

Chemical Organization Theory as a Theoretical Base for Chemical Computing

NAOKI MATSUMARU, FLORIAN CENTLER,
PIETRO SPERONI DI FENIZIO, 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/>

Submitted 14 November 2005

In chemical computing, the result of a computation appears as an emergent global behavior based on local reaction rules. For programming chemical systems a theoretical method to cope with that emergent behavior is desired. In this paper, we demonstrate how the chemical organization theory can help in designing and understanding chemical computing systems. After providing a recipe for mapping logic circuits to chemical reaction rules, we discuss reaction networks implementing various logic circuits: an XOR, a flip-flop, and a controllable oscillator. The theory decomposes reaction networks into a hierarchy of closed and self-maintaining sub-networks (called organizations) using stoichiometric information only. The dynamical behavior of a reaction system is then explained as a movement between organizations. We show how the theoretical analysis provides insight into the potential behavior of chemical reaction systems. The encouraging results suggest that the theory of chemical organizations contributes to a theoretical framework for chemical computing.

Key words: chemical programming, artificial chemistry, chemical information processing, organic computing, controlling emergence, amorphous computing, complex networks, distributed systems, molecular 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, adaptable, decentralized, asynchronous, fault-tolerant, and evolvable. These mechanisms of biological information processing are now exploited to cope with the fast-growing complexity of technical information processing systems [23, 29, 30]. Since all known life forms process information using chemical processes [19], the chemical reaction metaphor has been proposed as a source of inspiration for a novel computation paradigm [4, 12]. Using chemical reactions for formal computations has initially been suggested by Banâtre and Métayer [4]. In their GAMMA system [5], 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 the chemical abstract machine (CHAM) [9], P-Systems [24, 25] (stressing the importance of membranes), and *MGS* [15, 21] allowing arbitrary topologies [3].

In a chemical reaction process, the solution of a computation appears as an emergent global behavior based on a manifold of local interactions [6]. For its heavy nonlinearity such behavior is hard to analyze and in general impossible to predict by methods that are more efficient than simulations.

Yet emergent behavior occurs in biological systems, for example, by combining simple biochemical signaling pathways [10]. As recently demonstrated by Tsuda, Aono, and Gunji with *Physarum* [28], the discrepancy between local and global behavior may also be problematic in practice. There is also a common agreement that a satisfying theory of emergence is lacking [22].

This paper contributes to the establishment of a theoretical analysis of emergent behavior in chemical computing, which should lead to a deeper understanding of the micro-macro link between reaction rules and resulting behavior. The ability to predict how a chemical program (*e.g.*, a list of reaction rules) behaves is a prerequisite for programming by construction [32]. Note that there is a fundamentally different approach for creating chemical information processing systems, namely programming by evolution [8, 11, 17]. This latter approach is not considered here. However, our methods might be applied to analyze the outcome of an evolutionary approach.

We suggest chemical organization theory [13, 26] 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. Following the line of [2] and in contrast to methods like Ref. [18], the theoretical analysis assumes only stoichiometric information, *i.e.*, the structure of the reaction network. Although other algebraic approaches divide biochemical reaction network into sub-networks, *e.g.*, in terms of functional modules [7], a rigorously proven relation between sub-networks and potential emergent dynamics is usually 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 [14], Dittrich and Speroni di Fenizio [13] defined a chemical organization as a set of molecular species that is (algebraically) closed and (stoichiometrically) self-maintaining. 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 [26].

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, adopted from Ref. [13]. A general procedure of converting a logic circuit into a chemical reaction network is described in Section 3. Like others (*cf.*, [1, 28, 31]), a simple non-linear logical operation XOR is implemented first, in Section 4. In Section 5, the target example of logical operation is scaled up by linking multiple NAND gates. Another example is a flip-flop logic circuit (Section 6) and a controllable oscillator (Section 7). Both circuits contain a simple feedback loop, which is an important building block in biological signaling networks to achieve robustness [27] or multi-stationarity. Finally, in Section 8, we discuss the potential of the theory as a theoretical base for the analysis of emergent chemical computing.

2 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 [13]). *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 [3] 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 $\{s_1, s_2, \dots, s_n\}$, the set is written as $s_1 + s_2 + \dots + s_n$ 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 [14]). *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 [13]). *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* in reaction j , a set of molecular species $S \subseteq \mathcal{M}$ is self-maintaining,*

* Formally, this can be defined as $m_{i,j} = \#(i \in B_j) - \#(i \in A_j)$, where $\#(i \in A_j)$

if there exists a flux vector $\mathbf{v} = (v_{A_1 \rightarrow B_1}, \dots, v_{A_j \rightarrow B_j}, \dots, v_{A_{|\mathcal{R}|} \rightarrow B_{|\mathcal{R}|}})^T$ 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 $(f_1, \dots, f_i, \dots, f_{|\mathcal{M}|})^T = \mathbf{M}\mathbf{v}$.

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 [13, 14]). *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.

2.1 Dynamics

For deriving the Hasse diagram of organizations no detailed knowledge concerning the dynamics is required. Only stoichiometric information, *i.e.*, the set of reaction rules, is sufficient. Therefore we refer to that part of chemical organization theory as the static part.

In the “dynamical part”, the set of organizations is used to describe the dynamics of a reaction system as a movement between organizations. The strength of this method lies in the analysis of chemical processes where molecular species appear (*i.e.*, their concentration becomes positive) and disappear (*i.e.*, their concentration becomes zero) at some point in time. The movement between organizations can take place spontaneously (down-movement) or can be triggered by external events, *e.g.* by adding some input molecules. For further details see Ref. [13].

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 .

Finally, a relevant theorem from Ref. [13] states that given a differential equation describing the dynamics of a chemical reaction system and the algebraic chemistry corresponding to that system. Assume further a fixed point (*i.e.*, stationary state) of that differential equation, then the set of molecules with positive concentrations in that fixpoint 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.

3 A RECIPE FOR A CHEMICAL LOGIC CIRCUIT

In this section we present a procedure for designing chemical reaction networks implementing a logic circuit (see Table 1 for a non-formal recipe). A logic circuit is a composition of logic gates. As such it can be fully described by a set of boolean functions and boolean variables, forming a boolean network [16]. Let the boolean network be defined by a set of M boolean functions and a set of N ($\geq M$) boolean variables:

$$\{\mathbf{b}_1, \dots, \mathbf{b}_M, \dots, \mathbf{b}_N\} \quad (1)$$

where $\{\mathbf{b}_j | 1 \leq j \leq M\}$ are determined by the boolean functions (*internal variables*) and the remaining variables $\{\mathbf{b}_j | M < j \leq N\}$ are the input variables of the boolean network. The set of boolean functions is

$$\{\mathbf{b}_i = F_i(\mathbf{b}_{q(i,1)}, \dots, \mathbf{b}_{q(i,n_i)}) \mid i = 1, \dots, M\} \quad (2)$$

where $\mathbf{b}_{q(i,k)}$ indicates the boolean variable listed as the k -th argument of the i -th function. Since the i -th boolean function F_i takes n_i boolean variables as arguments, there are 2^{n_i} possible inputs. Thus the truth table T_i for function F_i has 2^{n_i} rows and $n_i + 1$ columns:

$$T_i : \begin{bmatrix} t_{1,1}^i & \cdots & t_{1,n_i}^i & t_{1,n_i+1}^i \\ \vdots & \ddots & \vdots & \vdots \\ t_{2^{n_i},1}^i & \cdots & t_{2^{n_i},n_i}^i & t_{2^{n_i},n_i+1}^i \end{bmatrix} \quad (3)$$

where $t_{h,k}^i \in \{0, 1\}$ is the boolean value of the k -th argument in the h -th input case for the i -th boolean function. The $(n_i + 1)$ -th column contains the output of F_i .

