Common Organizational Structures within Two Chemical Flip-Flops

Naoki Matsumaru[1], Thorsten Lenser[1], Florian Centler[2],
Pietro Speroni di Fenizio[3], Thomas Hinze[1], and Peter Dittrich[1]

[1] Bio Systems Analysis Group
Jena Centre for Bioinformatics (JCB) and Institute of Computer Science
Friedrich Schiller University Jena, 07743 Jena, Germany
`{naoki,thlenser,hinze,dittrich}@minet.uni-jena.de`
`http://www.minet.uni-jena.de/csb`
[2] Department of Environmental Microbiology
Helmholtz Centre for Environmental Research - UFZ
Permoserstr. 15 04318 Leipzig, Germany
`florian.centler@ufz.de`
[3] Artificial Life Group, School of Electronic Engineering
Dublin City University, Dublin 9, Ireland

Abstract. Chemical information processing found in living systems has inspired the design of various organic computing systems. However, finding the right reaction rules appears difficult, because the desired behavior emerges from a large number of microscopic molecular interactions. Here, we follow an approach that applies chemical organization theory to bridge the gap between the microscopic reaction rules and the potential macroscopic behavior implied by them. We analyze two chemical flip-flops originating from two fundamentally different chemical programming methods: manual design and artificial evolution. Although both networks implement the same function, they hardly show no structural similarity and it is quite difficult to understand their function by looking at the network. We show that by applying chemical organization theory to the networks, we can explain their operation and reveal their common behavioral structure.

Key words: Chemical organization theory, chemical computing, natural computing, organization-oriented programming, organic computing

1 Introduction

Chemical reactions are elementary components for information processing in life forms [1]. In bacteria, environmental chemical signals are processed to coordinate movements (i.e., chemotaxis [2]). Gene expression controls morphogenesis [3, 4]. Further examples are found in defense coordination and adaptation in the immune system and information broadcasting by the endocrine system. In biology, particularly systems biology [5], chemical reactions are assumed to be a main control mechanisms of biological phenomena, and the language to describe mechanisms is defined as a composition of chemical reactions as specified in Systems Biology Markup Language (SBML) [6]. When exploiting natural, biological systems for computation, chemical reactions are regarded as the basic operation.

Employing molecules and reaction rules as a metaphor, novel computation paradigms have been explored [7, 8, 9, 10, 11, 12, 13]. Essentially, these chemical computing models refer the elementary units as molecules and the operations are described in the form of reactions among those molecules. Given the inputs of the computation as the initial configuration of reaction vessels, the output emerge from local interactions in accordance with the reaction rules given [14]. Although there are approaches such that spatial configuration of the reaction vessel is an important factor [15, 16, 17], we here focus on computation models with a well-stirred reaction vessel. In this context, programming means to specify reaction rules forming a chemical reaction network.

Dissimilarity between natural chemical systems and conventional digital computers is emphasized with respect to levels of programmability [18]. The conventional digital computers are contrived to achieve high programmability by restricting the behaviors of computational entities to be context insensitive. In natural molecular systems, on the other hand, the context sensitivity of computational entities are preserved to be highly evolvable or adaptable and computationally efficient. According to the trade-off principles developed by Conrad [19], programmability is disregarded for those two qualities. Conrad [20] and also Teuscher [21] claimed that different programming principles from those of silicon-based computers are desired for such (bio)chemical systems.

One of fundamental difficulties of programming chemical reaction systems is the gap between reaction rules specified at the microscopic level and global systems' behavior emerging at the macroscopic level. Chemical reaction rules are the fundamental components of the programs, and a global systems' state is interpreted as outcomes of computation. The relation between those two levels is not trivial. Reaction rules are likely to be contingent on each other, and the tangled reactions are operated in parallel. It seems scarcely possible, in general, to predict the global behavior from local reaction rules. Since that predictability is required for programming in a constructive way [22], this micro-macro gap has to be bridged. In passing these difficulties are generally an issue when a system consists of a numerous number of entities, and theoretical understanding of emergent behaviors becomes promoted [23].

