Organization-oriented chemical programming for the organic design of distributed computing systems

Naoki Matsumaru and Peter Dittrich
Bio Systems Analysis Group
Jena Centre for Bioinformatics (JCB) and
Department of Mathematics and Computer Science
Friedrich-Schiller-University Jena
{naoki, dittrich}@minet.uni-jena.de

***Abstract*— Biochemical information processing found in nature is known to be robust, self-organizing, adaptive, decentralized, asynchronous, fault-tolerant, and evolvable. A couple of approaches are already using the chemical metaphor, such as, Gamma, MGS, amorphous computing, membrane computing, and reaction-diffusion processors. However, in accordance with Conrad's tradeoff principle, programming a chemical computer appears to be difficult. Therefore, in order to further exploit the mentioned properties new programming techniques are required. Here we describe how chemical organization theory can serve as a tool for chemical programming. The theory allows to predict the potential behavior of a chemical program and thus supports a programmer in the design of a chemical-like control system. The approach is demonstrated by applying it to the maximal independent set problem. We show that the desired solutions are predicted by the theory as chemical organizations. Furthermore the theory uncovers "undesirable" organizations, representing uncompleted halting computations due to insufficient amount of molecules. Finally we discuss an architecture for a "chemical virtual machine".**

I. Introduction

With respect to computation, dissimilarity between conventional computers and brains can be studied in the light of programmability and evolvability [1]. Digital computers are the realization of a highly programmable computational model, while the brain is not programmable or at least less programmable. High programmability is achieved by conventional computer languages, implying constraints on a system's operations [2]. However, evolvability is remarkably impaired since a small random change of a program causes the system to behave entirely differently or, in most cases, to stop working due to invalid operations. The robustness of a deliberately programmed systems against internal modification or even environmental change is one of the factors organic computing is aiming at [3]. Looking at the nature or natural organisms, evolvability is inevitable to cope with the ever-changing environment. Evolutionary processes are the main force for biological organisms to adapt to the environmental change. The brain model as a neural network is characterized by the continuous evolution of connections between its neurons. The achievement of high evolvability has contributed to the successful existence of biological systems.

Considering any computational system, efficiency in computation is also important. For natural systems, for example, time to process perceived information is critical. In engineering, systems are carefully designed for fast computation and low energy consumption. A computational system is regarded to be programmable, evolvable or adaptable, and computational efficient. However, Conrad's trade-off principle [4] states that it is impossible to achieve those three properties at the same time at high level. Hence, outstanding achievement of computational efficiency on a basis of programmable components results in little evolutionary adaptable systems. In other words, programmability is the cost for efficiency and adaptability found in biological information processing [4]. It should be noted that the discussion about the trade-off is rather informal and conceptual.

Since all known life forms process information using chemical processes [5], the chemical reaction metaphor has been proposed as a source of inspiration for a novel computation paradigm [6], [7]. In chemical computing, the solution appears as an emergent global behavior based on a manifold of local interactions [8]. For its heavy nonlinearity such behavior is hard to analyze and in general impossible to predict by methods that are more efficient than simulations (proof by reduction to the halting problem). There is also a common agreement that a satisfying theory of emergence is lacking [9]. Since a prerequisite for programming by construction is the ability to predict how a chemical program (e.g., a list of reaction rules) behaves [10], a theoretical analysis of emergent behavior in chemical computing is necessary. This analysis should lead to a deeper understanding of the micro-macro link between reaction rules and resulting behavior. We have suggested chemical organization theory [11] as a tool helping to construct (i.e., program) and analyze (i.e., describe and understand) chemical computing systems, which might also contribute towards establishing a theory of (chemical) emergence [12]. In this paper, we show how chemical organization theory helps programming distributed processes of chemical computing. The maximum independent set problem serves as an example.

Before we briefly review chemical organization theory we note that we are interested in approaches where chemistry stimulates the development of new computational paradigms. These approaches can be distinguished whether real or artificial chemistries are used. *Real chemical computing* employs real molecules and chemical processes to compute. For example, the simplest nonlinear function XOR can implemented with reaction-diffusion behavior of palladium chloride [13] or with the context sensitive enzyme malate dehydrogenase [14]. Here we focus on *artificial chemical computing* where the chemical metaphor is utilized to program or to build electronic computational systems. This takes the chemical metaphor as a design principle for new software or hardware architectures built on conventional silicon devices. Artificial chemical computing, thus, includes constructing chemical-like formal system in order to model and master concurrent processes [15], [16], [7].

