in: Workshop on Unconventional Computing
C. Teuscher, A. Adamtzky, Editors,
pp. 71-83, Luniver Press, Beckington, 2005, ISBN 978-0-9551170-0-8

Chemical Organization Theory as a Theoretical Base for Chemical Computing

Naoki Matsumaru, Florian Centler, and Peter Dittrich

Bio Systems Analysis Group
Jena Centre for Bioinformatics (JCB) and
Department of Mathematics and Computer Science
Friedrich Schiller University Jena
D-07743 Jena, Germany
`http://www.minet.uni-jena.de/csb/`

Abstract. In chemical computing, a theoretical analysis to cope with the emergent behavior is required. In this paper, we exemplified with XOR logic gate and flip-flop circuit how chemical organization theory can help in designing and understanding chemical computing systems. The reaction network implementing the boolean computation is decomposed into a hierarchical self-maintaining sub-network (organization), and the dynamical behavior is explained as a movement between the organizations. The theoretical analysis provides an insight into potential emergent behaviors. The results are encouraging so that the theory of chemical organization contributes to a theoretical base for emergent computing.

1 Introduction

By employing a large number of simple components interacting with each other in an orchestrated way, biological systems invented a variety of information processing mechanisms, which are robust, self-organizing, adaptive, decentralized, asynchronous, fault-tolerant, and evolvable. This principle of biological information processing has been exploited to cope with the fast-growing complexity of technical information processing systems [1–3]. Since all known life forms process information using chemical processes [4], the chemical reaction metaphor has been proposed as a source of inspiration [5, 6]. Using chemical reactions for formal computations has initially been suggested by Banâtre and Métayer [5]. In their GAMMA system [7], a chemical reaction is defined as a rewriting operation on a multiset, mimicking a well-stirred reaction vessel. In order to capture the spatial context of chemical systems, chemical rewriting systems have been extended to chemical abstract machines (CHAM) [8], P-Systems [9, 10] stressing the importance of membranes, and MGS allowing arbitrary topologies [11].

The difficulty of engaging the chemical reaction process for computing is that the solution appears as an emergent global behavior based on the manifold local interactions [12]. As, for example, recently demonstrated by Tsuda et al. with *Physarum* [14], the discrepancy between local and global behavior may be problematic in a practical case. A satisfying theory of emergence is lacking [15].

This paper should contribute towards the establishment of a theoretical analysis of the emergent behavior in chemical computing. We suggest chemical organization theory [20, 16] as a tool helping to construct (program) and analyze (describe and understand) chemical computing systems. Chemical organization theory allows to decompose a reaction network into a hierarchy of self-maintaining sub-networks, called organizations, only using stoichiometric information. Although other approaches to divide biochemical reaction network into sub-networks in terms of the functional moduls have been investigated [17], a rigorously proven relation between sub-networks and potential emergent dynamics is mostly lacking. Since emergent properties appear in a dynamical situation, the theoretical method for analyzing emergence must take it into consideration.

Inspired by Fontana and Buss [19], we define a (chemical) organization as a set of molecular species that is (algebraically) closed and (stoichiometrically) self-maintaining [16]. It is important to note, that, when we talk about organizations, we abstract details like concentration levels or the spatial distribution of a chemical species. On this relatively high level of abstraction, a system state is characterized only by the molecular species present and we can describe the dynamics of a system more qualitatively, namely, as a movement between sets of species instead of a movement in a more complex state space [20].

Borrowing the notion of chemical organizations defined as closed and self-maintaining sets of molecular species, we demonstrate in this paper, how the algebraic analysis of chemical reaction networks helps to understand the emergent dynamical behavior of (artificial) chemical computing. In Section 2, we describe the concepts from chemical organization theory needed here. As an example of chemical computing, the XOR logic gate is implemented using an (artificial) chemical reaction network with the help of the chemical organization theory in Section 3. A flip-flop logic circuit consisting of two NAND gates serves as another example in Section 4. Finally, in Section 5, we discuss the potential of the theory as a theoretical base for emergence analysis.