The notion of chemical organizations developed by Dittrich and Speroni di Fenizio [24] has been employed as a guide for programming of chemical reaction systems [25]. Following Fontana and Buss [26], an organization is defined as a set of molecular species that is closed and self-maintaining. The hierarchy of all organizations in a reaction network represents its organizational structure, which can be used to describe the dynamical (qualitative) behavior of a reaction system as a movement between organizations [27]. Dittrich and Speroni di Fenizio [24] have shown as a theorem that only species that form an organization can makeup a stationary state. Regarding this theorem as a bridge between local reaction rules and the global systems' behavior, analyzing a reaction network with respect to the organizational structure embedded within helps to predict computational behavior of the reaction systems.

We further demonstrate in this paper the usefulness of the organizational analysis for programming chemical reaction systems. For that purpose, two reaction networks are adopted from [28] and [29]. Those were individually developed using the different programming strategies but for the same computational task, namely implementing a RS flip-flop. Although the intended task is the same, the resulting reaction networks are distinctive. By analyzing organizations within the networks, we argue that the target function is reflected to the common organizational structures, which are not visible from the normal graph figure. In the next section, we briefly describe the theory of chemical organizations adopted from [24]. The reaction networks adopted and the organizational structures within are compared (Section 3). Finally, in Section 4 we further discuss opportunities of organizational analysis with respect to chemical programming.

2 Chemical Organization Theory

The aims of the chemical organization theory have been to derive dynamical behaviors of reaction systems from algebraic analysis of the underlying reaction network. Given a **reaction network** $\langle \mathcal{M}, \mathcal{R} \rangle$ where $\mathcal{M}$ is a set of molecular species and $\mathcal{R}$ is a set of reaction rules, a **reaction system** is an instance of dynamical systems based on the given network and realized by putting a multiple copies (**molecules**) of species from set $\mathcal{M}$ into a reaction vessel. The molecules react according to reaction rules given in $\mathcal{R}$, and dynamics of the reactor and reactions are additionally defined [30]. Dynamical behaviors of the reaction systems are represented by dynamical changes of the concentration profile in the state space.

A **chemical organization** is defined as a set of molecular species that is closed and self-maintaining [24, 26], and those properties are determined only from algebraic analysis on network topology. This definition implies that existence of exact organizations in the dynamical systems are algebraically supported by the network structure. In other words, organizations are sets of species existing more likely in the dynamical systems with positive concentrations when the system is in a stationary state. The closure property prevents generation of extra species, and the self-maintenance property keeps all species of organizations from disappearing. Identifying which species combination satisfies the criteria to be an organization, the reaction network is decomposed into over-lapping sub-networks. We visualize the set of all organizations with a Hasse-diagram, in which organizations are arranged vertically according to their size in terms of the number of their members. Two organizations are connected by a line if the upper organization contains all species of the lower organization and there is no other organization between them. The Hasse-diagram represents the hierarchical **organizational structure** of the reaction network under study.

A **reaction rule** $\rho \in \mathcal{R}$ can be written according to the chemical notation:

$$l_{a_1,\rho}\, a_1 + l_{a_2,\rho}\, a_2 + \ldots \to r_{a_1,\rho}\, a_1 + r_{a_2,\rho}\, a_2 + \ldots . \tag{1}$$

Note that "+" is not an operator but a separator of elements. **Stoichiometric coefficients** $l_{a,\rho}$ and $r_{a,\rho}$ describe the amount of molecular species $a \in \mathcal{M}$ in reaction $\rho \in \mathcal{R}$ on the left-hand and right-hand side, respectively. The stoichiometric coefficients define the **stoichiometric matrix**

