Physarum Machines. Selected Works

Andrew Schumann, Krzysztof Pancerz

UNIVERSITY of INFORMATION
TECHNOLOGY and MANAGEMENT
in Rzeszow, POLAND

Physarum Machines. Selected Works

Andrew Schumann, Krzysztof Pancerz

Reviewers:
prof. Andrew Adamatzky (Bristol, UK)
dr Andrzej Szelc (Rzeszow, Poland)

Rzeszow – Szczecin 2016

The publication has been financed by the
University of Information Technology and Management in Rzeszow

Title: Physarum Machines. Selected Works

Authors:
Andrew Schumann, Krzysztof Pancerz

Reviewers:
prof. Andrew Adamatzky (Bristol, UK)
dr Andrzej Szelc (Rzeszow, Poland)

Publishers:
University of Information Technology and Management in Rzeszow
St. Sucharskiego 2, 35-225 Rzeszów, Poland
www.wsiz.rzeszow.pl, www.ksiegarnia.wsiz.pl
email: wsiz@wsiz.rzeszow.pl

Scientific Publishing House IVG
St. Cyfrowa 6, 441 Szczecin, Poland
www.wydawnictwoivg.pl
email: biuro@wydawnictwoivg.pl

ISBN 978-83-62062-74-4 eBook
ISBN 978-83-64286-59-9 eBook

Printed in Poland by Scientific Publishing House IVG

UNIVERSITY of INFORMATION TECHNOLOGY and MANAGEMENT in Rzeszow

Scientific Publishing House IVG

Table of contents

Preface

The *Physarum polycephalum* slime mould is a large single cell capable for distributed sensing, concurrent information processing, parallel computation, and decentralized actuation. The ease of culturing and experimenting with *Physarum polycephalum* makes this slime mould an ideal substrate for real-world implementations of unconventional sensing and computing devices. The project *Physarum Chip: Growing Computers from Slime Mould* (its leader is prof. Andrew Adamatzky, the research is supported by the Seventh Framework Programme FP7-ICT-2011-8) focused on theoretical and experimental laboratory studies on sensing and computing properties of slime mould, and development of mathematical and logical theories of *Physarum* behaviour. The project resulted in the design and prototyping of a *Physarum* Chip.

This book contains some selected works prepared by us within the *Physarum* Chip project.

We would like to express our warm gratitude to all our collaborators in the *Physarum* Chip project, but, first of all, to Andrew Adamatzky, Ludmila Akimova and Jeff Dale Jones with whom the collaboration was really significant for us. This book includes some papers written jointly with them.

Andrew Schumann,
Krzysztof Pancerz

Introduction

Physarum polycephalum belongs to the species of order *Physarales*, subclass *Myxogastromycetidae*, class *Myxomycetes*, division *Myxostelida*. Plasmodium is a 'vegetative' phase of *P. polycephalum*, it is a single cell with a myriad of diploid nuclei. It behaves and moves as a giant amoeba. Typically, the plasmodium forms a network of protoplasmic tubes connecting the masses of protoplasm at the food sources which has been shown to be efficient in terms of network length and resilience.

A *Physarum* Chip is a biological sensing and computing device implemented in vegetative stage of acellular slime mould *Physarum polycephalum*. The *Physarum* Chip is programmed by spatio-temporal configurations of repelling and attracting gradients. There are several classes of *Physarum* Chips: morphological processors, sensing devices, frequency-based, bio-molecular and microfluidic logical circuits, and electronic devices.

In the project, we have proposed several *logical methods* for *designing the Physarum Chip*:

p-Adic valued gates

The slime mould is considered a natural fuzzy processor with fuzzy values on the set of p-adic integers. The point is that in any experiment with the slime mould we deal with attractants which can be placed differently to obtain different topologies and to induce different transitions of the slime mould. If the set A of attractants, involved into the experiment, has the cardinality number $p - 1$, then any subset of A can be regarded as a condition for the experiment such as "Attractants occupied by the plasmodium". These conditions change during the time, $t = 0, 1, 2, \ldots$, and for the infinite time, we obtain p-adic integers as values of fuzzy (probability) measures defined on conditions (properties) of the experiment. This space is a semantics for p-adic valued fuzzy syllogistic we constructed for describing the propagation of the slime mould. Also, we propose an implementation of finite part of p-adic arithmetic by means of p-adic valued logic gates on the slime mould. So, each experiment with *Physarum* may be understood as a p-adic valued fuzzy controller on *Physarum* with calculations over p-adic integers.

Syllogistic systems

To make computations on the plasmodium tree more expressive we propose a syllogistic system of propagation. This system can logically simulate a massive-parallel behaviour in the propagation of any swarm. In particular, this system simulates the behaviour of collectives of Trematode larvae (miracidia and cercariae). Also, this syllogistic system of propagation can be used as basic logical theory for quantum logic (without logical atoms). In this theory we can build non-well-founded trees for which there cannot be logical atoms. This theory is much more expressive than standard spatial algorithms in simulating the plasmodium motions. We define some unconventional algorithms on non-well-founded trees to make calculations on plasmodia more effective. These algorithms are implemented on plasmodia by means of reversible logic gates.

Kolmogorov-Uspensky machine

By implementing the double-slit experiment in *Physarum*, we showed that conventional approaches to computation such as Kolmogorov-Uspensky machines can be implemented only in a form of approximation, because we cannot extract individual actions from the slime mould's behaviour. Therefore we constructed the syllogistic that models *Physarum* simultaneous propagations in all directions (i.e. it is massive-parallel), the cellular-automatic game theory that describes reflexive and context-based games, where players are presented by different localization of plasmodia, the unconventional logic combining different games into one complex game by their different logical superpositions.

Syllogistic interpretation of storage modification machine

We constructed syllogistic versions of storage modification machine in *Physarum*. We implemented two syllogistics in the biological behaviour of *Physarum*: the Aristotelian syllogistic and a non-Aristotelian syllogistic. While Aristotelian syllogistic may describe concrete directions of *Physarum* spatial expansions, non-Aristotelian syllogistic may describe *Physarum* simultaneous propagations in all directions.

Non-well-founded calculi

We constructed an unconventional logic, which deals with non-well-founded data, namely with infinite streams and wave sets (sets of mutually defined infinite streams). Within this logic we can logically combine cellular automata, where Boolean functions are regarded as local transition rules which can change at in time. This symbolic logic can simulate reflexive games of different levels and can simulate the *Physarum* propagations with much localization of plasmodia.

Computing with storage modification machines

We eliminated notions of apriority (axioms, theorems) in symbolic logic to show how we can build up formal theories using only notions of aposteriority. The new kind of logic is said to be anti-Platonic. It satisfies basic presuppositions of unconventional computing. Due to the double-slit experiment with *Physarum*, we know that we cannot obtain a partition of subsets of *Physarum* actions, therefore we appeal to the new probability theory, where there cannot be a partition in an algebra of subsets, too. This theory has probability measures defined on infinite streams or p-adic numbers.

Petri-nets based graphical language

We constructed an abstract graphical language in the form of Petri nets to describe *Physarum*. Such a tool for modelling behaviour of *Physarum* has several advantages. Petri nets are a graphical tool with well-defined elements which can be easily implemented in a new object-oriented programming language defined by us. Petri nets are also a powerful tool for modelling concurrent systems. Moreover, Petri nets with inhibitor arcs can be used to model semantics of other types of computations, for example, pi-calculus or process algebra.

Process algebra

To formalize *Physarum* computing we appeal first to some timed and probabilistic extensions of standard process algebra to implement timed and rough set models of behaviour of *Physarum* machines in our new object-oriented programming language. In this language we define Bayesian networks on rough sets. Computations on the basis of rough set models enable us to take into consideration some ambiguities in *Physarum* motions. We propose to use rough set models created over transition systems. Second, we define an extension of process algebra, where simple actions of labelled transition systems cannot be atomic; consequently, their compositions cannot be inductive. Their informal meaning is that in one simple action we can suppose the maximum of its modifications. Such actions are called hybrid. Then we propose two formal theories on hybrid actions (the hybrid actions are defined there as non-well-founded terms and non-well-founded formulas): group theory and Boolean algebra.

Theories on hybrid actions

The group theory proposed by us can be used as the new design method to construct reversible logic gates on plasmodia. In this way, we should appeal to the so-called non-linear permutation groups. These groups contain non-well-founded objects such as infinite streams and their families. The theory of non-linear permutation groups proposed by us can be used for designing reversible logic gates on any behavioural systems. The simple versions of these gates are represented by logic circuits constructed on the basis of the performative syllogistic. It seems to be natural for behavioural systems and these circuits have very high accuracy in implementing. Our general motivation in designing logic circuits in behavioural systems without repellents is as follows: in this way, we can present behavioural systems as a calculation process more naturally; we can design devices, where there are much more outputs than inputs for performing massive-parallel computations in the bio-inspired way; we can obtain unconventional (co)algorithms by programming behavioural systems. Computations on protoplasmic tree are understood as a kind of extension of concurrent processes defined in concurrent games. This extension is called context-based processes and they are defined in the theory of context-based games proposed by us.

We have designed a *programming language* for the *storage modification machines on plasmodia*:

Physarum programming language

We obtained a basis of new object-oriented programming language for *Physarum* computing. Within this language we can check possibilities of practical implementations of storage modification machines on plasmodia and their applications to behavioural science such as behavioural economics and game theory. The proposed language can be used for developing programs for *Physarum* by the spatial configuration of stationary nodes. Geometrical distribution of stimuli can be identified with a low-level programming language for *Physarum* machines. Our programming language uses the prototype-based approach called also the class-less or instance-based approach. There are inbuilt sets of prototypes corresponding to both the high-level models used for describing behaviour of *Physarum* (e.g., ladder diagrams, transition systems, Petri nets) and the low-level model (distribution of stimuli). According to the prototype-based approach, objects representing individuals can be created without reference to class-defining via the “new” keyword using defined constructors. Methods are used to manipulate features of the objects and create relationships between objects. The objects can be manipulated at runtime.

Symbolic transformations

We examined the potential to transform symbolic representations of logic circuits to the spatial computing domain. By utilising transformations between ladder diagrams and Petri nets we were able to automatically generate spatial representations of a de-multiplexer circuit which was successfully implemented and assessed on a model of *Physarum*.

Decision making

We analysed biological mechanisms for *Physarum* decision making and selected the mechanisms (e.g. chemo-attraction, self-avoidance, competition between different species of moulds) as basis for design of novel computing devices using *Physarum*.

Strategies

Within the unconventional logic, we introduced the notion of payoff cellular automata instead of payoff matrices. By using these automata we can formalize context-based reflexive games for *k* players on different finite or infinite levels of reflexion. We defined games as interactions among rational players, where decisions impact the payoffs of others, but players are limited by contexts that permanently change. A game is described by (i) its players who are presented in appropriate *transition rules of cellular automata*; (ii) players’ possible strategies which are supposed known before the game and a combination of all possible payoffs from each strategy outcome gives the resulting payoffs which are collected as a *set of states of cellular automata*; (iii) a *neighbourhood of cellular automata* that makes some strategies actual and others non-actual (i.e. accepts the most important strategies in the given context at time *t*) and also changes or correct strategies. So, in this form of game description, players analyze strategies not purely logically, but contextually. Therefore players take decisions not only in an environment given by the payoff that corresponds to each possible outcome, but also in an environment of different other circumstances, e.g. by defining: which strategies can be accepted in this context, how they can be changed by the given context, how past contexts have influenced present contexts, whether some public announcements are false in fact, etc.

Probabilistic behaviour

In the stream-valued or p-adic valued probability theory, we can disprove the Aumann's agreement theorem (the so-called reflexion disagreement theorem). This result can be obtained if (1) we assume that rational agents can become unpredictable for each other and try to manipulate; (2) we define probabilities on streams (e.g. on hypernumbers or p-adic numbers); (3) games are presented as coalgebras. This new theorem is an important statement within the new mathematics (coalgebras, transition systems, process calculi, etc.) which has been involved into game theory recently. Instead of the agreement theorem, the reflexion disagreement theorem is valid if we cannot obtain inductive sets, e.g. in case of sets of streams. The reflexion disagreement theorem opens the door for new mathematics in game theory and decision theory; in particular it shows that it has sense to use stream calculus, non-Archimedean mathematics, and p-adic analysis there. Within this mathematics we can formalize reflexive games of different reflexive levels (up to the the infinite reflexive level). And these results we use for formalizing the game theory of plasmodia.

Zero-sum games in plasmodia

We have designed the zero-sum game between plasmodia of *Physarum polycephalum* and *Badhamia utricularis*, the so-called PHY game. To simulate *Physarum* games, we have created a specialized software tool. It is a part of the larger platform developed, using the Java environment, under the support of the PhyChip project. The tool works under the client-server paradigm. The server window contains:

- a text area with information about actions undertaken,
- a combox for selecting one of two defined situations,
- start and stop server buttons.

Communication between clients and the server is realized through text messages containing statements of a new object-oriented programming language, called the *Physarum* language, created by us for *Physarum polycephalum* computing. The locations of attractants and repellents are determined by the players during the game. At the beginning, origin points of *Physarum polycephalum* and *Badhamia utricularis* are scattered randomly on the plane. During the game, players can place stimuli. New veins of plasmodia are created. The server sends to clients information about the current configuration of the *Physarum* machine (localization of origin points of *Physarum polycephalum* and *Badhamia utricularis*, localization of stimuli as well as a list of edges, corresponding to veins of plasmodia, between active points).

So, we have proposed a game-theoretic visualization of morphological dynamics with non-symbolic interfaces between living objects and humans. These non-symbolic interfaces are more general than just sonification and have a game-theoretic form. The user interface for this game is designed on the basis of the following game steps: first, the system of *Physarum* language generates locations of attractants and repellents; second, we can chose *n* plasmodia/agents of *Physarum polycephalum* and *m* plasmodia/agents of *Badhamia utricularis*; third, we obtain the task, for example to reach as many as possible attractants or to construct the longest path consisting of occupied attractants, etc.; fourth, we can choose initial points for *Physarum polycephalum* transitions and initial points for *Badhamia utricularis* transitions; fifth, we start to move step by step; sixth, we define who wins, either *Physarum polycephalum* or *Badhamia utricularis*.

We assume that bio-inspired games wake new interests in designing new games and new game platforms.

For more details please see http://www.phychip.eu/.

Andrew Schumann,
Krzysztof Pancerz

Originally published: Schumann, A., *Anti-Platonic Logic and Mathematics*, Journal of Multiple -Valued Logic and Soft Computing 21(1-2), pp. 53–88, 2013.

I
Anti-Platonic Logic and Mathematics

Andrew Schumann

Abstract

In this paper, I show that we can absolutely eliminate notions of apriority (axioms, theorems) in symbolic logic and mathematics and build up formal theories using only notions of aposteriority. The new kind of logic and mathematics is said to be anti-Platonic. Both satisfy basic presuppositions of unconventional computing.

1.1. Platonism in logic and mathematics

According to Platonism, mathematical entities, which we obtain from primary terms and axioms, are eternal and unchanging. This means that, on the one hand, they exist in fact and, on the other hand, they are abstract and have supranatural properties instead of spatiotemporal or causal ones. Plato says about as follows:

> The sphere of the intelligible will also have two divisions, — one of mathematics, in which there is no ascent but all is descent; no inquiring into premises, but only drawing of inferences. In this division the mind works with figures and numbers, the images of which are taken not from the shadows, but from the objects, although the truth of them is seen only with the mind's eye; and they are used as hypotheses without being analysed. Whereas in the other division reason uses the hypotheses as stages or steps in the ascent to the idea of good, to which she fastens them, and then again descends, walking firmly in the region of ideas, and of ideas only, in her ascent as well as descent, and finally resting in them. 'I partly understand', he replied; 'you mean that the ideas of science are superior to the hypothetical, metaphorical conceptions of geometry and the other arts or sciences, whichever is to be the name of them; and the latter conceptions you refuse to make subjects of pure intellect, because they have no first principle, although when resting on a first principle, they pass into the higher sphere.' You understand me very well, I said. And now to those four divisions of knowledge you may assign four corresponding faculties — pure intelligence to the highest sphere; active intelligence to the second; to the third, faith; to the fourth, the perception of shadows — and the clearness of the several faculties will be in the same ratio as the truth of the objects to which they are related... (*The Republic* (Jowell translation). Book VII).

The main question of mathematical ontology and epistemology, whether mathematics is discovered or invented, was answered by Plato and Platonists rigorously. They claimed that mathematical entities exist indeed as abstract eternal objects that were discovered, not invented. According to this claim, the mathematical realm is considered as plenitudinous. Such an approach is regarded by Marc Balaguer as full-blooded Platonism. In his words, the pure Platonistic point of view is expressed as follows: *"all the mathematical objects (logically possibly) could exist actually do exist"* (Balaguer). Formally:

$$\forall x\ [(x \text{ is a mathematical object } \&\ x \text{ is logically possible}) \Rightarrow x \text{ exists}]$$

This approach was fully embodied in modern logic and mathematics. Since David Hilbert, logicians and mathematicians have been sure that if a mathematical entity is correctly defined within axiomatic method, then it really exists as eternal object:

> An example of this way of setting up a theory can be found in Hilbert's axiomatization of geometry. If we compare Hilbert's axiom system to Euclid's, ignoring the fact that the Greek geometer fails to include certain [necessary] postulates, we notice that Euclid speaks of figures to be constructed, whereas, for Hilbert, system of points, straight lines, and planes exist from the outset. Euclid postulates: One can join two points by a straight line; Hilbert states the axiom: Given any two points, there exists a straight line on which both are situated. "Exists" refers here to existence in the system of straight lines (Bernays).

The world of eternal objects that has been assumed in modern logic and mathematics was considered by Gottlob Frege as *das dritte Reich* (the third world) that exists independently on the physical world (the first world) and our subjective impressions, fictions and imaginations (the second world). *Das dritte Reich* contains all interpersonal senses (Sinne):

> Die Gedanken sind weder Dinge der Außenwelt noch Vorstellungen. Ein drittes Reich muß anerkannt werden.
> The thoughts neither are things of physical world nor impressions. A third world should be accepted (Frege 1918/19).

However, we could prove that modern logic and mathematics may be established without *das dritte Reich*, i.e. without axiomatic method and Platonic presuppositions. In the beginning, let us show, where exactly there are Platonic assumptions in modern logic and mathematics in order to avoid them.

1.2. Classical logic

Let us take a set consisting of k_1 one-placed predicates, k_2 two-placed predicates, k_n n-placed predicates, P_{11}, P_{12}, …, P_{1k1}, P_{21}, …, P_{2k2}, …, P_{nkn}, i.e. P_{ij} means a j-th i-placed predicate. Let x, y, z, … are individual variables, i.e. their values are individuals (specific names). A proposition of the form P_{ij} $(x_1, x_2, …, x_i)$ is said to be *atomic*. Take now logical connectives ε (' … is …'), & (conjunction), ∨ (disjunction), ⇒, (implication), ¬ (negation). Then *formulas* are defined as follows:

- Each atomic formula is a formula.

- If P_{il}, P_{im} are atomic formulas, then $P_{il}(x_1, x_2, \ldots, x_i)$ ε $P_{im}(x_1, x_2, \ldots, x_i)$ is a formula (e.g. let H be a predicate to be a human being and M to be mortal. Both predicates are one-placed. Then the formula H (x) ε M (x) has the meaning: "The human being is mortal").
- If Φ_1, Φ_2 are formulas, then $\Phi_1 \,\&\, \Phi_2$, $\Phi_1 \vee \Phi_2$, $\Phi_1 \Rightarrow \Phi_2$, $\neg\Phi_1$, $\neg\Phi_2$ are formulas.

A formula of the form $P_{il}(x_1, x_2, \ldots, x_i)$ ε $P_{im}(x_1, x_2, \ldots, x_i)$ is called *quasi-atomic*.

Let us take an algebraic system M consisting of k_1 one-placed relations, k_2 two-placed relations, k_n n-placed relations, R_{11}, R_{12}, …, R_{1k1}, R_{21}, …, R_{2k2}, …, R_{nkn}, i.e. R_{ij} means j-th i-placed relations. We assume that these relations are defined on a set A. Then we can assign values on the set A to individual variables x, y, z, … by means of an interpretation I. In other words, $I(x) \in A$. Let Γ be the set of propositions of our formal language. Taking into account that for each predicate P_{ij} there is a relation R_{ij} of algebraic system M, we can consider M as a model of the set Γ, it is denoted by $M \models \Gamma$, if the following conditions are satisfied for all formulas Φ_1, Φ_2 of Γ:

$I(P_{ij}(x_1, x_2, \ldots, x_i)) = R_{ij}(I(x_1), I(x_2), \ldots, I(x_i)) \in A^i$, i.e. $M \models R_{ij}(I(x_1), I(x_2), \ldots, I(x_i))$;

$M \models \neg\,\Phi_1$ iff it is false that $M \models \Phi_1$;

$M \models \Phi_1 \,\&\, \Phi_2$ iff $M \models \Phi_1$ and $M \models \Phi_2$;

$M \models \Phi_1 \vee \Phi_2$ iff $M \models \Phi_1$ or $M \models \Phi_2$;

$M \models \Phi_1 \Rightarrow \Phi_2$ iff from $M \models \Phi_1$ it follows that $M \models \Phi_2$;

$M \models P_{il}(x_1, x_2, \ldots, x_i)$ ε $P_{im}(x_1, x_2, \ldots, x_i)$ iff $M \models P_{il}(x_1, x_2, \ldots, x_i)$ and $M \models P_{il}(x_1, x_2, \ldots, x_i) \Rightarrow P_{im}(x_1, x_2, \ldots, x_i)$.

A *proof of formula* Φ is a sequence of formulas $\Phi_1, \ldots, \Phi_n$ such that $\Phi_n = \Phi$ and for every i ($i \le n$) the formula Φ satisfies one of the following conditions:

- Φ_i is an axiom;
- Φ_i is entailed from Φ_j $\{i > j\}$ by one of the inference rules.

If for the formula Φ there is its proof, then Φ is called a *theorem* or *provable formula*. In this case we write $\vdash \Phi$.

A *deduction of formula* Φ from the set of formulas Γ is a sequence of formulas $\Phi_1, \ldots, \Phi_n$ such that $\Phi_n = \Phi$ and for every i ($i \le n$) the formula Φ satisfies one of the following conditions:

- Φ_i is provable;
- Φ_i belongs to the set of formulas Γ;
- Φ_i is entailed from Φ_j $\{i > j\}$ by one of the inference rules.

If there is a deduction of formula Φ from the set Γ, then the formula Φ is called *deducible* from Γ, and the set of formulas Γ is called the *set of hypotheses.* It is obvious that the provability of formula is equivalent to its deducibility from the empty set of hypotheses. A deducibility of formula Φ from the set Γ is denoted by $\Gamma \vdash \Phi$.

If Γ is a set of propositions, then by Th(Γ) we denote the set of all propositions deducible from Γ, i.e. $\text{Th}(\Gamma) = \{\Phi : \Gamma \vdash \Phi\}$. Instead of $\Gamma \vdash \Phi$ it is possible to write $\Phi \in \text{Th}(\Gamma)$. In particular, if Φ is a theorem and Φ is a proposition, then $\Phi \in \text{Th}(\varnothing)$. The set $\text{Th}(\varnothing)$ consists of axioms and theorems of the first-order logic.

The set Γ of propositions such that $\Gamma = \text{Th}(\Gamma)$ is called the *elementary theory*. Members of Γ are called *theorems of theory* Γ.

If Γ is an elementary theory, $A \subseteq \Gamma$ and $\Gamma = \text{Th}(\Gamma)$, then the set A is called the *set of axioms of theory* Γ or simply the *axiomatics* for Γ, and its members are *axioms.* In particular, Γ is an axiomatics for Γ.

If Γ_1 and Γ_2 are elementary theories and $\Gamma_1 \subseteq \Gamma_2$, then Γ_2 is called the *overtheory* (*extension* of theory Γ_1) and Γ_1 the *subtheory* (*restriction*) of theory Γ_2. Theories Γ_1 and Γ_2 are called *equivalent (deductively equal)* if everyone of them is an extension for another.

Let us get a class of algebraic systems, K, consisting of the same set of relations $R_{11}, R_{12}, \ldots, R_{1k1}, R_{21}, \ldots, R_{2k2}, \ldots, R_{nkn}$. The proposition Φ is called:

- *true on a class* K if Φ is true on each system of a class K;
- *realizable on a class* K if in K there is a system in which Φ is true.

Clearly, valid propositions (propositions from $\text{Th}(\varnothing)$) are true on any class of algebraic systems. Also, if a proposition Φ is true on any class of algebraic systems, then the proposition Φ is valid.

Let Γ be a collection of propositions. A class of all algebraic systems such that on each of them all propositions from Γ are true is denoted by Mod (Γ). Evidently, if $\Gamma_1 \subseteq \Gamma_2$, then Mod ($\Gamma_1$) $\supseteq$ Mod (Γ_2).

