STUDIES IN LOGIC, GRAMMAR AND RHETORIC 17 (30) 2009

**Andrew Schumann**
Belarusian State University, Minsk, Belarus

# CIRCULAR PROOFS IN PROOF-THEORETIC SIMULATION OF BELOUSOV-ZHABOTINSKY REACTION

In the paper I consider abstract machine of reaction-diffusion computing. This machine is constructed by using process algebra. Further, I propose proof-theoretic interpretation of process calculus within the framework of Belousov-Zhabotinsky reaction. I show that some proofs simulating the behaviour of Belousov-Zhabotinsky reaction are circular. This means that the derivation tree simulating the behaviour tree of Belousov-Zhabotinsky reaction has cycles, i.e. some derivable formulas occur among top premisses.

## 1. Introduction

The reaction-diffusion approach to computing was first proposed by Alan Turing in his famous paper titled "The Chemical Basis of Morphogenesis." In his version, design patterns were considered as a conceptual framework for transferring knowledge from biology to computing. The importance of this idea is that reaction-diffusion processes are observed throughout nature in all spatiotemporal pattern formation and self-organization far from equilibrium coupled with diffusion.

After Turing, the interest in controlling self-organized systems and in building up computational models based on them in order to achieve their specific desired behavior has been increasing. Control of self-organization in reaction-diffusion frames is an ambitious goal allowing to simulate all massive-parallel natural processes: chemical reactions, morphogenesis, population dynamics, biological, economic and social evolution and other phenomena that produce a complex network of interactions among the component species.

In the initial meaning, reaction-diffusion systems involve constituents locally transformed into each other by chemical reactions and transported in space by diffusion. The simplest kind of a chemical wave emerges when reactants are converted into products as the front propagates through the reac-

ISBN 978–83–7431–231–8 ISSN 0860-150X 

tion mixture. Usually, chemical reaction-diffusion systems eventually evolve to the state of chemical equilibrium, but there is a possibility of thermodynamic equilibrium, like pattern formation in Belousov-Zhabotinsky reaction. In the latter case, equilibrium of self-organized patterns occur due to interaction of autocatalytic compensated by reaction step blocking the unbounded growth.

Wave is a key notion for analyzing reaction-diffusion processes. It is a form of propagating the spatial coupling of elements from the stable or the most stable state toward the unstable or the least stable one. Due to waves of reagent diffusions there is a chemical reaction. Furthermore, a wave can cause composite patterns in a regime of periodic or chaotic oscillations, like that we observe in Belousov-Zhabotinsky reaction. If a wave causes pattering and regulation of timing independently of input conditions and the letter are determined by the system itself, then this is called auto-wave. Its typical instance is described in Belousov-Zhabotinsky reaction. Auto-waves may be detected in any self-organized systems, some examples of auto-waves are given below. As a result, we can claim that reaction-diffusion nature is observed not just within chemical reactions, but also in different pattering and interactions like as follows:

- Auto-waves of morphogenesis consisting in the processes that generate tissue organization and shape and are usually the downstream response to the timing and patterning.
- Auto-waves in an ecological system that change the network of interacting populations.
- Auto-waves of epidemics, they are observed in the rapid increase of the size of infected population due to the interaction between infected and susceptible individuals.
- Auto-waves of opinion dynamics, i.e. changes in opinions caused by interaction of people.
- Auto-waves of aggregated individual amoebae of the cellular slime mold Dictyostelium discoideum to form a multicellular migrating slug, which moves toward a region suitable for culmination.

As we see, thanks to the above-mentioned examples, the majority of physical, chemical, biological and sociological processes can be described in terms of auto-waves, i.e. propagating fronts of one nature or another.

The dynamics of complex reaction-diffusion frames can be described in terms of activator and inhibitor. Activator, $A$, is a short-range autocatalytic substance, inhibitor, $I$, is its long-range antagonist. $A$ autocatalytically promotes its own synthesis. $I$ inhibits synthesis of $A$, and $I$ can diffuse faster than $A$. The pattern of peaks of $A$ is dependent upon

the reaction-diffusion frame. The dynamics of a self-organizing system is typically non-linear, because of circular or feedback relations between the components. In other words, in complex reaction-diffusion systems there are circular cause-and-effect relations: each component affects the other components, but these components in turn affect the first component. Feedback can have two basic values: positive if activator diffuses faster or negative if inhibitor diffuses faster. In the positive feedback, the recurrent influence reinforces the initial change. In the negative feedback, the reaction is opposite to the initial action, so change is counteracted. The aim of negative feedback is to stabilize the system, by returning to its original states. On the other hand, positive feedback makes changes grow in an explosive manner. It leads to accelerated development, which ends when all components have been absorbed into the new configuration, leaving the system in a stable state.