$$\mathbf{S} = (s_{a,\rho}) = (r_{a,\rho} - l_{a,\rho}). \tag{2}$$

An entry $s_{a,\rho}$ of the matrix denotes the net amount of molecules of type a produced in reaction ρ. By multiplying flux vector $\mathbf{v} = (v_\rho) \in \mathbb{R}^{|\mathcal{R}|}_{\geq 0}$ to the stoichiometric matrix, a **production rate** f_a of species $a \in \mathcal{M}$ is given as a differential equation of the form $f_a = \dot{x}_a = (\mathbf{S}\mathbf{v})_a$. A **flux vector** $\mathbf{v} = (v_\rho)$ is specified by kinetic laws of a reaction $\rho \in \mathcal{R}$ and indicates how fast the reaction occurs depending on the current concentration profile or vector $\mathbf{x} \in \mathbb{R}^{|\mathcal{M}|}_{\geq 0}$. We also define mappings $\mathrm{LHS}(\rho) \equiv \{a \in \mathcal{M} : l_{a,\rho} > 0\}$ and $\mathrm{RHS}(\rho) \equiv \{a \in \mathcal{M} : r_{a,\rho} > 0\}$, returning the species with a positive coefficient on the left-hand and right-hand side, respectively. Reaction ρ can take place in a subset of species $A \subseteq \mathcal{M}$ only when $\mathrm{LHS}(\rho) \subseteq A$.

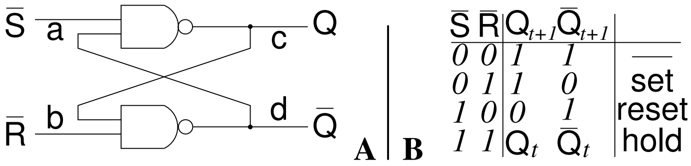
Next, we give the formal definition of two properties of organizations.

Definition 1 (closure). *Given a reaction network* $\langle \mathcal{M}, \mathcal{R} \rangle$*, a set of species* $A \subseteq \mathcal{M}$ *is closed, if for all reactions* ρ *with* $\mathrm{LHS}(\rho) \subseteq A$*, the products are also contained in* A*, that is,* $\mathrm{RHS}(\rho) \subseteq A$*.*

This closure property ensures that there exists no reaction in A producing new species not yet present in the organization using only species of that organization. The other property is a theoretical capability of an organization to maintain all of its members. Since the maintenance possibly involves complex reaction pathways, the stoichiometry of the whole reaction network must be considered.

Definition 2 (self-maintenance). *A set of species* $B \subseteq \mathcal{M}$ *is self-maintaining, if there exists a flux vector* $\mathbf{v} \in \mathbb{R}^{|\mathcal{R}|}$ *such that the following three conditions apply: (1) for all reactions* ρ *that can take place in* B *(i.e.,* $\mathrm{LHS}(\rho) \subseteq B$*) the flux is fixed strictly positive* $v_\rho > 0$*; (2) for all remaining reactions* ρ *(i.e.,* $\mathrm{LHS}(\rho) \nsubseteq B$*), the flux is fixed to zero* $v_\rho = 0$*; and (3) for all species* $a \in B$*, the production rate is non-negative* $(\mathbf{S}\mathbf{v})_a \geq 0$*.*

These conditions read: the reactions that can take place are forced to occur (Condition 1) and the other reactions are forced being deactivated (Condition 2). Then, there still exists a flux vector such that all species in the set are produced at a positive or zero rate (Condition 3). Roughly speaking, in self-maintaining set molecular species consumed by reactions are reproduced by some reaction pathways possibly containing multiple reactions. Combining these, a chemical organization is defined as a set of molecular species that is closed and self-maintaining.



$\bar{S}$	$\bar{R}$	Q_{t+1}	$\bar{Q}_{t+1}$	
0	0	1	1	—
0	1	1	0	set
1	0	0	1	reset
1	1	Q_t	$\bar{Q}_t$	hold

Fig. 1. Circuit diagram and operation mode of flip-flop.

3 Organizational analysis of the chemical flip-flop

Utilizing the notion of chemical organizations described above, chemical programs in the form of reaction networks are evaluated whether the reaction system based on the network functions properly. In order to demonstrate the evaluation with the organizational analysis, in this section, we compare organizational structures embedded within two chemical programs prepared for the same functional operation, implementing a RS flip-flop.