II. Chemical Organization Theory

The target of chemical organization theory are reaction networks. A reaction network consists of a set of molecules $\mathcal{M}$ and a set of reaction rules $\mathcal{R}$. Therefore, we define a reaction network formally as a tuple $\langle \mathcal{M}, \mathcal{R} \rangle$ and call this tuple an algebraic chemistry in order to avoid conflicts with other formalizations of reaction networks.

Definition 1 (algebraic chemistry [11]): Given a set $\mathcal{M}$ of molecular species and a set of reaction rules given by the relation $\mathcal{R} : \mathcal{P}_M(\mathcal{M}) \times \mathcal{P}_M(\mathcal{M})$. We call the pair $\langle \mathcal{M}, \mathcal{R} \rangle$ an *algebraic chemistry,* where $\mathcal{P}_M(\mathcal{M})$ denotes the set of all multisets with elements from $\mathcal{M}$.

A multiset differs from an ordinary set in that it can contain multiple copies of the same element. A reaction rule is similar to a rewriting operation [15] on a multiset. Adopting the notion from chemistry, a reaction rule is written as $A \rightarrow B$ where both A and B are multi sets of molecular species. The elements of each multi set are listed with "+" symbol between them. Instead of writing $\{s_1, s_2, \ldots, s_n\}$, the set is written as $s_1 + s_2 + \cdots + s_n$ in the context of reaction rules. We also rewrite $a + a \rightarrow b$ to $2a \rightarrow b$ for simplicity. Note that "+" is not an operator but a separator of elements.

A set of molecular species is called an organization if the following two properties are satisfied: closure and self-maintenance. A set of molecular species is closed when all reaction rules applicable to the set cannot produce a molecular species that is not in the set. This is similar to the algebraic closure of an operation in set theory.

Definition 2 (closure [17]): Given an algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$, a set of molecular species $C \subseteq \mathcal{M}$ is closed, if for every reaction $(A \rightarrow B) \in \mathcal{R}$ with $A \in \mathcal{P}_M(C)$, also $B \in \mathcal{P}_M(C)$ holds.

The second important property, self-maintenance, assures, roughly speaking, that all molecules that are consumed within a self-maintaining set can also be produced by some reaction pathways within the self-maintaining set. The general definition of self-maintenance is more complicated than the definition of closure because the production and consumption of a molecular species can depend on many molecular species operating as a whole in a complex pathway.

Definition 3 (self-maintenance [11]): Given an algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$, let i denote the i-th molecular species of $\mathcal{M}$ and the j-th reaction rules is $(A_j \rightarrow B_j) \in \mathcal{R}$. Given the stoichiometric matrix $\mathbf{M} = (m_{i,j})$ that corresponds to $\langle \mathcal{M}, \mathcal{R} \rangle$ where $m_{i,j}$ denotes the number of molecules of species i produced[1] in reaction j, a set of molecular species $S \subseteq \mathcal{M}$ is self-maintaining, if there exists a flux vector $\mathbf{v} = (v_{A_1 \rightarrow B_1}, \ldots, v_{A_j \rightarrow B_j}, \ldots, v_{A_{|\mathcal{R}|} \rightarrow B_{|\mathcal{R}|}})^T$ satisfying the following three conditions:

1) $v_{A_j \rightarrow B_j} > 0$ if $A_j \in \mathcal{P}_M(S)$
2) $v_{A_j \rightarrow B_j} = 0$ if $A_j \notin \mathcal{P}_M(S)$
3) $f_i \geq 0$ if $s_i \in S$ where $(f_1, \ldots, f_i, \ldots, f_{|\mathcal{M}|})^T = \mathbf{M}\mathbf{v}$.

These three conditions can be read as follows: When the j-th reaction is applicable to the set S, the flux $v_{A_j \rightarrow B_j}$ must be positive (Condition 1). All other fluxes are set to zero (Condition 2). Finally, the production rate f_i for all the molecular species $s_i \in S$ must be nonnegative (Condition 3). Note that we have to find only one such flux vector in order to show that a set is self-maintaining.

Taking closure and self-maintenance together, we arrive at an organization:

Definition 4 (organization [11], [17]): A set of molecular species $O \subseteq \mathcal{M}$ that is closed and self-maintaining is called an organization.

We visualize the set of all organizations by a Hasse diagram, in which organizations are arranged vertically according to their size in terms of the number of their members (e.g. Fig. 1). Two organizations are connected by a line if the lower organization is contained in the organization above and there is no other organization in between.

Finally, a relevant theorem from Ref. [11] states that given a differential equation describing the dynamics of a chemical reaction system and the algebraic chemistry corresponding to that system, then the set of molecular species with positive concentrations in a fixed point (*i.e.,* stationary state), if there exists a fixed point, is an organization. In other words, we can only obtain a stationary behavior with a set of molecular species that are both closed and self-maintaining.