Diffusion waves of activator and inhibitor interact according to reaction rules. Solutions of those rules are represented by multisets of waves: this accounts for the associativity and commutativity of parallel composition, that is the implicit stirring mechanism. Setting diffusion waves by rules allows us to define reaction-diffusion processes as a kind of computing. This computing is fundamentally similar to parallel computing in that it takes advantage of the many different waves to try many different possibilities of diffusion at once.

Setting up reaction-diffusion computing is a way of simulating physics by a universal computer. This aim to build up a "life computer" is the most ambitious task attempted in unconventional computing, e.g. see for more details [13]. Its difficulties consist in that in conventional approach to computing control is centralized. On the other hand, in self-organizing systems control of the organization is typically distributed over the whole of the system. Due to a centralized control, software is written for serial computation and a problem is broken into a discrete series of executed instructions so that only one instruction may execute at any moment in time. Due to a decentralized, distributed control in parallel computing, a computational problem is solved by the simultaneous use of multiple compute resources and a problem is broken into discrete parts that can be solved concurrently. We can add that reaction-diffusion computing is not just parallel, but massive-parallel. This means that parts of a problem are broken into can be indiscrete and infinite many.

The sequential model of computation called von Neumann paradigm is unapplied, broken in reaction-diffusion computing. The reason is that the latter computing may be presented as a massive-parallel locally-connected mathematical machine with circular and cyclic processes. These machines

cannot have a single centralized source exercising precise control over vast numbers of heterogeneous devices. Interactive-computing paradigm is able to describe concurrent (parallel) computations whose configuration may change during the computation and is decentralized as well. Within the framework of this paradigm, one proposed a lot of so-called concurrency calculi also called process algebras. They are typically presented using systems of equations. These formalisms for concurrent systems are formal in the sense that they represent systems by expressions and then reason about systems by manipulating the corresponding expressions.

One of the unconventional, nature inspired models similar to reaction-diffusion computing is chemical machine in that molecules are viewed as computational processes supplemented with a minimal reaction kinetics. Berry and Boudol first built up a chemical abstract machine [6] as an example of how a chemical paradigm of the interactions between molecules can be utilized in concurrent computations (in algebraic process calculi). In the **second section** of this paper we will consider more general abstract machine of reaction-diffusion computing. This machine is constructed by using process algebra. In the **third section** we will propose proof-theoretic interpretation of process calculus for reaction-diffusion computing within the framework of Belousov-Zhabotinsky reaction. We will show that some proofs simulating the behaviour of Belousov-Zhabotinsky reaction are circular. This means that the derivation tree simulating the behaviour tree of Belousov-Zhabotinsky reaction has cycles (circles), i.e. some derivable formulas themselves occur among top premisses.

## 2. Process calculus of reaction-diffusion computing

A behaviour in reaction-diffusion systems is thought as the total of actions that a system can perform taking into account that we describe certain aspects of behaviour by introducing an abstraction or idealization of the 'real' behaviour. The simplest model of behaviour we will use is to consider behaviour as an input/output function. Evidently, it is a restriction, idealization we will deal with. Usually, such input/output function approach is implemented in automata theory, where a process is modeled as an automaton. An automaton has a number of states and a number of transitions, going from one state to another state. A behaviour is a run, i.e. a path from initial state to final state. There are different ways of modeling reaction-diffusion behaviour within this approach. The most popular is to simulate the behaviour by cellular automata.

In the present paper we will use more general approach presented by process algebra – the study of the behaviour of parallel or distributed systems by algebraic means. It offers tools to describe or specify transition systems, and thus it has tools to talk about parallel composition in general.

