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,thlens,nak,dittrich}
@minet.uni-jena.de
www.minet.uni-jena.de/csb
Outline

Efficient Chemical Computing using Deterministic Reaction Systems with Prioritisation of Rules

1. Introduction:
   Chemical reaction networks

2. P systems $\Pi_{PR}$:
   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
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

⇒ Models suitable for simulations of computational behaviour?

Applications

- Synthetic biology
- Molecular computing
- Organic computing
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

- difference equations
- term rewriting systems
- cellular automata / X machines
- state-based machines / automata
- Petri nets
- Boolean nets
- pi-calculus
- ambient calculus
- stochastic simulation algorithms
- Bayesian networks
- master eqn.
- Markov chain
- Gillespie
- analytic
- algebraic
- wetlab experimental data
- verification
- simulation
- modelling
- time and/or space

Efficient Chemical Computing

T. Hinze  R. Faßler  T. Lenser  N. Matsumaru  P. Dittrich
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** $\Pi_{PR}$

- 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

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

- $V$ ................................................. 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:

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

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

Efficient Chemical Computing  

T. Hinze  R. Faßler  T. Lenser  N. Matsumaru  P. Dittrich
**System Definition**

**Components**

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

- \( V \) ..................................................... 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 \( E_i \), 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:

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

\[ \implies \text{Index } i \text{ specifies priority of } r_i: r_1 > r_2 > \ldots > r_k.\]
System Definition

**Behaviour**

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

$$L_{t+1} = \left\{ (a, \alpha_{a,k}) \mid \forall a \in V \right\}$$

$$\land \alpha_{a,0} = |L_t \cap \{(a, \infty)\}|$$

$$\land \beta_{a,i} = |E_i \cap \{(a, \infty)\}|$$

$$\land \gamma_{a,i} = |P_i \cap \{(a, \infty)\}|$$

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

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

System output (distinction empty/nonempty configuration):

$$\text{supp} \left( \bigcup_{t=0}^{\infty} (L_t \cap \{(w, \infty) \mid w \in T\}) \right) \subseteq T$$
Programming $\Pi_{PR}$ 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** $\Pi_{PR}$

\[
\Pi_{PR} = (V, T, [1]_1, L_0, \{r_1, \ldots, r_{|\delta|}\})
\]

\[
V = Z \cup \Sigma \cup \{\Phi, C\}, \text{ w.l.o.g. } \Phi, C \notin Z \cup \Sigma
\]

\[
T = \{\Phi\}
\]

\[
L_0 = \{(z_0, q) \mid q = |\{(z_0, x, q') \in \delta\}|\} \cup
\]

\[
\{(w, a_w) \mid w \in \Sigma \land a_w = |\{(z_0, w, q') \in \delta\}|\} \cup
\]

\[
\{(C, \tau)\}, \text{ with tractability parameter } \tau \in \mathbb{N} \cup \{\infty\}
\]

\[
r_{1\ldots m} : q + x + C \longrightarrow f + \Phi + x \quad \forall (q, x, f) \in \delta : f \in F
\]

\[
r_{m+1\ldots n} : z_0 + x + C \longrightarrow q' + z_0 + x \quad \forall (z_0, x, q') \in \delta
\]

\[
r_{n+1\ldots p} : q + x + C \longrightarrow q' + x \quad \forall (q, x, q') \in \delta : q \in Z \setminus (F \cup \{z_0\})
\]

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

$\Phi$: particles indicating configurations in final state
Solution to the Knapsack Problem: Strategy

Dynamic programming approach $\rightarrow$ finite automaton

Example instance: $n = 3, a_1 = 3, a_2 = 1, a_3 = 2, b = 3$
Chemical Description and Simulation Results

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

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

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

Dynamical simulation was carried out using Copasi.

Particles \( \Phi \) 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 $\rightarrow$ flip-flops $\rightarrow$ registers $\rightarrow$ automata
- Comparison: hand-constructed and evolved networks for several computational tasks
- First promising results
- www.esignet.net
- www.minet.uni-jena.de/csb