2 Chemical Organization Theory

The target of chemical organization theory are reaction networks. We represent a reaction network formally by a tuple $\langle \mathcal{M}, \mathcal{R} \rangle$, called an algebraic chemistry:

Definition 1 (algebraic chemistry). *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 $\mathcal{M}$.*

For simplicity, we adopt a notion from chemistry to write reaction rules. Instead of writing $(\{s_1, s_2, \ldots, s_n\}, \{s'_1, s'_2, \ldots, s'_{n'}\}) \in \mathcal{R}$ we write: $s_1 + s_2 + \cdots + s_n \to s'_1 + s'_2 + \cdots + s'_{n'}$. And instead of writing $a + a \to b$ we can also equivalently write $2a \to b$. Note that "+" is not an operator here, but is used to separate the elements on both sides. Given the left hand side species $A = \{s_1, s_2, \ldots, s_n\}$ and the right hand side species $B = \{s'_1, s'_2, \ldots, s'_{n'}\}$, we write $(A \to B) \in \mathcal{R}$

instead of $(A, B) \in \mathcal{R}$. $A \to B$ represents a chemical reaction equation where A is the multiset of species on the left hand side (also called **reactants**) and B the multiset of species on the right hand side (also called **products**).

So, in an algebraic chemistry, the reaction system is specified by a set of molecular species $\mathcal{M}$, and a set of reaction rules $\mathcal{R}$. A reaction rule is similar to a rewriting operation [22] on a multiset where multiple occurrences of the same element are allowed. Reaction rules are given as a subset $\mathcal{R} \subseteq \{A \to B : A, B \in \mathcal{P}_M(\mathcal{M})\}$ where $\mathcal{P}_M(\mathcal{M})$ is the set of all multisets with any size using the elements in $\mathcal{M}$. By applying a rule $A \to B$, molecules $a_i \in A$ are replaced with molecules $b_i \in B$.

To reveal the organizational structure of a reaction network, we identify all its organizations, i.e. those sets of molecular species that are closed and self-maintaining. 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. We formally define closure as:

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

A set of molecules is called self-maintaining, if, loosely speaking, all molecules that are used up within the set can also be reproduced by reactions among molecules of that set. For general reaction networks the formal definition of self-maintenance reads:

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

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

Loosely speaking, these three conditions can be read as follows: We have to find a flux vector, such that all fluxes within the species set S are greater zero (Condition 1), all other fluxes are zero (Condition 2), and all production rates of molecules within S are nonnegative (Condition 3). When a set of molecular species is self-maintaining, it is possible to construct a dynamical system (i.e. a chemical differential equation) based on the given algebraic structure of the reaction network, so that all molecular species in the set are maintained in terms of their concentration, resp. the number of molecules present. Taking closure and self-maintenance together, we arrive at an organization:

Definition 4 (organization). *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., Figure 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.

3 Case Study I: A Chemical XOR

To demonstrate how chemical organization theory can be used for chemical computing, an (artificial) chemical reaction network is designed to implement an XOR logic gate.

The two input and one output variables of the XOR gate are labeled as a, b, and c, respectively. Two molecular species are assigned to each variable to represent its boolean value. For example, the lowercase species a represents the state $\mathsf{a} = 0$. while the uppercase species A stands for $\mathsf{a} = 1$. Thus, the XOR reaction network consists of 6 molecular species $\{a, A, b, B, c, C\} = \mathcal{M}$. Note that with this kind of coding, the logic state of a variable, say a, can be undefined, namely if the reaction vessel contains molecules of type a and A at the same time, or if the reaction vessel does not contain a nor A.

The reaction rules representing an XOR gate are derived directly from the truth table: A boolean computation $1 \oplus 0 = 1$, for example, is converted to the reaction rule $A + b \rightarrow C$, so that we arrive at four reaction rules: $\mathcal{R}_1 = \{a + b \rightarrow c,\ a + B \rightarrow C,\ A + b \rightarrow C,\ A + B \rightarrow c\}$. The Hasse diagram in Figure 1 (A) shows the hierarchical organizational structure of this reaction network with 28 organizations. All organizations consist of combinations of molecules that do not react with each other. A set of molecules that does not react is obviously closed. Furthermore, it is self-maintaining since with no reactions among the set species, a zero flux vector $\boldsymbol{v} = \mathbf{0}$ fulfills all required conditions of Definition 3. With the species set of an organization being closed and self-maintaining, it is more likely to observe the presence of molecular species of an organization than of an other species combination in the reaction vessel. If the dynamics of the reaction network is modelled using ordinary differential equations, there exists a related organization for every fixed point of the system [16].

When implementing XOR using only the four reactions mentioned above, we can see in Figure 1 (A) that half of the organizations represent contradictory situations, e.g. a and A being present in an organization at the same time. If both the uppercase and lowercase species of a variable are present, the variable is computed to have a value of 0 and 1 at the same time. In order to avoid these cases, destruction reactions of the form $a + A \rightarrow \emptyset$ are added for each boolean variable a, b, and c. These reactions ensures that contradictory situations vanish quickly in the reaction vessel. The resulting set of reactions for the chemical XOR gate is: $\mathcal{R}_2 = \mathcal{R}_1 \cup \{a + A \rightarrow \emptyset,\ b + B \rightarrow \emptyset,\ c + C \rightarrow \emptyset\}$. Rule set $\mathcal{R}_2$ represents the XOR gate without any input specified. Its Hasse diagram of organizations is shown in Figure 1 (B). The number of organizations is reduced from 28 to 15.