Let K be an arbitrary class of algebraic systems. The collection of all propositions that are true on the class K is an elementary theory and it is called the *elementary theory of class* K. The elementary theory of class K is denoted by Th(K). Let us notice that if $K_1 \subseteq K_2$, then $\text{Th}(K_1) \supseteq \text{Th}(K_2)$. Obviously also that $K \subseteq \text{Mod}(\text{Th}(K))$. If M is an algebraic system, then Th({M}) is called the *elementary theory of system* M and it is denoted by Th(M). The class K of systems is called axiomatized if K = Mod(Th(K)), i.e. the class K of algebraic systems is axiomatized if and only if there exists an elementary theory Th(K) such that K is *a set of all* models for the set of propositions Th(M).

In the *Critique of Pure Reason*, Kant defines the following four kinds of quasi-atomic propositions:

Analytic proposition: a quasi-atomic proposition $P_{il}(x_1, x_2, \ldots, x_i)\ \varepsilon\ P_{im}(x_1, x_2, \ldots, x_i)$ whose predicate concept (i.e. $P_{im}(x_1, x_2, \ldots, x_i)$) is a genus for its subject concept (i.e. for $P_{il}(x_1, x_2, \ldots, x_i)$) is analytic, i.e. it is analytic if the formula $P_{il}(x_1, x_2, \ldots, x_i) \Rightarrow P_{im}(x_1, x_2, \ldots, x_i)$ is true in any class K that contains $R_{il}(I(x_1), I(x_2), \ldots, I(x_i))$, $R_{im}(I(x_1), I(x_2), \ldots, I(x_i))$.

Kant’s example of analytic propositions/judgements is based on the presupposition that individuals of algebraic models are presented by real things. Now let us take the predicate, H, to be a human being and the predicate, R, to be rational. Then the quasi-atomic formula $H(x)\ \varepsilon\ R(x)$ is analytic, becouse $H(I(x)) \Rightarrow R(I(x))$ holds everywhere.

Synthetic proposition: a quasi-atomic proposition $P_{il}(x_1, x_2, \ldots, x_i)\ \varepsilon\ P_{im}(x_1, x_2, \ldots, x_i)$ whose predicate concept is not contained in its subject concept in general case is synthetic.

Kant's example is as follows. Let H(x) mean that x is a human being and F(x) mean that x is featherless bipeds. Then $H(I(x)) \Rightarrow F(I(x))$ holds just in some models.

A priori proposition: a quasi-atomic proposition $P_{il}(x_1, x_2, \ldots, x_i)$ ε $P_{im}(x_1, x_2, \ldots, x_i)$ whose justification does not rely upon experience is a priori, i.e. $P_{il}(x_1, x_2, \ldots, x_i)$ ε $P_{im}(x_1, x_2, \ldots, x_i)$ is a priori iff $P_{il}(x_1, x_2, \ldots, x_i) \vdash P_{im}(x_1, x_2, \ldots, x_i)$, namely iff $P_{im}(x_1, x_2, \ldots, x_i)$ is deduceble from $P_{il}(x_1, x_2, \ldots, x_i)$.

A posteriori proposition: a quasi-atomic proposition $P_{il}(x_1, x_2, \ldots, x_i)$ ε $P_{im}(x_1, x_2, \ldots, x_i)$ whose justification does rely upon experience is a posteriori, i.e. $P_{il}(x_1, x_2, \ldots, x_i)$ ε $P_{im}(x_1, x_2, \ldots, x_i)$ is a posteriori iff $P_{im}(x_1, x_2, \ldots, x_i)$ is not deduceble from $P_{il}(x_1, x_2, \ldots, x_i)$.

We can extend these definitions up to the case of all formulas.

Analytic proposition: a proposition Φ is analytic iff it is true in any class K. On the basis of implication 'if $\Gamma_1 \subseteq \Gamma_2$, then $\mathrm{Mod}(\Gamma_1) \supseteq \mathrm{Mod}(\Gamma_2)$' it is possible to claim that the more propositions are contained in a theory, the less it is analytic.

Synthetic proposition: a proposition Φ is synthetic iff it is not true in any class K.

A priori proposition: a proposition Φ is a priori iff $\vdash \Phi$. On the basis of implication 'if $K_1 \subseteq K_2$, then $\mathrm{Th}(K_1) \supseteq \mathrm{Th}(K_2)$' it is possible to claim that the greater the class of algebraic systems K is, the more the theory containing of propositions true on the class K is a priori.

A posteriori proposition: a proposition Φ is a posteriori iff $\Gamma \vdash \Phi$ and $\Gamma \neq \varnothing$.

Kant combines his distinction between analytic and synthetic propositions with the distinction between a priori and a posteriori propositions to explain Platonic presuppositions that there are abstract objects, pure senses. As a result of that combination, he obtaines the following kinds of propositions: (i) analytic a priori; (ii) synthetic a priori, (iii) analytic a posteriori; (iv) synthetic a posteriori. Kant claims that the third type is self-contradictory. Indeed, if a formula Φ is analytic (holds in any class K), then it is a priori (is a theorem). Could we say at the same time that if it is a priori, then it is analytic? In Kant's view, the answer is negative. In modern logic it is too, namely there are theorems that do not hold in any class K. Hence, in order to explicate Platonic assumptions in modern logic we should consider the meaning of analytic a priori and synthetic a priori propositions more in details and explain, why there are synthetic a priori propositions.

The family of propositions $\mathrm{Th}(\varnothing)$ is analytic, because it holds in any class K. This family is included in any theory Γ: $\forall\Gamma. (\Gamma = \mathrm{Th}(\Gamma) \Rightarrow \mathrm{Th}(\varnothing) \subseteq \Gamma)$. It means that this set is a priori. This $\mathrm{Th}(\varnothing)$ is called the set of logical axioms. A theory Γ without the propositions of $\mathrm{Th}(\varnothing)$ (formally: $\Gamma / \mathrm{Th}(\varnothing)$, where $\Gamma = \mathrm{Th}(\Gamma)$) is called the set of non-logical axioms. It is denoted by NonL(Γ). The propositions of NonL(Γ) are not analytic, but they are a priori.

Thus, Kant's distinction between analytic a priori and synthetic a priori propositions could be explicated as follows: (i) analytic a priori propositions are logical axioms of any formal theory (i.e. belong to the set $\mathrm{Th}(\varnothing)$) and (ii) synthetic a priori propositions are non-logical axioms of an apropriate theory (i.e. they belong to NonL(Γ)).

Now we could find out, where *das dritte Reich* (the world of pure senses) in modern logic has been hidden. In order to ground pure senses (Sinne), Gottlob Frege provides the distinction between sense (Sinn) and reference (*Bedeutung*). He takes into account that propositions $a = a$ and $a = b$ are obviously of differing cognitive value. Propositions of the form $a = a$ are to be labeled analytic a priori, while propositions of the form $a = b$ are synthetic.

The first kind of propositions has the same sense, the second kind of propositions expresses two different senses (‘*a*’ and ‘*b*’), but have the same reference that makes them true:

> When we found ‘$a = a$’ and ‘$a = b$’ to have different cognitive values, the explanation is that for the purpose of knowledge, the sense of the sentence, viz., the thought expressed by it, is no less relevant than its reference, i.e. its truth value. If now $a = b$, then indeed the reference of ‘*b*’ is the same ‘*a*’, and thereby the sense expressed in ‘$a = b$’ differs from that of ‘$a = a$’. In that case the two sentences do not have the same cognitive value. If we understand by 'judgment' the advance from the thought to its truth value, as in the above paper, we can also say that the judgments are different (Frege 1892).

In the *Meaning and Necessity* Rudolf Carnap explicates the two kinds of meaning proposed by Frege as follows:

Extension: it is a corresponding class of references (denotations, *Bedeutungen*) for a predicate or proposition. Let Γ be a family of propositions and M be a model for Γ, i.e. $M \in Mod(\Gamma)$. Then two propositions Φ_1, Φ_2 of Γ have the same extension iff $M \models (\Phi_1 \Rightarrow \Phi_2)\ \&\ (\Phi_2 \Rightarrow \Phi_1)$. If H(x) means that *x* is a human being and F(x) means that *x* is featherless bipeds, then there is a model, where $(H(x) \Rightarrow F(x))\ \&\ (F(x) \Rightarrow H(x))$ is true. In other words, H and F have the same extension in M. For Frege the extension is expressed in propositions of the form $a = b$.

Intension: it is a corresponding family of properties (senses, *Sinne*) for a predicate or proposition. Let Γ be a collection of propositions derivated with the use of predicates $P_{11}, P_{12}, \ldots, P_{1k1}, P_{21}, \ldots, P_{2k2}, \ldots, P_{nkn}$, and K be a class of algebraic systems consisting of the same set of relations $R_{11}, R_{12}, \ldots, R_{1k1}, R_{21}, \ldots, R_{2k2}, \ldots, R_{nkn}$. Then two propositions Φ_1, Φ_2 of Γ have the same intension iff $(\Phi_1 \Rightarrow \Phi_2)\ \&\ (\Phi_2 \Rightarrow \Phi_1)$ is true on K. For instance, if we take the predicate, H, to be a human being and the predicate, R, to be rational, then we obtain the formula $(H(x) \Rightarrow R(x))\ \&\ (R(x) \Rightarrow H(x))$ is true everywhere. This means that H and R have the same intension. For Frege the intension is expressed in propositions of the form $a = a$.

> Now we shall introduce the terms ‘extension’ and ‘intension’ with respect to predicators. If two predicators apply to the same individuals in other words, if they are equivalent it is sometimes said that they are coextensive or that they have the same extension (in one of the various customary uses of this term). The use of ‘intension’ varies still more than that of ‘extension’. Two predicators have the same intension if and only if they are *L*-equivalent (Carnap).

1.3. Analytic a posteriori

Thus, *das dritte Reich* as the Platonic world of pure senses consists of analytic a priori, synthetic a priori and intensions. However, the basic idea of this approach that allows us to build up classical logic assumes that analytic a posteriori does not exist. Is it really so? In the *Naming and Necessity* Saul Kripke states that there exist ‘contingent a priori’ (instead of this term it is more presicely to say analytic a posteriori) propositions as well. They are constructed by using indexicals like ‘I’, ‘here’, ‘now’, ‘actually’, ‘you’. The example of analytic a posteriori is Descartes' famous *cogito* argument:

> To say ‘I think, therefore I am’ expresses a necessary truth only if it is taken as an analytic a posteriori principle: ‘I know from experience (a posteriori) that I can and do think, and this implies (analytically) that I exist’ (Palmquist).

According to Platonic presuppositions, which Kant, Frege and Carnap shared, logical relations do not depend upon our experience (the actual world), they are defined everywhere, i.e. on any class of algebraic systems (since they are analytic a priori). However, as Kripke showes, there are logical relations, namely modal operators (‘possible’, ‘necessary’), that can not be interpreted on any model. In defining them we can dot the I’s and cross the T’s only if we appreciate the so-called Kripke models. Modal relations being logical ones are to be labelled contingent a priori (as Kripke says) or analytic a posteriori (as Palmquist says and we are saying).

Kripke proposes to differ necessary truth from a priori knowable truth:

> Philosophers have talked ... [about] various categories of truth, which are sometimes called ‘a priori’, ‘analytic,’ ‘necessary’ ... these terms are often used as if whether there are things answering to these concepts is an interesting question, but we might as well regard them all as meaning the same thing.
> ... First the notion of a prioricity is a concept of epistemology. I guess the traditional characterization from Kant goes something like: a priori truths are those which can be known independently of any experience.
> ...The second concept which is in question is that of necessity ... what I am concerned with here is a notion which is not a notion of epistemology but of metaphysics ...We ask whether something might have been true, or might have been false. Well, if something is false, it’s obviously not necessarily true. If it is true, might it have been otherwise? Is it possible that, in this respect, the world should have been different than the way that it is? ... This in and of itself has nothing to do with anyone’s knowledge of anything. It’s certainly a philosophical thesis, and not a matter of obvious definitional equivalence, either that everything a priori is necessary or that everything necessary is a priori ... at any rate they are dealing with two different domains, two different areas, the epistemological and the metaphysical (Kripke).

Necessary truth may be a posteriori. This means that logical relations may be a posteriori, too. From this it follows that we could design logic as theory of necessary truth without Platonic presuppositions, i.e. without axioms. As a result, we could destroy *das dritten Reich*. So, we could hit Goliath’ head, i.e. the big head of Platonism.

Stephen Palmquis claims that analytic a posteriori takes place if conceptual (analytic) content satisfies not general, but particular conditions:

> What then is the difference between the 'analytic a posteriori' and Kant's special 'synthetic a priori' class of knowledge? The analytic a posteriori and synthetic a priori are similar classes of knowledge insofar as both are concerned with conditions imposed on the world by the subject (in contrast to the analytic a priori and synthetic a posteriori types, which are concerned with information that can be drawn out of, or deduced from, what we find in experience), but they differ by virtue of the fact that the former imposes *particular* conditions (a posteriori) with conceptual (analytic) content, whereas the latter imposes *general* conditions (a priori) with intuitive (synthetic) content (Palmquist).

This understanding of analytic a posteriori is too metaphysic. It is very far from logic and mathematics. In order to formally explicate the notion of analytic a posteriori, instead of primary terms and axioms as basic, initial notions of logic and mathematics let us consider transition and behavior as ground concepts. In this connection we can assume that there are no unchangeable things. Everything changes and nothing remains still (*πάντα χωρεῖ καὶ οὐδὲν μένει*). Entities that are felt eternal by us are an iteration in fact, i.e. a permanent iterance of the same. Eternal entities also move and vary, but in the modification they come back to the initial states.

The eternal existence in logic and mathematics, treated as iteration or recursion, will correspond to the notion 'analytic a posteriori'. On the one hand, iteration or recursion are a posteriori, because they stay in the eternal transition that is detected by experience. On the other hand, they are analytic, because they are eternal and contain a permanent iterance in the modification.

If we admit analytic a posteriori in logic and mathematics, we will already be able to deny notions 'analytic a priori' and 'synthetic a priori'. These notions are exhaling. The eternal in its pure sense disappears. We face just changeable things of two sorts: analytic a posteriori and synthetic a posteriori. The first notion expresses iteration in transitions, the second a permanent modification in transitions.

Logic and mathematics where all propositions (mathematical entities) are either analytic a posteriori or synthetic a posteriori are called anti-Platonic.

Anti-Platonic logic and mathematics (as the novel program of foundations of mathematics without Platonism) satisfy the following basic presuppositions:

- there are no propositions that are true on any class K (i.e. there are no logical axioms);
- there are no non-logical axioms, we draw conclusions from premises having the equal status, all formulas are infered from hypotheses;
- finally, any object has only a relative existence.

Within these assumptions we build up derivations without using axioms, therefore there is no sense in distinguishing logic and theory (i.e. logical and non-logical axioms), derivable and provable formulas, etc. In such a formal theory the conclusion is understood as massive-parallel computing, i.e. the deduction is considered as a transition of automaton. This formal proof theory is reduced to an appropriate cellular automaton which is called a *proof-theoretic cellular automaton*, whose cell states are regarded as well-formed formulas of a logical language. As a result, in deduction we do not obtain derivation trees and instead of the latter we find out derivation traces, i.e. a linear evolution of each singular premise. Thereby some derivation traces may be circular, i.e. some premises could be derivable from themselves, and hence some derivation traces may be infinite. Propositions from circular derivations are said to be analytic a posteriori. Other propositions are called synthetic a posteriori.

1.4. Proof-theoretic cellular automata

For any logical language L we can construct a *proof-theoretic cellular automaton* (instead of conventional deductive systems) simulating massive-parallel proofs.

Definition 4.1. *A proof-theoretic cellular automaton is a 4-tuple* $\mathsf{A} = \langle \mathsf{Z}^d, S, N, \delta \rangle$*, where:*

- $d \in \mathsf{N}$ is a number of dimensions and the members of Z^d are referred as cells,
- S is a finite or infinite set of elements called the states of an automaton A, the members of Z^d take their values in S, the set S is collected from well-formed formulas of a language L,
- $N \subset \mathsf{Z}^d \setminus \{0\}^d$ is a finite ordered set of *n* elements, N is said to be a neighborhood,
- $\delta : S^{n+1} \to S$ that is δ is the inference rule of a language L, it plays the role of local transition function of an automaton A.

As we see, an automaton is considered on the endless *d*-dimensional space of integers, i.e. on Z^d. Discrete time is introduced for $t = 0,1,2,\ldots$ fixing each step of inferring.

For any given $z \in \mathsf{Z}^d$, its neighborhood is determined by $z + N = \{z + \alpha : \alpha \in N\}$. There are two often-used neighborhoods:

- Von Neumann neighborhood $N_{VN} = \{z \in \mathsf{Z}^d : \sum_{k=1}^{d} | z_k | = 1\}$
- Moore neighborhood $N_M = \{z \in \mathsf{Z}^d : \max\limits_{k=1,d} | z_k | = 1\} = \{-1,0,1\}^d \setminus \{0\}^d$

For example, if $d = 2$, N_{VN} = {(–1, 0), (1, 0), (0, –1), (0, 1)}; N_M = {(–1, –1), (–1, 0), (–1, 1), (0, –1), (0, 1), (1, –1), (1, 0), (1, 1)}.

In the case $d = 1$, von Neumann and Moore neighborhoods coincide. It is easily seen that $| N_{VN} | = 2d$, $| N_M | = 3^d - 1$.

At the moment t, the *configuration* of the whole system (or the *global state*) is given by the mapping $x^t : \mathsf{Z}^d \to S$, and the *evolution* is the sequence $x^0, x^1, x^2, \ldots$ defined as follows: $x^{t+1}(z) = \delta(x^t(z), x^t(z+\alpha_1), \ldots, x^t(z+\alpha_n))$, where $\langle \alpha_1, \ldots, \alpha_n \rangle \in N$. Here x^0 is the initial configuration, and it fully determines the future behavior of the automaton. It is the set of all premises (not axioms).

We assume that δ is an inference rule, i.e. a mapping from the set of premises (their number cannot exceed $n = | N |$) to a conclusion. For any $z \in \mathsf{Z}^d$ the sequence $x^0(z)$, $x^1(z)$, … , $x^t(z)$, is called a *derivation trace from a state* $x^0(z)$. If there exists *t* such that $x^t(z) = x^l(z)$ for all $l > t$, then a derivation trace is *finite*. It is *circular/cyclic* if there exists l such that $x^t(z) = x^{t+l}(z)$ for all *t*.

Definition 4.2. *In case all derivation traces of a proof-theoretic cellular automaton* A *are circular, this automaton* A *is said to be reversible.*

Notice that x^{t+1} depends only upon x^t, i.e. the previous configuration. It enables us to build the function $G_{\mathsf{A}}: C_{\mathsf{A}} \to C_{\mathsf{A}}$, where C_{A} is the set of all possible configurations of the cellular automaton A (it is the set of all mappings $\mathsf{Z}^d \to S$, because we can take each element of this set as the initial configuration x^0, though not every element can arise in the evolution of some other configuration). G_{A} is called the *global function* of the automaton.

1. Example of modus ponens *Consider a propositional language* L *that is built in the standard way with the only binary operation of implication* $\supset$. *Let us suppose that well-formed formulas of that language are used as the set of states for a proof-theoretic cellular automaton* A. *Further, assume that modus ponens is a transition rule of this automaton* A *and it is formulated for any* φ, $\psi \in \mathsf{L}$ *as follows:*

$$x^{t+1}(z) = \begin{cases} \psi, & \textit{if } x^t(z) = \varphi \supset \psi \textit{ and } \varphi \in (z+N); \\ x^t(z), & \textit{otherwise} \end{cases}$$

The further dynamics will depend on the neighborhood. If we assume the Moor neighborhood in the 2-dimensional space, this dynamics will be exemplified by the evolution of cell states in fig. 4.1 – fig. 4.3.

Figure 4.1. An initial configuration of a proof-theoretic cellular automaton A with the Moor neighborhood in the 2-dimensional space, its states run over formulas set up in a propositional language L with the only binary operation $\supset$, $t = 0$. Notice that p, q, r are propositional variables.

$(p\supset q)\supset r$	$p\supset(p\supset q)$	$p\supset q$	$(p\supset q)\supset(p\supset q)$	$(r\supset p)\supset r$
$(p\supset r)\supset(q\supset r)$	$p\supset q$	p	$p\supset(p\supset q)$	$r\supset p$
$p\supset r$	p	$p\supset(q\supset(p\supset q))$	p	r
$p\supset(q\supset r)$	$p\supset p$	$p\supset q$	$(p\supset r)\supset(q\supset p)$	$p\supset r$
$p\supset q$	$p\supset(q\supset p)$	q	$p\supset r$	p

Figure 4.2. An evolution of A described in fig. 4.1 at the time step $t = 1$.

r	$p\supset q$	q	$p\supset q$	r
$q\supset r$	q	p	$p\supset q$	p
r	p	$q\supset(p\supset q)$	p	r
$q\supset r$	p	q	$q\supset p$	r

$p \supset q$	$p \supset (q \supset p)$	q	r	p

Figure 4.3. An evolution of $\mathcal{A}$ described i fig. 4.1 at the time step $t = 3$. Its configuration cannot vary further.

r	q	q	q	r
r	q	p	q	p
r	p	q	p	r
r	p	q	p	r
q	p	q	r	p

This example shows that first we completely avoid axioms and secondly we take premisses from the cell states of the neighborhood according to a transition function. As a result, we do not come across proof trees in our novel approach to deduction taking into account that a cell state has just a linear dynamics (the number of cells and their location do not change). This allows us evidently to simplify deductive systems.

Now we are trying to consider a cellular-automaton presentation of two basic deductive approaches: Hilbert's type and sequent ones.

2. Example of Hilbert's inference rules *Suppose a propositional language $\mathcal{L}$ contains two basic propositional operations: negation and disjunction. As usual, the set of all formulas of $\mathcal{L}$ is regarded as the set of states of an appropriate proof-theoretic cellular automata. In that we will use the exclusive disjunction of the following five inference rules converted from Joseph R. Shoenfield's deductive system:*

$$x^{t+1}(z) = \begin{cases} \psi \vee \varphi, & if\ \ x^t(z) = \varphi; \\ \varphi, & if\ \ x^t(z) = \varphi \vee \varphi; \\ (\chi \vee \psi) \vee \varphi, & if\ \ x^t(z) = \chi \vee (\psi \vee \varphi); \\ \chi \vee \psi, & if\ \ x^t(z) = \varphi \vee \chi\ and\ (\neg\varphi \vee \psi) \in (z + N); \\ \chi \vee \psi, & if\ \ x^t(z) = \neg\varphi \vee \psi\ and\ (\varphi \vee \chi) \in (z + N). \end{cases}$$

3. Example of sequent inference rules *Let us take a sequent propositional language $\mathcal{L}$, in which the classical propositional language with negation, conjunction, disjunction and implication is extended by adding the sequent relation $\circ$. Recall that a sequent is an expression of the form $\Gamma_1 \circ \Gamma_2$, where $\Gamma_1 = \{\varphi_1, \ldots, \varphi_j\}$, $\Gamma_2 = \{\psi_1, \ldots, \psi_i\}$ are finite sets of well-formed formulas of the standard propositional language, that has the following interpretation: $\Gamma_1 \circ \Gamma_2$ is logically valid iff*

$$\bigwedge_j \varphi_j \supset \bigvee_i \psi_i$$

is logically valid. Let $\mathcal{S}$ denote the set of all sequents of $\mathcal{L}$, furthermore let us assume that this family $\mathcal{S}$ is regarded as the set of states for a proof-theoretic cellular automaton $\mathcal{A}$.