Assume that the computational domain $\Omega$ is partitioned into computational cells $c_j = \overline{1,K}$ such that $c_i \cap c_j = \emptyset, i \neq j$ and $\bigcup_{j=1}^{K} c_j = \Omega$. Further, suppose that in the $K$ cells, there are $N$ chemically active species and the state of species $i$ in cell $j$ is denoted by $p_{ij}, i = \overline{1,N}$, $j = \overline{1,K}$. These states are time dependent and they are changed by reactions occurring between waves in the same cell and by diffusion where waves move to adjacent cells. In reaction-diffusion, the species interact concurrently and in a parallel manner.

Our process calculus contain the following basic operators: $Nil$ (inaction), '$\bullet$' (prefix), '$|$' (cooperation), '$\backslash$' (hiding), '$\&$' (reaction/fusion), '$\oplus$' (choice), $a$ (constant or restriction to a stable state). Let $N = \{a, b, \ldots\}$ be a set of names and $L = \{a, \overline{a} \colon a \in N\}$, where $a$ is considered as activator and $\overline{a}$ as inhibitor for $a$, be the set of labels built on $N$. We use the symbols $\alpha$, $\beta$, etc., to range over labels, with $\alpha = \overline{\overline{\alpha}}$, and the symbols $P$, $Q$, etc., to range over states $p_{ij}, i = \overline{1,N}$, $j = \overline{1,K}$. The behaviour of states is given by the syntax:

$$P ::= Nil \mid \alpha \bullet P \mid (P|P) \mid P\backslash X \mid P\&P \mid P \oplus P \mid a$$

An operational semantics for this syntax is defined in Fig. 1. The informal meanings of basic operations are as follows:

$Nil$ This is the empty process which does nothing. In other words, $Nil$ represents the component which is not capable of performing any activities: a deadlocked component.

$\alpha \bullet P$ A process $\alpha \in L$ followed by the process $P$: $P$ becomes activity only after the action $\alpha$ has been performed.

$P|Q$ This is a parallel composition (commutative and associative) of actions: $P$ and $Q$ are performed in parallel.

$P\backslash X$ This restriction operator allows us to force some of $P$'s actions not to occur; all of the actions in the set $X$ are prohibited, i.e. the component $P\backslash X$ behaves as $P$ except that any activities of types within the set $X$ are hidden, meaning that their type is not visible outside the component upon completion.

**Prefix** $\dfrac{}{\alpha \bullet P \longrightarrow^{\alpha} P}$

**Constant** $\dfrac{P \longrightarrow^{\alpha} P'}{a \longrightarrow^{\alpha} P'}\ (a := P)$

**Choice** $\dfrac{P \longrightarrow^{\alpha} P'}{P \oplus Q \longrightarrow^{\alpha} P'}, \quad \dfrac{Q \longrightarrow^{\alpha} Q'}{P \oplus Q \longrightarrow^{\alpha} Q'}$

**Cooperation** $\dfrac{P \longrightarrow^{\alpha} P'}{P|Q \longrightarrow^{\alpha} P'|Q}, \quad \dfrac{Q \longrightarrow^{\alpha} Q'}{P|Q \longrightarrow^{\alpha} P|Q'}$

**Hiding** $\dfrac{P \longrightarrow^{\alpha} P'}{P\backslash X \longrightarrow^{\alpha} P'\backslash X}\ (\alpha \notin X), \quad \dfrac{P \longrightarrow^{\alpha} P'}{P\backslash X \longrightarrow^{\beta} P'\backslash X}\ (\alpha \in X)$

**Fusion** $\dfrac{}{\alpha \bullet P\&\overline{P} \longrightarrow^{\alpha} Nil}, \quad \dfrac{P \longrightarrow^{\alpha} P' \quad Q \longrightarrow^{\alpha} P'}{P\&Q \longrightarrow^{\alpha} P'}, \quad \dfrac{P \longrightarrow^{\alpha} P'}{P\&Q \longrightarrow^{\alpha} P' \oplus Q'}$

**Figure 1.** Operational semantics: inference rules for basic operations. The ternary relation $P \longrightarrow^{\alpha} P'$ means that the initial state $P$ is capable of engaging in action $\alpha$ and then behaving like $P'$

$P\&Q$ This is the fusion of $P$ and $Q$; $P\&Q$ represents a system which may behave as both component $P$ and $Q$. For instance, $Nil$ behaves as $P\&\overline{P}$, where $P$ is an activator and $\overline{P}$ an appropriate inhibitor respectively.