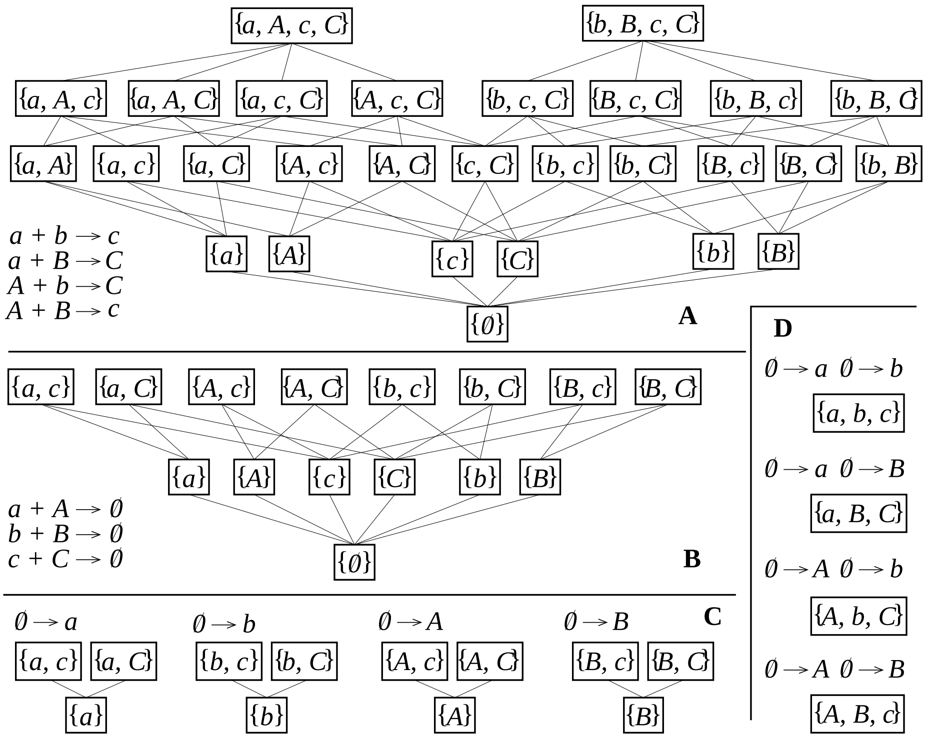

Fig. 1. Hierarchy of organizations for the chemical reaction network implementing an XOR logic gate. (A) The network consists only of the reactions directly derived from the truth table (see bottom left). (B) Destruction reactions are added to exclude contradictions. (C) Defining one input. (D) Adding the second input. The hierarchy of organizations collapses from (A) to (D), with the desired output as the only organization left in (D).

Setting an input variable to a certain value is equivalent to adding an inflow reaction to the network. The inflow reaction creates a molecule representing the desired value of the variable from “nothing”, e.g., $\emptyset \rightarrow a$ corresponds to setting $\mathsf{a} = 0$. Note that the network structure is changed due to the different input reactions, so that the result of the algebraic analysis changes for each input. We start with providing one input only, leaving the other input variable undefined. Figure 1 (C) shows the results for the four resulting networks $\mathcal{R}_2 \cup \{\emptyset \rightarrow a\}$, $\mathcal{R}_2 \cup \{\emptyset \rightarrow A\}$, $\mathcal{R}_2 \cup \{\emptyset \rightarrow b\}$, and $\mathcal{R}_2 \cup \{\emptyset \rightarrow B\}$, respectively. We can see that providing one input signal has further reduced the behavioral freedom of the reaction system. Only three combinations of molecules are left, which may be encountered in the reaction vessel in a stationary state. Furthermore we can see that – in this special case – the output is not determined from a stoichiometric point of view, since in all four Hasse diagrams an organization containing c and C is present.

When we finally provide both inputs, the Hasse diagram of organizations collapses, so that only one organization remains for every input condition (Figure 1 (D)). This implies that, no matter how we chose the dynamics, no other molecular species than those of the organization can sustain themselves in the reactor regardless of the initial state. We can see that the remaining organization contains the desired output molecule c or C, respectively. The analyzed networks are $\mathcal{R}_2 \cup \{\emptyset \rightarrow a, \emptyset \rightarrow b\}$, $\mathcal{R}_2 \cup \{\emptyset \rightarrow A, \emptyset \rightarrow b\}$, $\mathcal{R}_2 \cup \{\emptyset \rightarrow a, \emptyset \rightarrow B\}$, and $\mathcal{R}_2 \cup \{\emptyset \rightarrow A, \emptyset \rightarrow B\}$.