The transition rule of L *is an exclusive disjunction of the 14 singular rules (6 structural rules and 8 logical rules):*

$$x^{t+1}(z) = \Gamma_1 \circ \Gamma_2, \begin{cases} \textit{if } \Gamma_1 \circ \Gamma_2 \textit{ is a result of applying to } x^t(z) \\ \textit{eather one of structural rules} \\ \textit{or the left (right) introduction of negation} \\ \textit{or the left introduction of conjunction} \\ \textit{or the right introduction of disjunction} \\ \textit{or the right introduction of implication.} \end{cases}$$

$$x^{t+1}(z) = \begin{cases} \Gamma \circ \Gamma', \psi \wedge \chi, & \textit{if } x^t(z) = \Gamma \circ \Gamma', \psi \textit{ and} \\ & (\Gamma \circ \Gamma', \chi) \in (z+N); \\ \Gamma, \psi \vee \chi \circ \Gamma', & \textit{if } x^t(z) = \Gamma, \psi \circ \Gamma' \textit{ and} \\ & (\Gamma, \chi \circ \Gamma') \in (z+N); \\ \psi \supset \chi, \Gamma, \Delta \circ \Gamma', \Delta', & \textit{if } x^t(z) = \Gamma \circ \Gamma', \psi \textit{ and} \\ & (\chi, \Delta \circ \Delta') \in (z+N). \end{cases}$$

4. Example of Brotherston's cyclic proofs *The sequent language used in the previous example we extend by adding predicates N, E, O and appropriate inference rules of Fig. 4.4 for them. Further, let us extend also the automaton of Example 3 in the same way by representing inference rules of Fig.4.4 in the cellular-automatic form.*

Now we assume that a cell has an initial state $[\Gamma, N(z) \circ \Delta, O(z), E(z)]$ and its neighbor cell an initial state $[\Gamma, z = 0 \circ \Delta]$ that is equal to $[\Gamma, z = 0 \circ \Delta, O(z), E(z)]$ for any $t = 4,14,24,\ldots$ and to $[\Gamma, z = 0 \circ \Delta, E(z), O(z)]$ for any $t = 9,19,29,\ldots$. Then we will have the following infinite cycle:

$[\Gamma, N(z) \circ \Delta, O(z), E(z)] \rightarrow^{(substitution)} [\Gamma, N(y) \circ \Delta, O(y), E(y)] \quad \rightarrow [\Gamma, N(y) \circ \Delta, O(y), O(y+1)]$
$\rightarrow [\Gamma, N(y) \circ \Delta, E(y+1), O(y+1)]$
$\rightarrow [\Gamma, z = (y+1), N(y) \circ \Delta, O(z), E(z)] \rightarrow^{(case N)} [\Gamma, N(z) \circ \Delta, E(z), O(z)] \rightarrow \ldots$

Figure 4.4. Inference rules for predicates N ('being a natural number'), E ('being an even number'), O ('being an odd number'), see *Brotherston*.

$$\frac{\Gamma^\circ N(x)}{\Gamma^\circ N(x+1)}, \quad \frac{\Gamma^\circ\Delta}{\Gamma^\circ\Delta, N(0)}, \quad \frac{\Gamma^\circ E(x)}{\Gamma^\circ O(x+1)}, \quad \frac{\Gamma^\circ O(x)}{\Gamma^\circ E(x+1)}, \quad \frac{\Gamma^\circ\Delta}{\Gamma^\circ\Delta, E(0)},$$

$$\frac{N(x)^\circ\Delta}{N(x+1)^\circ\Delta}, \quad \frac{\Gamma^\circ\Delta}{\Gamma, N(0)^\circ\Delta}, \quad \frac{E(x)^\circ\Delta}{O(x+1)^\circ\Delta}, \quad \frac{O(x)^\circ\Delta}{E(x+1)^\circ\Delta}, \quad \frac{\Gamma^\circ\Delta}{\Gamma, E(0)^\circ\Delta},$$

$$\frac{\Gamma, t = 0^\circ\Delta \qquad \Gamma, t = x+1, N(x)^\circ\Delta}{\Gamma, N(t)^\circ\Delta}(Case N), where x \notin FV(\Gamma \cup \Delta \cup \{N(t)\}),$$

$$\frac{\Gamma^\circ\Delta}{\Gamma[x]^\circ\Delta[x]}(Substitution).$$

As we see, the possibility of cyclic derivation traces depends on configuration of cell states.

Definition 4.3. *Initial states of cyclic derivation traces are called analytic a posteriori propositions. All other states are called synthetic a posteriori propositions. For instance, the premise* $[\Gamma, N(z)^\circ\Delta, O(z), E(z)]$ *of Example 4 is an analytic a posteriori proposition.*

From this it follows immediately that all premisses of a reversible proof-theoretic cellular automaton are analytic *a posteriori* propositions.

Definition 4.4. *A set of all premisses of a cyclic derivation trace is called a logical biota. For instance, the premises* $[\Gamma, N(z)^\circ\Delta, O(z), E(z)]$ *and* $[\Gamma, z = 0^\circ\Delta]$ *of Example 4 forms a logical biota.*

Each logical biota contains at least one analytic a posteriori proposition.

Proposition 4.1. *Let us assume that a proof-theoretic cellular automaton* A *is not reversible and contains at least one logical biota. Then there is a natural number n such that at the time step n the configuration of logical biota cannot vary further.*

A logical biota is a kind of self-organization in deductions and may be considered as a dissipative system in proof-theoretic cellular automata. Also here, it is possible to say that for logical biotas the dissipation (energy dispersion) takes place. It is related by that logical biotas under the influence of an environment can disintegate, too. In other words, their dynamics can stop at the time step *n*.

A vivid example of dissipative systems in the nature is presented by an auto-oscillation, namely by an undamped wave supported from outside in non-linear system with the energy damping. In spite of the fact that auto-oscillations are provided with exterior effects, their properties are determined by parametres of the system. We observe a version of auto-oscillations in logical biotas.

The evolutionary character of behaviour is a major property of the dissipative structure. Meanwhile in the class of evolutionary processes it is possible to distinguish various types of dissipative systems. The first type includes systems whose evolution moves to a stationary state, the second one contains systems whose evolution goes through a sequence of nonequilibrium stationary states of an open system. Systems of the first type are termed as simple structures, systems of the second type as difficult ones.

Let us notice that an absolute stability in front of exterior effects is achievable only at very great values of entropy and it is equivalent to the thermal death. For cellular automata the state, when the configuration of cellular automata cannot vary further, is an analogue with the thermal death. In other cases the system stays in states which have no ultimate balance. Nevertheless, entropy can have a certain relative maximum. In this situation, the system appears as a dissipative structure having a complex organisation (and it is capable to adaptation, evolution, etc.).

The common feature of self-organized processes is in an increase of the order from the chaos due to the occurrence of inconvertible thermodynamic balance with the environment, which is opposite, however, to the thermal death (a balance of pices of environment that have inetracted in the independent and unstructured way). Self-organized processes of dissipative systems proceed together with processes that lead to a degradation and decay of system as a whole.

In proof-theoretic cellular automata we can consider dissipative and self-organized phenomena as well. The notion of analytic a posteriori allows us to do it. At the same time, taditional tasks concerning proof theory like completeness and independence of axioms lose their sense in massive-parallel proof theory, although it can be readily shown that we can speak about consistency:

Proposition 4.2. *Proof theories given in Examples from 1 to 4 are consistent, i.e. we cannot deduce a contradiction within them.*

1.5. Unconventional (non-well-founded) logic

We have just constructed the theory of proofs in which there is no place for theorems or provable formulas (a priori propositions). But in this theory, there appears the notion of cyclic proofs (analytic a posteriori propositions). We can go further and construct a formal language in which there is no basic notions of classical semantics, e.g. there are no valid formulas (analytic a priori propositions) and propositions that are true on a class of models (synthetic a priori propositions).

The modern language will be called *unconventional* or *non-well-funded*. Its formulas are defined as follows:

- Each atomic formula is a formula.
- If $\Phi_1 \varepsilon \Phi_2$ or $\Phi_1 \& \Phi_2$ or $\Phi_1 \vee \Phi_2$ or $\Phi_1 \Rightarrow \Phi_2$ or $\neg\Phi_1$, $\neg\Phi_2$ are formulas, then Φ_1, Φ_2 are formulas.

The above definition assumes that the formula is a syntactic object which does not satisfy the set-theoretic axiom of foundation. It means that the given syntactic objects can already be of an infinite length and comprise cycles. For example, cyclic expressions $(\Phi_1 \Rightarrow (\Phi_2 \Rightarrow (\Phi_1 \Rightarrow (\Phi_2 \Rightarrow (\ldots)))))$ or $(\Phi_1 \,\&\, (\Phi_1 \,\&\, (\Phi_1 \,\&\, (\Phi_1 \,\&\, (\ldots)))))$ are well formed formulas. These expressions can be defined by recursion, too:

The formula of infinite length $(\Phi_1 \Rightarrow (\Phi_2 \Rightarrow (\Phi_1 \Rightarrow (\Phi_2 \Rightarrow (\ldots)))))$ is equivalent to the cyclic definition $\Phi = (\Phi_1 \Rightarrow (\Phi_2 \Rightarrow \Phi))$.

The formula of infinite length $(\Phi_1 \,\&\, (\Phi_1 \,\&\, (\Phi_1 \,\&\, (\Phi_1 \,\&\, (\ldots)))))$ is equivalent to the cyclic definition $\Phi = (\Phi_1 \,\&\, \Phi)$.

Meanings of unconventional-logical connectives ε, &, ∨, ⇒, ¬ are defined not by true tables as this proceeds usually, but by hybrid cellular automata. Let us take now a cellular automaton $\mathsf{A} = \langle \mathsf{Z}^d, S, N, \delta \rangle$, where:

- $d \in \mathsf{N}$ is a number of dimensions and the members of Z^d are referred as cells,
- S is a finite or infinite set of elements called the states of an automaton A, the members of Z^d take their values in S, the set S is collected from two truth values 1 and 0 for well-formed formulas of a language L,
- $N \subset \mathsf{Z}^d \setminus \{0\}^d$ is a finite ordered set of *n* elements, N is said to be a neighborhood,
- $\delta : S^{n+1} \to S$ that is $\delta \in \{\varepsilon, \&, \vee, \Rightarrow, \neg\}$ is the truth valuation for each logical operation of a language L, it plays the role of local transition function of an automaton A.

Definition 5.1. *A truth valuation of conjunction is a transition rule of the automaton* A*, where* $S = \{1, 0\}$*, and it is formulated for any formula* $\Phi_1 \,\&\, \Phi_2 \in \mathsf{L}$ *as follows:*

$$x^{t+1}(z) = \begin{cases} 1, & if\ x^t(z) = 1\ and\ 0 \notin (z+N); \\ 0, & otherwise \end{cases}$$

Definition 5.2. *A truth valuation of disjunction is a transition rule of the automaton* A*, where* $S = \{1, 0\}$*, and it is formulated for any formula* $\Phi_1 \vee \Phi_2 \in \mathsf{L}$ *as follows:*

$$x^{t+1}(z) = \begin{cases} 1, & if\ x^t(z) = 1\ or\ 1 \in (z+N); \\ 0, & otherwise \end{cases}$$

Definition 5.3. *A truth valuation of implication (respectively affirmation) is a transition rule of the automaton* A*, where* $S = \{1, 0\}$*, and it is formulated for any formula* $\Phi_1 \Rightarrow \Phi_2 \in \mathsf{L}$ (*respectively* $\Phi_1\ \varepsilon\ \Phi_2 \in \mathsf{L}$) *as follows:*

$$x^{t+1}(z) = \begin{cases} 0, & if\ x^t(z) = 0\ and\ 0 \notin (z+N); \\ 1, & otherwise \end{cases}$$

Definition 5.4. *A truth valuation of negation is a transition rule of the automaton* $\mathcal{A}$, *where* $S = \{1, 0\}$, *and it is formulated for any formula* $\neg\Phi \in \mathcal{L}$ *as follows:*

$$x^{t+1}(z) = \begin{cases} 0, & if\ x^t(z) = 1; \\ 1, & otherwise \end{cases}$$

Algorithm of truth valuation *Let* Φ *be a formula. If it is atomic, then its truth value is an initial configuration of a cellular automaton with the set of states* $\{1, 0\}$. *If it is not atomic, then we start the evaluation with the most outside connective* k_1. *By using one of definitions 5.1 – 5.4 that corresponds to* k_1, *we transform an initial configuration of a cellular automaton with the set of states* $\{1, 0\}$. *This transformation is fixed by steps* $t_{k1} = 1, 2, 3, \ldots$ *Then we move to a more inside connective* k_2. *For each step* $t_{k1} = 1, 2, 3, \ldots$ *we transform a configuration at the step* t_{k1} *in accordance with one of definitions 5.1 – 5.4 that corresponds to the new connective* k_2. *This means that for each step* t_{k1} *we obtain a new cellular automaton for the truth evaluation of* k_2. *Each transformation of that cellular automaton is fixed by new steps* $t_{k2} = 1, 2, 3, \ldots$ *Notice that it is possible that simultaneously we have two inside connectives* k_2 *and* k_2' *of the same range. In this case we do all the same, but in parallel manner. Further, we can move to a more inside connective* k_3. *For each step* $t_{k2} = 1, 2, 3, \ldots$ *we transform a configuration at the step* t_{k2} *in accordance with one of definitions 5.1 – 5.4 that corresponds to the connective* k_3 *and so on.*

To provide an example, let us evaluate the formula ($\Phi_1 \Rightarrow (\Phi_2\ \&\ \Phi_1)$) in a cellular automaton $\mathcal{A}$ with the Moor neighborhood in the 2-dimensional space:

(I) Initial configuration, $t = 0$

1	0	0
0	1	1
0	1	0

We begin our evaluation with the connective $\Rightarrow$ as most outside. Thereby we are using definition 3:

(II) $t = 1$

1	1	1
1	1	1
1	1	0

So, we have only one transformation fixed by step 1. Then we move to the inside connective & and by definition 5.1 we obtain the following data over (II):

(III) $t = 2$

1	1	1
1	0	0
1	0	0

⇓

(IV) $t = 3$

0	0	0
0	0	0
0	0	0

The automaton cannot vary further. If we had an additional more inside connective, then we would have two automata (III) and (IV) for further modification in accordance with this more inside connective.

Definition 5.5. *The formula is called analytic a posteriori if its truth valuation cellular automaton is reversible. The most simple case of analytic a posteriori formula is* $\neg\Phi$ *as it follows from definition 5.4 directly*.

As models of unconventional (non-well-founded) formulas we will use coalgebraic systems, e.g. labelled transition systems. The latter are defined as systems <S, $\to_S$, A>, consisting of a set S of states, a transition relation $\to_S \subseteq S \times A \times S$, and a set A of labels. As always, $s \to_S^a s'$ is used to denote <s, a, s'> $\in \to_S$.

Cellular automata $\mathcal{A} = \langle \mathbb{Z}^d, S, N, \delta \rangle$ may be regarded as a kind of labelled transition systems of the form $\langle S, \to_S, \{\delta_i\}_i \rangle$, where S is a set of states, $\to_S \subseteq S \times \{\delta_i\}_i \times S$, and $\{\delta_i\}_i$ is a set of labels for the local transition function δ of an automaton $\mathcal{A}$. The members of $\{\delta_i\}_i$ is obtained as follows. We have *i* different combinations of neighbor states. This number depends on the number of neighborhood $n = |N|$ and the number of states $k = |S|$, namely $i = k^n$. Hence, $|\{\delta_i\}_i| = i$. Each combination of *i* has its own label $\delta_{1 \le j \le i}$.

Further let us define:

$$B(X) = P(A \times X) = \{V : V \subseteq A \times X\}, \text{ for any set } X .$$

Then a labeled transition system <S, $\to_S$, A> can be represented as a B-coalgebra <S, α_S> by defining:

$$\alpha_S : S \to B(S);\ s \to \{\langle a, s' \rangle : s \to_S^a s'\}$$

Notice that the class of all labelled transition systems coincides with the class of all B-coalgebras.

Let us consider now two B-coalgebras $<S, \alpha_S>$ and $<T, \alpha_T>$. A B-bisimulation between S and T is said to be a relation $R \subseteq S \times T$ satisfying, for all $<s, t> \in R$, the following conditions:

1. for all s' in S, if $s \ \to_S^a \ s'$, then there is t' in T with $t \ \to_T^a \ t'$ and $<s', t'> \in R$,
2. for all t' in T, if $t \ \to_T^a \ t'$, then there is s' in S with $s \ \to_S^a \ s'$ and $<s', t'> \in R$

Instead of the induction proof principle used in algebras, the coinduction proof principle is involved into coalgebras: for every bisimulation R on S, $R \subseteq \Delta_S$ (where $\Delta_S = \{<s, s> : s \in S\}$. Equivalently, for all s and s' in S, if s and s' are bisimilar in S, then s = s'.

Let M be a B-coalgebra $<S, \alpha_S>$. Let us tale the set consisting of k_1 one-placed relations, k_2 two-placed relations, k_n n-placed relations, $R_{11}, R_{12}, \ldots, R_{1k_1}, R_{21}, \ldots, R_{2k_2}, \ldots, R_{nk_n}$, i.e. R_{ij} means j-th i-placed relations. Then an i-placed relation in M is defined as a B-coalgebra over S^i, namely $\alpha_S : S^i \to B(S^i)$.

To individual variables $x, y, z, \ldots$ we can assign meanings in the set S by an interpretation I. In other words, $I(x) \in S$. To predicates P_{ij} we can assign relations R_{ij} understood as B-coalgebras over S^i.

Let Γ be a set of propositions of our formal language. Taking into account that for every predicate P_{ij} there exists a relation R_{ij} of B-coalgebra M, we can consider M as a model of Γ (it is denoted by $M \models \Gamma$) if the following conditions are satisfied for any formulas Φ_1, Φ_2 of Γ:

$I(P_{ij}(x_1, x_2, \ldots, x_i)) = R_{ij}(I(x_1), I(x_2), \ldots, I(x_i))$ is a B-coalgebra over S^i, i.e. $M \models R_{ij}(I(x_1), I(x_2), \ldots, I(x_i))$;

$M \models \neg \Phi_1$ iff the truth valuation of $\neg \Phi_1$ is bisimilar with M;

$M \models \Phi_1 \& \Phi_2$ iff the truth valuation of $\Phi_1 \& \Phi_2$ is bisimilar with M;

$M \models \Phi_1 \vee \Phi_2$ iff the truth valuation of $\Phi_1 \vee \Phi_2$ is bisimilar with M;

$M \models \Phi_1 \Rightarrow \Phi_2$ iff the truth valuation of $\Phi_1 \Rightarrow \Phi_2$ is bisimilar with M;

$M \models \Phi_1 \varepsilon \Phi_2$ iff iff the truth valuation of $\Phi_1 \varepsilon \Phi_2$ is bisimilar with M.

Let us assume that there is a class of coalgebras consisting of the same set of relations $R_{11}, R_{12}, \ldots, R_{1k_1}, R_{21}, \ldots, R_{2k_2}, \ldots, R_{nk_n}$. Then there is no proposition Φ that is true on any system of the class K. However, there exists a proposition Φ that is rearizable on the class K, i.e. K contains a system where Φ is true.

Let Γ be a family of propositions. The class of all coalgebraic systems, where all propositions of Γ are true, is denoted by $Mod(\Gamma)$.

Proposition 5.1. If $\Gamma_1 \subseteq \Gamma_2$, then $Mod(\Gamma_1) \subseteq Mod(\Gamma_2)$.

Let K be a class of coalgebraic systems. The family of all propositions that are true on the class K is an elementary theory and is called an elementary theory of the class K. It is denoted by Th(K).

Proposition 5.2. If $K_1 \subseteq K_2$, then $Th(K_1) \subseteq Th(K_2)$.

Propositions 5.1 – 5.2 demonstrate some features of anti-Platonic logic and mathematics: the more there are propositions, the more there are models and also the more there are models, the more there are propositions.

1.6. Analytic a posteriori and unconventional computing

Modern logic was finally formed in the beginning of the 20th century. At that time the formal language of logic has been created and since then its main properties have been studied. Then modern logic has had no key modifications. Since then and till now it has been constructed as axiomatic theory. There is, say, many-valued logic. However, the way of its designing does not differ from two-valued logic as a matter of principle. We have there an axiomatic system in which axioms differ, but the formal language is almost the same.

The main advantage of logic is that on its basis the computer architecture has been constructed (for example, circuits are modelled by logical relations), and also programming languages have been developed. Since then the computer equipment has not varied essentially from the logical point of view. For instance, due to computers new materials memory, speed are to increase, but the logic remained almost same.

In the last decades one has carried out many researches in direction of how it is possible to design logic which would describe dynamics of any natural process as computation. The given field of research is called unconventional computing. According to this approach any physical, chemical or biological process is an unconventional computer. In unconventional computers, the architecture ("iron") and the software coincide, i.e. during computations the architecture of the computer varies, too.

Conventional computers are based on logic that was obtained more than a century ago. Now the problem is set up, if unconventional computers (chemical, biological, etc.) are possible. For designing them we should know, what logic they need, whether this logic is conventional or not. Many (if not all) experts claim that logic of unconventional computing (e.g. logic of circuits) should differ from conventional one. Some new logic such as glider logic was proposed. However, universal logical methods for unconventional computing were not obtained still.

If one does the trick to build up universal logic of unconventional computers (and sooner or later it will be possible), then rational robots-humanoids round us will become a part of everyday's reality. Every thing in our world is an unconventional computer, any physical, chemical or biological system. And the human being is not an exception to the rule here. If we understand logic of these systems, it will make an enormous break-in in computer equipments and in all exact sciences. We will already live in another world. Our reality will vary cardinally.

One of the tasks of logic for unconventional computing consists in the logical formalization of self-organized phenomena. These phenomena can be understood as logical biotas, i.e. cyclic deductions.

As we see, logic will soon become a science that rocks the world. Logic will privatise life for good and all. All exact sciences including classical ontology will disappear. Please ponder what the project of logic for unconventional computing means in fact. It is a project of absorbing all exact sciences. The necessity for operating theoretical terms of physics, chemistry, biology will disappear. There will be no foggy notions such as “matter”, “energy”, “chemical reaction”, “morphology”, “physiology”. Instead of them there will be notions of computations describing processes in physical, chemical and biological systems. There will be a reduction of all natural-science knowledge to logic!

Logic has now grown up from children's panties of semiotics. In a few time, it will absorb all other sciences. Then we will speak not only about being, but also about any natural processes purely from the logical point of view. Logic will become a unified general science. As concerns engineering, we will face the most grandious revolutionary changes in the human civilisation that have never been still!

1.7. The proof-theoretic cellular automaton for Belousov-Zhabotinsky reaction

Unconventional computing assumes that massive-parallel computations are observed everywhere in natural systems. There are different approaches to nature-inspired computing: reaction-diffusion computing (Adamatzky, De Lacy Costello, Asai), chemical computing (Berry, Boudol), biological computing (Adamatzky 2007, Schumann, Adamatzky 2011), etc. In all those computational models parallel inferring and concurrency are assumed as key notions. In the paper (Khrennikov, Schumann 2009, Khrennikov 2010) a hypothesis was put forward that the paradigm of parallel and concurrent computation caused by rejecting the set-theoretic axiom of foundation can be widely applied in modern physics. In this section we are analyzing simulating Belousov-Zhabotinsky reaction within the framework of our theory of massive-parallel proofs.

Belousov-Zhabotinsky reaction is the best-known example of auto-wave phenomenon in chemistry, i.e. the most famous example of self-organizing system there. We could show that the Belousov-Zhabotinsky reaction can be understood as a logical biota in massively parallel proof theory. In other words, we need a proof-theoretic cellular automaton with circular proofs for formalizing the Belousov-Zhabotinsky reaction containing feedback relations. The mechanism of this reaction (namely, cerium (III) $\leftrightarrow$ cerium (IV) catalyzed reaction) is very complicated: its recent model contains 80 elementary steps and 26 variable species concentrations. Let us consider a simplification of Belousov-Zhabotinsky reaction assuming that the set of states consists just of the following reactants: Ce^{3+}, $HBrO_2$, BrO_3^-, H^+, Ce^{4+}, H_2O, $BrCH(COOH)_2$, Br^-, $HCOOH$, CO_2, $HOBr$, Br_2, $CH_2(COOH)_2$ which interact according to inference rules (reactions) (1) – (7). In this reaction we observe sudden oscillations in color from yellow to colorless, allowing the oscillations to be observed visually. In spatially non-homogeneous systems (such as a simple petri dish), the oscillations propagate as spiral wave fronts. The oscillations last about one minute and are repeated over a long period of time. The color changes are caused by alternating oxidation-reductions in which cerium changes its oxidation state from cerium (III) to cerium (IV) and vice versa: $Ce^{3+} \to Ce^{4+} \to Ce^{3+} \to \ldots$.

When Br^- has been significantly lowered, the reaction pictured by inference rule (1) causes an exponential increase in bromous acid ($HBrO_2$) and the oxidized form of the metal ion catalyst and indicator, cerium(IV). Bromous acid is subsequently converted to bromate (BrO_3^-) and $HOBr$ (the step (3)). Meanwhile, the step (2) reduces the cerium (IV) to cerium (III) and simultaneously increase bromide (Br^-) concentration. Once the bromide concentration is high enough, it reacts with bromate (BrO_3^-) and $HOBr$ in (4) and (6) to form Br_2, further Br_2 reacts with $CH_2(COOH)_2$ to form $BrCH(COOH)_2$ and the process begins again. Thus, parallel processes in (1) – (7) have several cycles which are performed synchronously.

The proof-theoretic simulation of Belousov-Zhabotinsky reaction can be defined as follows:

Definition 7.1. *Consider a propositional language* $\mathcal{L}$ *with the only binary operation* $\oplus$*, it is built in the standard way over the set of variables* S = {Ce^{3+}, $HBrO_2$, BrO_3^-, H^+, Ce^{4+}, H_2O, $BrCH(COOH)_2$, Br^-, $HCOOH$, CO_2, $HOBr$, Br_2, $CH_2(COOH)_2$}. *Let* $\mathcal{S}$ *be the set of states of proof-theoretic cellular automaton* $\mathcal{A}$. *The inference rule of the automaton is presented by the conjunction of singular inference rules (1) – (7):*

$$(1) \wedge (2) \wedge (3) \wedge (4) \wedge (5) \wedge (6) \wedge (7).$$

The operation $\oplus$ has the following meaning: $A \oplus B$ defines a probability distribution of events A and B in neighbor cells participated in a reaction caused the appearance of $A \oplus B$. Then $\mathcal{A}$ simulates the Belousov-Zhabotinsky reaction.