Given the boolean network, an algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$ is designed as follows. For each boolean variable \mathbf{b}_j we assign two molecular species s_{2j-1} and s_{2j} representing the value 0 and the value 1 in it, respectively. Thus the set of molecular species \mathcal{M} contains $2N$ molecular species as follows:

$$\mathcal{M} = \{s_{2j-1}, s_{2j} \mid j = 1, \dots, N\} \quad (4)$$

The set of reaction rules can be decomposed into two sets of reactions:

$$\mathcal{R} = \mathcal{L} \cup \mathcal{D}. \quad (5)$$

Set of reactions \mathcal{L} is derived from the logical operations of the boolean functions with $\mathcal{L} = \bigcup_{i=1}^M \mathcal{L}^i$ where \mathcal{L}^i is a set of *logical reactions* associated with the truth table T_i of boolean function F_i . For each input case h (each row of the truth table), one reaction rule is created:

$$\mathcal{L}^i = \{A_{i,h} \rightarrow B_{i,h} \mid h = 1, \dots, 2^{n_i}\}. \quad (6)$$

The lefthand side is a set of *reactants* $A_{i,h} = \{a_{i,1,h} + \dots + a_{i,k,h} + \dots + a_{i,n_i,h}\}$ where $a_{i,k,h}$ is a molecular species representing the boolean variable that is taken as the k -th argument of function F_i and thus $b_{q(i,k)}$. Since two molecular species $s_{2q(i,k)-1}$ and $s_{2q(i,k)}$ are assigned to boolean variable $b_{q(i,k)}$ depending on its content, the truth table T_i is used to select from the two. If the entry $t_{h,k}^i$ of the truth table is equal to 0, $b_{q(i,k)}$ must be set to 0 in the h -th input case, and thus $s_{2q(i,k)-1}$ is chosen as the reactant. Otherwise, $a_{i,k,h}$ is $s_{2q(i,k)}$:

$$a_{i,k,h} = \begin{cases} s_{2q(i,k)-1} & \text{if } t_{h,k}^i = 0, \\ s_{2q(i,k)} & \text{if } t_{h,k}^i = 1. \end{cases} \quad (7)$$

Similarly, the righthand side is a set of *products* $B_{i,h} = \{b_{i,h}\}$, and

$$b_{i,h} = \begin{cases} s_{2i-1} & \text{if } t_{h,n_i+1}^i = 0, \\ s_{2i} & \text{if } t_{h,n_i+1}^i = 1, \end{cases} \quad (8)$$

since the $(n_i + 1)$ -th column of truth table T_i contains the output.

The other component of set \mathcal{R} is the set of *destructive reactions* \mathcal{D} . Since binary states of a boolean variable b_j are coded with two molecular species s_{2j-1} and s_{2j} , the state becomes undefined when both or neither of the species are present. In order to avoid such a case, the two opposite molecular species are defined to vanish upon collision:

$$\mathcal{D} = \{s_{2j-1} + s_{2j} \rightarrow \emptyset \mid j = 1, \dots, N\}. \quad (9)$$

The resulting algebraic chemistry $\langle \mathcal{M}, \mathcal{R} \rangle$ implements the logic circuit without any input specified. The input variables of the boolean network

$\{b_j | M < j \leq N\}$ must be initialized externally because they are not set by the boolean functions. The initialization of the input variables is encoded by an inflow reaction, which is a zero-order reaction producing substances from the empty set. If an input variable b_j is initialized to 0, for example, the algebraic chemistry is changed to $\langle \mathcal{M}, (\mathcal{R} \cup \{\emptyset \rightarrow s_{2j-1}\}) \rangle$. It is possible for more than one variable to be initialized in this manner as it is possible for more than one molecular species to be injected by the influx.

3.1 Implementing logic circuits with periodic attractors

At last, we would like to point out that while converting boolean networks into chemical reaction networks, feedback loops need special treatment. When considering boolean networks in general, the network can form feedback loops by connecting an output to an input so that the input is dependent on the output. This configuration can give rise to attractors having a period greater than one so that the system starts to oscillate between two (or more) states. An example of such a periodic attractor is an oscillator. When an oscillator is implemented with a reaction network, the complementary molecular species are generated alternatively and decay instantaneously. To delay the complete destruction of the two species, an amplification process has to be introduced for variables that change in the periodic attractors. A detailed description of the implementation can be found in Section 7.

4 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 XOR logic gate is defined as a set of three boolean variables $\{a, b, c\}$ and a set of one boolean function $\{F_c\}$ where the function is: $c = F_c(a, b)$, and the truth table is:

$$T_c : \begin{array}{c} \begin{array}{ccc} a & b & c \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \end{array} \quad (10)$$

Since boolean variable c is the internal variable which is determined by a boolean function, the set of boolean variable should be listed as $\{c, a, b\}$ according to the recipe in Section 3. Boolean variables and molecular species

Input: Boolean network given by two sets: a set of M boolean functions $\{F_1, \dots, F_M\}$ and a set of N boolean variables $\{b_1, \dots, b_M, \dots, b_N\}$. Variables $\{b_1, \dots, b_M\}$ are determined by the boolean functions (*internal variables*); the remaining variables $\{b_{M+1}, \dots, b_N\}$ are input variables of the boolean network.

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 boolean network without any input variable specified.^a

Algorithm:

1. For each boolean variable b_j :
 - (a) Add two molecular species, b_j and B_j , to \mathcal{M} ;^b
 - (b) Add one *destructive reaction* of the form $b_j + B_j \rightarrow \emptyset$ to \mathcal{R} ;
2. For each boolean function F_i :
 - (a) Create the truth table of F_i with 2^{n_i} input cases (where n_i is the arity of F_i);
 - (b) For each input case, create a *logical reaction*.^c
 - i Lefthand side (*reactants*) corresponds to the input of F_i .
 - ii Righthand side (*products*) consists of one molecular species representing the respective boolean output of F_i .

^aSpecifying an input variable of the boolean network is coded by an inflow reaction.

^bAs a naming convention of molecular species in this paper, the lowercase species represents value 0 in the boolean variable, and the uppercase stands for 1.

^cFor example, the XOR-function is converted into reactions as follows:

b_2	b_3	$b_1 = F_1(b_2, b_3)$		Reactants	\rightarrow	Products
0	0	0	\Rightarrow	$b_2 + b_3$	\rightarrow	b_1
0	1	1		$b_2 + B_3$	\rightarrow	B_1
1	0	1		$B_2 + b_3$	\rightarrow	B_1
1	1	0		$B_2 + B_3$	\rightarrow	b_1

TABLE 1
Recipe for mapping a boolean circuit to a chemical reaction network.

are ordered alphabetically for readability, however. Furthermore, the variable name is adopted as an index of functions.

Given the definition of the XOR boolean network, an algebraic chemistry $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \rangle$ is generated to implement the logic gate. Since there are $N = 3$ boolean variables, the set of molecular species consists of six molecular species:

$$\mathcal{M}_{\text{XOR}} = \{a, A, b, B, c, C\} \quad (11)$$

where the lower- and uppercase version of the variable name are assigned to the boolean variable of that name. For example, molecular species a represents boolean variable $a = 0$, and A stands for $a = 1$.

