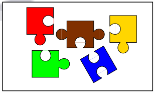
17. Theorietag
Automaten und Formale Sprachen

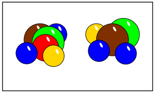
ESIGNET

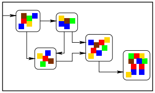
Outline

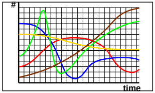
Efficient Chemical Computing using Deterministic Reaction Systems with Prioritisation of Rules

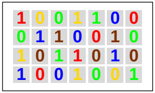
1. **Introduction:**
   Chemical reaction networks
2. **P systems Π_{PR}:**
   Motivation, definition, behaviour, properties
3. **Programming the system:**
   Emulation of NFAs
4. **Solution to the knapsack problem:**
   Idea, algorithm, simulation
5. **Conclusions and further work**











1	0	0	1	1	0	0
0	1	1	0	0	1	0
1	0	1	1	0	1	0
1	0	0	1	0	0	1

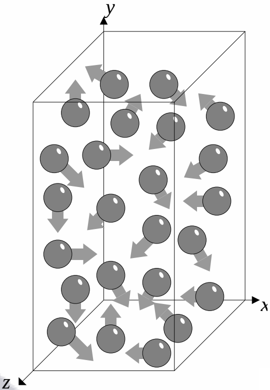
Motivation

Programming Techniques for Chemical Reaction Systems

Chemical reaction systems

- Many autonomous units (particles, molecules, agents, ...)
- Decentralisation
- Nondeterminism
- Stochasticity
- Massive data parallelism
- Communication as computation

$\Longrightarrow$ Models suitable for simulations of computational behaviour?



Applications

- Synthetic biology
- Molecular computing
- Organic computing

ESIGNET

Main Classes of Modelling Approaches

Specific Advantages and Preferred Applications

Analytic approaches

- Primarily adopted from chemical reaction kinetics
- Macroscopic view on species concentrations
- Differential equations from generation and consumption rates
- Deterministic **monitoring of temporal or spatial system behaviour**

Stochastic approaches

- Aspects of uncertainty: incorporating randomness and probabilities
- Ranges of possible scenarios and their statistical distribution
- Facilitating direct comparison with wetlab experimental data
- Statistics: **discovering correlations between network components**

Algebraic approaches

- Discrete principle of operation
- Embedding/evaluating structural information
- Modularisation, hierarchical graduation of provided system information
- **Flexible instruments** regarding level of abstraction

Main Classes of Modelling Approaches

Specific Advantages and Preferred Applications

Analytic approaches

- Primarily adopted from chemical reaction kinetics
- Macroscopic view on species concentrations
- Differential equations from generation and consumption rates
- Deterministic **monitoring of temporal or spatial system behaviour**

Stochastic approaches

- Aspects of uncertainty: incorporating randomness and probabilities
- Ranges of possible scenarios and their statistical distribution
- Facilitating direct comparison with wetlab experimental data
- Statistics: **discovering correlations between network components**

Algebraic approaches

- Discrete principle of operation
- Embedding/evaluating structural information
- Modularisation, hierarchical graduation of provided system information
- **Flexible instruments** regarding level of abstraction

Main Classes of Modelling Approaches

Specific Advantages and Preferred Applications

Analytic approaches

- Primarily adopted from chemical reaction kinetics
- Macroscopic view on species concentrations
- Differential equations from generation and consumption rates
- Deterministic **monitoring of temporal or spatial system behaviour**

Stochastic approaches

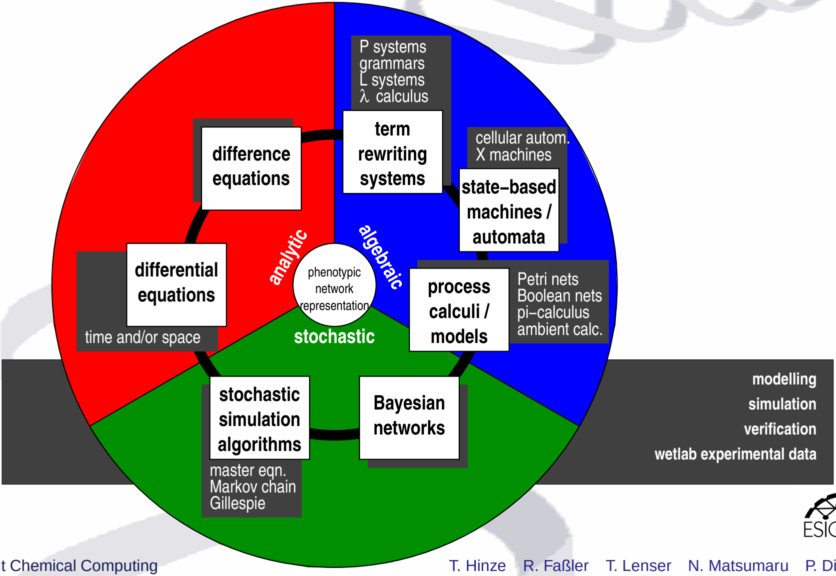
- Aspects of uncertainty: incorporating randomness and probabilities
- Ranges of possible scenarios and their statistical distribution
- Facilitating direct comparison with wetlab experimental data
- Statistics: **discovering correlations between network components**