Definition 7.2. *Let* p, s_i, s_{i+1} $\in$ {Ce^{3+}, $HBrO_2$, BrO_3^-, H^+, Ce^{4+}, H_2O, $BrCH(COOH)_2$, Br^-, $HCOOH$, CO_2, $HOBr$, Br_2, $CH_2(COOH)_2$}. *A state* p *is called a premise for deducing* s_{i+1} *from* s_i *by the inference rule* $(1) \wedge (2) \wedge \ldots \wedge (7)$ *iff*

- p is s_i or,
- in a neighbor cell we find out an expression of the form $p \oplus A, B \oplus C$, where A, B, C are propositional metavariables, i.e. they run over either the empty set or the set of states closed under the operation $\oplus$. Thus, we assume that each premise should occur in a separate cell. This means that if we find out an expression $p_i \oplus p_j, B \oplus C$ or $p_i \oplus A, p_j \oplus C$ in a neighbor cell and both p_i and p_j are needed for deducing, whereas p_i, p_j do not occur in other neighbor cells, then p_i, p_j could not be considered as premises.

$$x^{t+1}(z)=\begin{cases} Ce^{4+}\oplus HBrO_2\oplus H_2O, if\ \ x^t(z)\in\{Ce^{3+}\}\ and \\ \ \ premises\, HBrO_2, BrO_3^-, H^+\in(z+N); \\ \\ x^t(z), otherwise. \end{cases} \tag{1}$$

$$x^{t+1}(z)=\begin{cases} Br^-\oplus Ce^{3+}\oplus HCOOH\oplus CO_2\oplus H^+, if\ \ x^t(z)\in\{Ce^{4+}\} \\ and\ \ premises\, BrCH(COOH)_2, H_2O\in(z+N); \\ \\ x^t(z), otherwise. \end{cases} \tag{2}$$

$$x^{t+1}(z)=\begin{cases} HOBr\oplus BrO_3^-\oplus H^+, & if\ \ x^t(z)\in\{HBrO_2\}; \\ \\ x^t(z), & otherwise. \end{cases} \tag{3}$$

$$x^{t+1}(z)=\begin{cases} HOBr\oplus HBrO_2, & if\ \ x^t(z)\in\{BrO_3^-\}\ and \\ & premises\, Br^-, H^+\in(z+N); \\ \\ x^t(z), & otherwise. \end{cases} \tag{4}$$

$$x^{t+1}(z)=\begin{cases} HOBr, & if\ \ x^t(z)\in\{Br^-\}\ and \\ & premises\, HBrO_2, H^+\in(z+N); \\ \\ x^t(z), & otherwise. \end{cases} \tag{5}$$

$$x^{t+1}(z)=\begin{cases} Br_2\oplus H_2O, & if\ \ x^t(z)\in\{HOBr\}\ and \\ & premises\, Br^-, H^+\in(z+N); \\ \\ x^t(z), & otherwise. \end{cases} \tag{6}$$

$$x^{t+1}(z) = \begin{cases} Br^- \oplus H^+ \oplus BrCH(COOH)_2, if\ x^t(z) \in \{Br_2\} \\ and\ premises CH_2(COOH)_2 \in (z+N); \\ \\ x^t(z), otherwise. \end{cases} \tag{7}$$

Example of Belousov-Zhabotinsky's cyclic proofs *We can simplify the automaton defined above assuming that* $\oplus$ *is a metatheoretic operation with the following operational semantics:*

$$\frac{A \oplus B}{A} \qquad \frac{A \oplus B}{B},$$

where A and B are metavariables defined on S. The informal meaning of that operation is that we can ignore one of both variables coupled by $\oplus$. In the cellular automaton A this metaoperation will be used as follows:

$$x^{t+1}(z) = \begin{cases} X, Y, & if\ x^t(z) = A \oplus B\ and\ according\ to\ rules\ (1)-(7), \\ & X\ changes\ from\ A\ and\ Y changes\ from\ B; \\ X, & if\ x^t(z) = A \oplus B\ and\ according\ to\ rules\ (1)-(7), \\ & X\ changes\ from\ A\ and\ B\ does\ not\ change; \\ Y, & if\ x^t(z) = A \oplus B\ and\ according\ to\ rules\ (1)-(7), \\ & Y\ changes\ from\ B\ and\ A\ does\ not\ change; \\ A \oplus B, & if\ x^t(z) = A \oplus B\ and\ rules\ (1)-(7) \\ & cannot\ be\ applied\ to\ A\ or\ B. \end{cases} \tag{8}$$

Let us suppose now that X, Y run over the set of states closed under the operation $\oplus$.

$$x^{t+1}(z) = \begin{cases} X, Y, & if\ (i)\ x^t(z) = X, Y\ and\ (ii)\ both\ X\ and\ Y \\ & are\ simultaneously\ usable\ (not\ usable) \\ & as\ premises\ in\ at\ least\ two\ different\ rules \\ & of\ (1)-(7)\ (see\ definition\ 2); \\ X, & if\ (i)\ x^t(z) = X, Y\ and\ (ii)\ only\ X\ is\ usable \\ & as\ a\ premise\ in\ at\ least\ one\ rule\ of\ (1)-(7); \\ Y, & if\ (i)\ x^t(z) = X, Y\ and\ (ii)\ only\ Y\ is\ usable \\ & as\ a\ premise\ in\ at\ least\ one\ rule\ of\ (1)-(7). \end{cases} \tag{9}$$

$$indempotency: A ::= A, A. \tag{10}$$

$$\textit{commutativity}: A, B ::= B, A. \tag{11}$$

Hence, we cannot ignore one of both variables coupled by $\oplus$ and should accept both them if in the neighborhood there are reactants that catenate both variables and change them. This rule is the simplest interpretation of $A \oplus B$ in definition 7.1. We have three cases: (i) both variables are catenated with reactants from the neighborhood, in this case we mean that the probability distribution of events A and B is the same and equal to 0.5 and, as a result, we cannot choose one of them and accept both; (ii) only A is catenated with reactants from the neighborhood, then the probability distribution of event A is equal to 1.0 and that of B to 0.0; (iii) only B is catenated with reactants from the neighborhood, then the probability distribution of event B is equal to 1.0 and that of A to 0.0. Thus, $A \oplus B$ is a function that associates either exactly one value with its arguments (i.e. either A or B) or simultaneously both values (i.e. A and B).

This simplified version of the automaton A is exemplified in fig. 7.1.

Evidently, reducing the complicated dynamics of Belousov-Zhabotinsky reaction to conventional logical proofs is a task that cannot be solved in easy way differently from simulating within massive-parallel proofs.

Figure 7.1. The evolution of a reversible proof-theoretic cellular automaton A with the Moor neighborhood in the 2-dimensional space for the Belousov-Zhabotinsky reaction. This automaton simulates the circular feedback $Ce^{3+} \to Ce^{4+} \to Ce^{3+} \to \ldots$ (more precisely temporal oscillations in a well-stirred solution): Ce^{3+} is colorless and Ce^{4+} is yellow. The initial configuration of A - cells described in (I) occurs in the same form at the further steps and the cycle repeats several times. For entailing (I) $\to$ (II) we have just used inference rule (1) (row 2, column 2) and inference rule (3) (row 1, column 1), for entailing (II) $\to$ (III) inference rules (2), (3) and (8) (row 2, column 2), for entailing (III) $\to$ (IV) inference rule (4) (row 1, column 2) and inference rules (4), (6), (8) and (9) (row 2, column 2), for entailing (IV) $\to$ (V) inference rules (3) and (6) (row 1, column 2) inference rules (1), (3), (5), (6), (7), (8) and (9) (row 2, column 2), for entailing (V) $\to$ (VI) inference rules (4), (6) (row 1, column 1), inference rules (4), (6), (10) (row 1, column 2), inference rules (2), (3), (4), (6), (7), (9), (10), (11) (row 2, column 2).

(I) Initial configuration, $t = 0$

$HBrO_2$	BrO_3^-	H^+
$BrCH(COOH)_2$	Ce^{3+}	H_2O
Br^-	$CH_2(COOH)_2$	BrO_3^-

$\Downarrow$

(II) $t = 1$

$HOBr \oplus BrO_3^- \oplus H^+$	BrO_3^-	H^+
$BrCH(COOH)_2$	$Ce^{4+} \oplus HBrO_2 \oplus H_2O$	H_2O
Br^-	$CH_2(COOH)_2$	BrO_3^-

$$\Downarrow$$

(III) $t = 2$

$HOBr \oplus BrO_3^- \oplus H^+$	BrO_3^-	H^+
$BrCH(COOH)_2$	$Br^- \oplus Ce^{3+} \oplus HCOOH \oplus CO_2 \oplus H^+$, $HOBr \oplus BrO_3^- \oplus H^+$	H_2O
Br^-	$CH_2(COOH)_2$	BrO_3^-

$$\Downarrow$$

(IV) $t = 3$

$HOBr \oplus BrO_3^- \oplus H^+$	$HOBr \oplus HBrO_2$	H^+
$BrCH(COOH)_2$	$Br^- \oplus Ce^{3+} \oplus HCOOH \oplus CO_2 \oplus H^+$, $Br_2 \oplus H_2O$, $HOBr \oplus HBrO_2$	H_2O
Br^-	$CH_2(COOH)_2$	BrO_3^-

$$\Downarrow$$

(V) $t = 4$

$HOBr \oplus BrO_3^- \oplus H^+$	$Br_2 \oplus H_2O$, $HOBr \oplus BrO_3^- \oplus H^+$	H^+
$BrCH(COOH)_2$	$Ce^{4+} \oplus HBrO_2 \oplus H_2O$, $Br_2 \oplus H_2O$, $HOBr \oplus BrO_3^- \oplus H^+$, $HOBr$, $Br^- \oplus H^+ \oplus BrCH(COOH)_2$	H_2O
Br^-	$CH_2(COOH)_2$	BrO_3^-

$$\Downarrow$$

(VI) $t = 5$

$Br_2 \oplus H_2O$, $HOBr \oplus HBrO_2$	$Br_2 \oplus H_2O$, $HOBr \oplus HBrO_2$	H^+
$BrCH(COOH)_2$	$Br_2 \oplus H_2O$, $Br^- \oplus Ce^{3+} \oplus HCOOH \oplus CO_2 \oplus H^+$, $HOBr \oplus BrO_3^- \oplus H^+$, $HOBr \oplus HBrO_2$, $Br^- \oplus H^+ \oplus BrCH(COOH)_2$	H_2O
Br^-	$CH_2(COOH)_2$	BrO_3^-

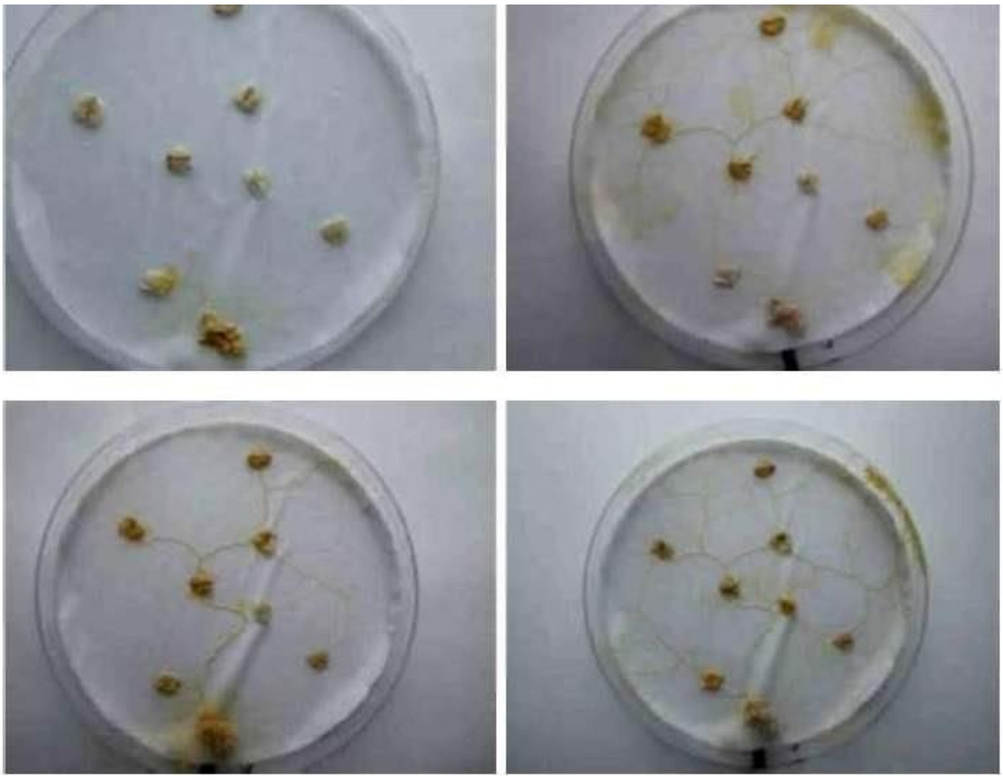
$$\Downarrow$$

…

1.8. The proof-theoretic cellular automaton for dynamics of plasmodium of *Physarum polycephalum*

The dynamics of plasmodium of *Physarum polycephalum* could be regarded as another simple example of the natural proof-theoretic automata. The point is that when the plasmodium is cultivated on a nutrient-rich substrate (agar gel containing crushed oat flakes) it exhibits uniform circular growth similar to the excitation waves in the excitable Belousov-Zhabotinsky medium (Fig. 8.1). If the growth substrate lacks nutrients, e.g. the plasmodium is cultivated on a non-nutrient and repellent containing gel, a wet filter paper or even glass surface localizations emerge and branching patterns become clearly visible (Fig. 8.2, 8.3).

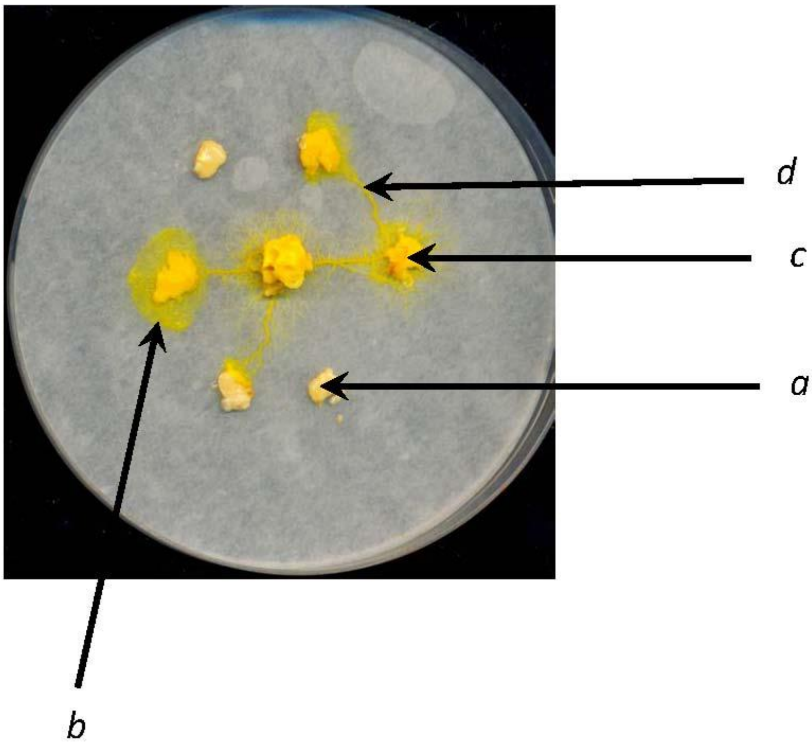
Figure 8.1. An example of computational process in *Physarum* machine. Photographs are taken with time lapse circa 24 hours. *Courtesy of Andy Adamatzky*.



The plasmodium continues its spreading, reconfiguration and development as long as there are enough nutrients. When the supply of nutrients is over, the plasmodium either switches to fructification state (if level of illumination is high enough), when sporangia are produced, or forms sclerotium (encapsulates itself in hard membrane), if in darkness.

The pseudopodium propagates in a manner analogous to the formation of wave-fragments in sub-excitable Belousov-Zhabotinsky systems. Starting in the initial conditions the plasmodium exhibits foraging behavior, searching for sources of nutrients (Fig. 8.1). When such sources are located and taken over, the plasmodium forms characteristic veins of protoplasm, which contracts periodically. Belousov-Zhabotinsky reaction and plasmodium are light-sensitive, which gives us the means to program them. *Physarum* exhibits articulated negative phototaxis, Belousov-Zhabotinsky reaction is inhibited by light. Therefore, by using masks of illumination one can control the dynamics of localizations in these media. Light-sensitivity of Plasmodium has been already explored in the design of robotics controllers (Adamatzky 2007, Adamatzky, Jones 2009).

Figure 8.2. Basic components of *Physarum* and its environment: (a) oat flake, (b) propagating pseudopodium, plasmodium's wave-fragment, (c) oat flake colonized by plasmodium, (d) protoplasmic tube. *Courtesy of Andy Adamatzky*.

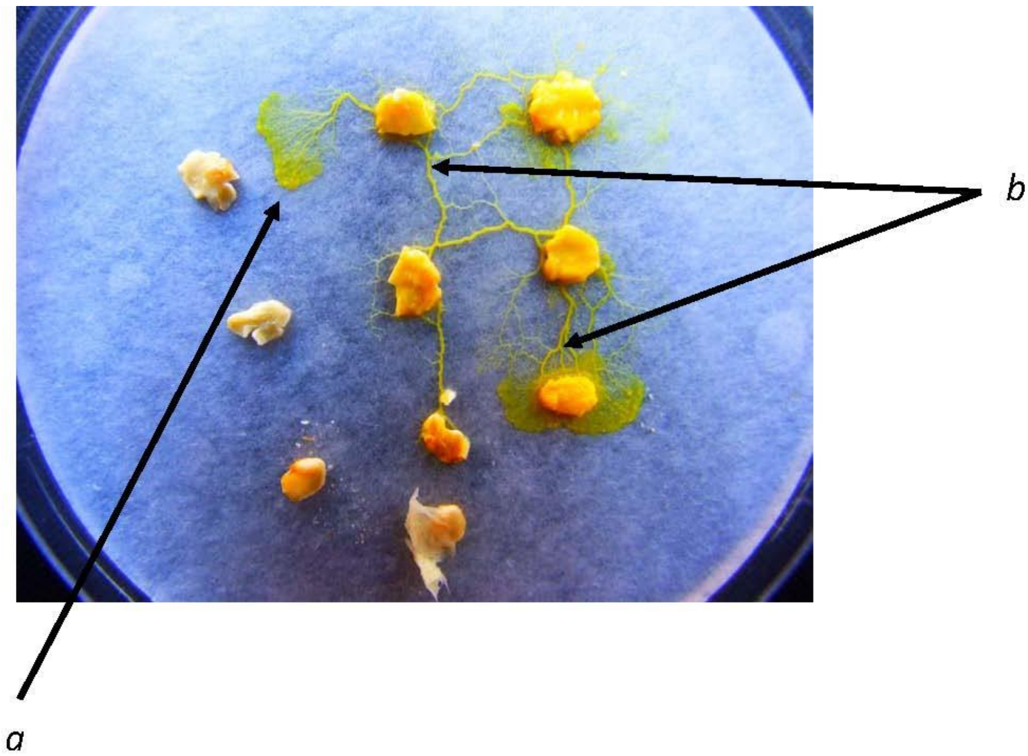


Experiments with *Physarum polycephalum* were carried out by Prof. Adamatzky as follows. The plasmodia of *Physarum polycephalum* were cultured on wet paper towels, fed with oat flakes, and moistened regularly. We subcultured the plasmodium every 5 – 7 days.

Experiments were performed in standard Petri dishes, 9 cm in diameter. Depending on particular experiments we used 2% agar gel or moisten filter paper, nutrient-poor substrates, and 2% oatmeal agar, nutrient-rich substrate (Sigma-Aldrich). All experiments were conducted in a room with diffusive light of 3 – 5 cd/m, 22^0C temperature. In each experiment an oat flake colonized by the plasmodium was placed on a substrate in a Petri dish, and few intact oat flakes distributed on the substrate. The intact oat flakes acted as source of nutrients, attractants for the plasmodium. Petri dishes with plasmodium were scanned on a standard HP scanner. The only editing done to scanned images is color enhancement: increase of saturation and contrast.

Repellents were implemented with illumination domains using blue electroluminescent sheets, see details in *Adamatzky 2010*. Masks were prepared from black plastic, namely the triangle was cut in the plastic, when this mask was placed on top of the electro-luminescent sheet, the light was passing only through the cuts (Adamatzky 2010).

Figure 8.3. Snapshot of experimental dish with propagating plasmodium, new activated propagating zone (a) and sites of branching pseudopodia, junctions of protoplasmic tubes (b) are shown by arrows. *Courtesy of Andy Adamatzky*.



Results of experiments may be described in terms of proof-theoretic cellular automata. Let us assume that its set of states consists of the entities from the following sets (Fig. 8.2).

- The set of *neutral zones*, $\{Z_1, Z_2, \ldots\}$, where nothing goes.
- The set of *growing pseudopodia*, $\{P_1, P_2, \ldots\}$, localized in *active zones* (see an example on Fig. 8.2a). On a nutrient-rich substrate plasmodium propagates as a typical circular, target wave, while on the nutrient-poor substrates localized wave-fragments are formed.
- The set of *attractants* $\{A_1, A_2, \ldots\}$, they are sources of nutrients, on which the plasmodium feeds. It is still subject of discussion how exactly plasmodium feels presence of attracts, indeed diffusion of some kind is involved. Based on previous experiments by Prof. Adamatzky we can assume that if the whole experimental area is about $8-10 \approx cm$ in diameter then the plasmodium can locate and colonize nearby sources of nutrients.
- The set of *repellents* $\{R_1, R_2, \ldots\}$. Plasmodium of *Physarum* avoids light. Thus, domains of high illumination are repellents such that each repellent R is characterized by its position and intensity of illumination, or force of repelling.
- The set of *protoplasmic tubes* $\{C_1, C_2, \ldots\}$. Typically, plasmodium spans sources of nu-

trients with protoplasmic tubes/veins (Fig. 8.2). The plasmodium builds a planar graph, where nodes are sources of nutrients, e.g. oat flakes, and edges are protoplasmic tubes.

Hence, the set of states in the proof-theoretic cellular automaton for dynamics of plasmodium of *Physarum polycephalum* is equal to:

$$\{Z_1,\ Z_2,\ldots\}\cup\{P_1,P_2,\ldots\}\cup\{A_1,\ A_2,\ldots\}\cup\{R_1,R_2,\ldots\}\cup\{C_1,C_2,\ldots\}.$$

The proof-theoretic simulation of *Physarum polycephalum* is defined as follows:

Definition 8.11. *Consider a propositional language* $\mathcal{L}$ *with the only binary operation* $\oplus$*, it is built in the standard way over the set of variables:*

$$S=\{Z_1,\ Z_2,\ldots,P_1,P_2,\ldots,A_1,\ A_2,\ldots,R_1,R_2,\ldots,C_1,C_2,\ldots\}.$$

Let S *be the set of states of proof-theoretic cellular automaton* $\mathcal{A}$*. The inference rule of the automaton is presented by the conjunction of singular inference rules (1) – (7):*

$$(1)\wedge(2)\wedge(3)\wedge(4)\wedge(5)\wedge(6)\wedge(7).$$

The operation $\oplus$ has the following meaning: $A\oplus B$ defines a probability distribution of events A and B in cells. The operation $\oplus$ is idempotent, commutative and associative. Then $\mathcal{A}$ simulates the dynamics of Plasmodium of *Physarum polycephalum*.

Definition 8.2. *Let* $p,s_i,s_{i+1}\in\{Z_1,\ Z_2,\ldots,P_1,P_2,\ldots,A_1,\ A_2,\ldots,R_1,R_2,\ldots,C_1,C_2,\ldots\}$.
A state p *is called a premise for deducing* s_{i+1} *from* s_i *by inference rules (1) – (7) iff*

- p is s_i or,
- in a neighbor cell we find out an expression of the form $p\oplus X$, where X is a propositional metavariable, i.e. it runs over either the empty set or the set of states closed under the operation $\oplus$. Thus, we assume that each premise should occur in a separate cell. This means that if we find out an expression $p_i\oplus p_j\oplus X$ in a neighbor cell and both p_i and p_j are needed for deducing, whereas p_i, p_j do not occur in other neighbor cells, then p_i, p_j could not be considered as premises. This restriction is just for rule (7).

$$x^{t+1}(z)=\begin{cases}X\ if\ \ x^t(z)=X\oplus A_i\ \ and\ \ X=Y\oplus P_j\\[2ex] x^t(z), otherwise\end{cases}\tag{1}$$

$$x^{t+1}(z) = \begin{cases} X \oplus P_j \ if \ x^t(z) = X \oplus C_n \oplus A_i \\ \\ x^t(z), otherwise. \end{cases} \tag{2}$$

$$x^{t+1}(z) = \begin{cases} X \oplus P_j \ if \ x^t(z) = X \oplus P_j \oplus A_i \\ \\ x^t(z), otherwise. \end{cases} \tag{3}$$

$$x^{t+1}(z) = \begin{cases} X \oplus P_i \vee X \oplus P_j \ if \ x^t(z) = X \oplus P_i \oplus P_j \\ \\ x^t(z), otherwise. \end{cases} \tag{4}$$

$$x^{t+1}(z) = \begin{cases} X \oplus C_i \vee X \oplus C_j \ if \ x^t(z) = X \oplus C_i \oplus C_j \\ \\ x^t(z), otherwise. \end{cases} \tag{5}$$

$$x^{t+1}(z) = \begin{cases} (X \oplus C_i, & if \ x^t(z) = X \ and \\ & premises\, R_i \notin (z+N), \\ & premises\, C_j, A_i \in (z+N); \\ \\ x^t(z), & otherwise. \end{cases} \tag{6}$$

$$x^{t+1}(z) = \begin{cases} X \oplus C_i, & if \ x^t(z) = X \ and \\ & premises\, R_i \notin (z+N), \\ & premises\, P_j, A_i \in (z+N); \\ \\ x^t(z), & otherwise. \end{cases} \tag{7}$$

Figure 8.4. The evolution of a proof-theoretic cellular automaton A with the Moor neighborhood in the 2-dimensional space for *Physarum polycephalum*.