The set of reaction rules \mathcal{R}_{XOR} is decomposed into two parts:

$$\mathcal{R}_{\text{XOR}} = \mathcal{L}_{\text{XOR}} \cup \mathcal{D}_{\text{XOR}} \quad (12)$$

where \mathcal{L}_{XOR} is a set of reactions for the logical operation and \mathcal{D}_{XOR} is a set of destructive reactions. Since there is only one function in the boolean network, $\mathcal{L}_{\text{XOR}} = \mathcal{L}_{\text{XOR}}^c$ where $\mathcal{L}_{\text{XOR}}^c$ is a set of logical reactions constructed from the boolean function F_c . From the truth table T_c , four logical reactions are derived:

$$\mathcal{L}_{\text{XOR}} = \mathcal{L}_{\text{XOR}}^c = \{a + b \rightarrow c, a + B \rightarrow C, A + b \rightarrow C, A + B \rightarrow c\}. \quad (13)$$

The Hasse diagram in Figure 1 (A) shows the hierarchy of organizations of the reaction network that includes only the logical reactions \mathcal{L}_{XOR} . Twenty-eight sets of molecular species are found to be organizations. The remaining 36 sets do not satisfy either the closure or the self-maintenance property.

The set $\{a, b\}$, for example, is not an organization because it is not closed. The reaction $a + b \rightarrow c$ is applicable and produces a new molecular species c that is not a member of the set $\{a, b\}$. The set $\{a, b, c\}$ is closed but not an organization because it is not self-maintaining. A production rate vector \mathbf{f} is calculated as follows:

$$\mathbf{f} = \begin{pmatrix} f_a \\ f_A \\ f_b \\ f_B \\ f_c \\ f_C \end{pmatrix} = \mathbf{M}\mathbf{v} = \begin{pmatrix} -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -v_1 \\ 0 \\ -v_1 \\ 0 \\ v_1 \\ 0 \end{pmatrix} \quad (14)$$

where a stoichiometric matrix \mathbf{M} is multiplied by the flux vector \mathbf{v} with $v_1 > 0$ satisfying the condition 1 and condition 2 from the definition of self-maintenance. The third condition cannot be satisfied because the production

rates f_a for molecular species a and f_b for molecular species b cannot be greater or equal than 0 at the same time.

In this particular case of the reaction network, all organizations consist of combinations of molecular species that do not react with each other. A set of molecular species where no reaction can take place[†] is obviously closed and self-maintaining. Provided that a set contains molecular species with no reactions among them, Condition 1 of Definition 3 is automatically fulfilled. According to Condition 2 of Definition 3, a zero flux vector $\mathbf{v} = \mathbf{0}$ is multiplied by the stoichiometric matrix M . The result is a zero production rate vector $\mathbf{f} = \mathbf{0}$. The zero vector fulfills Condition 3 of Definition 3, and thus all conditions for self-maintenance are satisfied.

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 any 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 [13].

The second part of the set \mathcal{R}_{XOR} is a set of destructive reactions:

$$\mathcal{D}_{\text{XOR}} = \{a + A \rightarrow \emptyset, b + B \rightarrow \emptyset, c + C \rightarrow \emptyset\}. \quad (15)$$

Combining \mathcal{D}_{XOR} and \mathcal{L}_{XOR} the algebraic chemistry $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \rangle$ implements the XOR logic 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.

Now we set the input variables of the boolean network a and b to initiate the computational process. For the initialization, an inflow reaction is added to the reaction network. We start with providing one input only, leaving the other input variable undefined. Figure 1 (C) shows the results for the four resulting algebraic chemistries $\langle \mathcal{M}_{\text{XOR}}, (\mathcal{R}_{\text{XOR}} \cup \{\emptyset \rightarrow a\}) \rangle$, $\langle \mathcal{M}_{\text{XOR}}, (\mathcal{R}_{\text{XOR}} \cup \{\emptyset \rightarrow A\}) \rangle$, $\langle \mathcal{M}_{\text{XOR}}, (\mathcal{R}_{\text{XOR}} \cup \{\emptyset \rightarrow b\}) \rangle$, and $\langle \mathcal{M}_{\text{XOR}}, (\mathcal{R}_{\text{XOR}} \cup \{\emptyset \rightarrow B\}) \rangle$, respectively. We can see that providing one input signal has further reduced the behavioral freedom of the reaction system. Only three combinations of molecular species are left, which may be encountered in the reaction vessel as 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, sets containing c and C are found to be closed and self-maintaining.

[†] This means that there is no reaction whose lefthand side is a subset of that set. This includes also zero-order (influx) and first-order (e.g. dilution flow) reactions.

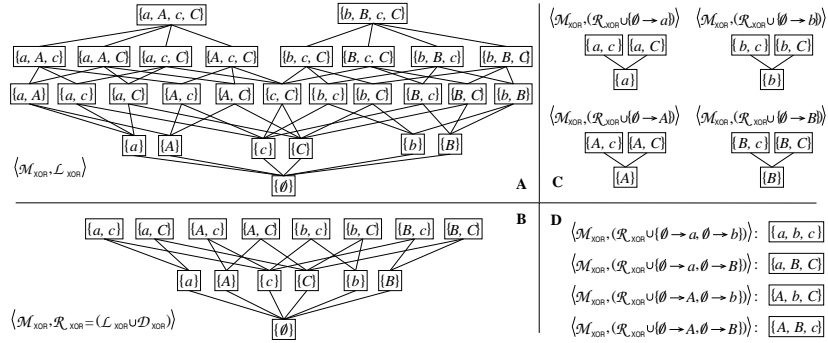


FIGURE 1

Hierarchy of organizations for the chemical reaction network implementing an XOR logic gate. (A) The network consists only of the logical reactions \mathcal{L}_{XOR} . (B) Destructive reactions \mathcal{D}_{XOR} are added to exclude contradictions. The resulting algebraic chemistry $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \rangle$ implements the XOR logic gate without any input specified. (C) One input is defined by adding one influx reaction. (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).

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 be sustained in the reaction vessel regardless of the initial state. We can see that the remaining organization contains the desired output molecular species c or C , respectively. The analyzed algebraic chemistries are $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \cup \{\emptyset \rightarrow a, \emptyset \rightarrow b\} \rangle$, $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \cup \{\emptyset \rightarrow a, \emptyset \rightarrow B\} \rangle$, $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \cup \{\emptyset \rightarrow A, \emptyset \rightarrow b\} \rangle$, and $\langle \mathcal{M}_{\text{XOR}}, \mathcal{R}_{\text{XOR}} \cup \{\emptyset \rightarrow A, \emptyset \rightarrow B\} \rangle$.

4.1 Dynamical Simulation

To validate the results from applying organization theory to the XOR reaction network, stochastic simulations are performed using the simulator packages *MGS* [15, 21] and *Copasi* [20].