III. A Recipe for Chemical Programming

In this section we present a procedure for designing chemical reaction networks solving the maximal independent set problem (see Table I for a short recipe). Let an undirected graph $G = \langle V, E \rangle$ be defined by a set of N vertexes:

$$V = \{v_1, \ldots, v_N\} \tag{1}$$

and a set of edges E. When two vertexes v_p and v_q are connected, the pair of the vertexes are in the set: $(v_p, v_q) \in E$. Note that the order of the pair is insignificant, that is, $(v_p, v_q) = (v_q, v_p)$. A set of vertex $I \subset V$ is independent if no two vertexes in the set are adjacent: $(\forall v_p, v_q \in I : (v_p, v_q) \notin E)$. An independent set is maximal if no vertex can be added to the set and the set is still an independent set. A chemical reaction system is programmed to find a maximal independent set in an undirected graph.

[1] Formally, this can be defined as $m_{i,j} = \#(i \in B_j) - \#(i \in A_j)$, where $\#(i \in A_j)$ denotes the number of occurrence of species i on the lefthand side of reaction j and $\#(i \in B_j)$ the number of occurrence of species i on the righthand side of reaction j.

Given the undirected graph G, an algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$ is designed as follows. For each vertex v_j, we assign two molecular species s_j^0 and s_j^1 representing the membership of the vertex in the maximal independent set. The subscript of the species name corresponds to the index number of the vertex. High concentration, higher than a threshold chosen to be smaller than any positive coordinate of any fixed point, of species s_j^1 means that the vertex v_j is included in the maximal independent set. High concentration of species s_j^0 represents that the vertex v_j is *not* included in the maximal independent set. Thus the set of molecular species $\mathcal{M}$ contains $2N$ molecular species:

$$\mathcal{M} = \{s_j^0, s_j^1 \mid j = 1, \ldots, N\} \tag{2}$$

The set of reaction rules $\mathcal{R}$ is constructed by assembling reactions for each vertex:

$$\mathcal{R} = \bigcup_{i=1}^{N} \mathcal{R}^i = \bigcup_{i=1}^{N} (\mathcal{V}^i \cup \mathcal{N}^i \cup \mathcal{D}^i). \tag{3}$$

For each reaction set $\mathcal{R}^i$, there are three sorts of reactions. The first two sorts are adapted from two predicates constituting a program for any distributed processor to solve maximal independent set problem under a central scheduler [18]. A reaction rule to produce species s_i^1 is the first:

$$\mathcal{V}^i = (\overbrace{s_j^0 + s_k^0 + \cdots + s_l^0}^{n_i} \rightarrow n_i s_i^1) \tag{4}$$

where n_i is the number of vertexes connected to vertex v_i and $v_j, v_k, \ldots, v_l$ are these neighboring vertices, that is, $(v_i, v_j), (v_i, v_k), \ldots, (v_i, v_l) \in E$. The left hand side of the reaction contains n_i terms, and this reaction is interpreted as follows: When no neighboring vertex is included in the maximal independent set, the target vertex v_i should be included in the set.

The negation of this predicate is considered by a set of n_i reactions:

$$\mathcal{N}^i = \{s_j^1 \rightarrow s_i^0 | (v_i, v_j) \in E\}. \tag{5}$$

This is the second type of the reactions, which produce species s_i^0 from any species corresponding to the neighboring vertexes with superscript 1. This rule can be interpreted as follows: If there exists at lease one neighboring vertex included in the maximal independent set, then the target vertex v_i should be excluded from the maximal independent set (otherwise the definition of the maximal independent set would be violated). Generating species s_i^0 forces vertex v_i not to be included in the set.

The last component of set $\mathcal{R}^i$ is a *destructive reaction*. Since the membership of the maximal independent set is a binary state, the state becomes undefined when neither or both of the species are present. In order to avoid the latter case, the two opposite molecular species are defined to vanish upon collision:

$$\mathcal{D}^i = s_i^0 + s_i^1 \rightarrow \emptyset. \tag{6}$$

TABLE I

RECIPE FOR MAPPING AN UNDIRECTED GRAPH TO A CHEMICAL REACTION NETWORK.