(I) Initial configuration, $t=0$

A_{11}	Z_{12}	A_{31}
R_{12}	Z_{22}	Z_{32}
R_{13}	Z_{23}	P_{33}

$\Downarrow$

(II) $t=1$

A_{11}	Z_{12}	A_{31}
R_{12}	Z_{22}	$Z_{32} \oplus C_{32}$
R_{13}	Z_{23}	P_{33}

$\Downarrow$

(III) $t=2$

A_{11}	Z_{12}	$A_{31} \oplus C_{31}$
R_{12}	Z_{22}	$Z_{32} \oplus C_{32}$
R_{13}	Z_{23}	P_{33}

$\Downarrow$

(IV) $t=3$

A_{11}	Z_{12}	P_{31}
R_{12}	Z_{22}	$Z_{32} \oplus C_{32}$
R_{13}	Z_{23}	P_{33}

1.9. Unconventional computing as novel paradigm in natural sciences

In natural sciences, there are many theoretical terms which as it seems to us allow to understand reality better, but actually they replace true reality by images made up. For example, in understanding the Belousov-Zhabotinsky reaction the following theoretical terms of chemistry are used: matter, chemical element, chemical reaction, autowave, valence etc. In understanding the dynamics of plasmodium of *Physarum polycephalum* the following theoretical terms of biology are assumed: contiguous living system, cell, protoplasm, physiology, morphology etc.

If we share positions of unconventional computing and agree that any natural process is a computation and nature as a whole is a computer, theoretical terms of physics, chemistry and biology will lose a sense. We can eliminate them from the language of scientific theories.

Theoretical terms cannot be verified, i.e. they are not reduced to observation terms. However, within the limits of unconventional computing we can substitute logical terms simulating the process of computation for theoretical terms. Logical terms do not require already verification. They form only the structure and algorithmization of observation terms.

Let us suppose that we created the universal language L_{U} of *unconventional computing*, i.e. the universal logical language that allows us to consider any natural process as computation process. Then the language L_{U}, first, eliminates all theoretical terms of natural sciences, replacing them by logical terms of massively parallel proof theory and, second, instead of huge number of scientific theories of physics, chemistry, biology we obtain an opportunity to have the universal logical theory that simulates the Universe.

Let us show, how we could eliminate theoretical terms in the language L_{U}. Take *n* theoretical terms: $T_1, T_2, \ldots, T_n$, linked to observation terms by correspondence rules of an appropriate theory and *m* observation terms, i.e. their references are found out immediately: $O_1, O_2, \ldots, O_m$. The complete theoretical statement is a union consisting of T - and O - terms:

$$\langle T_1, T_2, \ldots, T_n; O_1, O_2, \ldots, O_m \rangle.$$

This formula can be replaced by another, identical to the first in the way that all theoretical terms are replaced by logical terms $U_1, U_2, \ldots, U_k$ which describe the process of computation of an appropriate automata with features $\langle O_1, O_2, \ldots, O_m \rangle$:

$$\langle U_1, U_2, \ldots, U_k; O_1, O_2, \ldots, O_m \rangle.$$

In such a statement, empirical values of theoretical terms are positioned, i.e. they obtain references. These references are expressed by observation terms. Such a representation of theoretical expressions has a number of advantages, e.g. sophisticated theoretical terms of natural sciences could find the classes of references. For example, it is obvious that the statements describing Earth's rotation have the set {the Earth} as a class of references. However, it is not clearly, which set of physical bodies realises the statements describing the concept ‘electron’.

In a proposition of language L_{U} the sophisticated expression setting theoretical terms is transformed into an expression in which there are only observation terms as well as terms of anti-Platonic (a posteriori) logic and mathematics, therefore the concept ‘electron’ may be eliminated from theory expressions.

In our simulation of Belousov-Zhabotinsky reaction as observation terms the following states of an appropriate cellular automaton were used {Ce^{3+}, $HBrO_2$, BrO_3^-, H^+, Ce^{4+}, H_2O, $BrCH(COOH)_2$, Br^-, $HCOOH$, CO_2, $HOBr$, Br_2, $CH_2(COOH)_2$}, and in the simulation of the dynamics of Plasmodium of *Physarum polycephalum* the following states: $\{Z_1, Z_2, \ldots, P_1, P_2, \ldots, A_1, A_2, \ldots, R_1, R_2, \ldots, C_1, C_2, \ldots\}$.

The theoretical terms needed for understanding the Belousov-Zhabotinsky reaction were replaced by inference rules of definition 7.1, and theoretical terms needed for understanding the dynamics of plasmodium of *Physarum polycephalum* by inference rules of definition 8.1.

References

[1] Adamatzky, A., Jones, J., *Programmable reconfiguration of Physarum machines,* Natural Computation, accepted, arXiv:0901.4556, 2011, 2009.

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

[3] Adamatzky, A., *Physarum machine: implementation of a Kolmogorov-Uspensky machine on a biological substrate,* Parallel Processing Letters, 17, pp. 455–467, 2007.

[4] Adamatzky, A., *Physarum machines: encapsulating reaction-diffusion to compute spanning tree,* Naturwisseschaften, 94, pp. 975–980, 2007.

[5] Adamatzky, A., *Steering Plasmodium with light: Dynamical programming of Physarum machine,* New Mathematics and Natural Computation, 2011.

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

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

[8] Balaguer, Mark, *Platonism and Anti-Platonism in Mathematics,* Oxford, Oxford University Press, 1998.

[9] Bernays, P., *Sur le platonisme dans les mathematiques. L'enseignement mathematique,* 34, pp. 52 – 69, 1935.

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

[11] Brotherston, J., *Cyclic proofs for first-order logic with inductive definitions,* [in:] Beckert, B. (ed.), *TABLEAUX 2005*, vol. 3702 of LNAI, Springer-Verlag, pp. 78–92, 2005.

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

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

[14] Carnap, Rudolf, *The Meaning and Necessity: A Study in Semantics and Modal Logic*, University of Chicago Press, 1988.
[15] Frege, Gottlob, *Foundations of Arithmetic*, Oxford, Blackwell, Transl. by Austin, J.L., 1953.
[16] Frege, Gottlob, *Der Gedanke. Eine logische Untersuchung,* [in:] *Beiträge zur Philosophie des deutschen Idealismus* 2, pp. 58–77, 1918/19.
[17] Frege, Gottlob, *On Sense and Reference,* translated by Max Black, Über Sinn und Bedeutung, 1892.
[18] Hilbert, David, *Mathematical problems*, [in:] Ewald, W. (ed.), *From Kant to Hilbert*, Oxford, Oxford University Press, vol. 2, pp. 1096–1105, 1996.
[19] 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, pp. 214–217, 1984.
[20] Kant, I., *Critique of Pure Reason, Cambridge,* Cambridge University Press, 1999.
[21] Khrennikov, A., *Ubiquitous quantum structure: from psychology to finances,* Springer, Berlin-Heidelberg-New York, 2010.
[22] Khrennikov, A., Schumann, A., *Physics Beyond The Set-Theoretic Axiom of Foundation,* [in:] *AIP Conf. Proc*, March 10, vol. 1101, pp. 374–380, 2009.
[23] Khrennikov, A., Schumann, A., *p-Adic Physics, Non-well-Founded Reality and Unconventional Computing, p-Adic Numbers, Ultrametric Analysis, and Applications,* 1(4), pp. 297–306, 2009.
[24] Kripke, S., *Naming and Necessity*. Cambridge, Mass., Harvard University Press, 1980.
[25] Nakagaki, T., Yamada, H., Toth, A., *Path finding by tube morphogenesis in an amoeboid organism,* Biophysical Chemistry, 92, pp. 47–52, 2001.
[26] Nakagakia, T., Yamada, H., Ueda, T., *Interaction between cell shape and contraction pattern in the Physarum plasmodium,* Biophysical Chemistry, 84, pp. 195–204, 2000.
[27] Balaguer, Marc, *Platonism and Anti-Platonism in Mathematics,* Oxford University Press, 1998.
[28] Palmquis, Stephen, *Kant's system of perspectives,* University Press of America, 1993.
[29] Prajer, M., Fleury, A., Laurent, M., *Dynamics of calcium regulation in Paramecium and possible morphogenetic implication*, *Journal Cell Sci*., 110, pp. 529–535, 1997.
[30] Restall, Greg, *Just what Is full-blooded platonism?,* Philosophia Mathematica, 11(1), pp. 82–91, 2003.
[31] Santocanale, L., *A calculus of circular proofs and its categorical semantics,* [in:] Nielsen, M., Engberg, U. (ed.), *Proc. of FoSSaCS 2002*, Grenoble, Springer-Verlag LNCS 2303, April, pp. 357–371, 2002.
[32] Schumann, A., *Non-well-founded probabilities on streams,* [in:] Dubois, D. (et al., ed.), *Soft Methods for Handling Variability and Imprecision*, Advances in Soft Computing 48, pp. 59–65, 2008.
[33] Schumann, A., *Towards Theory of Massive-Parallel Proofs. Cellular Automata Approach,* Bulletin of the Section of Logic, 39/3-4, 2010.
[34] Schumann, A., Adamatzky, A., *Physarum Spatial Logic*, New Mathematics and Natural Computation, 7 (3), pp. 483 – 498, 2011.

[35] Schumann, A., Adamatzky, A., *Towards Semantical Model of Reaction-Diffusion Computing*, Kybernetes, 38 (9), pp. 1518 – 1531, 2009.
[36] Schumann, A., *Qal Wa-Homer and the Theory of Massive-Parallel Proofs*, History and Philosophy of Logic, 32(1), pp. 71 – 83, 2011.

Originally published: Schumann, A., Adamatzky, A., *Logical Modelling of Physarum Polycephalum*, Analele Universitatii de Vest, Timisoara, Seria Matematica Informatica XLVIII, 3, arXiv:1105.4060, pp. 175-190, 2010.

II
Logical Modelling of *Physarum Polycephalum*

Andrew Schumann, Andrew Adamatzky

Abstract

In the paper, we proposed a novel model of unconventional computing where a structural part of computation is presented by dynamics of plasmodium of *Physarum polycephalum*, a large single cell. We sketch a new logical approach combining conventional logic with process calculus to demonstrate how to employ formal methods in design of unconventional computing media presented by *Physarum polycephalum*.

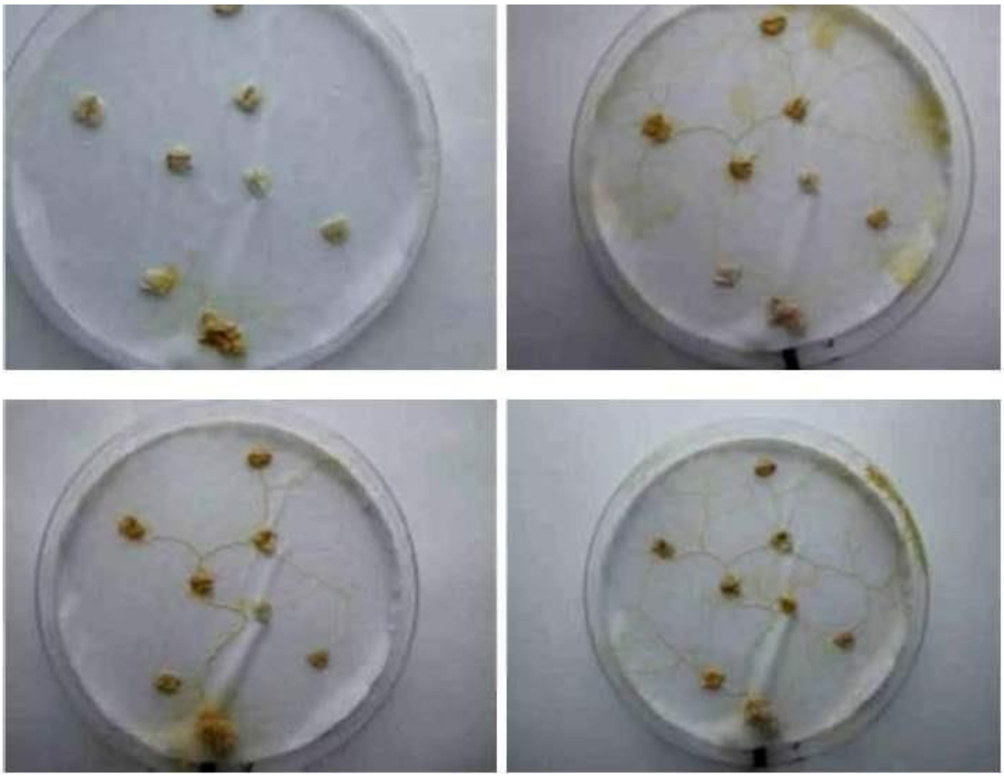
Introduction

In the paper, we are demonstrating how to design unconventional computing media taking into account the problem that in unconventional computing media both structural parts and computing data of computers are variable ones, in particular in reaction-diffusion processors [1] we are dealing with, both the data and the results of the computation are encoded as concentration profiles of the reagents (on the contrary, in conventional models of computation one makes a distinction between the structural part of a computer, which is fixed, and the data which are variable and on which the computer operates). As a result, while in conventional models circuits are fixed, in unconventional computing media circuits could be set up just as dynamically variable ones within spatio-temporal logic. Solving this task allows us to build up nature-inspired computer models and to consider biological and physical systems as computational models.

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 [4] as an example of how a chemical paradigm of the interactions between molecules can be utilized in concurrent computations (in algebraic process calculi). We are considering another abstract machine of reaction-diffusion computing exemplified by dynamics of plasmodium of *Physarum polycephalum*. This machine is constructed by using process algebra, too. Using it, we are studying a possibility of logical representation of the computation in reaction-diffusion systems. Using notions of space-time trajectories of local domains of a reaction-diffusion medium we will could the spatial logic of trajectories, where well-formed formulas and their truth-values are defined in the unconventional way.

Experimental studies and designs of reaction-diffusion computers could be traced back to the pioneer discovery of L. Kuhnert (1986). He demonstrated that some very basic image transformations can be implemented in the light-sensitive Belousov-Zhabotinsky system. The ideas by L. Kuhnert., K.L. Agladze, I. Krinsky (1989) on image and planar shape transformations in two-dimensional excitable chemical media were further developed and modified by N.G. Rambidi (1998). At that time, during the mid and late nineties, a range of chemical logical gates were experimentally built in the Showalter and Yoshikawa laboratories. The first chemical laboratory prototypes of precipitating chemical processors for computational geometry were developed by A. Adamatzky (1996). He also designed and studied a range of hexagonal cellular-automaton models of reaction-diffusion excitable chemical systems. Now reaction-diffusion computing is an extremely wide area of researches towards unconventional computing.

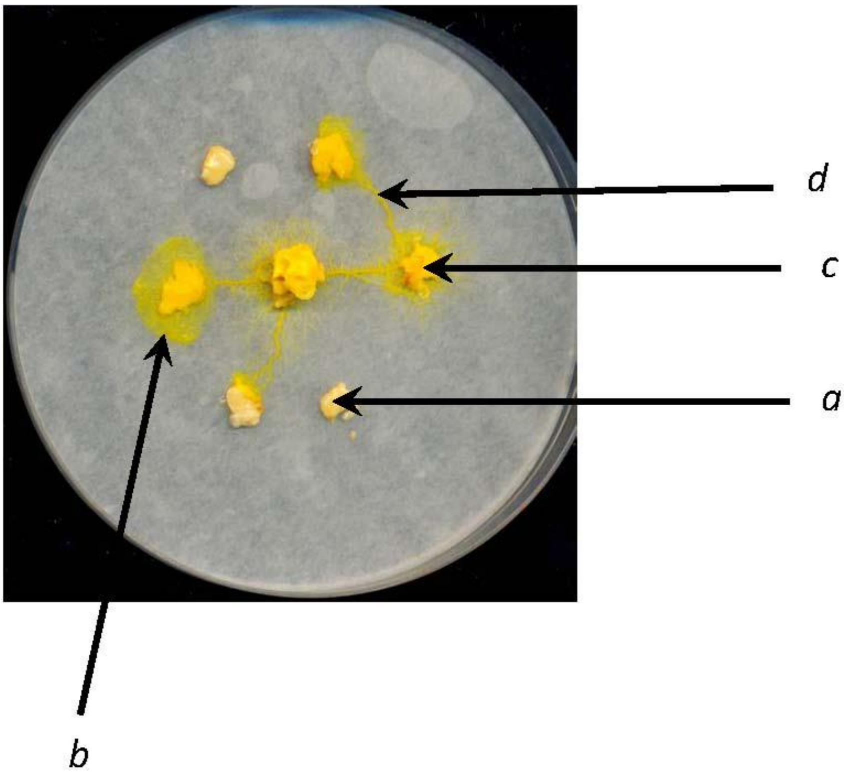
Figure 1. An example of computational process in *Physarum* machine. Photographs are taken with time lapse circa 24 hours.



The dynamics of plasmodium of *Physarum polycephalum* could be regarded as one of the natural reaction-diffusion computers. The point is that when the plasmodium is cultivated on a nutrient-rich substrate (agar gel containing crushed oat flakes) it exhibits uniform circular growth similar to the excitation waves in the excitable Belousov-Zhabotinsky medium (Fig. 1). If the growth substrate lacks nutrients, e.g. the plasmodium is cultivated on a non-nutrient and repellent containing gel, a wet filter paper or even glass surface localizations emerge and branching patterns become clearly visible (Fig. 2, 3).

The plasmodium continues its spreading, reconfiguration and development till there are enough nutrients. When the supply of nutrients is over, the plasmodium either switches to fructification state (if level of illumination is high enough), when sporangia are produced, or forms sclerotium (encapsulates itself in hard membrane), if in darkness.

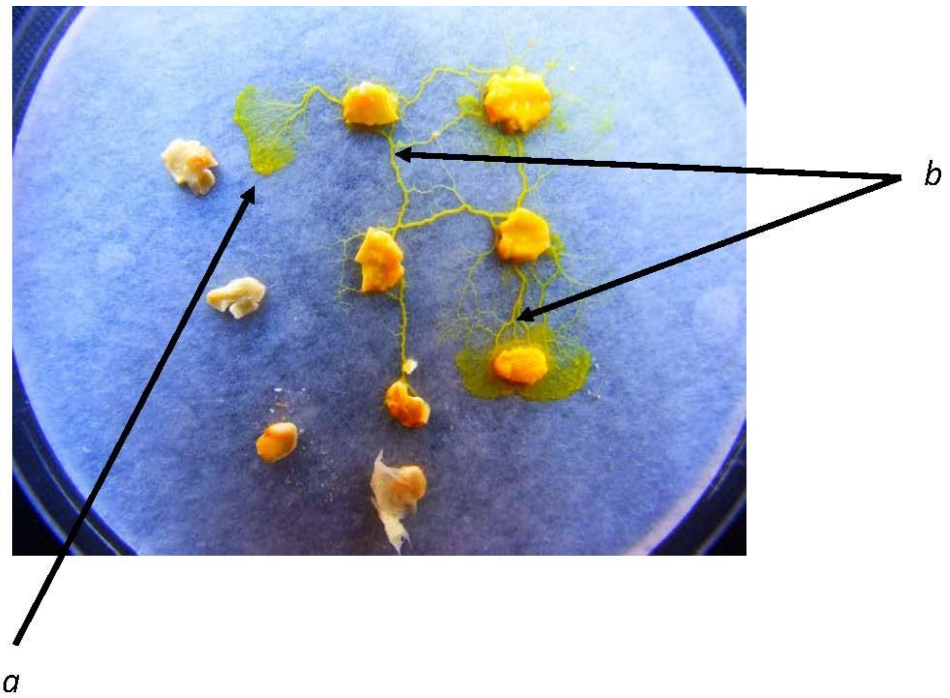
Figure 2. Basic components of *Physarum* and its environment: (a) oat flake, (b) propagating pseudopodium, plasmodium's wave-fragment, (c) oat flake colonized by plasmodium, (d) protoplasmic tube.



The pseudopodium propagates in a manner analogous to the formation of wave-fragments in sub-excitable Belousov-Zhabotinsky systems. Starting in the initial conditions the plasmodium exhibits foraging behavior, searching for sources of nutrients (Fig. 1). When such sources are located and taken over, the plasmodium forms characteristic veins of protoplasm, which contracts periodically. Belousov-Zhabotinsky reaction and plasmodium are light-sensitive, which gives us the means to program them. *Physarum* exhibits articulated negative phototaxis, Belousov-Zhabotinsky reaction is inhibited by light. Therefore, by using masks of illumination one can control dynamics of localizations in these media: change a signal's trajectory or even stop a signal's propagation, amplify the signal, generate trains of signals. Light-sensitive of plasmodium has been already explored in design of robotics controllers [2], [3].

Despite numerous experimental implementations of *Physarum* computers there is a lack of formalization of the plasmodium's behavior abstract enough to infer high-level principle of information transmission by the plasmodium and accurate enough to reflect peculiarities of the *Physarum* foraging behavior. In the paper we are trying to fill the gap and offer interpretation of *Physarum* behavior in a framework of process calculi. In the paper we are trying to define *Physarum* machine as a process calculus built on an unconventional interpretation of logical connectives.

Figure 3. Snapshot of experimental dish with propagating plasmodium, new activated propagating zone (a) and sites of branching pseudopodia, junctions of protoplasmic tubes (b) are shown by arrows.



2.1. A logical approach to analyzing *Physarum* machine

Let us set up the problem how to define logical connectives in *Physarum* machine. The matter is that both structural parts and computing data of *Physarum* computers are variable.

Physarum machine may be viewed as a labelled transition system, which consists of a collection of states $\mathsf{L} = \{p_{ij}\}_{i=\overline{1,N},\, j=\overline{1,K}}$ and a collection T of transitions (processes, actions) over them. Assume $\mathsf{T} : \mathsf{L} \mapsto \mathsf{P}(\mathsf{L})$, where $\mathsf{P}(\mathsf{L}) = \{T : T \subseteq \mathsf{L}\}$. This means that $\mathsf{T}(p)$ consists of all states that a reachable from p. The transition system is understood as a triple

$$\langle \mathsf{L}, \mathsf{T}, \rightarrow \rangle,$$

where $\rightarrow \subseteq \mathsf{L} \times \mathsf{T} \times \mathsf{L}$ is a transition relation that models how a state $p \in \mathsf{L}$ can evolve into another state $p' \in \mathsf{L}$ due to an interaction $\sigma \in \mathsf{T}$. Usually, $\langle p, \sigma, p' \rangle \in \rightarrow$ is denoted by $p \xrightarrow{\sigma} p'$. So, a state p' is reachable from a state p if $p \xrightarrow{\sigma} p'$.

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

$$\rho = p_0 \alpha_1 p_1 \alpha_2 \ldots \alpha_n p_n \; such that \; p_i \xrightarrow{\alpha_{i+1}} p_{i+1} \; for all \; 0 \leq i < n.$$

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

$$\rho = p_0 \alpha_1 p_1 \alpha_2 p_2 \alpha_3 p_3 \ldots such that \; p_i \xrightarrow{\alpha_{i+1}} p_{i+1} \; for all \; 0 \leq i.$$

The word $\alpha_1 \alpha_2 \cdots$ is denoted by $trace(\rho)$ too.

Definition 2.1.2. *An infinite (resp. finite) trace of state* p *denoted by* $\rho(p)$ *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.

Conventionally, logical connectives are defined in the algebraic way that is broken within transition systems. Therefore, in *Physarum* machine we will distinguish two kinds of logical connectives:

- *logical connectives defined co-algebraically*, these ones are closed to conventional logical connectives: they are “eternal,” because they are defined over traces (i.e. for any future states);
- *logical connectives defined as transitions over states*, these ones differ from conventional logical connectives: they are defined over states, therefore their values could change for some future states.

2.1.1. Logical connectives defined co-algebraically

First, let us consider logical connectives defined co-algebraically, i.e. by coinduction.

An infinite trace of state $\rho(p)$ may be presented as a kind of stream. For a trace $\rho(p)$, we call $\rho_p(0)$ the initial value of $\rho(p)$. We define the *derivative* $\rho_p(0)$ of a trace $\rho(p)$, for all $n \geq 0$, by $\rho_{p'}(n) = \rho_p(n+1)$. For any $n \geq 0$, $\rho_p(n)$ is called the *n*-th state of $\rho(p)$. It can also be expressed in terms of higher-order trajectory derivatives, defined, for all

$k \geq 0$, by $\rho_p^{(0)} = \rho(p)$; $\rho_p^{(k+1)} = (\rho_p^{(k)})'$. In this case the *n*-th state of a trace $\rho(p)$ is given by $\rho_p(n) = \rho_p^{(n)}(0)$. So, the trajectory is understood as an infinite sequence of derivatives: $\rho(p) = \rho_p(0) :: \rho_p(1) :: \rho_p(2) :: \ldots :: \rho_p(n-1) :: \rho_p^{(n)}$, or $\rho(p) = \langle \rho_p(0), \rho_p(1), \rho_p(2), \ldots \rangle$.