Figure 2 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. The parameters for the destructive reactions \mathcal{D}_{XOR} are set to $k = 0.1$, and those for the logical reactions \mathcal{L}_{XOR} are set to $k = 0.001$. At several simulation times, the input is changed in order

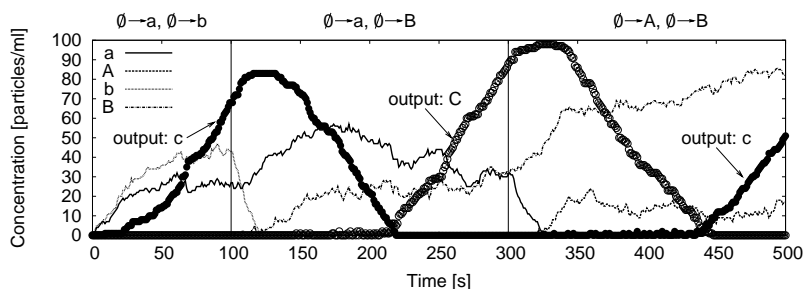


FIGURE 2

Dynamic behavior of the chemical reaction network implementing an XOR logic gate. The time course of all 6 molecular species is shown. Irreversible mass action kinetics are assumed for all reactions. Reaction rates are set to $k = 0.001$ for logical reactions. Reaction rates of destruction reactions are set to $k = 0.1$. For all irreversible constant influxes (e.g., $\emptyset \rightarrow A$), the rates are set to $k = 1$. The reaction system is stochastically simulated with the biochemical network simulator *Copasi* [20] using a compartment size of 1 ml. See text for details.

to observe the switching of the XOR gate. 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. Since molecular species c is generated, the output is computed to $c = 0$.

At simulation time 100 s, the content of input variable b is switched to 1 by replacing the influx of molecular species b with the influx $\emptyset \rightarrow B$. The molecular particles b and c , whose concentrations are still high from the previous computation, deteriorate and finally vanish. The desired output C does not appear until the time point of approximately 200 s. Then, instead of a , the molecular species A is applied as an input starting from simulation time 300 s. The remaining molecules of species a and C from the previous computation decay first and the desired answer c appears in the end.

As seen from the dynamical simulation, the computational result represented by the qualitative final state of the reaction vessel is independent of the initial state. The applied continuous input is the only factor deciding on the final state. The output molecules are generated continuously while undesired species are removed from the reaction vessel by collisions with their anti-particles. When applying two inputs, the analysis of the reaction network revealed that only one organization exists, predicting only one species composition (the species of that organization) to be closed and self-maintaining,

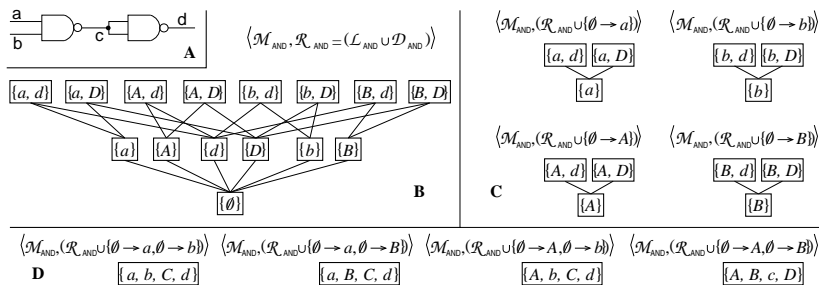


FIGURE 3

Results of the theoretical analysis of the chemical reaction network implementing the logic circuit consisting of multiple gates. (A) Circuit diagram of an AND gate with two NAND gates. (B) The network consists of six logical reactions \mathcal{L}_{AND} and four destructive reactions \mathcal{D}_{AND} . (C) An influx is added to define one input. (D) Another influx is added so that both inputs are defined. Despite the combination of two chemical logic gates, only the organization containing the desired output species is left in (D).

and thus likely to be observed in the reactor. The stochastic simulation confirms the result.

5 CASE STUDY II: MULTIPLE LOGIC GATES

Extensibility and scalability is an advantage of conventional logic gates. Multiple logic gates can be easily connected to realize different forms of computation. In this section, we demonstrate the connectivity of chemical logic gates and scalability of the theoretical analysis. As an example, we implement an AND and an OR gate by combining NAND gates.

5.1 AND Gate by Connecting Two NAND Gates

An AND gate can be constructed by sequentially connecting two NAND gates (Figure 3 (A)). The single logic NAND gates are chemically implemented in the same way as the XOR gate in the previous example.

The boolean network is defined by a set of four boolean variables $\{a, b, c, d\}$ and a set of two boolean functions $\{c = F_c(a, b), d = F_d(c)\}$. The first NAND is associated with F_c and the second is with F_d . The truth table T_c for the first NAND gate has four rows. On the other hand, the truth table T_d of the second NAND gate has only two rows, since the function F_d requires only one

argument. The algebraic chemistry $\langle \mathcal{M}_{\text{AND}}, \mathcal{R}_{\text{AND}} \rangle$ is constructed as follows:

$$\mathcal{M}_{\text{AND}} = \{a, A, b, B, c, C, d, D\} \quad (16)$$

and

$$\mathcal{R}_{\text{AND}} = \mathcal{L}_{\text{AND}} \cup \mathcal{D}_{\text{AND}} = (\mathcal{L}_{\text{AND}}^c \cup \mathcal{L}_{\text{AND}}^d) \cup \mathcal{D}_{\text{AND}} \quad (17)$$

where

$$\begin{aligned} \mathcal{D}_{\text{AND}} &= \{a + A \rightarrow \emptyset, b + B \rightarrow \emptyset, c + C \rightarrow \emptyset, d + D \rightarrow \emptyset\}, \\ \mathcal{L}_{\text{AND}}^c &= \{a + b \rightarrow C, a + B \rightarrow C, A + b \rightarrow C, A + B \rightarrow c\}, \\ \mathcal{L}_{\text{AND}}^d &= \{2c \rightarrow D, 2C \rightarrow d\}. \end{aligned}$$

The two reaction rules in $\mathcal{L}_{\text{AND}}^d$ are equivalent to a NOT operation.

The algebraic chemistry $\langle \mathcal{M}_{\text{AND}}, \mathcal{R}_{\text{AND}} \rangle$ with six reactions and four destructive outflows is analyzed for organizations (closed and self-maintaining sets of molecular species), and the result is shown as the Hasse diagram in Figure 3 (B) depicting a hierarchy of organizations in the reaction network. The algebraic chemistry implements the AND gate without any input specified. Initialization of input variables a and b is represented by adding inflows to the set of reactions. In Figure 3 (C), hierarchies of organizations in the reaction network are shown when one inflow is provided. Hasse diagrams in Figure 3 (D) show the hierarchy of organizations in the reaction network with two input fluxes in which both input variables are defined. The same discussion as in the previous XOR logic gate example can be applied. When both inputs are provided, only one organization remains for every input condition and the organization contains the desired output molecular species d or D , respectively. The theoretical analysis suggests that AND behavior emerges regardless of an initial state and regardless of the dynamics chosen (cf. Section 2.1).