3.1 Chemical flip-flop

Figure 2 shows two chemical reaction networks for implementing the same RS flip-flop operation (see Figure 1). These networks are designed with two different programming strategies: compilation (left panel in Figure 2 and 3) and evolution (right panel in the figures). The former is adopted from [28] and was constructed such that each logical operation composing the flip-flop logic circuit is translated systematically to the reaction rule format. Evolutionary algorithms for the other network are described in [29].

Comparison of these two networks is possible because the same coding scheme is used to represent a logical operation in the form of a chemical reaction rule. To code the four binary variables `a`, `b`, `c` and `d`, making up this flip-flop, to a chemical format, we employ two opposing species x^0 and x^1 for each binary variable `x`, where the presence of x^0 denotes the value `x = 0`, and x^1 denotes `x = 1`. When both or neither species are present in the reaction vessel, the state is interpreted as an undecided output. To help maintaining a valid state inside the reaction system, four destructive reactions $x^0 + x^1 \rightarrow \emptyset$ for all four species pairs $x^i = a^i, b^i, c^i, d^i$ $(i \in \{0, 1\})$ are predefined for both networks. Inputs to logic circuits are induced as influxes of species.

Except for the predefined cooperative decays, there are no reactions in common at all even though those reaction networks are intended to implement the identical flip-flop behavior. As can be seen from the figures, detecting the flip-flop capability is hardly possible. Instead, by analyzing the organizational structure within the reaction networks, a common structure becomes visible.

3.2 Organizational structure

In Figure 3, the organizational structures within those reaction networks are compared. The relevant common structure consists of four organizations labeled

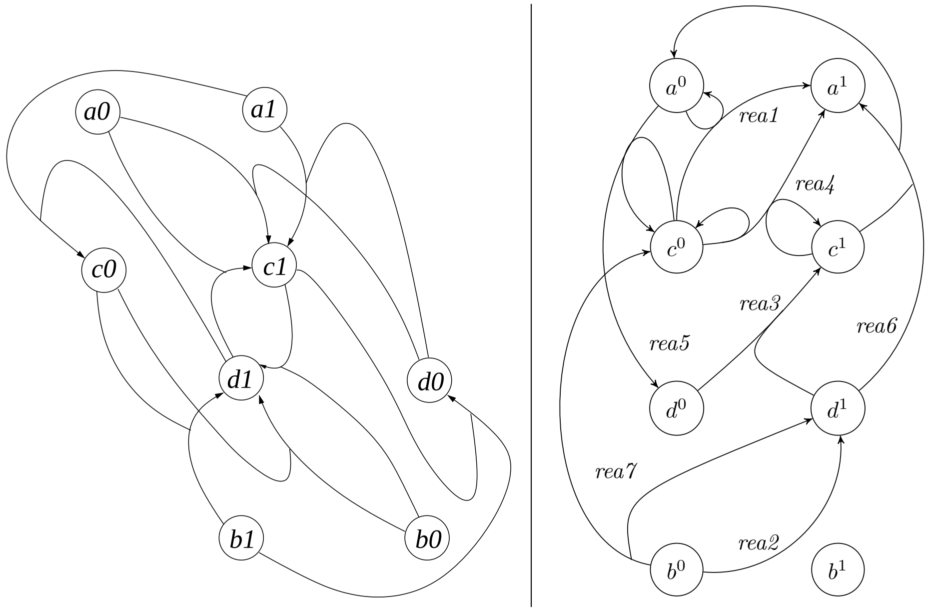


Fig. 2. Chemical reaction network implementing RS flip-flop circuits, designed manually (left) and through an evolutionary process (right). Cooperative decay reactions ($a^1 + a^0 \to \emptyset$, $b^1 + b^0 \to \emptyset$, $c^1 + c^0 \to \emptyset$, $d^1 + d^0 \to \emptyset$) are omitted. These base reaction networks are extended to include inflow reactions, representing the inputs to the flip-flop circuit, depending on the operations. For the hold operation, two reactions $\emptyset \to a^1$ and $\emptyset \to b^1$ are added. The other two operations (set and reset) are initiated by $\emptyset \to a^0$ and $\emptyset \to b^1$, and $\emptyset \to a^1$ and $\emptyset \to b^0$, respectively.