Algebraic approaches

- Discrete principle of operation
- Embedding/evaluating structural information
- Modularisation, hierarchical graduation of provided system information
- **Flexible instruments** regarding level of abstraction

Modelling (Bio)Chemical Reaction Networks

Main Concepts within Modelling Approaches



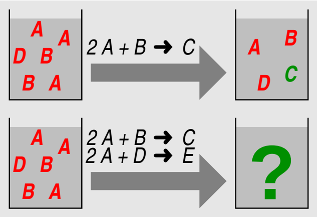
Description of Chemical Reaction Systems: Decisions

P systems in general (G. Păun)

- Algebraic denotation of system components and behaviour
- Discrete principle of operation
- Regulated term rewriting based on multisets and compartmental structures

Aspects considered for systems Π_{PR}

- Suitability for small amounts of reacting particles (e.g. cell signalling)
- Compliance with mass conservance for undersatisfied reaction scenarios
- Determinisation by consistent prioritisation of rewriting rules
- Obtaining simple computational units



System Definition

Components

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

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 E_i, products 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:

$$r_i : a_1\,A_1 + \ldots + a_h\,A_h \longrightarrow b_1\,B_1 + \ldots + b_v\,B_v$$

$\Longrightarrow$ Index i specifies priority of r_i: $r_1 > r_2 > \ldots > r_k$.

System Definition

Components

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

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 E_i, products 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:

$$r_i : a_1 A_1 + \ldots + a_h A_h \longrightarrow b_1 B_1 + \ldots + b_v B_v$$

$\Longrightarrow$ Index i specifies priority of r_i: $r_1 > r_2 > \ldots > r_k$.

System Definition

Behaviour

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

$$\begin{aligned} L_{t+1} \;=\; \{(a,\alpha_{a,k}) \mid \forall a \in V \\ & \wedge\; \alpha_{a,0} = |L_t \cap \{(a,\infty)\}| \\ & \wedge\; \beta_{a,i} = |E_i \cap \{(a,\infty)\}| \\ & \wedge\; \gamma_{a,i} = |P_i \cap \{(a,\infty)\}| \\ & \wedge\; \alpha_{a,i} = \begin{cases} \alpha_{a,i-1} + \gamma_{a,i} - \beta_{a,i} & \text{iff } \forall a \in V \;:\; \alpha_{a,i-1} \geq \beta_{a,i} \\ \alpha_{a,i-1} & \text{else} \end{cases} \\ & \wedge\; i \in \{1,\ldots,k\}\} \end{aligned}$$

System output (distinction empty/nonempty configuration):

$$\mathrm{supp}\left(\biguplus_{t=0}^{\infty}\left(L_t \cap \{(w,\infty) \mid w \in T\}\right)\right) \subseteq T$$

Programming Π_{PR} using Finite Automata

Transforming NFA $M = (Z, \Sigma, \delta, z_0, F)$ with transition table $\delta \subseteq Z \times \Sigma \times Z$, initial state $z_0 \in Z$, final states $F \subseteq Z$, $Z \cap \Sigma = \emptyset$.

Corresponding system Π_{PR}

$$
\begin{array}{rcl}
\Pi_{\text{PR}} & = & (V, T, [_1]_1, L_0, \{r_1, \ldots, r_{|\delta|}\}) \\
V & = & Z \cup \Sigma \cup \{\Phi, C\}, \text{ w.l.o.g. } \Phi, C \notin Z \cup \Sigma \\
T & = & \{\Phi\} \\
L_0 & = & \{(z_0, q) \mid q = |\{(z_0, x, q') \in \delta\}|\} \cup \\
 & & \{(w, a_w) \mid w \in \Sigma \ \wedge \ a_w = |\{(z_0, w, q') \in \delta\}|\} \cup \\
 & & \{(C, \tau)\}, \text{ with tractability parameter } \tau \in \mathbb{N} \cup \{\infty\} \\
r_{1\ldots m} & : & q + x + C \longrightarrow f + \Phi + x \quad \forall (q, x, f) \in \delta \ : \ f \in F \\
r_{m+1\ldots n} & : & z_0 + x + C \longrightarrow q' + z_0 + x \quad \forall (z_0, x, q') \in \delta \\
r_{n+1\ldots p} & : & q + x + C \longrightarrow q' + x \quad \forall (q, x, q') \in \delta \ : \ q \in Z \setminus (F \cup \{z_0\})
\end{array}
$$