5.2 OR Gate by Connecting Three NAND Gates

Another example of connecting chemical logic gates is an OR circuit with three NAND gates (Figure 4 (A)). The logic circuit can be defined by five boolean variables $\{a, b, c, d, e\}$ and three boolean functions $\{c = F_c(a), d = F_d(b), e = F_e(c, d)\}$. The algebraic chemistry $\langle \mathcal{M}_{\text{OR}}, \mathcal{R}_{\text{OR}} \rangle$ implementing the logic circuit (without any input specified) consists of ten molecular species:

$$\mathcal{M}_{\text{OR}} = \{a, A, b, B, c, C, d, D, e, E\}. \quad (18)$$

The set of reaction rules is

$$\mathcal{R}_{\text{OR}} = \mathcal{L}_{\text{OR}} \cup \mathcal{D}_{\text{OR}} = (\mathcal{L}_{\text{OR}}^c \cup \mathcal{L}_{\text{OR}}^d \cup \mathcal{L}_{\text{OR}}^e) \cup \mathcal{D} \quad (19)$$

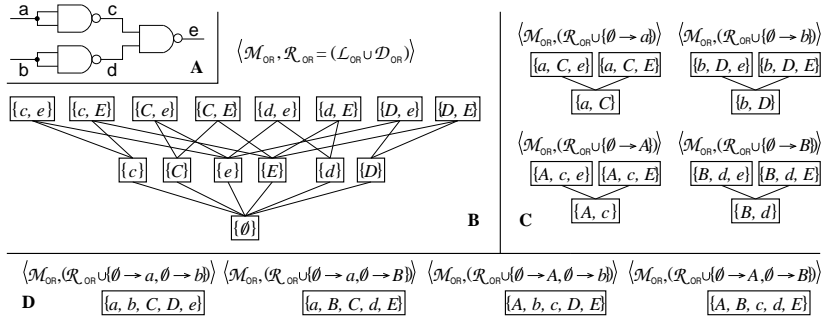


FIGURE 4

Results of the theoretical analysis of a chemical reaction network implementing the logic circuit consisting of multiple gates. (A) Circuit diagram of an OR gate with three NAND gates. (B) The network consists of eight logical reactions \mathcal{L}_{OR} and five destructive reactions \mathcal{D}_{OR} . (C) An influx is added. (D) Two inflows are added, specifying two input values. Despite the combination of three chemical logic gates, only the organization including the desired output species is left in (D).

where

$$\begin{aligned} \mathcal{L}_{\text{OR}}^c &= \{2a \rightarrow C, 2A \rightarrow c\}, \\ \mathcal{L}_{\text{OR}}^d &= \{2b \rightarrow D, 2B \rightarrow d\}, \\ \mathcal{L}_{\text{OR}}^e &= \{c + d \rightarrow E, c + D \rightarrow E, C + d \rightarrow E, C + D \rightarrow e\}, \\ \mathcal{D}_{\text{OR}} &= \{a + A \rightarrow \emptyset, b + B \rightarrow \emptyset, c + C \rightarrow \emptyset, d + D \rightarrow \emptyset, e + E \rightarrow \emptyset\}. \end{aligned}$$

Given the algebraic chemistry, the reaction network is analyzed with chemical organization theory and the result is shown in Figure 4 (B). In Figure 4 (C) and (D), Hasse diagrams depicting the hierarchy of organizations in the chemical reaction network including influxes are shown. As the other cases, one inflow is not enough to determine the output since output molecular species e and E are both found to be a member of the organizations. Defining a value for both input variables, by adding two influxes to the reaction network, reduces the number of organizations in the network to one, and the only organization consists of the desired combination of molecular species.

It is interesting to note that in our current implementation of a chemical OR gate, the output is not determined by a single input flux like $\emptyset \rightarrow B$ ($b = 1$), while input a is unspecified (Figure 4 (C), right). Theoretically, for $b = 1$ the output should be 1, independently of a . We can now use chemical organization theory to search for chemical networks that are also able to cope

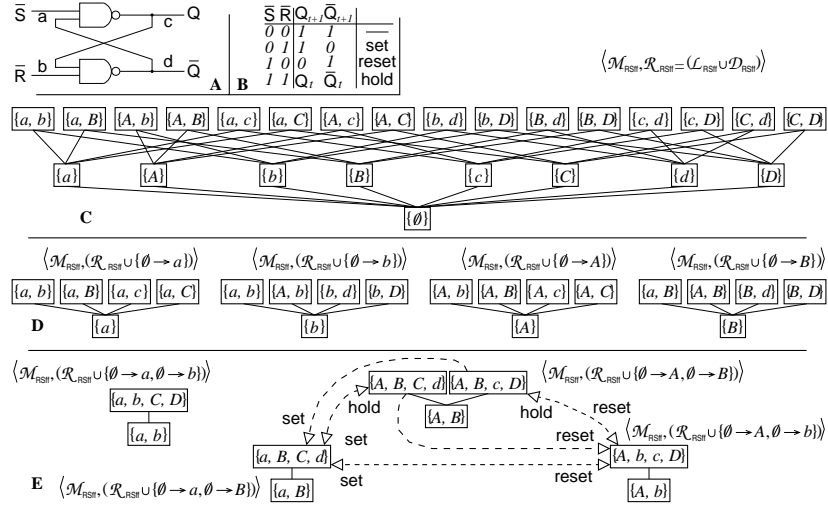


FIGURE 5

Analysis of a chemical reaction network implementing an RS flip-flop circuit with respect to its emergent behavior at the systems level. (A) Circuit diagram of the RS flip-flop. (B) Truth table describing its behavior. (C) Hierarchy of organizations of the reaction network. (D) An influx is added to define one input. (E) Two inflows are added, specifying two input values. The analysis using chemical organization theory reveals that we can expect a dynamical behavior corresponding to the operation of a flip-flop circuit. See text for details.

with unspecified inputs (not shown here).

6 CASE STUDY III: 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. When we analyze the organizations of our chemical instantiation of the flip-flop, the bistability of the circuit will also become apparent. 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 5 (A). The behavior can be described by the truth table as shown in Figure 5 (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 $(\bar{S}, \bar{R}) = (0, 1)$ changes the output Q to 1, and the “reset” operation $(\bar{S}, \bar{R}) = (1, 0)$ 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, i.e. $(\bar{S}, \bar{R}) = (1, 1)$. Normally, the input $(\bar{S}, \bar{R}) = (0, 0)$ is prohibited because the circuit will go into a state where $Q = 1$ and $\bar{Q} = 1$. Application examples for the flip-flop are memory and counter circuits.

The flip-flop logic circuit can be defined by the set of four boolean variables $\{a, b, c, d\}$ and the set of two boolean functions $\{c = F_c(a, d), d = F_d(b, c)\}$. Variables a and b are input variables for the boolean network and the internal variables are c and d . According to the recipe described in Section 3, the algebraic chemistry $\langle \mathcal{M}_{RSff}, \mathcal{R}_{RSff} \rangle$ is constructed. The set of molecular species consists of eight molecular species

$$\mathcal{M}_{RSff} = \{a, A, b, B, c, C, d, D\}. \quad (20)$$

The set of reaction rules is composed of three sets

$$\mathcal{R}_{RSff} = \mathcal{L}_{RSff} \cup \mathcal{D}_{RSff} = (\mathcal{L}_{RSff}^c \cup \mathcal{L}_{RSff}^d) \cup \mathcal{D}_{RSff} \quad (21)$$

where

$$\begin{aligned} \mathcal{L}_{RSff}^c &= \{a + d \rightarrow C, a + D \rightarrow C, A + d \rightarrow C, A + D \rightarrow c\}, \\ \mathcal{L}_{RSff}^d &= \{b + c \rightarrow D, b + C \rightarrow D, B + c \rightarrow D, B + C \rightarrow d\}, \\ \mathcal{D}_{RSff} &= \{a + A \rightarrow \emptyset, b + B \rightarrow \emptyset, c + C \rightarrow \emptyset, d + D \rightarrow \emptyset\}. \end{aligned}$$