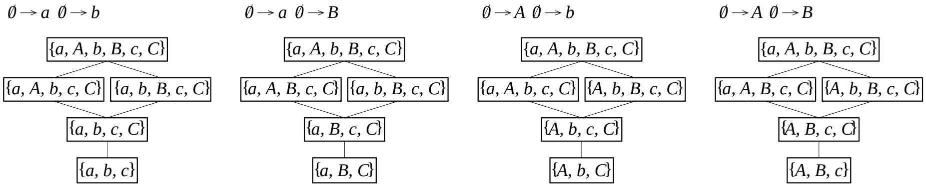

Fig. 2. Hierarchical structure of closed sets of molecular species in the reaction network implementing an XOR logic gate. In addition to 7 reaction rules (4 reactions from the truth table and 3 for simultaneous destruction to avoid contradictory situation), two influxes are included in the reaction network. Since the destruction reactions do not violate the closure property of a set, contradictory situations such as co-existence of c and C in closed sets are observed.

3.1 Is closure sufficient? (Moving to discussion?)

As described in Section 2, an organization is defined as a set of molecular species that is closed and self-maintaining. To address the question whether the property of closure is already sufficient for the analysis or not, we determine all closed sets of the reaction network containing 7 reaction rules and two influxes. Figure 2 shows the resulting hierarchical structure of closed sets. Comparing this hierarchy with the hierarchy of closed *and* self-mainaining sets in Figure 1 (D), we find that the smallest closed sets coincide with the only organizations found in the network. But we find that other closed sets contain contradictory situations, e.g., c and C being present at the same time. The destruction reactions designed to eliminate this cases do not work as they do not effect the closure property of any species set. Only requiring self-maintenance leads to the desired exclusion of contradictions.

3.2 Dynamic Simulation

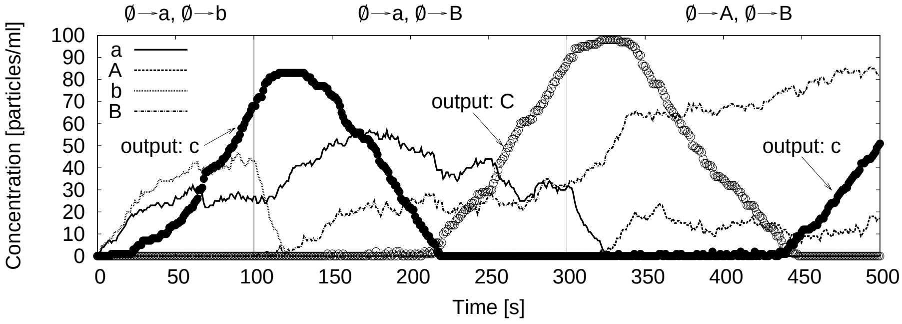

Fig. 3. Dynamical concentration change of 6 molecular species is shown. The reaction network composed of 7 reaction rules (4 reactions from the truth table and 3 decaying reactions) and two influx to implement a XOR logic gate. The reactions are defined as irreversible mass action kinetics, and the influx is an irreversible constant flux. The dynamical behavior of the reaction system is stochastically simulated with the biochemical network simulator *Copasi* [23]. The kinetic parameters for logic computation reactions is set to 0.001, and 0.1 for decay reactions. The parameter for constant influx is set to 1, and the volume of the compartment is 1 ml. See text for details.

When providing each chemical reaction rule with a kinetic rate, it is possible to analyze the dynamical behavior of the reaction system. The influxes are assumed to be an irreversible constant flux, and the other reaction rules work in accordance with irreversible mass action kinetics. Figure 3 shows a dynamical behavior of a reactor in terms of the concentration trajectory of the molecular species from the XOR reaction network. The reaction system is stochastically simulated with the biochemical reaction network simulator *Copasi* [23], a descendant of Gepasi[24]. Initially, there exist no molecular particles in the reactor, and two influxes of a and b are present. This corresponds to the case in which both the input variables a and b are set to 0. The output calculates to be $\mathsf{c} = 0$ with the generation of molecular species c.

At the time point of 100 s, the contents of input variable b is switched to 1 so that the influx of molecular species b is replaced by the influx $\emptyset \rightarrow B$. Since the reactor is not initialized, the molecular particles b and c from the previous computation are deteriorating first. The desired output C does not appear until the time point of approximately 200 s. Then, instead of a, the molecular particles of species A are injected continuously starting from the time point of 300 s. The remaining molecular species a and C from the previous computation are deleted in the beginning, and the desired answer emerges in the end.