Input: Undirected graph $G = \langle V, E \rangle$ where V is a set of N vertexes $V = \{v_1, \ldots, v_N\}$ and E is a set of edges. When two vertexes v_p and v_q are connected, $(v_p, v_q) \in E$.
Output: Algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$ (a set of molecular species $\mathcal{M}$ and a set of reaction rules $\mathcal{R}$) representing the chemical program to solve maximal independent set problem.
Algorithm:

```
1. For each vertex $v_j$:
(a) Add two molecular species, $s_j^0$ and $s_j^1$, to $\mathcal{M}$;[a]
(b) Add one *destructive reaction* of the form $s_j^0 + s_j^1 \rightarrow \emptyset$ to $\mathcal{R}$;
(c) Add one reaction to $\mathcal{R}$ of the form:
        $(\cdots + s_i^0 + \ldots \rightarrow n_j s_j^1)$
    where $n_j$ is the number of edges connected to vertex $v_j$ and $(v_j, v_i) \in E$.
(d) Add a set of $n_j$ reactions to $\mathcal{R}$:
        $\{s_i^1 \rightarrow s_j^0 | (v_i, v_j) \in E\}$.
```

[a] As a naming convention of molecular species in this paper, the superscript indicates the membership for the maximal independent set.

Note that the algebraic chemistry is defined such that molecules react only if they are located on the same vertex or are neighbors. Thus, the resulting (artificial) chemical system can be interpreted as a spatially distributed compartmentalized reaction system, where a compartment j holds only the two chemical species representing a vertex v_j, namely s_j^0 and s_j^1 and where the topological structure of the compartments is equivalent to the undirected graph.

Algorithms to solve the maximal independent set problem are theoretically studied (e.g., [19]) and discussed especially in the context of distributed processors [18], [20]. In the next section, we exemplify how the chemical organization theory helps to understand the potential dynamical behavior of the chemical program for the maximal independent set problem.

IV. EXAMPLE OF CHEMICAL PROGRAMMING TO SOLVE MAXIMAL INDEPENDENT SET PROBLEM

To demonstrate how chemical organization theory can be used to understand the potential behavior of a chemical program, various reaction networks are designed for specific instances of the maximal independent set problem in accordance with the recipe shown in Table I.

A. Linear graph with three nodes

Provided that an undirected graph $G = \langle V, E \rangle$ consists of three vertexes and those vertexes are connected linearly as shown in Fig. 1 (A):

$$G = \langle V = \{v_1, v_2, v_3\}, E = \{(v_1, v_2), (v_2, v_3)\rangle. \tag{7}$$

Following the recipe, an algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$ is constructed. The set of molecular species $\mathcal{M}$ consists of six species because the graph contains $N = 3$ vertexes:

$$\mathcal{M} = \{s_1^0, s_1^1, s_2^0, s_2^1, s_3^0, s_3^1\}. \tag{8}$$

Our naming convention for the species is that the subscript of the species name is associated with the index of the graph vertex and that the superscript stands for the membership of the maximal independent set. For example, species s_2^1 stands for vertex v_1 is included in the maximal independent set, and s_2^0 represents otherwise for the same vertex.

For each vertex v_1, v_2, and v_3, reaction rules are constructed. The destructive reactions are:

$$\mathcal{D} = \bigcup_{i=1}^{3} \mathcal{D}^i = \{s_1^0 + s_1^1 \to \emptyset, s_2^0 + s_2^1 \to \emptyset, s_3^0 + s_3^1 \to \emptyset\}.$$

The reaction rules to produce positive membership species are composed of three reactions:

$$\mathcal{V} = \bigcup_{i=1}^{3} \mathcal{V}^i = \{s_2^0 \to s_1^1, s_1^0 + s_3^0 \to 2s_2^1, s_2^0 \to s_3^1\}$$

Finally, the non-membership species are also produced:

$$\mathcal{N} = \bigcup_{i=1}^{3} \mathcal{N}^i = \{s_2^1 \to s_1^0, s_1^1 \to s_2^0, s_3^1 \to s_2^0, s_2^1 \to s_3^0\}$$

The whole set of reactions $\mathcal{R}$ results in:

$$\begin{aligned} \mathcal{R} &= \mathcal{V} \cup \mathcal{N} \cup \mathcal{D} \\ &= \{s_2^0 \to s_1^1, s_2^1 \to s_1^0, s_1^0 + s_3^0 \to 2s_2^1, \\ &\quad s_1^1 \to s_2^0, s_3^1 \to s_2^0, s_2^0 \to s_3^1, s_2^1 \to s_3^0, \\ &\quad s_1^0 + s_1^1 \to \emptyset, s_2^0 + s_2^1 \to \emptyset, s_3^0 + s_3^1 \to \emptyset\} \end{aligned} \quad (9)$$

The algebraic chemistry is analyzed for hierarchical organizational structure within the reaction network. When applying chemical organization theory (Section II), the chemical reaction network is decomposed into a hierarchy of overlapping sub-networks, called organizations. These organizations provide an overview of the potential (emergent) behavior of the system because only a set of molecular species forming an organization can be stable [11]. Furthermore, the dynamics of the system can be explained as a transition between organizations instead of a movement in the potentially more complex state space.