When we apply our analysis to the algebraic chemistry $\langle \mathcal{M}_{RSff}, \mathcal{R}_{RSff} \rangle$ implementing the RS flip-flop without any input specified, we found 25 organizations consisting of up to two molecular species, which do not react (Figure 5 (C)). If values of the two input variables are defined, two influxes are added to the set of reaction rules \mathcal{R}_{RSff} so that four algebraic chemistries are analyzed $\langle \mathcal{M}_{RSff}, \mathcal{R}_{RSff} \cup \{\emptyset \rightarrow a, \emptyset \rightarrow b\} \rangle$, $\langle \mathcal{M}_{RSff}, \mathcal{R}_{RSff} \cup \{\emptyset \rightarrow a, \emptyset \rightarrow B\} \rangle$, $\langle \mathcal{M}_{RSff}, \mathcal{R}_{RSff} \cup \{\emptyset \rightarrow A, \emptyset \rightarrow b\} \rangle$, and $\langle \mathcal{M}_{RSff}, \mathcal{R}_{RSff} \cup \{\emptyset \rightarrow A, \emptyset \rightarrow B\} \rangle$. As seen in Figure 5 (E), the number of organizations found in the network is reduced to two or three for each input case. Since the output species $c, C, d,$ and D are in the set of the reactants, no reaction occurs when those species are not present in the reaction vessel. Thus, the smallest organization contains only the two inflow species. Above it, the designated output species are included in the organizations. This implies that the presence of the output species $c, C, d,$ or D in the reaction vessel is necessary for the flip-flop operation. In other words, the input molecular species

alone cannot generate the organization representing an operational mode of the flip-flop.

The operation of the flip-flop can be described by transitions between organizations containing output species: 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 $\{\emptyset \rightarrow a, \emptyset \rightarrow B\}$ and $\{\emptyset \rightarrow A, \emptyset \rightarrow b\}$ to the set of reaction rules, respectively.

For the hold operation (including $\emptyset \rightarrow A, \emptyset \rightarrow B$), the flip-flop has two stable states represented by the organizations $\{A, B, C, d\}$ and $\{A, B, c, D\}$. If the reaction vessel had been in organization $\{a, B, C, d\}$ previously, 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 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 5 (D). A set of molecular species that no reaction rule (including decay reaction) is applicable is an organization because no molecular species is produced (closed) or consumed (self-maintaining). The smallest organizations contain only the input species with the influx. Adding one species that does not interact with the input species forms another organization. Since adding another species makes a reaction rule applicable and molecular species are used up with no reproduction, there exists no organization of size greater than two.

6.1 Dynamical Simulation

In order to validate the discussion of the previous section we performed stochastic simulations (using *MGS* [15, 21] and *Copasi* [20]) of reaction systems implementing the chemical flip-flop. Figure 6 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. The kinetic parameters are set to 0.1 for the second-order reactions that produce output species c, C, d , or D . For destructive reactions, the kinetic parameters are set to 0.001. 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 reaction vessel. The system eventually reaches a state in which only members of the organization

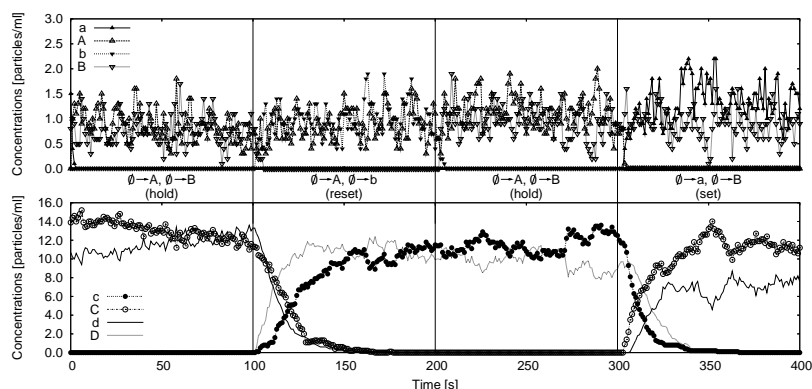


FIGURE 6

Dynamic behavior of the chemical reaction network implementing a RS flip-flop logic circuit. 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. The kinetic parameters are set to $k = 1$ for all zero-order reactions (e.g., $\emptyset \rightarrow A$). The kinetic parameter is set to $k = 0.001$ for destructive reactions. For the other second-order reactions producing output species c , C , d , or D , the kinetic parameter is $k = 0.1$. The reaction system is stochastically simulated with the biochemical network simulator *Copasi* [20] using a compartment size of 10 ml.

$\{A, b, c, D\}$ 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 states of the reaction vessel in terms of molecular species present 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.

7 CASE STUDY IV: AN OSCILLATOR

The final case study should elucidate how our method behaves when applied to boolean circuits exhibiting periodic attractors. With a direct feedback, a NAND logic gate can be configured as a controllable oscillator. Analyzing this system shows that the two alternating states are represented by only one organization in the corresponding chemical system. The organization is the union of the sets of molecular species representing each of these states. We will show that, compared to organizations representing fixed points, the organization representing an oscillation contains “contradicting” molecular pairs like a and A .

Figure 7 (A) shows a circuit diagram of the oscillator with a NAND gate (decomposed into an AND and a NOT gate) and a truth table describing the oscillatory behavior. A feedback loop is formed by feeding the output from the NOT gate to one of the inputs. The dynamical behavior has two operational modes depending on the value of the input variable a , which is the open input of the circuit. When $a = 0$, output variable d and the linked input variable b become 1, independently of an initial value of the other variable b .

The stationary state with $b = d = 1$ is one operational mode of the circuit, while the other is an oscillation between two states. Setting $a = 1$ causes the output variable d and linked variable b to alternate between 0 and 1. Provided that b contained 0 at time t , output variable d becomes 1. Since variable d is connected to variable b , the contents of variable b at time $t + 1$ is switched to 1 which is \bar{b} at time t . When the value of b becomes 1 at time t , variable d will get a value of 0, and so does variable b at time $t + 1$. Repeating the process successively, the value of boolean variables b and d will oscillate between 0 and 1 for each time step.

7.1 Chemical Implementation without Amplified Feedback

The oscillator can be defined by a set of three boolean variables $\{a, b, c\}$ and a set of two boolean functions $\{c = F_c(a, b), b = F_b(c)\}$. According to the recipe given in Section 3, an algebraic chemistry $\langle \mathcal{M}_{osc1}, \mathcal{R}_{osc1} \rangle$ is designed as follows:

$$\mathcal{M}_{osc1} = \{a, A, b, B, c, C\} \quad (22)$$

and

$$\mathcal{R}_{osc1} = \mathcal{L}_{osc1} \cup \mathcal{D}_{osc1} \quad (23)$$

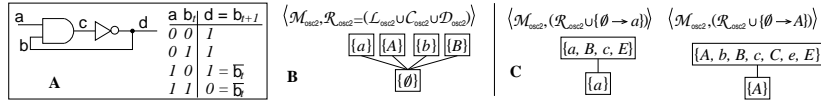


FIGURE 7

Result of analyzing a chemical reaction network implementing a controllable oscillator using chemical organization theory. (A) Circuit diagram and the dynamical oscillatory behavior described as a truth table. (B) Catalytic reactions \mathcal{C}_{osc2} for amplifying feedback signals are introduced into the set of reactions \mathcal{R}_{osc2} . (C) When an influx $\emptyset \rightarrow A$ is added, a set $\{A, b, B, c, C, e, E\}$ is also found to be an organization which could be interpreted as the oscillatory behavior since pair-wise molecular species (e.g., b and B) are both in the organization. The destructive reactions \mathcal{D}_{osc2} allow alternative dis-/appearance of the two species.

where

$$\begin{aligned} \mathcal{L}_{osc1} &= \{a + b \rightarrow c, a + B \rightarrow c, A + b \rightarrow c, A + B \rightarrow C, c \rightarrow B, C \rightarrow b\}, \\ \mathcal{D}_{osc1} &= \{a + A \rightarrow \emptyset, b + B \rightarrow \emptyset, c + C \rightarrow \emptyset\}. \end{aligned}$$

The algebraic chemistry $\langle \mathcal{M}_{osc1}, \mathcal{R}_{osc1} \rangle$ implements the oscillator circuit without input a specified so far. Without input, there are five organizations each containing not more than one molecular species: $\{\emptyset\}$, $\{a\}$, $\{A\}$, $\{b\}$, and $\{B\}$. As a result, there cannot be any oscillation.