As seen from the dynamical simulation, the emergent behavior in terms of the qualitative final state of the reactor is independent of the initial state due to the continuous injection of input particles. The molecular particles of output species are generated continuously, and unnecessary species are degraded from the collision with the anti-particle. From the algebraic analysis, the independence of the emergent behavior is predicted as one organizational structure in the reaction network. Since no other qualitatively stable sets are recognized in the XOR reaction network with two influx, the qualitative emergent behavior mode is structurally constrained to one. When the network does not have the influx, on the other hand, the number of organizations is 15. Results of the computation as a qualitative emergent behavior cannot be conclusive until the initial condition of the reactor is given. (X)

4 Case Study II: A Chemical Flip-Flop

In this section, we apply our approach to a more complicated example: the flip-flop logic circuit. As opposed to the previous example, a flip-flop circuit is bistable, which is achieved by two feedback connections. The fact that the circuit is bistable will also become apparent when we analyze the organizations of our chemical instantiation of the flip-flop. This allows us to explain the dynamical behavior of the chemical flip-flop in terms of chemical organization theory on an abstract level, which does not need to refer to concentration levels.

The RS (Reset and Set) flip-flop circuit consists of two NAND gates connected in parallel as shown in Figure 4 (A). The behavior can be described by the truth table as shown in Figure 4 (B). The output of one logic gate is connected to one of the two inputs of the other gate, forming a feedback loop. The “set” operation $(\bar{\mathsf{S}}, \bar{\mathsf{R}}) = (0, 1)$ changes the output Q to 1, and the “reset” operation $(\bar{\mathsf{S}}, \bar{\mathsf{R}}) = (1, 0)$ sets Q to 0. When both inputs are set to 1, the output is kept as in the previous state. The one-bit information whether the output Q has been 0 or 1 is stored by the “hold” operation, ie. $(\bar{\mathsf{S}}, \bar{\mathsf{R}}) = (1, 1)$. Normally, the input $(\bar{\mathsf{S}}, \bar{\mathsf{R}}) = (0, 0)$ is prohibited, because the circuit will go into a state where $\mathsf{Q} = 0$ and $\bar{\mathsf{Q}} = 0$. Application examples for the flip-flop are memory circuits and circuits that can count.

Starting from the logic circuit diagram, a chemical reaction network is constructed in the same way as for the XOR gate. The input variables of the circuit are labeled a and b, and the output variables are c and d. For one NAND gate, the input variable is a and d, and the output is c. In the same way as for the

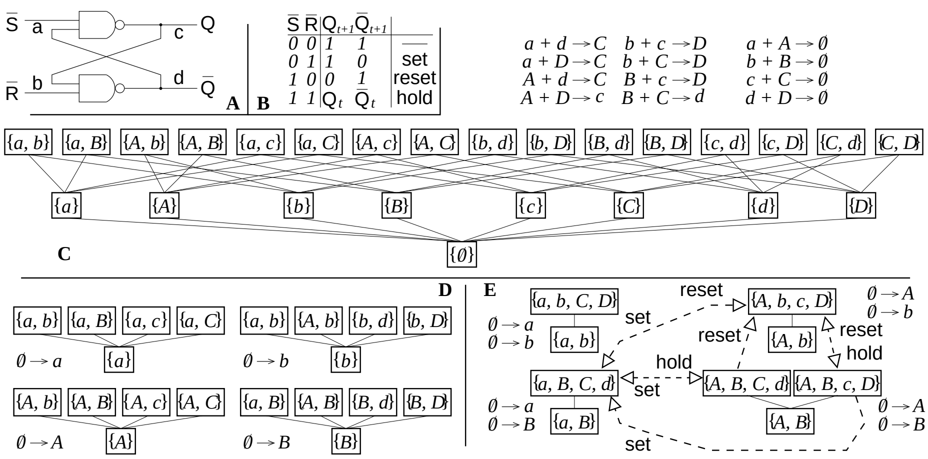

Fig. 4. A chemical reaction network is constructed to implement the RS flip-flop logic circuit with the help of chemical organization theory. The logic structure of the circuit is shown in (A), and the behavior is shown in (B) as a truth table. The chemical reaction network implementing the flip-flop with 8 molecular species and 12 reaction rules, and the hierarchical organizational structure of the network is shown in (C). By adding influx to the network, the number of organizations is reduced as shown in (D) and (E). When the molecular species related to both input variables a and b are supplied continuously as an influx, the organizational structure corresponds to the operation of the flip-flop logic circuit. See text for details.