$P \oplus Q$ This is the choice between $P$ and $Q$; $P \oplus Q$ represents a system which may behave either as component $P$ or as $Q$. Thus the first activity to complete identifies one of the components which is selected as the component that continues to evolve; the other component is discarded.

$a$ constants are components whose meaning is given by equations such as $a := P$. Here the constant $a$ is given the behaviour of the component $P$. Constants can be used to describe infinite behaviours, via mutually recursive defining equations.

Thus, in this process calculus we have four kinds of transitions between states: (1) the internal transitions $p_{mj} \longrightarrow^{\alpha} p_{nj}$, i.e. a reaction $\alpha$ in a cell $j$ is a transition from one state $p_{mj}$ before the reaction to the state $p_{nj}$ after the reaction, (2) the external transitions $p_{mk} \longrightarrow^{\alpha} p_{nl}$ (diffusion), i.e. a reaction $\alpha$ in a cell $l$ is a transition from one state $p_{mk}$ in a cell $k$ before the reaction to the state $p_{nl}$ in a cell $l$ after the reaction.

## 3. Proof-theoretic simulation of behaviour in reaction-diffusion systems

A behaviour of reaction-diffusion systems can be viewed as a labelled transition system, which consists of a collection of states $L$ and a collection $\Sigma$ of transitions between them (actions over them). Assume $\Sigma\colon L \mapsto \mathcal{P}(L)$, where $\mathcal{P}(L) = \{T\colon T \subseteq L\}$, i.e. $\Sigma = \bigcup\limits_{a\in L} \{\langle a, \alpha\rangle\colon \alpha \in T \subseteq L\}$. This means that $\Sigma(a)$ consists of all states that a reachable from $a$. The transition system is understood as a triple $\langle L, \Sigma, \longrightarrow\rangle$, where $\longrightarrow \subseteq L \times \Sigma \times L$ is a transition relation that models how a state $p \in L$ can evolve into another state $p' \in L$ due to an interaction $\sigma \in \Sigma$. Usually, $\langle p, \sigma, p'\rangle \in \longrightarrow$ is denoted by $p \longrightarrow^{a} p'$ under assumption that there exists $a$ such that $\langle p, a\rangle \in \Sigma$ and $\langle a, p'\rangle \in \Sigma$.

$$\frac{Ce^{3+}\&HBrO_2\&BrO_3^-\&H^+}{Ce^{4+} \oplus HBrO_2 \oplus H_2O} \tag{1}$$

$$\frac{Ce^{4+}\&BrCH(COOH)_2\&H_2O}{Br^- \oplus Ce^{3+} \oplus HCOOH \oplus CO_2 \oplus H^+} \tag{2}$$

$$\frac{HBrO_2}{HOBr \oplus BrO_3^- \oplus H^+} \tag{3}$$

$$\frac{BrO_3^-\&Br^-\&H^+}{HOBr \oplus HBrO_2} \tag{4}$$

$$\frac{HBrO_2\&Br^-\&H^+}{HOBr} \tag{5}$$

$$\frac{HOBr\&Br^-\&H^+}{Br_2 \oplus H_2O} \tag{5}$$

$$\frac{Br_2\&CH_2(COOH)_2}{Br^- \oplus H^+ \oplus BrCH(COOH)_2} \tag{6}$$

**Figure 2.** Axioms of proof-theoretic simulation of Belousov-Zhabotinsky reaction

The finite word $\alpha_1\alpha_2\ldots\alpha_n$ is a *finite trace of transition system* whenever there is a finite execution fragment of transition system

$$\varrho = p_0\alpha_1p_1\alpha_2\ldots\alpha_np_n \;\text{ such that }\; p_i \longrightarrow^{\alpha_{i+1}} p_{i+1} \;\text{ for all }\; 0 \leq i < n.$$

| | |
|---|---|
| **Choice:** | $\dfrac{P \oplus Q}{P}\ (CL), \qquad \dfrac{P \oplus Q}{Q}\ (CR),$ |
| **Fusion:** | $\dfrac{P}{P\&Q}\ (FL), \qquad \dfrac{Q}{P\&Q}\ (FR), \qquad \dfrac{}{P\&\overline{P}}\ (F),$ |
| **Cooperation:** | $\dfrac{P \mid Q}{P, Q}\ (C),$ |
| **Hiding:** | $\dfrac{P}{P'} \Rightarrow \dfrac{P \backslash X}{P' \backslash X}\ (H).$ |