The same is true, as expected, when the input variable a is initialized to 0. In that case the algebraic chemistry is modified to $\langle \mathcal{M}_{osc1}, (\mathcal{R}_{osc1} \cup \{\emptyset \rightarrow a\}) \rangle$. There are two organizations $\{a\}$ and $\{a, B, c\}$. The latter set of molecules corresponds to the expected (stationary) behavior of the boolean circuit.

For $a = 1$ the boolean circuit oscillates. When considering the corresponding algebraic chemistry $\langle \mathcal{M}_{osc1}, (\mathcal{R}_{osc1} \cup \{\emptyset \rightarrow A\}) \rangle$, the set $\{A\}$ is found to be the only organization. Hence, given dynamics, no matter how we initialize the reaction system, only molecules of species A and nothing else will inevitably remain after some transient, and there is obviously no oscillation possible. The reason for this behavior is that, apart from $\{A\}$, there is no set of molecular species that is self-maintaining. The lack of self-maintenance is due to the destruction of molecules through the reactions $b + B \rightarrow \emptyset$ and $c + C \rightarrow \emptyset$, as long as there are molecules of type b , B , c , and C left.

7.2 Chemical Implementation with Amplified Feedback

The preceding investigation showed that the naively derived chemical system cannot oscillate like the boolean circuit because the necessary molecular species are not self-maintaining. A solution to this problem is to counteract the consumption of molecules by introducing an amplification mechanism for each periodically changing variables, as already noted in Section 3.1

We chose variable b to be amplified, which is realized by a catalytic reaction. The new algebraic chemistry $\langle \mathcal{M}_{osc2}, \mathcal{R}_{osc2} \rangle$ contains two additional molecular species e and E , which are produced instead of b and B , respectively (*i.e.*, they replace b and B in the previous reaction rules). Molecular species b and B are now produced by catalytic reactions of the form $e \rightarrow e + b$ and $E \rightarrow E + B$. We can see that b and B can now be consumed by other reactions without causing a drain of the output of the NAND gate.

The resulting chemistry is defined as follows: the set of molecular species is

$$\mathcal{M}_{osc2} = \{a, A, b, B, c, C, e, E\} \quad (24)$$

and the set of reaction rules is

$$\mathcal{R}_{osc2} = \mathcal{L}_{osc2} \cup \mathcal{C}_{osc2} \cup \mathcal{D}_{osc2} \quad (25)$$

where \mathcal{C}_{osc2} is a set of catalytic reactions. The set of logical reactions becomes

$$\mathcal{L}_{osc2} = \{a + b \rightarrow c, a + B \rightarrow c, A + b \rightarrow c, A + B \rightarrow C, c \rightarrow E, C \rightarrow e\},$$

and the set of catalytic reactions is

$$\mathcal{C}_{osc2} = \{e \rightarrow e + b, E \rightarrow E + B\}.$$

Since there are four pairs of molecular species in the algebraic chemistry, the set of destructive reactions is now

$$\mathcal{D}_{osc2} = \{a + A \rightarrow \emptyset, b + B \rightarrow \emptyset, c + C \rightarrow \emptyset, e + E \rightarrow \emptyset\}.$$

Given the algebraic chemistry $\langle \mathcal{M}_{osc2}, \mathcal{R}_{osc2} \rangle$ implementing the controllable oscillator, chemical organization theory is applied to find organizations in the reaction network. The result of the analysis is shown in Figure 7 (B) as a hierarchy of organizations. The algebraic chemistry is extended by an influx to analyze the case in which variable a is initialized to 0 or 1. Figure 7 (C) shows the hierarchies of organizations found in the extended reaction networks. The smallest organizations for both input cases are composed of the

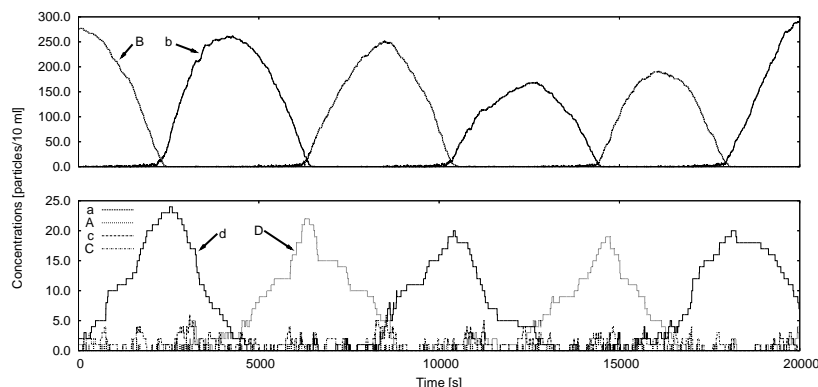


FIGURE 8

Dynamical oscillatory behavior of the chemical reaction network involving a feedback loop as shown in Figure 7. The loop is implemented with a catalytic reaction producing an input species using an output species as a catalyst. The upper figure shows the dynamical concentration changes of the species B and b , and those of the other species are shown in the lower figure. For all first- and second-order reactions, irreversible mass action kinetics is assumed, and the kinetic parameter is set to 0.01. An influx is assumed as an irreversible constant flux with a kinetic parameter of 0.001. The compartment volume is set to 10 ml.

single molecular species with influx because no reaction occurs without a feedback signal. For each input case, the biggest organizations correspond to the operational modes of the oscillator.

When influx $\emptyset \rightarrow A$ is added to the reaction network, the biggest organization is the set $\{A, b, B, c, C, e, E\}$. This implies that the pair-wise molecular species like b and B or c and C are sustained in a reaction vessel even though the two pair-wise molecules decay instantly upon collision due to the destructive reactions such as $b + B \rightarrow \emptyset$. An interpretation of the situation is the oscillating operational mode. Due to the amplified feedback reaction, coexistence of the pair-wise species is now possible.

Dynamical Simulation

To confirm the interpretation of the persistence of the pair-wise species in an organization, we stochastically simulated the reaction vessel using *Copasi* [20]. As an initial state of the reaction vessel, it is necessary to have a positive concentration of non-input molecular species because input molecules a and A cannot produce anything else without other molecular species (*cf.* Figure 7

(D)). We chose the concentration of molecular species B as approximately 25 molecules per ml. The dynamical concentration profile in the reaction compartment is shown in Figure 8. Alternative appearance of the pair-wise molecular species b and B in the upper graph or d and D in the lower graph is apparent.

In general, we can say that a boolean circuit that has periodic attractors (*e.g.*, a circuit that can oscillate) will lead to chemical organizations that contain “contradicting” molecular pairs such as b and B . Thus we can take those organizations as indicators for oscillatory behavior. However, our theory does not allow to say more about the nature of that oscillation. Actually, it is possible that for specific rate laws chosen, we might obtain a stationary state in the chemical system, whereas the corresponding attractor of the boolean network is periodic. Under which circumstance this is the case and how periodic attractors appear in the light of chemical organization theory has to be studied theoretically in more detail in the future.