In our example, the reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ possesses five organizations:

$$O = \{\emptyset, \{s_1^0\}, \{s_3^0\}, \{s_1^0, s_2^1, s_3^0\}, \{s_1^1, s_2^0, s_3^1\}\} \quad (10)$$

Figure 1 (B) visualizes these organizations as a Hasse diagram. In passing we note that the organizations do not form a lattice, because there is not a unique largest organizations.

The two largest organizations represent the two desired solution to the maximal independent set problem, namely “010” and “101”. This guartees that a dynamical reaction system implementing that algebraic chemistry can have stationary states representing the desired solutions (cf. [11]) and that other solutions that consists of species that are not an organization can not stably exist.

Interestingly the analysis has also uncovered three smaller organizations. These organizations represent uncompleted computation due to a lack of molecules. For example, the empty organization trivially implies: if there are no molecules in the system, no molecule will enter the system and there will be no computation. If we setup our chemical computing system such that these small organizations are avoided (e.g., by adding enough initial molecules), the system must produce a solution.

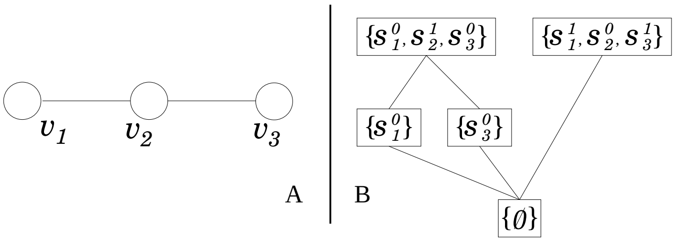


Fig. 1. Analysis of a chemical program with organization theory. (A) Graph structure and (B) hierarchy of organizations within the chemical reaction network for the maximal independent set problem for the linear 3-vertex graph.

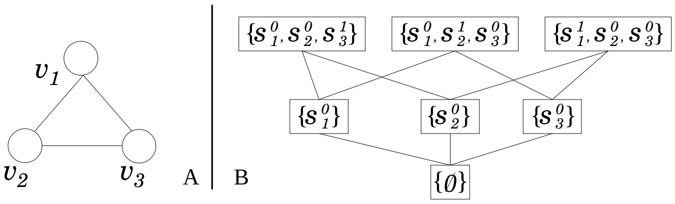

v_1
v_2
v_3
A
B
$\{s_1^0, s_2^0, s_3^1\}$
$\{s_1^0, s_2^1, s_3^0\}$
$\{s_1^1, s_2^0, s_3^0\}$
$\{s_1^0\}$
$\{s_2^0\}$
$\{s_3^0\}$
$\{\emptyset\}$

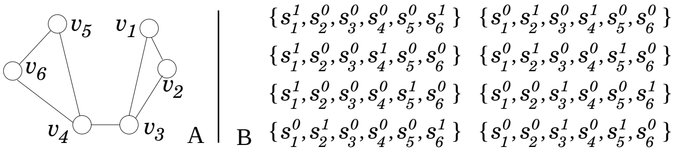
Fig. 2. Analysis of a chemical program with organization theory. (A) Graph structure and (B) hierarchy of organizations within the chemical reaction network for the maximal independent set problem for the circular 3-vertex graph.

We can now ask whether these solutions, organizations $\{s_1^0, s_2^1, s_3^0\}$ and $\{s_1^1, s_2^0, s_3^1\}\}$, are stable or whether the system, once they have been found, might move spontaneously down to a smaller organization below them. In general, this type of question requires to investigate the dynamics, such as, rate constants, in detail. Here, however, we can see already by looking at the reaction rules that organization $\{s_1^1, s_2^0, s_3^1\}\}$ must be stable, because all reactions are mass-conserving so that the empty organization (the only organization below) can never be reached. The situation with organization $\{s_1^1, s_2^0, s_3^1\}\}$ is more complicated, because it contains also the small organizations $\{s_1^0\}$ and $\{s_3^0\}$. So we can not use the same argument as before. The stability of that organization depends on the kinetics applied (not shown here).