A *bisimulation* on the set of traces is a relation R such that, for all $\rho(p)$ and $\rho(q)$, if $\langle \rho(p), \rho(q) \rangle \in R$ then (i) $\rho_p(0) = \rho_q(0)$ (this means that they have the same *initial value*) and (ii) $\langle \rho_{p'}, \rho_{q'} \rangle \in R$ (this means that they have the same *differential equation*).

If there *exists* a bisimulation relation R with $\langle \rho_p, \rho_q \rangle \in R$ then we write $\rho_p : \rho_q$ and say that ρ_p and ρ_q are *bisimilar*. In other words, the *bisimilarity* relation is the greatest bisimulation. In addition, the bisimilarity relation is an equivalence relation.

Theorem 2.1. (Coinduction) 3. *For all* $\rho(p), \rho(q)$, *if there exists a bisimulation relation* R *with* $\langle \rho(p), \rho(q) \rangle \in R$, *then* $\rho(p) = \rho(q)$.

This proof principle is called *coinduction*. It is a systematic way of proving the statement using bisimularity: instead of proving only the single identity $\rho(p) = \rho(q)$, one computes the greatest bisimulation relation R that contains the pair $\langle \rho(p), \rho(q) \rangle$. By coinduction, it follows that $\rho(p) = \rho(q)$ for all pairs $\langle \rho(p), \rho(q) \rangle \in R$.

Now consider logical connectives defined by coinduction over traces. Their syntax is as follows:

Variables: $\mathbf{p} ::= p \mid q \mid r \ldots,$

where *p*, *q*, *r* are states of *Physarum* machine presented as a labelled transition system.

Constants: $\mathbf{c} ::= \mathrm{T} \mid \perp$

where T means the truth (the ideal, universal trace) and $\perp$ means the falsity (the empty, impossible trace).

Formulas: $\varphi, \psi ::= p \mid \mathbf{c} \mid \neg\psi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \supset \psi$

These definitions are coinductive. For instance:

- a variable $\mathbf{p}$ is of the form of a trace $\mathbf{p} = \mathbf{p}(0) :: \mathbf{p}(1) :: \mathbf{p}(2) :: \ldots :: \mathbf{p}(n-1) :: \mathbf{p}^{(n)}$, where $\mathbf{p}(i) \in \{p, q, r, \ldots\}$ for each $i \in \omega$;
- a constant $\mathbf{c}$ is of the form of a trace $\mathbf{c} = \mathbf{c}(0) :: \mathbf{c}(1) :: \mathbf{c}(2) :: \ldots :: \mathbf{c}(n-1) :: \mathbf{c}^{(n)}$, where $\mathbf{c}(i) \in \{\mathrm{T}, \perp\}$ for each $i \in \omega$, a particular case is $[\mathrm{T}] = [\mathrm{T}](0) :: [\mathrm{T}](1) :: [\mathrm{T}](2) :: \ldots :: [\mathrm{T}]^{(n)}$, where $[\mathrm{T}](i) = \mathrm{T}$ for each $i \in \omega$;

- a formula $\neg\varphi$ has the differential equation $(\neg\varphi)' = \neg(\varphi')$ and its initial value is $(\neg\varphi)(0) = \neg\varphi(0)$, this formula will be understood as $\varphi \supset [\perp]$;
- a formula $\varphi \vee \psi$ has the differential equation $(\varphi \vee \psi)' = \varphi' \vee \psi'$ and its initial value is $(\varphi \vee \psi)(0) = \varphi(0) \vee \psi(0)$;
- a formula $\varphi \wedge \psi$ has the differential equation $(\varphi \wedge \psi)' = \varphi' \wedge \psi'$ and its initial value is $(\varphi \wedge \psi)(0) = \varphi(0) \wedge \psi(0)$;
- a formula $\varphi \supset \psi$ has the differential equation $(\varphi \supset \psi)' = \varphi' \supset \psi'$ and its initial value is $(\varphi \supset \psi)(0) = \varphi(0) \supset \psi(0)$.

2.1.2. Logical connectives defined as transitions over states

In analyzing the plasmodium we observe processes of inaction, fusion and choice, which could be interpreted as unconventional (spatial) falsity, conjunction and disjunction respectively, denoted by Nil, $\&$ and $+$. These operations differ from conventional ones, because they cannot have a denotational semantics in the standard way. However, they may be described as special transitions over states of *Physarum* machine:

- inaction (Nil) means that pseudopodia has just stopped to behave,
- fusion ($\&$) means that two pseudopodia come in contact one with another and then merge,
- choice (+) means a competition between two pseudopodia in their bahaviours.

Let us notice that a Boolean algebra may be extended up to the case of the system of logical connectives defined by coinduction [5] (see the previous subsection). However, if we define three basic logical connectives (falsity, conjunction, disjunction) as transitions over states of *Physarum* machine, they will be extremely non-classical and Boolean properties do not hold for them in general case.

2.2. *Physarum* process calculus

Further, let us try to build up a process calculus combining two approaches to logical connectives for describing the dynamics of *Physarum* machine, i.e. we are trying to show that indeed this machine could be presented as a labelled transition system with some logical relations.

Assume that the computational domain Ω is partitioned into computational cells $c_j = 1,\ldots,K$ such that $c_i \cap c_j = \varnothing$, $i \neq j$ and $\bigcup_{j=1}^{K} c_j = \Omega$. Then suppose that in the K cells, there are N active species or growing pseudopodia and the state of species i in cell j is denoted by p_{ij}, $i = 1,\ldots,N$, $j = 1,\ldots,K$. These states are time dependent and they are changed by plasmodium's active zones interacting with each other and affected by attractants or repellents. Plasmodium's active zones interact concurrently and in a parallel manner. Foraging plasmodium can be represented as a set of following abstract entities (Fig. 2).

- The set of *active zones* (growing pseudopodia or actions) $Z = \{a,b,\ldots\}$ (Fig. 2a). On a nutrient-rich substrate, plasmodium propagates as a typical circular, target, wave, while on the nutrient-poor substrates localized wave-fragments are formed. Each action *a* from Z belongs to a state p_{ij}, $i = 1,\ldots,N$, $j = 1,\ldots,K$ of a cell i, which is its current position, and says about a transition (propagation) of a state p_{ij} to another state of the same or another cell. Part of plasmodium feeding on a source of nutrients may not propagate, so its transition is nil, but this part can always start moving.
- The set of *attractants* $\{A_1, A_2,\ldots\}$ are sources of nutrients, on which the plasmodium feeds. It is still subject of discussion how exactly plasmodium feels presence of attracts, indeed diffusion of some kind is involved. Based on our previous experiments we can assume that if the whole experimental area is about $8-10\colon cm$ in diameter then the plasmodium can locate and colonize nearby sources of nutrients. Each attract $A(a)$ is a function from *a* to another action b.
- The set of *repellents* $\{R_1, R_2,\ldots\}$. Plasmodium of *Physarum* avoids light. Thus, domains of high illumination are repellents such that each repellent R is characterized by its position and intensity of illumination, or force of repelling. In other words, each repellent $R(a)$ is a function from *a* to another action b.
- The set of *protoplasmic tubes* $\{C_1, C_2,\ldots\}$. Typically plasmodium spans sources of nutrients with protoplasmic tubes/veins (Fig. 2). The plasmodium builds a planar graph, where nodes are sources of nutrients, e.g. oat flakes, and edges are protoplasmic tubes. $C(a)$ means a diffusion of growing pseudopodia by an action *a*.

Our process calculus contains the following basic operators: Nil (inaction), å (prefix), $|$ (cooperation), $\backslash$ (hiding), & (reaction/fusion), $+$ (choice), a (constant or restriction to a stable state), $A(\cdot)$ (attraction), $R(\cdot)$ (repelling), $C(\cdot)$ (spreading/diffusion). Let $\Lambda = \{a,b,\ldots\}$ be a set of names. With every $a \in \Lambda$ we associate a complementary action $\bar{a}$. Define $L = \{a, \bar{a} : a \in \Lambda\}$, where a is considered as *activator* and as *inhibitor* for a, be the set of labels built on Λ (under this interpretation, $a = \bar{\bar{a}}$). Suppose that an action a communicates with its complement $\bar{a}$ to produce the internal action τ. Define $L_\tau = L \cup \{\tau\}$.

We use the symbols α, β, etc., to range over labels (actions), with $a = \bar{\bar{a}}$, and the symbols P, Q, etc., to range over processes on states p_{ij}, $i = 1,\ldots,N$, $j = 1,\ldots,K$. The processes are given by the syntax:

$$P,Q ::= Nil \;\;|\;\; \alpha\text{å}P \;\;|\;\; A(\alpha)\text{å}P \;\;|\;\; R(\alpha)\text{å}P \;\;|\;\; C(\alpha) \;\;|\;\; (P\,|\,Q) \;\;|\;\; P \setminus Q \;\;|$$
$$P \,\&\, Q \;\;|\;\; P+Q \;\;|\;\; a$$

Each label is a process, but not vice versa. An operational semantics for this syntax is defined as follows:

$$\text{Prefix:} \quad \frac{}{\alpha\text{å}P \xrightarrow{\alpha} P},$$

$$\frac{}{A(\alpha)\text{å}P \xrightarrow{\beta} P} \; (A(\alpha)=\beta),$$

$$\frac{}{R(\alpha)\text{å}P \xrightarrow{\beta} P} \; (R(\alpha)=\beta),$$

(the conclusion states that the process of the form $\alpha\text{å}P$ (resp. $A(\alpha)\text{å}P$ or $R(\alpha)\text{å}P$) may engage in α (resp. $A(\alpha)$ or $R(\alpha)$) and thereafter they behave like P; in the presentations of behaviors as trees, $\alpha\text{å}P$ (resp. $A(\alpha)\text{å}P$ or $R(\alpha)\text{å}P$) is understood as an edge with two nodes: α (resp. $A(\alpha)$ or $R(\alpha)$) and the first action of P),

$$\text{Diffusion:} \quad \frac{P \xrightarrow{\alpha} P'}{P \xrightarrow{\alpha} C(\alpha)} \; (C(\alpha) ::= P'),$$

$$\text{Constant:} \quad \frac{P \xrightarrow{\alpha} P'}{a \xrightarrow{\alpha} P'} \; (a ::= P, a \in L_\tau),$$

$$\text{Choice:} \quad \frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}, \quad \frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'},$$

(these both rules state that a system of the form $P+Q$ saves the transitions of its subsystems P and Q),

$$\text{Cooperation:} \quad \frac{P \xrightarrow{\alpha} P'}{P\,|\,Q \xrightarrow{\alpha} P'\,|\,Q}, \quad \frac{Q \xrightarrow{\alpha} Q'}{P\,|\,Q \xrightarrow{\alpha} P\,|\,Q'},$$

(according to these rules, the cooperation | interleaves the transitions of its subsystems),

$$\frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\bar{\alpha}} Q'}{P\,|\,Q \xrightarrow{\tau} P'\,|\,Q'},$$

(i.e. subsystems may synchronize in the internal action τ on complementary actions α and α),

$$\text{Hiding}: \ \frac{P \xrightarrow{\alpha} P'}{P \setminus Q \xrightarrow{\alpha} P' \setminus Q} \ (\alpha \notin Q, Q \subseteq L),$$

(this rule allows actions not mentioned in Q to be performed by $P \setminus Q$),

$$\text{Fusion}: \ \frac{}{\alpha å P \,\&\, \overline{P} \xrightarrow{\alpha} Nil}$$

(the fusion of complementary processes are to be performed into the inaction),

$$\frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\alpha} P'}{P \,\&\, Q \xrightarrow{\alpha} P'}, \qquad \frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\alpha} P'}{Q \,\&\, P \xrightarrow{\alpha} P'}$$

(this means that if we obtain the same result P' that is produced by the same action α and evaluates from two different processes P and Q, then P' may be obtained by that action α started from the fusion $P \,\&\, Q$ or $Q \,\&\, P$),

$$\frac{P \xrightarrow{\alpha} P'}{P \,\&\, Q \xrightarrow{\alpha} Nil + C(\alpha) + P'}, \qquad \frac{P \xrightarrow{\alpha} P'}{Q \,\&\, P \xrightarrow{\alpha} Nil + C(\alpha) + P'}$$

(these rules state that if the result P' is produced by the action α from the processes P, then a fusion $P \,\&\, Q$ (or $Q \,\&\, P$) is transformed by that same α either into the inaction or diffusion or process P').

These are inference rules for basic operations. The ternary relation $P \xrightarrow{\alpha} P'$ means that the initial action P is capable of engaging in action α and then behaving like P'.

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.
- $\alpha å P$, a process $\alpha \in L$ followed by the process P : P becomes active only after the action α has been performed. An activator $\alpha \in L$ followed by the process P is interpreted as branching pseudopodia into two or more pseudopodia, when the site of branching represents newly formed process $\alpha å P$.

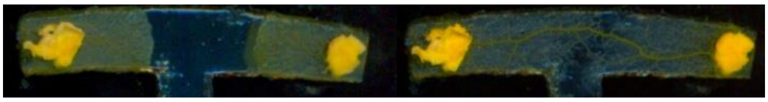
In turn, an inhibitor $\overline{\alpha} \in L$ followed by the process P is annihilating protoplasmic strands forming a process at their intersection.

- $A(\alpha)\to P$ denotes a process that waits for a value α and then continues as P. This means that an attractor A modifies propagation vector of action α towards P. Attractants are sources of nutrients. When such a source is colonized by plasmodium the nutrients are exhausted and attracts ceases to function: $A(\alpha)\to Nil$.
- $R(\alpha)\to P$ denotes a process that waits for a value α and then continues as P. This means that a repellent R modifies propagation vector of action α towards P. Process can be cancelled, or annihilated, by a repellent: $R(\alpha)\to Nil$. This happens when propagating localized pseudopodium α enters the domain of repellent, e.g. illuminated domain, and α does not have a chance to divert or split.
- $C(\alpha)$, a diffusion of activator $\alpha \in L$ is observed in placing sources of nutrients nearby the protoplasmic tubes belonging to α or inactive zone ($\alpha ::= Nil$). More precisely, diffusion generates propagating processes which establish a protoplasm vein (the case of activator α) or annihilate it (when source of nutrients exhausted, the case of inhibitor $\overline{\alpha}$).
- $P \mid Q$, this is a parallel composition (commutative and associative) of actions: P and Q are performed in parallel. The parallel composition may appear in the case, two more food sources are added to either side of the array and then the plasmodium sends two streams outwards to engulf the sources. When the food sources have been engulfed, the plasmodium shifts in position by redistributing its component parts to cover the area created by the addition of the two new processes P and Q that will already behave in parallel.
- Process P can be split, or multiplied, by two sources of attractants $(A_1 A_2)(P)\to P_1 \mid P_2$. Pseudopodium P approaches the site where distance to A_1 is the same as distance to A_2. Then P subdivides itself onto two pseudopodia P_1 and P_2. Each of the pseudopodia travels to its unique source of attractants. Also, process P can be split, or multiplied, by a repellent: $R(P)\to P_1 \mid P_2$. Biophysics of fission with illuminated geometrical shapes is discussed in [5]. The fission happens when a propagating pseudopodium 'hits' a repellent. The part of pseudopodium most affected by the repellent ceases propagating, while two distant parts continue their development. Thus, two separate pseudopodia are formed.
- $P \setminus Q$, this restriction operator allows us to force some of P's actions not to occur; all of the actions in the set $Q \subseteq L$ are prohibited, i.e. the component $P \setminus Q$ behaves as P except that any activities of types within the set Q are hidden, meaning that their type is not visible outside the component upon completion.
- $P \,\&\, Q$, 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. The fusion of P and Q is understood as collision of two active zones P and Q. When they collide they fuse and annihilate, $P \,\&\, Q \to Nil$. Depending on the particular circumstances the new active zone α

(the result of fusing) may become inactive (Nil), transform to protoplasmic tubes ($C(\alpha)$), or remain active and continue propagation in a new direction (the case of prefix å).

When two pseudopodia come in contact one with another, they do usually merge (Fig. 4). Thus by directing processes with attractors we can merge the processes: $A(P_1, P_2)$å$P_1 \& P_2$ (see details in [5]).

Figure 4. Merging of two plasmodium's wave-fronts. Photos are made with interval 9 hours.



- $P+Q$, this is the choice between P and Q; $P+Q$ 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. In *Physarum* calculi, the choice $P+Q$ between processes P and Q sometimes is represented by competition between pseudopodia tubes $C(P)$ and $C(Q)$, i.e. $C(P,Q) = C(P) + C(Q)$. In other words, two processes P and Q can compete with each, during this competition one process 'pulls' protoplasm from another process, thus making this another process inactive. The competition happens via protoplasmic tube.
- a, constants belonging to labels 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 two kinds of logical connectives.

- The group of connectives defined by coinduction. They are derivable from the hiding. Indeed, let 1 be a universal set of active zones, then the following equalities hold:

$$\neg P ::= 1 \setminus P \quad \text{negation}$$

$$P \wedge Q ::= P \setminus (1 \setminus Q) \quad \text{conjunction}$$

$$P \vee Q ::= 1 \setminus ((1 \setminus P) \setminus Q) \quad \text{disjunction}$$

$$P \supset Q ::= 1 \setminus (P \setminus Q) \quad \text{implication}$$

These connectives satisfy all properties of Boolean algebra.

- The group of connectives defined as transitions. It consists of three operations: **inaction**, **fusion** and **choice**. Their basic properties:

$$Nil \setminus P \cong Nil, \tag{1}$$

$$P \,\&\, \overline{P} \cong Nil, \tag{2}$$

$$P \,\&\, P \cong P, \tag{3}$$

$$P \,\&\, Nil \cong Nil, \tag{4}$$

$$(P+Q) \setminus P' \cong P \setminus P' + Q \setminus P', \tag{5}$$

$$(P \,\&\, Q) \setminus P' \cong P \setminus P' \,\&\, Q \setminus P', \tag{6}$$

$$P \,\&\, Q \cong Q \,\&\, P, \tag{7}$$

$$P \,\&\, (Q \,\&\, R) \cong (P \,\&\, Q) \,\&\, R, \tag{8}$$

$$P + P \cong P, \tag{9}$$

$$P + Nil \cong P, \tag{10}$$

$$P + Q \cong Q + P, \tag{11}$$

$$P + (Q + R) \cong (P + Q) + R, \tag{12}$$

$$P \,\&\, (Q + R) \cong (P \,\&\, Q) + (P \,\&\, R), \tag{13}$$

$$P + (Q \,\&\, R) \cong (P + Q) \,\&\, (P + R), \tag{14}$$

where $\cong$ is a congruence relation defined on the set of processes.

Conclusion

In the paper, we have just shown that the behavior of plasmodium of *Physarum polycephalum* could be considered as a kind of process calculus with several logical connectives defined in non-standard way. Thus, the media of *Physarum polycephalum* can be viewed as one of the natural unconventional (reaction-diffusion) computers. Its weakest point is that *the speed of computation is so slow*: each new state of *Physarum* dynamics may be observed just in hours (see Fig. 1).

References

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

[2] Adamatzky, A., *Physarum machines: encapsulating reaction-diffusion to compute spanning tree,* Naturwisseschaften, 94, pp. 975–980, 2007.

[3] Adamatzky, A., *Steering Plasmodium with light: Dynamical programming of Physarum machine,* New Mathematics and Natural Computation, 2009.

[4] Berry, G., Boudol, G., *The chemical abstract machine,* Theor. Comput. Sci., 96, pp. 217–248, 1992.

[5] Schumann, A., *Non-well-founded Probabilities and Coinductive Probability Logic, Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'08)*, IEEE, pp. 54–57, 2008.

[6] Schumann, A., Adamatzky, A., *Towards Semantical Model of Reaction-Diffusion Computing*, Kybernetes, 38(9), pp. 1518–1531, 2009.

Originally published: Schumann, A., *Probabilities on Streams and Reflexive Games*, Operations Research and Decisions, 24 (1), pp. 71-96, 2014.

III
Probabilities on Streams and Reflexive Games

Andrew Schumann

Abstract

In this paper, probability measures on streams (e.g. on hypernumbers and p-adic numbers) are defined. Then it is shown that these probabilities can be used for simulations of reflexive games. In particular, it can be proved that Aumann's Agreement Theorem does not hold on these probabilities. Instead of this theorem, there is a statement that is called Reflexion Disagreement Theorem. Due to this theorem, probabilistic and knowledge conditions can be defined for reflexive games at different reflexion levels up to the infinite level.

Introduction

For the first time probabilities on streams are defined in papers [46; 47]. They are a natural generalization of probabilities on hypernumbers and p-adic numbers. On the theory of p-adic valued probabilities there are surveys and many deatails in [19; 20; 21; 22; 23; 41]. Some basic properties of non-Archimedean (p-adic as well as hypernumber-valued) logical multiple-validity are considered in [41; 43, 44; 45; 48]. Recall that the fundamental work on non-Archimedean systems is [37].

All results of this paper are obtained due to some basic features of streams and coinductive probabilities on them. Streams refer to mathematical objects which cannot be generated as inductive sets. For more details please see [15; 18; 31; 32; 35; 36; 38; 39; 40]. Using these probabilities on streams, the reflexion disagreement theorem (theorem 2, section 3.3) can be readily proved, which contradicts to Aumann's agreement theorem proved on standard real probabilities. It is possible to simulate reflexive games on coinductive probabilities. It is assumed that the reader knows some basic notions of speech act theory (see [50; 51; 52]) such as performative, illocutionary, perlocutionary. In the paper, I propose a formalization of the notion ‘perlocutionary effect’ to coinductively define knowledge operators in reflexive games.

3.1. Why can we reject Aumann's agreement theorem?

Aumann's agreement theorem [3; 4] actually says that two agents acting rationally (according to Bayesian formulas) and with common knowledge of each other's beliefs cannot agree to disagree. More specifically, if two people share common priors, and have common knowledge of each other's current probability assignments (their posteriors for a given event *A* are common knowledge), then they must have equal probability assignments (these posteriors must be equal). It is one of the most important statement of game theory, epistemic logic and so on. For example, according to this statement, each rational player has to behave in the same manner under the same circumstances. Rational players have always a common knowledge, they know all parameters of the game and have to be sure that their opponents know that they know parameters of the game, that they know that they know and so on ad infinitum.

To prove his theorem, Aumann appeals to representing the possibility operator $\mathbf{P}_i(\omega)$ and the common knowledge operator K_i as the least fixed points, i.e. as *inductive sets*. Let us remember their definitions.

Let Ω be a finite set of possible states of the world which are called propositions, *N* be a set of agents, call them $i = 1, ..., N$. Agent *i*'s knowledge structure is a function $\mathbf{P}_i$ which assigns to each $\omega \in \Omega$ a non-empty subset of Ω. $\mathbf{P}_i$ is a partition of Ω: each world ω belongs to exactly one element of each $\mathbf{P}_i$, i.e. Ω is a set of mutually disjoint subsets $\mathbf{P}_i$ whose union is Ω. Then $\mathbf{P}_i(\omega)$ is called *i*'s knowledge state at ω. This means that if the true state is ω, the individual only knows that the true state is in $\mathbf{P}_i(\omega)$. We can interpret $\mathbf{P}_i(\omega)$ probabilistically as follows: $\mathbf{P}_i(\omega) = \{\omega' : P_i(\omega' \mid \omega) > 0\}$. Then all propositions in Ω in any of the *N* partitions form a σ-field A. $\mathbf{P}_i(\omega) \subseteq A$ is interpreted as meaning that at ω agent *i* knows that *A*, i.e. $\omega' \in A$ for all states ω' that *i* considers possible at ω.

For each *i*, the expression below defines a knowledge operator K_i which, applied to any set $A \in \mathsf{A}$, yields the set $K_i A \in \mathsf{A}$ of worlds in which *i* knows *A*:

$$K_i A = \{\omega : \mathbf{P}_i(\omega) \subseteq A\}.$$

The most important property of the knowledge operator is $K_i A \subseteq A$; i.e. if an agent knows an event *A* in state ω (i.e., $\omega \in K_i A$), then *A* is true in state ω (i.e., $\omega \in A$).

We can prove the following statements:

$$K_i \Omega = \Omega; \tag{1}$$

$$K_i(A \cap B) = K_i A \cap K_i B; \qquad (2)$$

$$A \subseteq B \Rightarrow K_i A \subseteq K_i B; \qquad (3)$$

$$K_i A \subseteq A; \qquad (4)$$

$$K_i K_i A = K_i A; \qquad (5)$$

$$\neg K_i \neg K_i A \subseteq K_i A. \qquad (6)$$

The properties of (1) – (6) are considered fundamental for defining knowledge operators in epistemic logic.

Nevertheless, we can define the possibility operator $\mathbf{P}_i(\omega)$ and the common knowledge operator K_i as the greatest fixed points as well, i.e. as *coinductive sets*. In this way we cannot prove Aumann's agreement theorem. Instead of the latter statement we then prove the reflection disagreement theorem as an appropriate negation of Aumann's theorem. While for Aumann's theorem we need the property $\mathbf{P}_i(\omega) = \bigcap\{A : \omega \in K_i A\}$, for its negation we need the property $\mathbf{P}_i(\omega) = \bigcup\{A : \omega \in K_i A\}$. In other words, this new statement can be proven if we change some standard philosophical presuppositions in game theory by the following new assumptions: each rational agent can cheat (disagree in his heart with) other rational agents, each player cannot know everything prior the game, each agent can try to foresee knowledge (beliefs) of his/her opponents and manipulate them, therefore the common knowledge does not mean that an agent cannot disagree and will be completely predictable for all others.