as *orgR*, *orgS*, *orgHR*, and *orgHS*. Both figures depict three panels, corresponding to three operation modes of the flip-flop, and the operations are also imposed on to the figures as a transition between organizations. By activating inflows of $\emptyset \to a^1$ and $\emptyset \to b^0$, the reset operation is implemented. In both networks, a set of molecular species labeled as *orgR*= $\{a^1, b^0, c^0, d^1\}$ becomes closed and self-maintaining. When activating the inflows of $\emptyset \to a^0$ and $\emptyset \to b^1$ for the set operation, both network topologies support the organizational properties of a set *orgS*= $\{a^0, b^1, c^1, d^0\}$. Changing inflow reactions to $\emptyset \to a^1$ and $\emptyset \to b^1$, the hold operation is achieved. Two organizations *orgHR*= $\{a^1, b^1, c^0, d^1\}$ and *orgHS*= $\{a^1, b^1, c^1, d^0\}$ are embedded within both of the reaction networks, reflecting the bistability of the flip-flop circuit. Depending on the state at the previous time step, the hold operation brings the system to a different state, either from *orgS* to *orgHS* or from *orgR* to *orgHR*. The dynamical simulations were, in fact, elaborated to use this common structure (See original papers [28] and [29] for dynamical simulations).

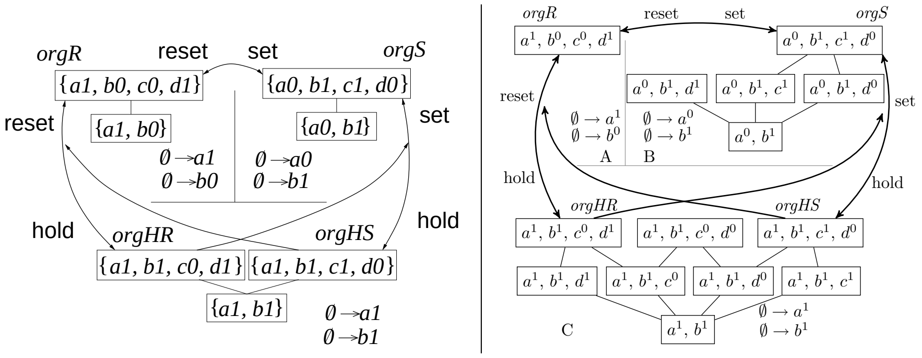


Fig. 3. Organizational structure in the reaction network shown in Figure 2, and the operation mode of the chemical flip-flops. Even though the network structures are different from each other, the relevant organizations *orgR*, *orgS*, *orgHR*, and *orgHS* are embraced within both of the networks. The sameness of the target computation is reflected to those common organizational structures.

4 Conclusion and discussion

In this paper, we illustrated the usefulness of evaluating chemical reaction networks using the notion of chemical organizations. Two reaction networks for the same computational task were adopted, and the organizational structures within are compared. From the normal graph structure, the intended computational task is not visible, and no common structure is realized. It is hardly feasible to detect that those networks are actually for the same computation. On the contrary, organizational analysis brings forward the common structure within those networks, and those common structures are, in fact, essential to implement the intended computational task.

These observations lead to a programming technique utilizing the theory of chemical organizations as a guide to design a reaction network. Organizational structures are first imagined how they are supposed to be for the desired computational task, and then chemical reactions are constructed such that the embedded organizational structures are fit to the desired ones. The organizational structure can be considered as a blueprint of the reaction network. As long as the organizational structure is in agreement, actual network topology can be more or less overlooked. The organizational analysis can be also applied interactively so that reactions are modified bit by bit toward eliminating extra organizations or generating expected ones. These techniques are summarized as general programming principles of *organization-oriented chemical programming* [31]. Incorporating organizational analysis with chemical programming process is, therefore, beneficial.