B. Circular graph with three nodes

The similar discussion is applicable to the circular graph structure. For instance, three vertexes are connected as depicted in Fig. 2 (A) to form a circular structure. The undirected graph can be defined as follows:

$$G = \langle V = \{v_1, v_2, v_3\}, E = \{(v_1, v_2), (v_2, v_3), (v_1, v_3)\}\rangle. \quad (11)$$



$\{s_1^1, s_2^0, s_3^0, s_4^0, s_5^0, s_6^1\}$ $\{s_1^0, s_2^1, s_3^0, s_4^1, s_5^0, s_6^0\}$
$\{s_1^1, s_2^0, s_3^0, s_4^1, s_5^0, s_6^0\}$ $\{s_1^0, s_2^1, s_3^0, s_4^0, s_5^1, s_6^0\}$
$\{s_1^1, s_2^0, s_3^0, s_4^0, s_5^1, s_6^0\}$ $\{s_1^0, s_2^1, s_3^0, s_4^0, s_5^0, s_6^1\}$
$\{s_1^0, s_2^1, s_3^0, s_4^0, s_5^0, s_6^1\}$ $\{s_1^0, s_2^0, s_3^1, s_4^0, s_5^1, s_6^0\}$

Fig. 3. Analysis of a chemical program with organization theory. (A) Graph structure with six vertexes and seven edges. (B) The largest organizations within the chemical reaction network for the maximal independent set problem for the graph. Each organization with the size of six corresponds to a solution to the maximal independent set problem.

According to the recipe, an algebraic chemistry is constructed, and the resulting algebraic chemistry is following:

$$\mathcal{M} = \{s_1^0, s_1^1, s_2^0, s_2^1, s_3^0, s_3^1\} \tag{12}$$

and

$$\begin{aligned} \mathcal{R} \;\; = \;\; & \{s_2^0 + s_3^0 \rightarrow 2s_1^1, s_2^1 \rightarrow s_1^0, s_3^1 \rightarrow s_1^0, \\ & s_1^0 + s_3^0 \rightarrow 2s_2^1, s_1^1 \rightarrow s_2^0, s_3^1 \rightarrow s_2^0, \\ & s_1^0 + s_2^0 \rightarrow s_3^1, s_1^1 \rightarrow s_3^0, s_2^1 \rightarrow s_3^0, \\ & s_1^0 + s_1^1 \rightarrow \emptyset, s_2^0 + s_2^1 \rightarrow \emptyset, s_3^0 + s_3^1 \rightarrow \emptyset, \}. \end{aligned} \tag{13}$$

Analyzing this reaction network reveals seven overlapping organizations as shown in Fig. 2 (B). The largest organizations are composed of three species, and each species specifies the different vertex state.

$$\{s_1^1, s_2^0, s_3^0\}, \{s_1^0, s_2^1, s_3^0\}, \{s_1^0, s_2^0, s_3^1\} \tag{14}$$

Apparently, each organization corresponds to a solution to the maximal independent set problem on this graph structure. When vertex v_1 is included in the maximal independent set, the other two vertexes should not be in the independent set.

C. Graph with 6 vertexes

Next instance is a combination of the previous examples. Two circular graph with three vertexes are connected as shown in Fig. 3 (A) so that both circles and lines are contained. Since the graph consists of six vertexes, the algebraic chemistry holds 12 molecular species. Twenty six reactions among those species constitute the reaction network. Within that reaction network, there are 49 organizations in total. In Fig. 4, a whole hierarchy of the organizations is shown, and only the largest organizations with six species are listed in Fig. 3 (B). Focusing on the largest organizations within the reaction network, only the set of species representing the solution to the maximal independent set problem is found to be the organization.

V. Discussion and Outlook

In chemical computing, the result emerges as a macroscopic phenomenon from many microscopic reaction events. It is, in general, very difficult to anticipate the macroscopic behavior from the microscopic interactions. Since programming chemical reaction system is to manipulate the reaction rules in the microscopic level, the ability to anticipate the behavior of a program in the macroscopic level is required, however. The micro-macro gap has to be bridged, at least partially, to allow programming the reaction systems.

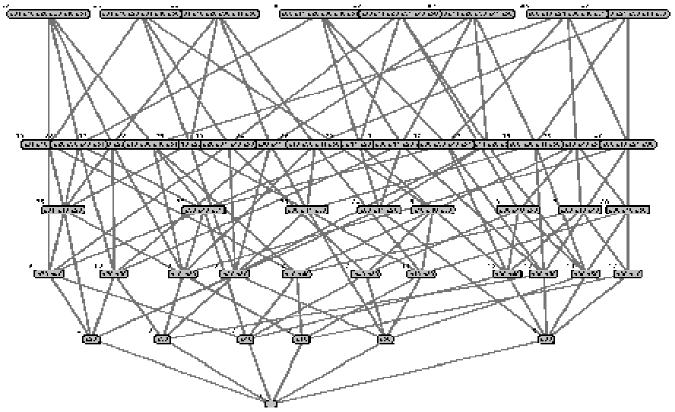


Fig. 4. Hierarchy of chemical organizations within the reaction network programmed to solve the maximal independent set problem in the graph structure depicted in Fig. 3 (A). There are 49 organizations in total, and eight organizations with six species are the largest. The potential dynamical behaviors of the reaction network to solve the maximal independent set problem appear as the largest organizations.