**Figure 3.** Inference rules of proof-theoretic simulation of Belousov-Zhabotinsky reaction. The variables $P$, $Q$ run over the set $\{Ce^{3+}, HBrO_2, BrO_3^-, H^+, Ce^{4+}, H_2O, BrCH\ (COOH)_2, Br^-, HCOOH, CO_2, HOBr, Br_2, CH_2(COOH)_2\}$

The word $\alpha_1\alpha_2 \ldots \alpha_n$ is denoted by $trace(\varrho)$. The infinite word $\alpha_1\alpha_2 \ldots$ is an infinite trace whenever there is an infinite execution fragment of of transition system

$$\varrho = p_0\alpha_1 p_1\alpha_2 p_2\alpha_3 p_3 \ldots \quad \text{such that} \quad p_i \longrightarrow^{\alpha_{i+1}} p_{i+1} \quad \text{for all} \quad 0 \leq i.$$

The word $\alpha_1\alpha_2 \ldots$ is denoted by $trace(\varrho)$ too. An infinite word $trace(\varrho)$ is *circular* if

$$\varrho = p_0\alpha_1 p_1\alpha_2 p_2 \ldots \alpha_n p_0\alpha_1 p_1\alpha_2 p_2 \ldots \alpha_n p_0\alpha_1 p_1\alpha_2 p_2 \ldots \alpha_n p_0 \ldots$$

such that $p_i \longrightarrow^{\alpha_{i+1}} p_{i+1}$ for all $0 \leq i < n-1$ and $p_{n-1} \longrightarrow^{\alpha_n} p_0$.

**Definition 1**

An infinite (resp. finite) trace of state $p$ denoted by $\varrho(p)$ is the trace of an infinite (resp. finite) execution fragment starting in $p$.

Each trace can be regarded as a graph, where nodes represent states and edges transitions. In this way, transition system is viewed as graph trees. On the other hand, in proof theory we can consider derivations as graph trees too. This allows us to simulate transition system within the framework of proof theory.

By a *derivation tree*, we mean a possibly infinite tree of formulas in which each parent formula is obtained as the conclusion of an inference rule with its children as premises.

**Definition 2**

Let $For$ denote the set of all well-formed formulas in some language and $Rul$ denote some set of rules. Let $n \in \mathbf{N}$ be the maximum number of premises of any $R \in Rul$. Then a derivation graph is given by $\langle V, s, r, \tau \rangle$, where: $V$ is a set of nodes, $s\colon V \mapsto For$, $r$ is a partial mapping from $V$ to $Rul$, and $\tau$ is a partial mapping from $V$ to $V^n$ (we write $\tau_j(v)$ for the $j$-th component of $\tau(v)$); for all $v \in V$, $\tau_j(v)$ is defined just in case $r(v)$ is a rule with $m$ premises, $1 \leq j \leq m$ and:

$$\frac{s(\tau_1(v)) \ldots s(\tau_m(v))}{s(v)}$$

is an instance of rule $r(v)$.

A derivation graph $\mathcal{G}$ consists of the set $V$ of nodes and the set of edges $E = \{\langle v, \tau_j, v' \rangle\colon v, v' \in V, \tau_j(v) = v'\}$. The finite word $\tau_1\tau_2 \ldots \tau_n$ is a *finite trace of derivation* whenever there is a finite execution fragment of transition system

$$\pi = v_0\tau_1 v_1\tau_2 \ldots \tau_n v_n \text{ such that } \tau_{i+1}(v_i) = v_{i+1} \text{ for all } 0 \leq i < n.$$

The word $\tau_1\tau_2 \ldots \tau_n$ is denoted by $trace(\pi)$. The infinite word $\tau_1\tau_2 \ldots$ is an *infinite trace of derivation* whenever there is an infinite execution fragment of of transition system

$$\pi = v_0\tau_1 v_1\tau_2 \ldots \text{ such that } \tau_{i+1}(v_i) = v_{i+1} \text{ for all } 0 \leq i.$$

The word $\tau_1\tau_2 \ldots$ is denoted by $trace(\pi)$ too. An infinite word $trace(\pi)$ is *circular* if

$$\pi = v_0\tau_1 v_1\tau_2 \ldots \tau_n v_0\tau_1 v_1\tau_2 \ldots \tau_n v_0\tau_1 v_1\tau_2 \ldots \tau_n v_0 \ldots$$

such that $\tau_{i+1}(v_i) = v_{i+1}$ for all $0 \leq i < n-1$ and $\tau_n(v_{n-1}) = v_0$. For more details about circular proofs see [9], [10], [11], [20].