An advantage of incorporating organizational analysis is that programmers can concentrate on global systems' behaviors, instead of actual network topology at the local level. This point may become critical when dealing with a real wet-lab chemical system. When modifying a single reaction in chemical reaction network, it is inevitable to cause side-effects and it is practically infeasible to modify a single reaction without affecting on the others. Viewing at the global systems' level, the programmers can decide whether the side-effects are acceptable. It could save much efforts to neutralize those effects.

Generally speaking, the term "programming" is often associated with modifying or altering systems. However, when considering "real" natural systems, there may be a programming technique not designing but rather detecting or exploring. Natural systems are very stable as it is, and small modification may cause unpredictable side-effects. To avoid such unpredictability and to even exploit innate stabilities, a chemical programming approach by detecting or exploring desired behaviors in the natural systems is worth investigating. The organizational analysis of biochemical reaction networks may expose another flip-flop module and further. This approach may be applied reversely, that is, to understand biological systems in terms of computation, and that may shed light on a computational aspect of biological phenomena. A trivial case is to translate biological reactions to a combination of logic gates. This line of research follows a philosophy of structural sciences (e.g., [32]), and the concepts of computation may promote the unveiling of a common structure among life forms.

Acknowledgments.

We acknowledge financial supports by the European Union, NEST-project ESIGNET no. 12789 and by the German Research Foundation (DFG) Grant DI 852/4-2.

Bibliography

[1] Küppers, B.O.: Information and the Origin of Life. MIT Press, Cambridge, MA (1990)
[2] Postma, M., Bosgraaf, L., Loovers, H.M., Haastert, P.J.M.V.: Chemotaxis: signalling modules join hands at front and tail. EMBO Rep **5** (2004) 35–40
[3] Jagla, K., Bellard, M., Frasch, M.: A cluster of drosophila homeobox genes involved in mesoderm differentiation programs. Bioessays **23** (2001) 125–133
[4] Lewis, E.B.: A gene complex controlling segmentation in drosophila. Nature **276** (1978) 565–570
[5] Kitano, H.: Systems biology: A brief overview. Science **295** (2002) 1662–1664
[6] Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A.P., Bornstein, B.J., Bray, D., Cornish-Bowden, A., Cuellar, A.A., Dronov, S., Gilles, E.D., Ginkel, M., Gor, V., Goryanin, I.I., Hedley, W.J., Hodgman, T.C., Hofmeyr, J.H., Hunter, P.J., Juty, N.S., Kasberger, J.L., Kremling, A., Kummer, U., Novère, N.L., Loew, L.M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E.D., Nakayama, Y., Nelson, M.R., Nielsen, P.F.,

Sakurada, T., Schaff, J.C., Shapiro, B.E., Shimizu, T.S., Spence, H.D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., Wang, J.: The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. Bioinformatics **19** (2003) 524–531

[7] 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, Tech. Rep. AIB-2004-04, Dept. of Comp. Sci., RWTH Aachen, Germany (2004) 98–108

[8] Banâtre, J.P., Métayer, D.L.: A new computational model and its discipline of programming. Tech. Rep. RR-0566, INRIA (1986)

[9] Banâtre, J.P., Métayer, D.L.: The GAMMA model and its discipline of programming. Sci. Comput. Program. **15** (1990) 55–77

[10] Berry, G., Boudol, G.: The chemical abstract machine. Theor. Comput. Sci. **96** (1992) 217–248

[11] Dittrich, P.: Chemical computing. In Banâtre, J.P., Fradet, P., Giavitto, J.L., Michel, O., eds.: Unconventional Programming Paradigms, International Workshop UPP 2004, Le Mont Saint Michel, France, September 15-17, 2004, Revised Selected and Invited Papers. Volume 3566 of LNCS., Springer, Berlin (2005) 19–32

[12] Păun, G.: Computing with membranes. J. Comput. Syst. Sci. **61** (2000) 108–143