We have shown that chemical organization theory can serve as a tool to predict the potential behavior of a chemical program given its “microscopic” reaction rules, without the need to know the kinetics in detail. The desired solutions to the maximal independent set problem appeared as organizations. Furthermore, the organizational analysis uncovered organizations representing incomplete computation due to a lack of molecules. Chemical organization theory can now guide further improvements of the chemical program, which aim at reducing or even removing completely these “undesired” organizations.

To allow not only qualitative but also quantitative evaluation of our approach, a benchmark problem is desirable. We envision as a (simple) benchmark a variant of the maximal independent set problem in a sensor network scenario as sketched in Fig. 5: In this scenario we assume that sensor nodes are arranged linearly. Specific molecules are distributed over the network. Then the network should self-organize such that pairwise neighboring nodes are in different states, for example, one class should perform a measurement at night the other at daytime. When nodes are removed or added dynamically, spontaneous reconfiguration should occur (self-repair). The recovery time or number of acceptable perturbations can serve as a quantitative measure of the systems performance.

For a concrete application we plan to implement a chemical programming environment and a runtime system as sketched in Fig. 6. It consists of a compiler that takes a high level description of a chemical program as input. A chemical program consists of a list of molecules and reaction rules including kinetic laws. The compiler generates “chemical byte

[2] For example, a simple language where molecules are just symbols and reaction rules are explicit transformation rules, or a more complex language where molecules posses a structure and reaction rules are defines implicitly by referring to that structure (e.g., prime number chemistry).

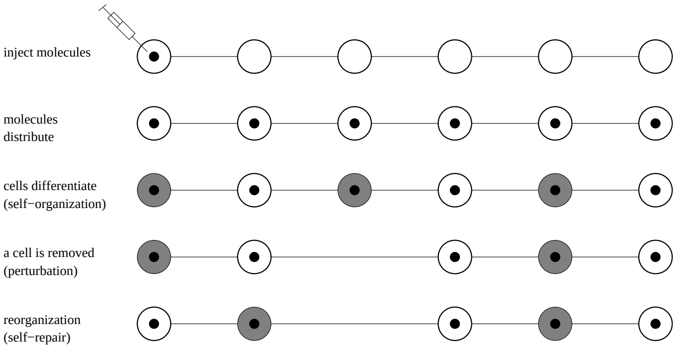

inject molecules
molecules distribute
cells differentiate (self−organization)
a cell is removed (perturbation)
reorganization (self−repair)

Fig. 5. Scenario of a benchmark problem, where a set of sensor nodes have to differentiate such that pairwise neighbors are in different states.

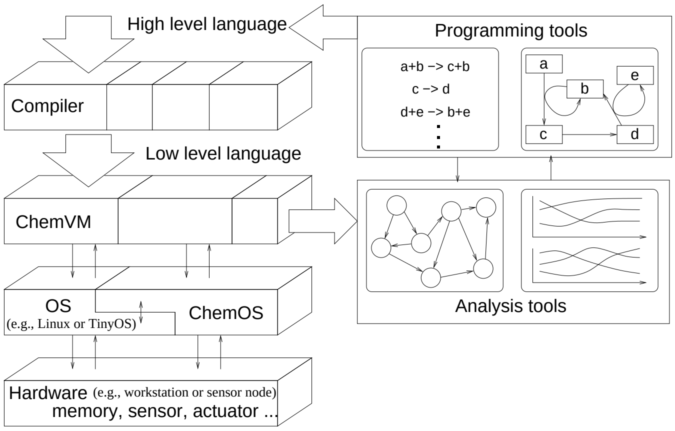


Fig. 6. Schematic representation of architecture of the chemical programming workbench. ChemVM: a virtual machine that is able to run a (low-level) chemical program. Compiler: compiles a high-level chemical language to a lower-level language that can be run on a ChemVM. ChemOS: its main task is to handle input-output to other (conventional) software processes running on the same system, or to hardware sensors and actuators. The architecture should allow to “plug in” different compilers, which may compile the same program to different virtual machines, e.g., a deterministic or a stochastic machine.

code”, which can be processed by the chemical virtual machine. The advantage of the compilation step is that different chemical languages[2]can be run on the same virtual machine. The virtual machine requires some input-output functionality, which is partly taken from the underlying operating system. Special communication between the chemical program and other hardware, such as sensors or actuators, is handled by the ChemOS (chemical operating system).