These philosophical presuppositions contradicting to Aumann's ideas were first formulated by Vladimir Lefebvre in his notion 'reflexive games' in 1965 [25; 27; 28]. A game is called reflexive if to choose the action the agent has to model (predict) actions of his/her opponents [33; 34], e.g. (s)he can try to manipulate them or cheat them. The Early Levebvre formulated reflexive games assuming many reflexion levels [28]. On the zero level I ignore beliefs of opponents, on the first level I take into account their beliefs, on the second level I take into account that they try to predict my beliefs, on the third level I foresee their beliefs in which my beliefs are foreseen by them, etc. The game-theoretic mathematics for ideas of the Early Levebvre has been devoloped by D.A. Novikov and A.G. Chkhirtishvili [11; 12; 13; 14; 33; 34]. The reflexion disagreement theorem that will be proved in the next section holds true for their approach, namely if we suppose a possibility of reflexive games on a reflexion level of any natural number. I was inspirated and suggested by the ideas of the Early Levebvre in the same measure as them.

The Later Levebvre tried to simulate decision making in reflexive games by means of Boolean functions [26]. The main disadvantages of approaches encouraged by the Later Lefebvre consist in that reflexive levels are ignored and agents are presented as automata. However, the self-estimation is a variable. Reflexion varies depending on characteristic moods (illocutionary acts) as well as the persuasiveness and emotionality of our interlocutors (perlocutionary effects). In this sense, the dynamism of reflexion quite corresponds to the well-known paradox of Chevalley and Belzung [10] which is formulated as such: an emotional response of the same person in the same situation can be various in different points of time. Thus, the simulation of decision making in reflexive games by means of Boolean functions is too speculative and cannot help in analyzing everyday situations. Within this approach we assume that reactions and evaluations of the same agent remain the same forever. Nevertheless, it is false. This approach can be useful only in explicating some basic features of reflexive management that take place in the given situation (such as the case of Soviet and American ethical patterns [26]).

The ideas of the Early Lefebvre which I try to develop in this paper are very close to ideas of metagame which were proposed by Nigel Howard [17]. According to him, for any game G and any player i there can be a metagame iG, in which player i chooses in knowledge of the choices of all the others. More formally, let $G = \langle S_1, S_2; M_1, M_2\rangle$ be the normal form of the game, where S_1 (resp. S_2) is the set of strategies for player 1 (resp. 2) and M_1 (resp. M_2) is his/her preference function. The set of outcomes is $S = S_1 \times S_2$, i.e. an outcome is an ordered pair $s = \langle s_1, s_2\rangle$. $M_i(s) = \{s'$: is not preferred to s by player $i\}$, $i = 1, 2$. Let $B(S_1)$ (resp. $B(S_2)$) be the set of non-null subsets of S_1 (resp. of S_2) and $K_1 \subseteq B(S_1)$ (resp. $K_2 \subseteq B(S_2)$). Then the *first level metagame* KG is defined as the normal form $KG = \langle X_1, X_2; M'_1, M'_2\rangle$, where $X_1 = \{x_1 : x_1 = \langle f_1, c_1\rangle; c_1 \in K_1; f_1 : K_2 \to c_1\}$, $X_2 = \{x_2 : x_2 = \langle f_2, c_2\rangle; c_2 \in K_2; f_2 : K_1 \to c_2\}$, and for $i = 1, 2$, M'_i satisfies the following property:

$$x' \in M'_i(x) \text{ iff } \beta x' \in M'_i(\beta x),$$

where $\beta(f_1, c_1; f_2, c_2) = (f_1(c_2), f_2(c_1))$.

By induction, we can obtain an n-th-level metagame $K_nK_{(n-1)}...K_1G$. The set of all metagames $K_nK_{(n-1)}...K_1G$ for any natural number n is called the *infinite metagame* based on G. This metagame corresponds to Lefebvre's *reflexive game of infinite level*.

Nigel Howard proposed to use metagames to have possibilities to require *"more of rationality than that each player should optimize given its beliefs about the others' choices"* [17]. Now (s)he should be able *"to know the others' choices, and know how the other would choose to react to such knowledge, and know each others' reactions to such reactions, and so on"* [17]. These ideas are almost the same as ideas introduced in reflexive games [11; 12; 13; 14; 33; 34]. The difference consists in other ways of defining preference functions.

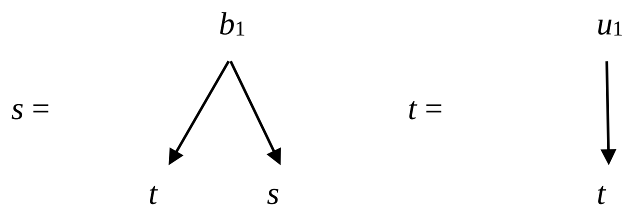
The reflection disagreement theorem holds true for the infinite metagame in Howard's meaning as well as for the reflexive game of infinite level in Lefebvre's meaning. This theorem shows *limits in infinite mutual predictions of others' knowledge*.

The mathematical meaning of the reflexion disagreement theorem is that we cannot prove the agreement theorem using probabilities running over streams (e.g. using probabilities with values on hypernumbers or p-adic numbers) in any way. In non-standard fields Aumann's theorem is false, because the powerset of any infinite set of streams is not a Boolean algebra and the Bayesian theorem does not hold in general case for streams [41; 46; 47]. Notice that we cannot avoid streams in the case of infinite metagame or reflexive game of infinite level, because we face there an infinite data structure consisting of streams. The fuzzy and probability logic with values on streams was obtained in [41; 42; 43; 44; 45; 46; 47]. This logic can be used for developing a probability theory and epistemic logic for the infinite metagame and reflexive game of infinite level.

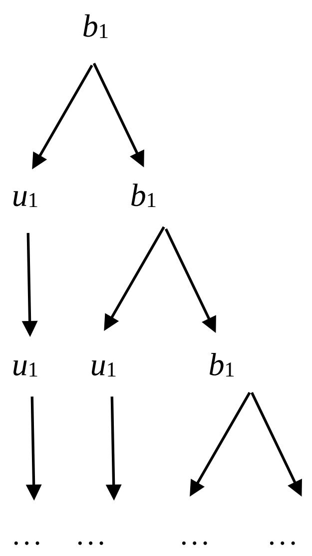
Mathematically, the infinite metagame is a coalgebra [18; 32; 36; 39]. Graphically, coalgebras (e.g. processes or games) can be represented as infinite trees. As an example, let us refer to the following definition of binary trees labeled with x, y, … and whose interior nodes are either unary nodes labeled with u_1, u_2, … or binary nodes labeled with b_1, b_2, …:

1. the variables x, y, … are trees;
2. if t is a tree, then adding a single node labeled with one of u_1, u_2, … as a new root with t as its only subtree gives a tree;
3. if s and t are trees, then adding a single node labeled with one of b_1, b_2, … as a new root with s as the left subtree and t as the right subtree again gives a tree;
4. trees may go on forever (i.e. trees satisfy the greatest fixed point condition).

This definition allows us to define some binary trees by circular definitions such as:



To sum up, we obtain the following infinite binary tree:



Let Tr be the set of trees that we have been defined. Then our definition introduces a coalgebra:

$$\mathrm{Tr} = \{x, y, \ldots\} \cup (\{u_1, u_2, \ldots\} \times \mathrm{Tr}) \cup (\{b_1, b_2, \ldots\} \times \mathrm{Tr} \times \mathrm{Tr}).$$

The reflection disagreement theorem is valid for games presented in the form of coalgebra. Recently many researchers [5; 24; 29; 53] have focused on the idea that in economics, in particular in decision theory, we cannot avoid coalgebraic notions such as process dynamics, behavioral instability, self-reference, or circularity. There exist many more cases of non-equilibriums in economics, because we engage coinductive databases more often as a matter of fact [5; 16]. For example, repeated games may be defined only coalgebraically [1; 30] and as well it is better to define epistemic games and belief functors as coalgebras [6; 7; 8; 9].

Thus, the reflexion disagreement theorem can be proved if (1) we assume that rational agents can become unpredictable for each other and try to manipulate; (2) we define probabilities on streams (e.g. on hypernumbers or p-adic numbers); (3) games are presented as coalgebras. As we see, this new theorem is a very important statement within the new mathematics (coalgebras, transition systems, process calculi, etc.) which has been involved into game theory recently. Sets of streams which have been modelled coalgebraically cannot generate inductive sets [2], therefore Aumann's agreement theorem is meaningless on these sets, but we face just the sets of streams in many kinds of games (e.g. if we deal with repeated games, games with infinite states, concurrent games, infinite metagame, reflexive game of infinite level, etc.). Instead of the agreement theorem, the reflexion disagreement theorem is valid if we cannot obtain inductive sets, e.g. in case of sets of streams. Notice that according to Aczel [2], the universum of coinductive sets is much larger than the universum of inductive sets.

3.2. Reflexion disagreement theorem

Let A be any set. We define the set A^{ω} of all streams over A as $A^{\omega} = \{\sigma : \{0,1,2,\ldots\} \to A\}$. For more details on stream calculus see [15; 35; 38; 39; 40]. For a stream σ, we call $\sigma(0)$ the initial value of σ. We define the *derivative* $\sigma(0)$ of a stream σ, for all $n \geq 0$, by $\sigma'(n) = \sigma(n+1)$. For any $n \geq 0$, $\sigma(n)$ is called the *n*-th element of σ. It can also be expressed in terms of higher-order stream derivatives, defined, for all $k \geq 0$, by $\sigma^{(0)} = \sigma$; $\sigma^{(k+1)} = (\sigma^{(k)})'$. In this case the *n*-th element of a stream σ is given by $\sigma(n) = \sigma^{(n)}(0)$. Also, the stream is understood as an infinite sequence of derivatives. It will be denoted by an infinite sequence of values or by an infinite tuple:

$$\sigma = \sigma(0) :: \sigma(1) :: \sigma(2) :: \ldots :: \sigma(n-1) :: \ldots,$$

$$\sigma = \langle \sigma(0), \sigma(1), \sigma(2), \ldots \rangle.$$

A *bisimulation* on A^ω is a relation $R \subseteq A^\omega \times A^\omega$ such that, for all σ and τ in A^ω, if $\langle \sigma, \tau \rangle \in R$ then (i) $\sigma(0) = \tau(0)$ and (ii) $\langle \sigma', \tau' \rangle \in R$.

Theorem 1. (Coinduction) *For all* $\sigma, \tau \in A^\omega$, *if there exists a bisimulation relation* $R \subseteq A^\omega \times A^\omega$ *with* $\langle \sigma, \tau \rangle \in R$, *then* $\sigma = \tau$. *In other words,* $\sigma \colon \tau \Rightarrow \sigma = \tau$.

This proof principle is called *coinduction*.

The repeated stream at each step is denoted by $[\sigma(0)]$ or by $[a]$. We can define addition and multiplication of streams as follows. The sum $\sigma + \tau$ and the product $\sigma \times \tau$ of streams σ and τ are element-wise:

$$\forall n \in \mathbf{N}, (\sigma + \tau)(n) = \sigma(n) + \tau(n).$$

$$\forall n \in \mathbf{N}, (\sigma \times \tau)(n) = \sum_{k=0}^{n} \sigma(k) \cdot \tau(n-k).$$

To set addition and multiplication by coinduction, we should use the following facts about differentiation of sums and products by applying the basic operations: $(\sigma + \tau)' = \sigma' + \tau'$, $(\sigma \times \tau)' = (|\sigma(0)| \times \tau') + (\sigma' \times \tau)$, where $|\sigma(0)| = \langle \sigma(0), 0, 0, 0, \ldots \rangle$. We see that the sum behaves exactly as in classical calculus, however multiplication does not. Now we can define them and as well as two other stream operations as follows:

Differential equation	Initial value	Name
$(\sigma + \tau)' = \sigma' + \tau'$	$(\sigma + \tau)(0) = \sigma(0) + \tau(0)$	Sum
$(\sigma \times \tau)' = (|\sigma(0)| \times \tau') + (\sigma' \times \tau)$	$(\sigma \times \tau)(0) = \sigma(0) \times \tau(0)$	Product
$(\sigma^{-1})' = |-1| \times |\sigma(0)^{-1}| \times \sigma' \times \sigma^{-1}$	$(\sigma^{-1})(0) = \sigma(0)^{-1}$	Inverse

We can embed the real numbers into the streams by defining the following constant stream. Let $r \in \mathbf{R}$. Then $|r| = \langle r, 0, 0, 0, \ldots \rangle$ is defined so: its differential equation is $|r|' = [0]$, its initial value is $|r|(0) = r$. This allows us to add and multiply real numbers and streams:

$$|r| + \sigma = \langle r + \sigma(0), \sigma(1), \sigma(2), \ldots \rangle,$$

$$|r| \times \sigma = \langle r \cdot \sigma(0), r \cdot \sigma(1), r \cdot \sigma(2), \ldots \rangle.$$

Taking into account these equalities we are to rely on our intuitions that it would be natural to define the positive real numbers to be less than the positive streams.

Consider the set of streams $[0,1]^{\omega}$ and extend the standard order structure on $[0,1]$ to a partial order structure on $[0,1]^{\omega}$. Further define this order as follows:

$\mathrm{O}_{[0,1]^{\omega}}$ (1) For any streams $\sigma,\tau\in[0,1]^{\omega}$, we set $\sigma\le\tau$ if $\sigma(n)\le\tau(n)$ for every $n\in\mathbf{N}$. For any streams $\sigma,\tau\in[0,1]^{\omega}$, we set $\sigma=\tau$ if σ, τ are bisimilar. For any streams $\sigma,\tau\in[0,1]^{\omega}$, we set $\sigma<\tau$ if $\sigma(n)\le\tau(n)$ for every $n\in\mathbf{N}$ and there exists n_0 such that $\sigma(n_0)\ne\tau(n_0)$. (2) Each stream of the form $|r|\in[0,1]^{\omega}$ (i.e. constant stream) is less than inconstant stream σ.

This ordering relation is not linear, but partial, because there exist streams $\sigma,\tau\in[0,1]^{\omega}$, which are incompatible.

Introduce two operations $\sup$, $\inf$ in the partial order structure $\mathrm{O}_{[0,1]^{\omega}}$. Assume that $\sigma,\tau\in[0,1]^{\omega}$ are either both constant streams or both inconstant streams. Then their supremum and infimum are defined by coinduction: the differential equation of supremum is $(\sup(\sigma,\tau))'=\sup(\sigma',\tau')$ and its initial value is $(\sup(\sigma,\tau))(0)=\sup(\sigma(0),\tau(0))$, the differential equation of infimum is $(\inf(\sigma,\tau))'=\inf(\sigma',\tau')$ and its initial value is $(\inf(\sigma,\tau))(0)=\inf(\sigma(0),\tau(0))$. Suppose now that one and only one of $\sigma,\tau\in[0,1]^{\omega}$ is constant, then an inconstant stream is greater than a constant one, therefore their supremum gives an inconstant stream, but their infimum gives a constant stream.

According to $\mathrm{O}_{[0,1]^{\omega}}$, there exist the maximal stream $[1]\in[0,1]^{\omega}$ and the minimal stream $[0]=|0|\in[0,1]^{\omega}$.

Each p-adic number has a unique expansion $n=\sum_{k=-N}^{+\infty}\alpha_k\cdot p^k$, where $\alpha_k\in\{0,1,\ldots,p-1\}$, $\forall k\in\mathbf{Z}$, and $\alpha_{-N}\ne0$, that is called the canonical expansion of p-adic number *n* (or p-adic expansion for *n*). p-Adic numbers can be identified with sequences of digits:

$$n=\ldots\alpha_2\alpha_1\alpha_0,\alpha_{-1}\ldots\alpha_{-N}$$

or with infinite tuples:

$$n=\langle\alpha_{-N},\ldots,\alpha_{-2},\alpha_{-1},\alpha_0,\alpha_1,\alpha_2,\ldots\rangle.$$

The set of such numbers is denoted by $\mathbf{Q}_p$.

The expansion:

$$n = \alpha_0 + \alpha_1 \cdot p + \ldots + \alpha_k \cdot p^k + \ldots = \sum_{k=0}^{\infty} \alpha_k \cdot p^k,$$

where $\alpha_k \in \{0,1,\ldots, p-1\}$, $\forall k \in \mathbf{N}$, is called the *expansion of p-adic integer n*. This number sometimes has the following notation: $n = \ldots\alpha_3\alpha_2\alpha_1\alpha_0$ or $n = \langle \alpha_0, \alpha_1, \alpha_2, \alpha_3, \ldots \rangle$. The set of such numbers is denoted by $\mathbf{Z}_p$.

It can be easily shown that p-adic numbers may be represented as potentially infinite data structures such as streams. Each stream of the form:

$$\sigma = \sigma(0) :: \sigma(1) :: \sigma(2) :: \ldots :: \sigma(n-1) :: ...,$$

where $\sigma(n) \in \{0,1,\ldots, p-1\}$ for every $n \in \mathbf{N}$, may be converted into a p-adic integer.

It is easily shown that the set A^{ω} of all p-adic streams includes the set of natural numbers. Let *n* be a natural number. It has a finite p-adic expansion $n = \sum_{k=0}^{m} \alpha_k \cdot p^k$. Then we can identify *n* with a p-adic stream $\sigma = \sigma(0) :: \sigma(1) :: \ldots :: \sigma(m) :: \sigma^{(m+1)}$, where $\sigma(i) = \alpha_i$ for $i = \overline{0,m}$ and $\sigma^{(m+1)} = [0]$.

Extend the standard order structure on $\mathbf{N}$ to a partial order structure on p-adic streams (i.e. on $\mathbf{Z}_p$).

- for any p-adic streams σ, $\tau \in \mathbf{N}$ we have $\sigma \leq \tau$ in $\mathbf{N}$ iff $\sigma \leq \tau$ in $\mathbf{Z}_p$,
- each p-adic stream $\sigma = \sigma(0) :: \sigma(1) :: \ldots :: \sigma(m) :: \sigma^{(m+1)}$, where $\sigma^{(m+1)} = [0]$ (i.e. each finite natural number), is less than any infinite number τ, i.e. $\sigma < \tau$ for any $\sigma \in \mathbf{N}$ and $\tau \in \mathbf{Z}_p \setminus \mathbf{N}$.

Define this partial order structure on $\mathbf{Z}_p$ as follows:

$\mathrm{O}_{\mathrm{Z}_p}$ Let $\sigma = \sigma(0) :: \sigma(1) :: \ldots :: \sigma(n-1) :: \sigma^{(n)}$ and $\tau = \tau(0) :: \tau(1) :: \ldots :: \tau(n-1) :: \tau^{(n)}$

be p-adic streams. (1) We set $\sigma < \tau$ if the following three conditions hold: (i) there exists *n* such that $\sigma(n) < \tau(n)$; (ii) $\sigma(k) \leq \tau(k)$ for all $k > n$; (iii) σ is a finite integer, i.e. there exists *m* such that $\sigma^{(m)} = [0]$. (2) We set $\sigma = \tau$ if σ and τ are bisimilar (see theorem 1). (3) Suppose that σ, τ are infinite integers. We set $\sigma \leq \tau$ by coinduction: $\sigma \leq \tau$ iff $\sigma(n) \leq \tau(n)$ for every $n \in \mathbf{N}$. We set $\sigma < \tau$ if we have $\sigma \leq \tau$ and there exists $n_0 \in \mathbf{N}$ such that $\sigma(n_0) < \tau(n_0)$.

The ordering relation $\mathsf{O}_{\mathbf{Z}_p}$ is not linear, but partial, because there exist p-adic streams $\sigma, \tau \in \mathbf{Z}_p$, which are incompatible. As an example, let $p = 2$ and let σ represents the p-adic integer $-\frac{1}{3} = \ldots 10101\ldots 101$ and τ the p-adic integer $-\frac{2}{3} = \ldots 01010\ldots 010$. Then the p-adic streams σ and τ are incompatible.

Now introduce two operations $\sup$, $\inf$ in the partial order structure on $\mathbf{Z}_p$. Suppose that p-adic streams σ, τ represent infinite p-adic integers. Then their $\sup$ and $\inf$ may be defined by coinduction as follows: the differential equation of supremum is $(\sup(\sigma, \tau))' = \sup(\sigma', \tau')$ and its initial value is $(\sup(\sigma, \tau))(0) = \sup(\sigma(0), \tau(0))$, the differential equation of infimum is $(\inf(\sigma, \tau))' = \inf(\sigma', \tau')$ and its initial value is $(\inf(\sigma, \tau))(0) = \inf(\sigma(0), \tau(0))$. Now suppose that at most one of two streams σ, τ represents a finite p-adic integer. In this case $\sup(\sigma, \tau) = \tau$ if and only if $\sigma \leq \tau$ under condition $\mathsf{O}_{\mathbf{Z}_p}$ and $\inf(\sigma, \tau) = \sigma$ if and only if $\sigma \leq \tau$ under condition $\mathsf{O}_{\mathbf{Z}_p}$.

It is important to remark that there exists the maximal p-adic stream $N_{max} \in \mathbf{Z}_p$ under condition $\mathsf{O}_{\mathbf{Z}_p}$. It is easy to see: $N_{max} = [p-1] = -1 = (p-1) + (p-1)\cdot p + \ldots + (p-1)\cdot p^k + \ldots$

Now, using the given notion of streams, let us prove the reflexion disagreement theorem. For Ω, the finite set of possible states of the world, and *N*, the set of agents, we can unconventionally define agent i's accepted performances as a function $\mathbf{Q}_i$ which assigns to each $o \in \Omega$ a non-empty subset of Ω, so that each world *o* belongs to one or more elements of each $\mathbf{Q}_i$, i.e. Ω is contained in a union of $\mathbf{Q}_i$, but $\mathbf{Q}_i$ are not mutually disjoint. Then $\mathbf{Q}_i(o)$ is called i's *accepted performative state* at *o*. If the successful performance is *o*, the individual knows (accepts) that the performative state is in $\mathbf{Q}_i(o)$. The elements of $\mathbf{Q}_i(o)$ are those states of the world that are considered as types of situations for performative states making the latter successful at *o*.

We can propose a stream interpretation of $\mathbf{Q}_i(o)$ and construct Ω^ω. We know that the set Ω^ω is much larger than Ω. According to the orders $\mathsf{O}_{[0,1]^\omega}$ and $\mathsf{O}_{\mathbf{Z}_p}$, we can identify all members of Ω with some streams of Ω^ω. Let the set of these streams be denoted by ${}^\sigma\Omega$. Evidently, ${}^\sigma\Omega \subset \Omega^\omega$. Assume that Ω^ω is a union of $\mathbf{Q}_i$. Therefore $\mathbf{Q}_i$ contains ${}^\sigma\Omega$ (it means that by our assumption Ω).

Now let us define probabilities on streams as follows: a *finitely additive probability measure* is a nonnegative set function $P(\cdot)$ defined for sets $A \subseteq \Omega^\omega$, running the set $[0, 1]^\omega$, and satisfying the following properties:

(i) $P(\varnothing) \geq |0|$ for all $A \subseteq \Omega^\omega$.

(ii) $P(\Omega^\omega) = [1]$ and $P(\varnothing) = |0|$.

(iii) If $A \subseteq \Omega^{\omega}$ and $B \subseteq \Omega^{\omega}$ are disjoint, i.e. $\inf(P(A), P(B)) = |0|$, then $P(A \cup B) = P(A) + P(B)$. Otherwise, $P(A \cup B) = P(A) + P(B) - \inf(P(A), P(B)) = \sup(P(A), P(B))$.

(iv) $P(\neg A) = [1] - P(A)$ for all $A \subseteq \Omega^{\omega}$, where $\neg A = \Omega^{\omega} \setminus A$.

(v) Relative probability functions $P(A \mid B) \in [0, 1]^{\omega}$ are characterized by the following constraint:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \tag{7}$$

where $P(B) \neq |0|$ and $P(A \cap B) = \inf(P(A), P(B))$. Notice that since there are no partitions of sets of streams in general case [41; 47], there are some problems in defining the conditional relation $P(A \mid B)$ between events also. There are much more dependent events than in the usual σ-field. For example, any real number of [0, 1] is less than any inconstant stream of $[0, 1]^{\omega}$. Let $P(B) = a$, $P(A) = b$, where a is a number of [0, 1] and b is an inconstant stream of $[0, 1]^{\omega}$. Then according to $\mathrm{O}_{[0,1]^{\omega}}$, $P(A \mid B) = 1$. However, this case cannot be defined within the traditional independence condition $P(B) = P(B|A)$. Instead of that we use the following condition: $P(A)P(B) = \inf(P(A), P(B))$.

The main originality of those probabilities is that conditions (iii), (iv) are independent. As a result, in a probability space $\langle \Omega^{\omega}, P \rangle$ some Bayes' formulas do not hold in the general case [41; 47] and Aumann's theorem cannot be proven.