XOR reaction network described in the previous section, four reaction rules are derived from the NAND truth table. Another four reaction rules are derived for the other NAND gate with `b` and `c` as input and `d` as output. Finally, destruction reactions for all species are added, so that contradictory states degenerate, as in the previous example. The flip-flop reaction network is constituted by 8 molecular species $\mathcal{M} = \{a,\, A,\, b,\, B,\, c,\, C,\, d,\, D\}$ and 12 reaction rules shown in Figure 4 (C).

When we applied our analysis to the flip-flop reaction network without any input flow, we found 25 organizations consisting of up to two molecules, which do not react (Figure 4 (C)). These organizations represent all possible "qualitative states" of the reaction vessel, provided that there is no input flow.

When two input species are injected into the system, the number of organizations found in the network is reduced to two resp. three for each input case as seen in Figure 4 (E). When the output species c, C, d, and D are not present in the reactor, no reaction occurs due to the feedback loops. Thus, the smallest organization contains only the two inflow species. Above it, the designated output species are included in the organizations. We can interprete this in the following way: The flip-flop exists if c, C, d, or D is present in the reaction vessel. Removing c, C, d, and D would also extinguish the chemical flip-flop. Or in other words, the input molecules alone can not generate the organization representing an operational mode of the flip-flop.

The operation of the flip-flop can be described by transitions between organizations: The set and reset operation move the reaction system to the states corresponding to organization $\{a,\, B,\, C,\, d\}$ (set) and $\{A,\, b,\, c,\, D\}$ (reset). Recall that for the set and reset operation we add $\{\emptyset \to a, \emptyset \to B\}$ and $\{\emptyset \to A, \emptyset \to b\}$ to the set of reaction rules, respectively.

For the hold operation $(\emptyset \to A, \emptyset \to B)$, the flip-flop has two stable states represented by the organizations $\{A,\, B,\, C,\, d\}$ and $\{A,\, B,\, c,\, D\}$. If the vessel had been in organization $\{a,\, B,\, C,\, d\}$ before, it will move into organization $\{A,\, B,\, C,\, d\}$; and if it had been in organization $\{A,\, B,\, c,\, D\}$ before, it will move into organization $\{A,\, B,\, c,\, D\}$. Symbolically speaking, the lowercase input species is replaced by its uppercase version due to the input change, but the output state remains unchanged.

For the sake of completeness, the cases in which only one influx is added to the network are shown in Figure 4 (D). If the reactor is initialized to be empty, the input species injected by the influx do not react. So, again, the smallest organizations contain only the input species. Adding one species that does not interact with the input species forms another organization. Since it does not react, it also does not need to be produced, since no species decays in the algebraic chemistry unless a decay reaction is explicitly defined. For example, no reaction rule is applicable in the organizations consisting of the input species a and the species related to variables `b` and `c`. Since adding other species makes a reaction rule applicable that uses up a molecular species that can not be reproduced, there exists no organization of size greater than two.