8 CONCLUDING REMARKS

In this paper, we propose a theoretical analysis method 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 an 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. Even though a few of the logic gates are connected, the hierarchy of organizations is helpful for analyzing the emergent dynamical behavior. 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 bistability is reflected by two organizations found in the network. Using chemical organization theory, we were able to explain the properties of the chemical flip-flop in a new, comprehensible way by referring to the Hasse diagram of organizations (Figure 5). Furthermore the “constructive” dynamics of the flip-flop could be described as a movement between organizations (Figure 5 (E)). This description is more compact than a classical description referring to the 8-dimensional concentration state space, as

demonstrated in Section 6.1.

Oscillatory behavior causes chemical organizations containing “contradictory” molecular pairs such as a and A . In this light it should be noted that the chemical system is more complex than the original boolean circuit because an on- and off-signal can be present at the same time. Furthermore, variables can be unspecified, e.g. representing an unspecified “open” input. Even in that case, the dynamics of the chemical system is well defined, as opposed to the boolean network.

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 repertoire of potential behavior patterns. Since only network structure is considered for the analysis, non-chemical reaction networks, e.g. social interaction networks, can also 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 thank Olivier Michel and Jean-Louis Giavitto for introducing us to the inspiring MGS system. We also would like to thank Bashar Ibrahim and Thorsten Lenser for valuable comments. 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] Adamatzky, A. and Costello, B.D.L. (2002). Experimental logical gates in a reaction-diffusion medium: The XOR gate and beyond. *Phys. Rev. E*, **66**(4 Pt 2):046112.
- [2] Bailey, J.E. (2001). Complex biology with no parameters. *Nat. Biotechnol.*, **19**(6):503–504.
- [3] Banâtre, J.P., Fradet, P., and Radenac, Y. (2004). Principles of chemical programming. In Abdennadher, S. and Ringeissen, C., editors, *RULE’04 Fifth International Workshop on Rule-Based Programming*, pages 98–108. Tech. Rep. AIB-2004-04, Dept. of Comp. Sci., RWTH Aachen, Germany.
- [4] Banâtre, J.P. and Métayer, D.L. (1986). A new computational model and its discipline of programming. Tech. Rep. RR-0566, INRIA.
- [5] Banâtre, J.P. and Métayer, D.L. (1990). The GAMMA model and its discipline of programming. *Sci. Comput. Program.*, **15**(1):55–77.

- [6] Banzhaf, W., Dittrich, P., and Rauhe, H. (1996). Emergent computation by catalytic reactions. *Nanotechnology*, **7**(1):307–314.
- [7] Barabási, A.L. and Oltvai, Z.N. (2004). Network biology: Understanding the cell’s functional organization. *Nat. Rev. Gen.*, **5**:101–113.
- [8] Bedau, M.A., Buchanan, A., Gazzola, G., Hanczyc, M., Maeke, T., McCaskill, J., Poli, I., and Packard, N.H. (2005). Evolutionary design of a DDPD model of ligation. In *7th International Conference on Artificial Evolution (EA’05)*, LNCS. Springer, Berlin. (submitted).
- [9] Berry, G. and Boudol, G. (1992). The chemical abstract machine. *Theor. Comput. Sci.*, **96**(1):217–248.
- [10] Bhalla, U.S. and Iyengar, R. (1999). Emergent properties of networks of biological signaling pathways. *Science*, **283**:381–387.
- [11] Deckard, A. and Sauro, H.M. (2004). Preliminary studies on the in silico evolution of biochemical networks. *Chembiochem.*, **5**(10):1423–1431.
- [12] Dittrich, P. (2005). The bio-chemical information processing metaphor as a programming paradigm for organic computing. In *Proceedings of ARCS: Architecture of Computing Systems*, LNCS. Springer, Berlin. to appear.
- [13] Dittrich, P. and Speroni di Fenizio, P., (2005). Chemical organization theory: towards a theory of constructive dynamical systems. arXiv:q-bio.MN/0501016.
- [14] Fontana, W. and Buss, L.W. (1994). "The arrival of the fittest": Toward a theory of biological organization. *B. Math. Biol.*, **56**:1–64.
- [15] Giavitto, J.L. and Michel, O. (2001). MGS: a rule-based programming language for complex objects and collections. In van den Brand, M. and Verma, R., editors, *Electr. Notes in Theor. Comput. Sci.*, volume 59. Elsevier Science Publishers.
- [16] Kauffman, S.A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**:437–467.
- [17] Koza, J.R., Bennett III, F.H., Andre, D., and Keane, M.A. (1999). *Genetic Programming III: Automatic Programming and Automatic Circuit Synthesis*. Morgan Kaufmann, San Francisco.
- [18] Kremling, A., Jahreis, K., Lengeler, J.W., and Gilles, E.D. (2000). The organization of metabolic reaction networks: A signal-oriented approach to cellular models. *Metab. Eng.*, **2**(3):190–200.
- [19] Küppers, B.O. (1990). *Information and the Origin of Life*. MIT Press, Cambridge, MA.
- [20] Mendes group at VBI and Kummer group at EML research, (2005). COPASI: Homepage. Retrieved May 31, 2005, from <http://www.copasi.org/>.
- [21] Michel, O., Giavitto, J.L., Cohen, J., and Spicher, A., (2005). The MGS home page. Retrieved Nov 14, 2005, from <http://mgs.lami.univ-evry.fr/>.
- [22] Müller-Schloer, C. (2004). 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*, pages 2–5. ACM Press, New York.
- [23] Müller-Schloer, C., Malsburg, M., and Würtz, R.P. (2004). Aktuelles Schlagwort: Organic Computing. *Inf. Spektr.*, **27**(4):332–336.
- [24] Păun, G. (2000). Computing with membranes. *J. Comput. Syst. Sci.*, **61**(1):108–143.
- [25] Păun, G. (2002). *Membrane Computing: An Introduction*. Nat. Comput. Ser. Springer, Berlin.

- [26] Speroni di Fenizio, P. and Dittrich, P. (2002). Artificial chemistry's global dynamics. movement in the lattice of organisation. *The Journal of Three Dimensional Images*, **16**(4):160–163.
- [27] Stelling, J., Sauer, U., Szallasi, Z., Doyle, F.J., and Doyle, J. (2004). Robustness of cellular functions. *Cell*, **118**:675–685.
- [28] Tsuda, S., Aono, M., and Gunji, Y.P. (January 2004). Robust and emergent *physarum* logical-computing. *Biosystems*, **73**(1):45–55.
- [29] von der Malsburg, C., (1999). The challenge of organic computing. Memorandum, Comp. Sci. Dept.
- [30] Würtz, R.P. (2004). Organic computing for face and object recognition. In Dadam, P. and Reichert, M., editors, *Informatik 2004*, volume 2, pages 636–640. Gesellschaft für Informatik.
- [31] Zauner, K.P. and Conrad, M. (2001). Enzymatic computing. *Biotechnology Progress*, **17**(3):553–559.
- [32] Zauner, K.P. (2005). From prescriptive programming of solid-state devices to orchestrated self-organisation of informed matter. In Banâtre, J.P., Giavitto, J.L., Fradet, P., and Michel, O., editors, *Unconventional Programming Paradigms: International Workshop UPP 2004*, volume 3566 of *LNCS*, pages 47–55. Springer, Berlin.