The sketched architecture and the theoretical approach exemplified along the maximum independent set problem should lead to a practical framework for “chemical programming”. By doing so, we expect to make available a technology that allows to create computational systems with the properties of their biological counterpart.

ACKNOWLEDGMENT

The authors would like to thank the members of our group for their support and useful advice. The acknowledgment also goes to Gerd Grüner and Christoph Kaleta for their help with the software development. We acknowledge financial support by the German Research Foundation (DFG) Grant DI 852/4-1, and by the Federal Ministry of Education and Research (BMBF) Grant 0312704A.

REFERENCES

[1] M. Conrad, “The brain-machine disanalogy,” *BioSystems*, vol. 22, pp. 197–213, 1989.

[2] ——, “Scaling of efficiency in programmable and non-programmable systems,” *BioSystems*, vol. 35, pp. 161–166, 1995.

[3] C. Müller-Schloer, M. Malsburg, and R. P. Würtz, “Aktuelles Schlagwort: Organic Computing,” *Inf. Spektr.*, vol. 27, no. 4, pp. 332–336, 2004.

[4] M. Conrad, “Molecular computing,” in *Advances in Computers*, M. C. Yovits, Ed. Boston: Academic Press, 1990, vol. 31, pp. 235–324.

[5] B.-O. Küppers, *Information and the Origin of Life*. Cambridge, MA: MIT Press, 1990.

[6] J.-P. Banâtre and D. L. Métayer, “A new computational model and its discipline of programming,” INRIA, Tech. Rep. RR-0566, 1986.

[7] P. Dittrich, “The bio-chemical information processing metaphor as a programming paradigm for organic computing,” in *Proceedings of ARCS: Architecture of Computing Systems*, ser. LNCS. Springer, Berlin, 2005, to appear.

[8] W. Banzhaf, P. Dittrich, and H. Rauhe, “Emergent computation by catalytic reactions,” *Nanotechnology*, vol. 7, no. 1, pp. 307–314, 1996.

[9] C. Müller-Schloer, “Organic computing: On the feasibility of controlled emergence,” in *Proceedings of the 2nd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS2004*. ACM Press, New York, 2004, pp. 2–5.

[10] K.-P. Zauner, “From prescriptive programming of solid-state devices to orchestrated self-organisation of informed matter,” in *Unconventional Programming Paradigms: International Workshop UPP 2004*, ser. LNCS, J.-P. Banâtre, J.-L. Giavitto, P. Fradet, and O. Michel, Eds., vol. 3566. Springer, Berlin, 2005, pp. 47–55.

[11] P. Dittrich and P. Speroni di Fenizio, “Chemical organization theory,” *Bull. Math. Biol.*, 2006, (accepted).

[12] N. Matsumaru, F. Centler, P. Speroni di Fenizio, and P. Dittrich, “Chemical organization theory as a theoretical base for chemical computing,” *International Journal of Unconventional Computing*, 2006, in print.

[13] A. Adamatzky and B. D. L. Costello, “Experimental logical gates in a reaction-diffusion medium: The XOR gate and beyond,” *Phys. Rev. E*, vol. 66, no. 4 Pt 2, p. 046112, 2002.

[14] K.-P. Zauner and M. Conrad, “Enzymatic computing,” *Biotechnology Progress*, vol. 17, no. 3, pp. 553–559, 2001.

[15] J.-P. Banâtre, P. Fradet, and Y. Radenac, “Principles of chemical programming,” in *RULE’04 Fifth International Workshop on Rule-Based Programming*, S. Abdennadher and C. Ringeissen, Eds. Tech. Rep. AIB-2004-04, Dept. of Comp. Sci., RWTH Aachen, Germany, 2004, pp. 98–108.

[16] G. Păun, *Membrane Computing: An Introduction*, ser. Nat. Comput. Ser. Berlin: Springer, 2002.

[17] W. Fontana and L. W. Buss, “”The arrival of the fittest”: Toward a theory of biological organization,” *B. Math. Biol.*, vol. 56, pp. 1–64, 1994.

[18] S. K. Shukla, D. J. Rosenkrantz, and S. S. Ravi, “Observations on self-stabilizing graph algorithms for anonymous networks,” in *Proceedings of the Second Workshop on Self-Stabilizing Systems*, 1995, pp. 7.1–7.15.

[19] M. Luby, “A simple parallel algorithm for the maximal independent set problem,” in *STOC*. ACM, 1985, pp. 1–10.

[20] T. Herman, “Models of self-stabilization and sensor networks,” in *IWDC*, ser. LNCS, S. R. Das and S. K. Das, Eds., vol. 2918. Springer, Berlin, 2003, pp. 205–214.