4.1 Dynamical Simulation

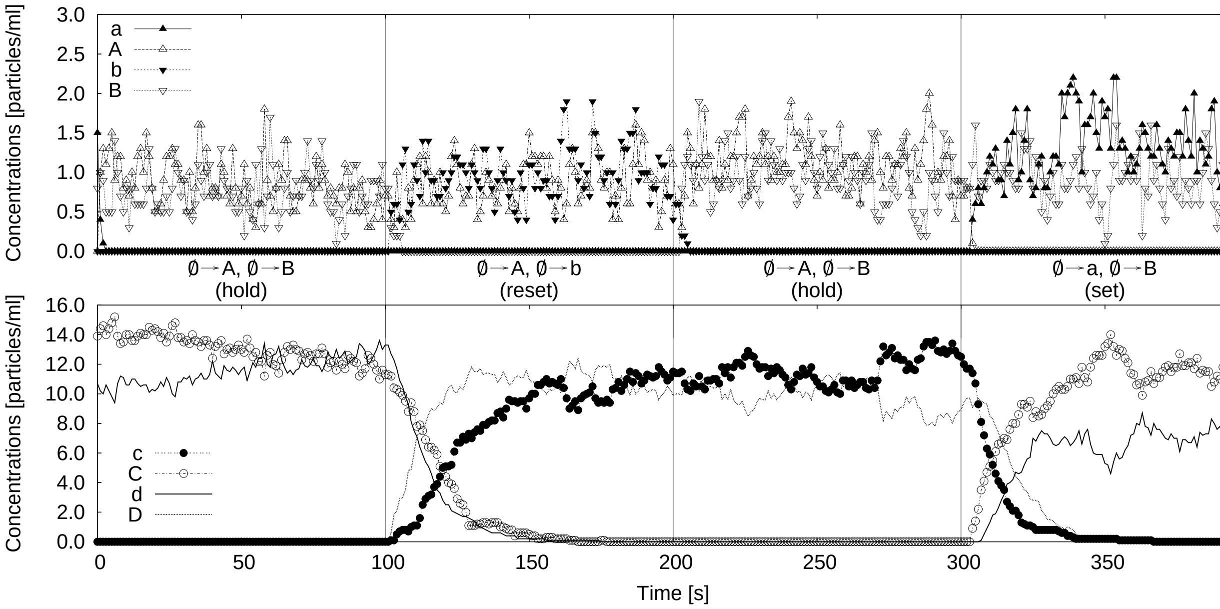

Fig. 5. Dynamic behavior of the chemical reaction network implementing a RS flip-flop logic circuit shown in Figure 4. The top figure shows the time course of the input species a, A, b, and B. The bottom figure shows the concentrations of the output species. Irreversible mass action kinetics is assumed for all reactions. Reaction rates are set to $k = 0.1$ for all second order reactions and $k = 1$ for all irreversible constant influxes like $\emptyset \rightarrow A$. The reaction system is stochastically simulated with the biochemical network simulator *Copasi* [23] using a compartment size of 10 ml.

In order to validate the discussion of the previous section we performed stochastic simulations (using MGS [11] and Copasi [23]) of reaction systems implementing the chemical flip-flop. Figure 5 shows a typical simulation run. The influx is defined as an irreversible constant flux with kinetic parameter set to 1. For all other reactions we chose irreversible mass action kinetics with the kinetic parameter set to 0.1. During the first "hold" phase (0 - 100 s), the concentration of C and d remain high. In the following "reset" phase (100 - 200 s), the input reactions $\emptyset \rightarrow A$ and $\emptyset \rightarrow b$ are added to "reset" the system so that the output variable c is set to 0. The concentration of C and d decreases gradually and species c and D accumulate in the reactor. The system eventually reaches a state in which only members of the organization $\{A, b, c, D\}$ are present as expected from the algebraic analysis. In the next phase (200 - 300 s), the input flow of b is replaced by that of B, $\emptyset \rightarrow B$, to "hold" the output of the previous phase. Although the input species have changed, no qualitative change is detected in the bottom graph. Finally, in the last phase (300 - 400 s), the "set" operation is executed by changing the influx $\emptyset \rightarrow A$ to $\emptyset \rightarrow a$. The transition to the state represented by the organization $\{a, B, C, d\}$ is observed.

Although the same input species are injected in the two "hold" phases, the qualitative states of the reactor are different depending on the initial conditions. The bistable behavior of the flip-flop circuit is implemented dynamically by the chemical reaction system, which we have expected from our theoretical analysis in the previous section. The reaction network with the two influxes $\emptyset \rightarrow A$ and $\emptyset \rightarrow B$ has two organizations with four species: The system state in the first "hold" phase corresponds to the organization $\{A, B, C, d\}$, and the members of the organization $\{A, B, c, D\}$ are present during the second "hold" phase.

5 Concluding Remarks

In this paper, a theoretical analysis method is proposed that helps to discover and implement computing capabilities in (artificial) chemical reaction networks. Given a list of molecular species and a list of reaction rules, the reaction network is decomposed into a hierarchy of closed and self-maintaining sub-networks called organizations. We have shown that the hierarchy of organizations helps to assess the emergent dynamical behavior of the chemical reaction network under study. When the approach is applied to a reaction network implementing a XOR logic gate, the hierarchy of organizations helps to predict its emergent dynamical behavior. Defining different inputs leads to different organizations corresponding to the various states of the gate. As another example, a flip-flop logic circuit in which two NAND gates are connected to each other via feedback loops is implemented by a chemical reaction network. From the theoretical analysis, the bistable behavior is reflected by two organizations found in the network. Using chemical organization theory, we where able to explain the properties of the chemical flip-flop in a new, comprehensible way by referring to the Hasse diagram of organizations (Figure 4). Furthermore the "constructive" dynamics of the flip-flop could be described as a movement between organizations (Figure 4 (E)). This description is more compact than a classical description referring to the 8-dimensional concentration state space, as demonstrated in Section 4.1.