[13] Tschudin, C.: Fraglets - a metabolic execution model for communication protocols. In: Proc. AINS'03. (2003)

[14] Banzhaf, W., Dittrich, P., Rauhe, H.: Emergent computation by catalytic reactions. Nanotechnology **7** (1996) 307–314

[15] Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T.F., Nagpal, R., Rauch, E., Sussman, G.J., Weiss, R., Homsy, G.: Amorphous computing. Commun. ACM **43** (2000) 74–82

[16] Adamatzky, A.: Universal dynamical computation in multidimensional excitable lattices. Int. J. Theor. Phys. **37** (1998) 3069–3108

[17] Tsuda, S., Aono, M., Gunji, Y.P.: Robust and emergent *p*hysarum logical-computing. Biosystems **73** (2004) 45–55

[18] Conrad, M.: The brain-machine disanalogy. BioSystems **22** (1989) 197–213

[19] Conrad, M.: The price of programmability. In Herken, R., ed.: The Universal Turing Machine: A Fifty Year Survey. Oxford University Press, New York (1988) 285–307

[20] Conrad, M.: On design principles for a molecular computer. Commun. ACM **28** (1985) 464–480

[21] Teuscher, C.: Computation in self-assembed avogadro-scale systems — challenges and opportunities. In Adamatzky, A., Bull, L., De Lacy Costello, B., Stepney, S., Teuscher, C., eds.: Unconventional computing 2007, Luniver Press, UK (2007) 315–316 Abstracts of invited talks presented at Unconventional Computing 2007, Bristol, United Kingdom.

[22] Zauner, K.P.: From prescriptive programming of solid-state devices to orchestrated self-organisation of informed matter. In Banâtre, J.P., Giavitto, J.L., Fradet, P., Michel, O., eds.: Unconventional Programming Paradigms:

International Workshop UPP 2004. Volume 3566 of LNCS., Springer, Berlin (2005) 47–55

[23] 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, ACM Press, New York (2004) 2–5

[24] Dittrich, P., Speroni di Fenizio, P.: Chemical organisation theory. Bull Math Biol **69** (2007) 1199–1231

[25] Matsumaru, N., Lenser, T., Hinze, T., Dittrich, P.: Toward organization-oriented chemical programming: A case study with the maximal independent set problem. In Dressler, F., Carreras, I., eds.: Advances in Biologically Inspired Information Systems. Volume 69 of Studies in Computational Intelligence. Springer, Berlin (2007) 147–163

[26] Fontana, W., Buss, L.W.: 'The arrival of the fittest': Toward a theory of biological organization. Bull Math Biol **56** (1994) 1–64

[27] Speroni di Fenizio, P., Dittrich, P., Banzhaf, W.: Spontanous formation of cells in a universal artificial chemistry on a planar graph. In Kelemen, J., Sosik, P., eds.: Advances in Artificial Life. Proc. 6th European Conference on Artificial Life (ECAL 2001), Prague, Czech Republic, September 10 - 14, 2001, LNCS 2159, Springer, Berlin (2001) 206–215

[28] Matsumaru, N., Centler, F., Speroni di Fenizio, P., Dittrich, P.: Chemical organization theory as a theoretical base for chemical computing. International Journal of Unconventional Computing **3** (2007) 285–309

[29] Lenser, T., Matsumaru, N., Hinze, T., Dittrich, P.: Tracking the evolution of chemical computing networks. In Bullock, S., Noble, J., Watson, R.A., Bedau, M.A., eds.: Proceedings of the Eleventh International Conference on Artificial Life, MIT Press, Cambridge, MA (2008) 343–350

[30] Dittrich, P., Ziegler, J., Banzhaf, W.: Artificial chemistries - a review. Artif. Life **7** (2001) 225–275

[31] Dittrich, P., Matsumaru, N.: Organization-oriented chemical programming. In: 7th International Conference on Hybrid Intelligent Systems (HIS). IEEE Conference Proceedings, IEEE (2007) 18–23

[32] Artmann, S.: Artificial life as a structural science. Philosophia naturalis **40** (2003) 183–205