**Definition 3**

A derivation graph $\mathcal{G}$ is a derivation tree if there is a distinguished node $v_0 \in V$ such that for all $v \in V$, there is a unique trace in $\mathcal{G}$ from $v_0$ to $v$. Then $v_0$ is called the root of the tree. It is denoted by $\pi(v_0)$.

Using the bisimulation principle [14], [16], [19], we can identify a transition system with a proof system, namely we can show that there exist

$$\frac{Ce^{3+}\&HBrO_2\&BrO_3^-\&H^+ \dagger}{Ce^{4+} \oplus HBrO_2 \oplus H_2O}\ Axiom\ (1),$$

$$\frac{Ce^{4+} \oplus HBrO_2 \oplus H_2O}{Ce^{4+}}\ Inf.rule\ 2 \times (CL),$$

$$\frac{Ce^{4+}}{Ce^{4+}\&BrCH(COOH)_2\&H_2O}\ Inf.rule\ 2 \times (FL),$$

$$\frac{Ce^{4+}\&BrCH(COOH)_2\&H_2O}{Br^- \oplus Ce^{3+} \oplus HCOOH \oplus CO_2 \oplus H^+}\ Axiom\ (2),$$

$$\frac{Br^- \oplus Ce^{3+} \oplus HCOOH \oplus CO_2 \oplus H^+}{Ce^{3+}}\ Inf.rules\ (CR), 3 \times (LR),$$

$$\frac{Ce^{3+}}{Ce^{3+}\&HBrO_2\&BrO_3^-\&H^+ \dagger}\ Inf.rule\ 3 \times (FL),$$

**Figure 4.** The derivation tree of proof-theoretic simulation of the Belousov-Zhabotinsky circular feedback $Ce^{3+} \longrightarrow Ce^{4+} \longrightarrow Ce^{3+} \longrightarrow \ldots$. We see that the pictured derivation tree has a *cycle*. This cycle presents the circular feedback in Belousov-Zhabotinsky reaction (more precisely temporal oscillations in a well-stirred solution): cerium(III) is colorless and cerium(IV) is yellow. Under some conditions, this cycle will repeat several times: in the beginning the solution is colorless, later it is yellow, later it is colorless, etc.

a transition system and a proof system such that for any $\varrho$ and $\pi$ (resp. $p$ and $v_0$) $trace(\varrho)$ and $trace(\pi)$ (resp. $\varrho(p)$ and $\pi(v_0)$) are bisimilar.

Let us return to process calculus of reaction-diffusion computing. Due to the bisimulation between a transition system and a proof system, we can set up proof-theoretic frameworks of this process calculus. For example, we can define the proof-theoretic simulation of Belousov-Zhabotinsky reaction as follows:

**Definition 4**

Let $L = \{Ce^{3+}, HBrO_2, BrO_3^-, H^+, Ce^{4+}, H_2O, BrCH\ (COOH)_2, Br^-, HCOOH, CO_2, HOBr, Br_2, CH_2(COOH)_2\}$ be the set of states. The set of initial transitions $\Sigma$ is defined by axioms in Fig. 2. The four basic operations $\oplus$ (choice), & (fusion), | (parallel), $\setminus$ (hiding) are defined by inference rules in Fig. 3, which describe general properties of transitions. Then the proof-theoretic simulation of Belousov-Zhabotinsky reaction is a tuple $\langle L, \Sigma, \oplus, \&, |, \setminus \rangle$.

In this system, transitions between states are identified with derivations of states. The example is given in Fig. 4. Each step of derivation means a transition. As a result, the circular trace of state $Ce^{3+}$ (resp. $Ce^{4+}$) has a meaning of circular proof, where the state $Ce^{3+}$ (resp. $Ce^{4+}$) is unfolded infinitely often among premisses and at the same time among derivable expressions.