When designing a system with numerous small, extensively interacting components, its global behavior cannot easily be predicted from the known local interactions. A general theory of emergence is desirable not only for analytical purposes, but also for engineering such systems. If local interactions are restricted to processes that are expressible as chemical reaction rules, the theory of chemical organization helps to determine the system's repertoir of potential behavior patterns. Since only network structure is considered for the analysis, also non-chemical reaction networks, e.g. social interaction networks, can be investigated. The encouraging results presented in this paper suggest that the theory of chemical organizations is a promising candidate to contribute to a general theoretical framework to master self-organization in complex chemical-like information systems.

Acknowledgment

We acknowledge financial support by the *Federal Ministry of Education and Research* (BMBF) Grant 0312704A to *Friedrich Schiller University Jena* and by the German Research Foundation (DFG) Grant Di 852/4-1 to PD.

References

1. Müller-Schloer, C., Malsburg, M., Würtz, R.P.: Aktuelles Schlagwort: Organic Computing. Informatik Spektrum **27** (2004) 332–336
2. von der Malsburg, C.: The challenge of organic computing (1999) Memorandum, Computer Science Department.
3. Würtz, R.P.: Organic computing for face and object recognition. In Dadam, P., Reichert, M., eds.: Informatik 2004. Volume 2., Gesellschaft für Informatik (2004) 636–640
4. Küppers, B.O.: Information and the Origin of Life. MIT Press, Cambridge, MA (1990)
5. Banâtre, J.P., Métayer, D.L.: A new computational model and its discipline of programming. technical report RR-0566, INRIA (1986)
6. Dittrich, P.: The bio-chemical information processing metaphor as a programming paradigm for organic computing. In: Proceeding of ARCS: Architecture of Computing Systems. LNCS, Berlin, Springer (2005) To appear.
7. Banâtre, J.P., Métayer, D.L.: The GAMMA model and its discipline of programming. Sci. Comput. Program. **15** (1990) 55–77
8. Berry, G., Boudol, G.: The chemical abstract machine. Theoretical Computer Science **96** (1992) 217–248
9. Păun, G.: Computing with membranes. Journal of Computer and System Sciences **61** (2000) 108–143
10. Păun, G.: Membrane Computing: An Introduction. Natural Computing Series. Springer (2002)
11. Giavitto, J.L., Michel, O.: MGS: a rule-based programming language for complex objects and collections. In van den Brand, M., Verma, R., eds.: Electronic Notes in Theoretical Computer Science. Volume 59., Elsevier Science Publishers (2001)
12. Banzhaf, W., Dittrich, P., Rauhe, H.: Emergent computation by catalytic reactions. Nanotechnology **7** (1996) 307–314
13. Zauner, K.P., Conrad, M.: Enzymatic computing. Biotechnology Progress **17** (2001) 553–559
14. Tsuda, S., Aono, M., Gunji, Y.P.: Robust and emergent *physarum* logical-computing. Biosystems **73** (2004) 45–55
15. Müller-Schloer, C.: 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, Stockholm, Sweden, September 8-10, 2004, ACM Press (2004) 2–5
16. Dittrich, P., Speroni di Fenizio, P.: Chemical organization theory: towards a theory of constructive dynamical systems (2005) arXiv:q-bio.MN/0501016.

17. Barabási, A.L., Oltvai, Z.N.: Network biology: Understanding the cell's functional organization. Nature Reviews Genetics **5** (2004) 101–113
18. Kremling, A., Jahreis, K., Lengeler, J.W., Gilles, E.D.: The organization of metabolic reaction networks: A signal-oriented approach to cellular models. Metabolic Engineering **2** (2000) 190–200
19. Fontana, W., Buss, L.W.: 'The arrival of the fittest': Toward a theory of biological organization. Bulletin of Mathematical Biology **56** (1994) 1–64
20. Speroni di Fenizio, P., Dittrich, P.: Artificial chemistry's global dynamics. movement in the lattice of organisation. The Journal of Three Dimensional Images **16** (2002) 160–163
21. Dittrich, P., Ziegler, J., Banzhaf, W.: Artificial chemistries - a review. Artificial Life **7** (2001) 225–275
22. Banâtre, J.P., Fradet, P., Radenac, Y.: Principles of chemical programming. In Abdennadher, S., Ringeissen, C., eds.: RULE'04 Fifth International Workshop on Rule-Based Programming, Technical Report AIB-2004-04, Department of Computer Science, RWTH Aachen, Germany (2004) 98–108
23. : Copasi (complex pathway simulator). URL: http://www.copasi.org/, visited: 31 of May, 2005, last updated: 29 of April, 2005 (2005)
24. Mendes, P.: Biochemistry by numbers: simulation of biochemical pathways with gepasi 3. Trends in Biochemical Sciences **22** (1997) 361–363