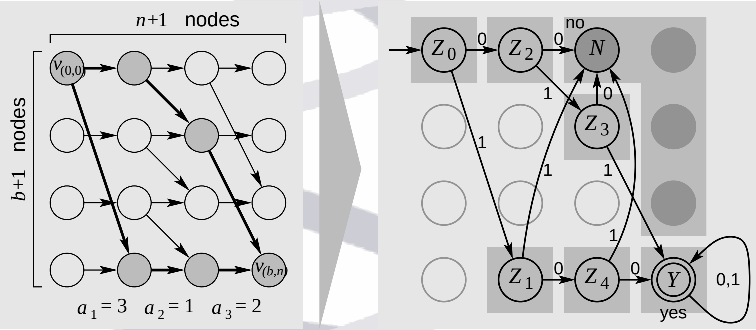
C: clock particles (enable enumeration of configurations over time)

Φ: particles indicating configurations in final state

Solution to the Knapsack Problem: Strategy

Dynamic programming approach ⟶ finite automaton

Example instance: $n = 3, a_1 = 3, a_2 = 1, a_3 = 2, b = 3$

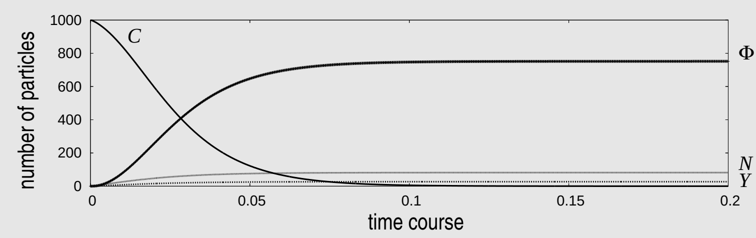


Chemical Description and Simulation Results

$$\Pi_{\mathrm{PR}} = (\{Z_0, Z_1, Z_2, Z_3, Z_4, N, Y, 0, 1, \Phi, C\}, \{\Phi\}, [_1]_1, L_0, \{r_1, \ldots, r_{12}\})$$

$$L_0 = \{(Z_0, 2), (0, 1), (1, 1), (C, \tau)\}$$

$$\begin{array}{lll}
r_1 : Z_3 + 1 + C \longrightarrow Y + \Phi + 1 & r_5 : Z_0 + 1 + C \longrightarrow Z_1 + 1 + Z_0 & r_9 : Z_2 + 1 + C \longrightarrow Z_3 + 1 \\
r_2 : Z_4 + 0 + C \longrightarrow Y + \Phi + 0 & r_6 : Z_0 + 0 + C \longrightarrow Z_2 + 0 + Z_0 & r_{10} : Z_2 + 0 + C \longrightarrow N + 0 \\
r_3 : Y + 1 + C \longrightarrow Y + \Phi + 1 & r_7 : Z_1 + 1 + C \longrightarrow N + 1 & r_{11} : Z_3 + 0 + C \longrightarrow N + 0 \\
r_4 : Y + 0 + C \longrightarrow Y + \Phi + 0 & r_8 : Z_1 + 0 + C \longrightarrow Z_4 + 0 & r_{12} : Z_4 + 1 + C \longrightarrow N + 1
\end{array}$$



Dynamical simulation was carried out using Copasi.

Particles Φ indicate problem solution ”yes”.

Conclusions

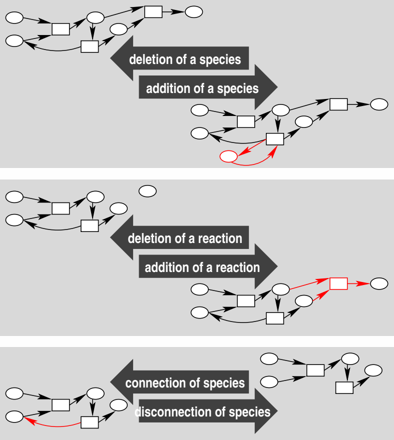
Efficient chemical computing

- Artificial chemistries / chemical computing provide efficient programming techniques
- Continuative static methods for chemical reaction networks available (e.g. fixed point analysis)
- Simulation of dynamical behaviour mostly based on deterministic model
- Consistent prioritisation of rewriting rules as one possible strategy for determinisation
- Small computational units (modules) as hierarchically evolvable components for complex systems

Further Work

Modelling and simulation of network evolution

- Artificial evolution: chemical reaction networks capable of performing computations
- Hierarchical composition: logic gates → flip-flops → registers → automata
- Comparison: hand-constructed and evolved networks for several computational tasks
- First promising results
- www.esignet.net
  www.minet.uni-jena.de/csb


deletion of a species
addition of a species
deletion of a reaction
addition of a reaction
connection of species
disconnection of species