## 4. Conclusion

Self-organization phenomena in nature assume circularity and cause-and-effect feedback relations: each component affects the other components, but these components in turn affect the first component. The most popular example of such self-organization is presented by Belousov-Zhabotinsky reaction. In this system we observe circularity in the interchange of solution color: in the beginning the solution is colorless, then it becomes yellow, then it becomes colorless, etc. *In logical simulation of Belousov-Zhabotinsky reaction we obtain circular proofs. This shows that reaction-diffusion computing cannot dispense with logical circularity like cyclic proofs and feedback relations in state transitions.* We can suppose that logical circularity should be a key notion of "life computer", i.e. of each self-organized reaction-diffusion system.

## References

[1] Adamatzky A. Reaction-diffusion algorithm for constructing discrete generalized Voronoi diagram. Neural Network World 6, 1994, 635–643.

[2] Adamatzky A. Computing in Nonlinear Media and Automata Collectives. Institute of Physics Publishing, 2001.

[3] Adamatzky A., De Lacy Costello B., Asai T. Reaction-Diffusion Computers, Elsevier, 2005.

[4] Adamatzky A., A. Wuensche, and B. De Lacy Costello, Glider-based computation in reaction-diffusion hexagonal cellular automata, Chaos, Solitons & Fractals 27, 2006, 287–295.

[5] Adamatzky A. and Wuensche A. Computing in spiral rule reaction-diffusion hexagonal cellular automaton. Complex Systems 16, 2007.

[6] Berry G., Boudol G. The chemical abstract machine, Teor. Comput. Sci., 96, 1992, 217–248.

[7] Belousov V. Synergetics and biological morphogenesis, Self-organization: Autowaves and Structures Far From Equilibrium. Ed. V. I. Krinsky. Heidelberg-Springer, 1984, 204–208.

[8] Boerlijst M. C., Lamers M. E., Hogeweg P. Evolutionary consequences of spiral waves in a horst-parasitoid system, Proc. R. Soc., B 253, 1993, 15–18.

[9] Brotherston, J., Cyclic proofs for first-order logic with inductive definitions [in:] B. Beckert, editor, *TABLEAUX 2005*, volume 3702 of LNAI, Springer-Verlag, 2005, 78–92.

[10] Brotherston, J., *Sequent Calculus Proof Systems for Inductive Definitions*. PhD thesis, University of Edinburgh, November 2006.

[11] Brotherston, J., Simpson, A., Complete sequent calculi for induction and infinite descent. *LICS-22*, IEEE Computer Society, July 2007, 51–60.

[12] Drodland G. W., Clausi D. A. Embryonic tissue morphogenesis modeled by FEM, Journal Biomech. Eng., 116, 1994, 146–155.

[13] Feynman R. P. Simulating physics with computers, Int. J. of Theor. Phys., 21, 1982.

[14] Fiore, M.P., A coinduction principle for recursive data types based on bisimulation. *Proc. 8th Conf. Logic in Computer Science (LICS'93)*. 1993, 110–119.

[15] Hafner G. S., Tokarski T. R. Morphogenesis and pattern formation in the retina of the crayfish, Procambarus clarkii Cell and Tissue Res., 293, 1998, 535–550.

[16] Howe, D. J., Proving congruence of bisimulation in functional programming languages, *Inf. and Comp.* 124, 1996.

[17] Ivanitsky G. R., Kunisky A. S., Tzyganov M. A. Study of 'target patterns' in a phage-bacterium system, Self-organization: Autowaves and Structures Far From Equilibrium. Ed. V. I. Krinsky. Heidelberg-Springer, 1984, 214–217.

[18] Prajer M., Fleury A., Laurent M. Dynamics of calcium regulation in Paramecium and possible morphogenetic implication, Journal Cell Sci., 110, 1997, 529–535.

[19] Rutten, J. J. M. M., Processes as terms: non-well-founded models for bisimulation. *Math. Struct. in Comp. Sci.*, 2 (3), 1992, 257–275.

[20] Santocanale L., A calculus of circular proofs and its categorical semantics [in:] M. Nielsen and U. Engberg, editors, *Proc. of FoSSaCS 2002*, Grenoble, Apr. 2002, Springer-Verlag LNCS 2303, 357–371.

[21] Vink E. P. de, and J. J. M. M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theor. Comp. Sci.*, 221, 1999, 271–293.

Andrew Schumann
Department of Philosophy and Science Methodology
Belarusian State University
Minsk, Belarus
Andrew.Schumann@gmail.com