A particular case of non-Archimedean probabilities is presented by p-adic probabilities. Let us define them on any subsets of Ω^{ω} as follows: a *finitely additive probability measure* is a set function $P(\cdot)$ defined for sets $A \subseteq \Omega^{\omega}$, running over the set $\mathbf{Z}_p$ and satisfying the following properties:

(i') $P(\Omega^{\omega}) = -1$ and $P(\varnothing) = 0$.

(ii') If $A \subseteq \Omega^{\omega}$ and $B \subseteq \Omega^{\omega}$ are disjoint, i.e. $\inf(P(A), P(B)) = 0$, then $P(A \cup B) = P(A) + P(B)$. Otherwise, $P(A \cup B) = P(A) + P(B) - \inf(P(A), P(B)) = \sup(P(A), P(B))$. Let us exemplify this property by 7-adic probabilities. Let $P(A) = \ldots 323241$ and $P(B) = \ldots 354322$ in 7-adic metrics. Then $P(A) + P(B) = \ldots 010563$; $\inf(P(A), P(B)) = \ldots 323221$; $P(A) + P(B) - \inf(P(A), P(B)) = \sup(P(A), P(B)) = \ldots 354342$.

(iii') $P(\neg A) = -1 - P(A)$ for all $A \subseteq \Omega^{\omega}$, where $\neg A = \Omega^{\omega} \setminus A$.

(iv') Relative probability functions $P(A \mid B) \in \mathbf{Q}_p$ are characterized by the following constraint:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)},$$

where $P(B) \neq 0$ and $P(A \cap B) = \inf(P(A), P(B))$.

Now we can interpret $\mathbf{Q}_i(|o|)$, where $|o| \in^\sigma \Omega$, probabilistically as follows: $\mathbf{Q}_i(|o|) = \{\tau : P_i(\tau \,||\, o|) > [0]\}$. These relative probabilities cannot set a partition of Ω^ω. In other words, using them we cannot define an equivalence relation corresponding to a partition. Instead of that the following properties hold, as we can prove on the basis of the orders $\mathrm{O}_{[0,1]^\omega}$ and $\mathrm{O}_{\mathbf{Z}_p}$:

- If $P_i(\tau \mid \pi) > [0]$, then there exists ρ such that $P_i(\rho \mid \pi) > [0]$ and $P_i(\tau \mid \rho) > [0]$. This property takes place instead of the usual transitivity in real probability logic: if $P_i(\rho \mid \pi) > 0$ and $P_i(\tau \mid \rho) > 0$, then $P_i(\tau \mid \pi) > 0$.
- $P_i(\tau \,\|\, \omega|) = |1|$, where $|\omega| \in^\sigma \Omega$ and $\tau \in \Omega^\omega \setminus^\sigma \Omega$.
- $P_i(\tau \mid \tau) > [0]$.

Thus, the possibility operator $\mathbf{Q}_i$ has the following properties: for all $\tau, \pi \in \Omega^\omega$:

$$\tau \in \mathbf{Q}_i(\tau) \tag{8}$$

$$\pi \in \mathbf{Q}_i(\tau) \Rightarrow \mathbf{Q}_i(\pi) = \mathbf{Q}_i(\tau) \tag{9}$$

Now we consider the relation $A \subseteq \mathbf{Q}_i(o)$, where $A \subseteq^\sigma \Omega$, as the statement that at *o* agent i accepts the performance A, i.e. $|o| \in A$ for all states $|\upsilon|$ that i considers possible at $|o|$:

$$K_i A = \{|o| : A \subseteq \mathbf{Q}_i(|o|)\}. \tag{10}$$

This set is another interpretation of knowledge operator which is coinductive now. If $A \subseteq \mathbf{Q}_i(|o|)$, an individual i who observes $|o|$, will accept a state of the performance *A*. The most important property of the knowledge operator is $A \subseteq K_i A$; i.e. if *A* is successful in state $|o|$ (i.e., $|o| \in A$), then an agent accepts a performance *A* in state $|o|$ (i.e., $|o| \in K_i A$).

The following statements can be proved in relation to coinductive knowledge operator defined in (10):

$$ {}^{\sigma}\Omega \subseteq K_i {}^{\sigma}\Omega \subseteq \Omega^{\omega}; \tag{11} $$

$$ (K_i A \cap K_i B) \Rightarrow K_i (A \cap B); \tag{12} $$

$$ K_i (A \cup B) \Rightarrow (K_i A \cup K_i B); \tag{13} $$

$$ K_i (A \cup B) = (K_i A \cap K_i B); \tag{14} $$

$$ A \subseteq B \Rightarrow K_i B \subseteq K_i A; \tag{15} $$

$$ A \subseteq K_i A; \tag{16} $$

$$ K_i K_i A = K_i A. \tag{17} $$

We can compare Aumann's statements (1) – (6) with statements (11) – (17) to notice that the latters assume a new epistemic logic with a stream interpretation. For example, it is possible to build up a kind of multy-valued illocutionary logic [50; 51; 52], where streams are values for perlocutionary effects. So, the coinductive knowledge operator K_i of (11) – (17) designates *perlocutionary effects of illocutionary acts*, i.e. it takes into account just successful performative propositions (it defines what influence was made on hearer's behavior). Let K_jA (respectively K_iA) means agent j's (respectively i's) performative (cognitive or emotional) estimations of states of affairs A with an expected perlocutionary effect of these estimations on agent j (respectively i). So, K_jA means 'j + performative verb + that A' (e.g. 'j thinks that A', 'j likes A', 'j hates A', etc.) and agent j follows this statement in his(her) behavior.

On the basis of the standard propositional language L with the set of values [0, 1] (or $\{0, 1, \ldots, p-1\}$) we can construct an extension L'' containing new modal operators $\mathbf{E}$, $\mathbf{E}_1$, $\mathbf{E}_2$, ... said to be *perlocutionary effects*. The semantics of L'' is defined in the following way. Assume that V is a valuation of well-formed formulas of L and it takes values in [0, 1] (or $\{0, 1, \ldots, p-1\}$). Let us extend V to V_e as follows:

a) If for $\varphi \in \mathsf{L}$, $V(\varphi) = r$, then $V_e(\mathbf{E}_i(\varphi)) = \langle \sigma(0) = r, \sigma(1), \sigma(2), \ldots \rangle$, i.e. $V_e(\mathbf{E}_i(\varphi))$ is a mapping from $V(\varphi)$ to an inconstant stream σ starting with $V(\varphi)$.
b) If for $\varphi \in \mathsf{L}$, $V(\varphi) = r$, then $V_e(\varphi) = |r|$.
c) For all $\varphi, \psi \in \mathsf{L}$, $V_e(\mathbf{E}_i(\varphi \vee \psi)) = V_e(\mathbf{E}_i(\varphi) \wedge V_e(\mathbf{E}_i(\psi))$.

In this semantics the following propositions will be perlocutionary tautologies, i.e. they will be true:

$$ \varphi \Rightarrow \mathbf{E}(\varphi), \tag{18} $$

$$\neg\varphi \Rightarrow \mathbf{E}(\neg\varphi), \tag{19}$$

$$\mathbf{E}(\neg\varphi) \Rightarrow \neg\mathbf{E}(\varphi), \tag{20}$$

$$(\mathbf{E}(\varphi) \wedge \mathbf{E}(\psi)) \Rightarrow \mathbf{E}(\varphi \wedge \psi), \tag{21}$$

$$\mathbf{E}(\varphi \vee \psi) \Rightarrow (\mathbf{E}(\varphi) \vee \mathbf{E}(\psi)), \tag{22}$$

$$(\mathbf{E}(\varphi \Rightarrow \psi)) \Rightarrow (\mathbf{E}(\varphi) \Rightarrow \mathbf{E}(\psi)), \tag{23}$$

$$(\mathbf{E}(\varphi \vee \psi)) \Leftrightarrow \mathbf{E}(\varphi) \wedge \mathbf{E}(\psi). \tag{24}$$

The epistemic logic closed under tautologies (18) – (24) is a kind of many-valued logic with values on non-Archimedean numbers [41; 43, 44; 45; 48].

Aumann's understanding of common knowledge satisfies the classical intuition of inductiveness of all logical entities, i.e. the presupposition that we can appeal only to inductive sets in our reasoning. For example, we can always find out an infinite intersection according to knowledge operators of different people. However, this intuition contradicts to the possibility of reflexive games when I can cheat or make false public announcements and should detect if I am cheated by other people.

Under conditions of reflexive games I cannot define common perlocutionary effects as the infinite intersection κA. An infinite mutual reflexion between two individuals assumes an infinite union: both have mutual knowledge of *A* or both know that both know *A* or both know that both know that both know *A* etc. ad infinitum. In other words, the *common perlocutionary operator* $\overline{KA}$ is defined as follows:

$$\overline{KA} = K_1 A \cup K_2 A \cup K_1 K_2 A \cup K_2 K_1 A \cup K_1 K_2 K_1 A \cup \ldots$$

For each natural number *n* an operator $\overline{M_n}$ expressing “*n*-th degree mutual reflexion for perlocutionary effects” is defined so:

$$\overline{M_0 A} = A; \ \overline{M_{n+1} A} = \bigcup_{i=1}^{N} K_i \overline{M_n A}$$

Common perlocutionary effect, $\overline{\kappa}$, is understood as a mutual reflexion for perlocutionary effects of all finite degrees:

$$\overline{\kappa A} = \bigcup_{n=0}^{+\infty} \overline{M_n A}.$$

Also, let us define for each natural number *n* an operator M_n expressing "*n*-th degree mutual reflexion":

$$M_0 A = A;\ M_{n+1} A = \bigcap_{i=1}^{N} K_i M_n A$$

and common knowledge, κ, as mutual reflexion for common knowledge of all finite degrees:

$$\kappa A = \bigcap_{n=0}^{+\infty} M_n A.$$

Lemma 1. *If* $|o| \in \kappa A$*, then for any* i*,* $\kappa A \subseteq \mathbf{Q}_i(|o|)$*. And if* $|o| \in \overline{\kappa A}$*, then for some* i*,* $\overline{\kappa A} \subseteq \mathbf{Q}_i(|o|)$*.*

Proof. If $|o| \in \kappa A$, then $|o| \in K_i M_n A$ for all agents i and degrees *n* of mutual reflexion for common knowledge. Therefore $\kappa A \subseteq \mathbf{Q}_i(|o|)$ for any i. If $|o| \in \overline{\kappa A}$, then $|o| \in K_i \overline{M_n A}$ for some agents i and degrees n of mutual reflexion for perlocutionary effects. Therefore $\overline{M_n A} \subseteq \mathbf{Q}_i(|o|)$ for some *n*, and thus $\overline{\kappa A} \subseteq \mathbf{Q}_i(|o|)$ for some i. Q.E.D.

Theorem 2. (Reflexion Disagreement Theorem) *Let us consider a hypothesis* H *of coinductive probability logic* [46; 47] *for which the various agents' coinductive probabilities are* $q_1, \ldots, q_N$ *after they condition* $P(\cdot)$ *on priors. The propositions* C *and* $\overline{C}$ *of coinductive probability logic identify these probabilities:*

$$C = \bigcap_{i=1}^{N} \{|o| : P(H \mid \mathbf{Q}_i(|o|)) = q_i\}.$$

$$\overline{C} = \bigcup_{i=1}^{N} \{|o| : P(H \mid \mathbf{Q}_i(\omega)) = q_i\}.$$

Let the coinductive probability space $\langle \Omega^{\omega}, P \rangle$ be closed under all of the operators K_i, M_n, κ, $\overline{K_i}$, $\overline{M_n}$, and $\overline{\kappa}$ and let P be the old probability measure that is common to all the agents. Assume that the possibility of C's and $\overline{C}$'s becoming common knowledge or common perlocution is not equal to zero: $P(\kappa C) \neq [0]$ and $P(\overline{\kappa C}) \neq [0]$, then:

$$P(H \mid \kappa C) \neq q_i \text{ for some } i.$$
$$P(H \mid \overline{\kappa C}) \neq q_i \text{ for some } i.$$

Proof. By lemma 1, $\kappa C = \bigcup_j D_{ij}$, where $\bigcup_j D_{ij}$ covers $\mathbf{Q}_i$, but it is not a partition of $\mathbf{Q}_i$ because of basic properties of coinductive probabilities. Then:

$$P(H \mid \kappa C) = \frac{P(H \cap \bigcup_j D_{ij})}{P(\bigcup_j D_{ij})} = \frac{\inf(P(H), \sup_j P(D_{ij}))}{\sup_j P(D_{ij})} \neq \frac{\sum_j P(H \mid D_{ij}) P(D_{ij})}{\sum_j P(D_{ij})} = \frac{\sum_j q_i P(D_{ij})}{\sum_j P(D_{ij})} = q_i$$

Thus, $P(H \mid \kappa C) \neq q_i$ in general case. In the same way we can show that $P(H \mid \overline{\kappa C}) \neq q_i$ in general case. Q.E.D.

3.3. Cellular-automatic reflexive games

The reflexion disagreement theorem is valid for games presented in the coalgebraic form. There are many kinds of such games: repeated, concurrent, etc. In this section a new game presentation in the coalgebraic form will be proposed on the basis of proof-theoretic cellular automata [49]. These automata can be used in formulating context-based decision rules in games. Usually, for representing databases of games, payoff matrices are involved (see Figure 1). However, in the case of coinductive databases we cannot appeal to payoff matrices. For example, we cannot appeal to them if we deal with games limited by some contexts or with infinite games. Some kinds of coinductive databases for making decisions could be presented by *payoff cellular automata*. These automata are constructed as follows. Cells of automata belong to the set Z^d and they take its value in S. The set S of states consists of payoff for all n players. The cardinality $|S|$ is equal $i_1 \cdot i_2 \cdot \ldots \cdot i_n$, where i_j is the number of all strategies for the j-th player, $j = 1, \ldots, n$. Each state has the form of n-tuple $\langle a_{ij\ldots k}, b_{ij\ldots k}, \ldots, c_{ij\ldots k}\rangle$, where

(1) $a_{ij\ldots k}$ is the payoff to player 1 when (1) he plays a_i; (2) player 2 plays b_j; …, (n) player n plays c_k;

(2) $b_{ij\ldots k}$ is the payoff to player 2 when (1) player 1 plays a_i; (2) player 2 plays b_j; …, (n) and player n plays c_k, etc.,

…

(n) $c_{ij\ldots k}$ is the payoff to player n when (1) player 1 plays a_i; (2) player 2 plays b_j; …, (n) and player n plays c_k.

Figure 1. An example of a payoff matrix showing the possible strategies available to player 1 (namely a_1 and a_2) and player 2 (namely b_1 and b_2) and the payoff that each player receives for his choice, depending on what the other players do. The payoff is in the form $\langle a_{ij}, b_{ij}\rangle$, where a_{ij} is the payoff to player 1 when he plays a_i and player 2 plays b_j and b_{ij} is the payoff to player 2 when he plays b_j and player 2 plays a_i.

$$\begin{array}{ccc} & & \textit{Player 1} \\ & & \begin{array}{cc} a_1 & a_2 \end{array} \\ \textit{Player 2} & \begin{array}{c} b_1 \\ \\ b_2 \end{array} & \left| \begin{array}{cc} \langle a_{11}, b_{11}\rangle & \langle a_{21}, b_{21}\rangle \\ & \\ \langle a_{12}, b_{12}\rangle & \langle a_{22}, b_{22}\rangle \end{array} \right| \end{array}$$

The local transition function, δ_j, for player j, where $j = 1, \ldots, n$, is presented by a decision rule based on the past payoff for all players. The rule δ_j can be the same for all players or different. The initial configuration of payoff cellular automaton is the set of all premises which fully determines the future behaviour of the automaton. They may be understood as the expected payoff for different contexts before the game. The *game context* is defined by the neighbourhood $N(z)$ for the cell z. The number of premises (the past payoff that we can take into account) cannot exceed the number $n = |N(z) \cup z|$. The decision rule δ_j is a mapping from the set of premises $N(z) \cup z$ to a conclusion. This rule generates the sequence $\boldsymbol{a}^0(z), \boldsymbol{a}^1(z), \ldots, \boldsymbol{a}^t(z), \ldots$ for any $z \in \mathbb{Z}^d$, where $\boldsymbol{a} = \langle a_{ij\ldots k}, b_{ij\ldots k}, \ldots, c_{ij\ldots k}\rangle$ and $\boldsymbol{a}^i(z)$ means the state of z at the i-th step of application of δ_j to $\boldsymbol{a}^0(z)$, the state of z at step 0. That sequence is called a *derivation trace from an initial state* $\boldsymbol{a}^0(z)$. Obviously, this sequence is an infinite stream.

Example 1. (Saddle Point).

Let us consider a simple payoff cellular automaton for the game with two players 1 and 2. Let a_{ij} be a payoff for the i-th strategy of player 1 and j-th strategy of player 2. If $\max_i \min_j a_{ij} = \min_j \max_i a_{ij}$ for $z \cup N(z)$ at step t, then a_{ij} is called a *saddle point* for z at time t. Thus, a saddle point is the element of the payoff cellular automaton at time t which is both a maximum of the minimums of each row within the neighbourhood $N(z)$ and a minimum of the maximums of each column within the same neighbourhood. The cells $z \cup N(z)$ may have no saddle points, one saddle point, or multiple saddle points. Let the payoff states pictured in figure 2 be an initial configuration of the automaton. The set S of states consists of the integers –5, –4, –3, …, 7, 8.

Figure 2. An initial configuration of a payoff cellular automaton $\mathcal{A}$ with the neighbourhood consisting of 8 members in the 2-dimensional space and with players 1 and 2.

8	2	3	2	3
–3	0	2	–5	–4
–2	–1	6	–1	8
4	1	9	2	4
5	–2	3	0	2

The local transition function is defined as follows:

$$\boldsymbol{a}^{t+1}(z)=\begin{cases} a_{ij}, & \text{if } a_{ij}=\max(a_{kl},a_{mn}), \text{where } a_{kl},a_{mn} \text{ are saddle points of } N(z),\\ \boldsymbol{a}^{t}(z), & \text{otherwise.}\end{cases}$$

At time $t = 1$, the configuration of figure 2 has the following form:

Figure 3. Values of A given in figure 2 at time $t = 1$.

2	2	2	2	2
2	2	2	2	2
1	1	1	2	2
1	1	1	2	2
1	1	1	2	2

Now let us define reflexive games within payoff cellular automata. Designate reflexive players by 1 and 2. Let A, a state of affairs, be identified with a set of payoffs within a game context (i.e. within a neighbourhood). In other words, let $A_{\langle aij,\, bij\rangle}$ be a set of payoffs at the point $z \in \mathbf{Z}^{d}$ consisting of all payoffs of $N(z) \cup z$, where z has a state $\langle a_{ij}, b_{ij}\rangle$, see figure 4.

Figure 4. An initial configuration of a payoff cellular automaton with the neighbourhood consisting of 8 members in a 2-dimensional space and with players 1 and 2. This configuration presents 9 states of affairs: $A_{\langle 3,\,3\rangle} = \{\langle 3, 3\rangle, \langle 12,-12\rangle, \langle -12, 12\rangle, \langle -5,-5\rangle\}$; $A_{\langle 12,\,-12\rangle} = \{\langle 3, 3\rangle, \langle 12,-12\rangle, \langle 13,-15\rangle, \langle -12, 12\rangle, \langle -5,-5\rangle, \langle -2,-2\rangle\}$, $A_{\langle 13,\,-15\rangle} = \{\langle 12,-12\rangle, \langle 13,-15\rangle, \langle -5,-5\rangle, \langle -2,-2\rangle\}$; $A_{\langle -12,\,12\rangle} = \{\langle 3, 3\rangle, \langle 12,-12\rangle, \langle -12, 12\rangle, \langle -5,-5\rangle, \langle -1, 1\rangle, \langle 0,-6\rangle\}$, ...

⟨3,3⟩	⟨12,−12⟩	⟨13,−15⟩
⟨−12,12⟩	⟨−5,−5⟩	⟨−2,−2⟩
⟨−1,1⟩	⟨0,−6⟩	⟨2,−3⟩

Let $\mathbf{B}^{1}{}_{1}\, A_{\langle aij,\, bij\rangle}$ (accordingly, $\mathbf{B}^{2}{}_{1}\, A_{\langle aij,\, bij\rangle}$) mean agent 1's (accordingly, agent 2's) Boolean superpositions of 1's payoffs of $A_{\langle aij,\, bij\rangle}$ (accordingly, 2's payoffs) for each first (accordingly, second) projection of all points of $A_{\langle aij,\, bij\rangle}$. Then $K_1A_{\langle aij,\, bij\rangle} = A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{1}{}_{1}\, A_{\langle aij,\, bij\rangle}$ and $K_2A_{\langle aij,\, bij\rangle} = A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{2}{}_{1}\, A_{\langle aij,\, bij\rangle}$. Let $\mathbf{B}^{1}{}_{2}\, A_{\langle aij,\, bij\rangle}$ (accordingly, $\mathbf{B}^{2}{}_{2}\, A_{\langle aij,\, bij\rangle}$) mean agent 1's (accordingly, agent 2's) Boolean superpositions of $\mathbf{B}^{1}{}_{1}\, A_{\langle aij,\, bij\rangle}$ and $\mathbf{B}^{2}{}_{1}\, A_{\langle aij,\, bij\rangle}$ for each first (accordingly, second) projection of all points of $A_{\langle aij,\, bij\rangle}$. Then $K_1K_2A_{\langle aij,\, bij\rangle} = A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{2}{}_{1}\, A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{1}{}_{2}\, A_{\langle aij,\, bij\rangle}$ and $K_2K_1A_{\langle aij,\, bij\rangle} = A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{1}{}_{1}\, A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{2}{}_{2}\, A_{\langle aij,\, bij\rangle}$. Let $\mathbf{B}^{1}{}_{3}\, A_{\langle aij,\, bij\rangle}$ (accordingly, $\mathbf{B}^{2}{}_{3}\, A_{\langle aij,\, bij\rangle}$) mean agent 1's (accordingly, agent 2's) Boolean superpositions of $\mathbf{B}^{1}{}_{2}\, A_{\langle aij,\, bij\rangle}$ and $\mathbf{B}^{2}{}_{2}\, A_{\langle aij,\, bij\rangle}$ for each first (accordingly, second) projection of all points of $A_{\langle aij,\, bij\rangle}$. Then $K_2K_1K_2A_{\langle aij,\, bij\rangle} = A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{2}{}_{1}\, A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{1}{}_{2}\, A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{2}{}_{3}\, A_{\langle aij,\, bij\rangle}$ and $K_1K_2K_1A_{\langle aij,\, bij\rangle} = A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{1}{}_{1}\, A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{2}{}_{2}\, A_{\langle aij,\, bij\rangle} \cup \mathbf{B}^{1}{}_{3}\, A_{\langle aij,\, bij\rangle}$. And so on.

Example 2. (Reflexive Game of the Second Level).

Let us consider a payoff cellular automaton of figure 4 where the set S of states consists of all pairs $\langle a_{ij}^t, b_{ij}^t\rangle$, where a_{ij}^t, b_{ij}^t at time $t = 0,1,2,\ldots$ are integers of $[-15,13]$ and the local transition function is as follows: $a^{t+1}(z) = \langle a_{ij}^{t+1}, b_{ij}^{t+1}\rangle$ where $a_{ij}^{t+1} = ((\bigvee_m b_m^t \Rightarrow \bigvee_k a_k^t) \wedge (a_{ij}^t \wedge b_{ij}^t))$ and $b_{ij}^{t+1} = (\bigvee_k b_k^t \Rightarrow \bigvee_m a_m^t)$ and $\bigvee_k a_k^t$, $\bigvee_m b_m^t$ are maximal payoffs of player 1 and player 2 respectively at cells $N(z) \cup z$ at time *t*, the logical operations are understood thus: $a \Rightarrow b := 13 - \max(a,b) + b$; $a \vee b := \max(a,b)$.

This automaton simulates the reflexive game, where player 1 has the second level of reflexion, while player 2 has the first level of reflexion. Its evolution at time $t = 1$ is pictured in figure 5.

Figure 5. The configuration of the payoff cellular automaton of figure 4 at time $t = 1$.

$\langle 3,13\rangle$	$\langle -12,13\rangle$	$\langle -15,13\rangle$
$\langle -12,13\rangle$	$\langle -5,13\rangle$	$\langle -2,13\rangle$
$\langle -1,1\rangle$	$\langle -6,3\rangle$	$\langle -3,13\rangle$

A reflexion of agent i at the *n*-th level in bimatrix games is expressed by $(n+1)$- order knowledge operators $K_i^{n+1}A = K_i K_j K_i \ldots A$, where on the right side there are $n+1$ K_m- operators ($m = i, j$). Let us consider two agents i and j and suppose that the reflexive game takes place at level *n*. This means that we have $K_i^{n+1}A$ and/or $K_j^{n+1}A$ which are understood as perlocutionary effects of illocutionary acts and satisfy requirements (12) – (17). We know that $A \subseteq \ldots \subseteq K_j^n A \subseteq K_i^{n+1}A$ and $A \subseteq \ldots \subseteq K_i^n A \subseteq K_j^{n+1}A$. Therefore $K_i^{n+1}A \cap K_j^{n+1}A \neq \varnothing$.

The payoff of reflexive game at the *n*-th level in accordance with $K_i^{n+1}A$ or $K_j^{n+1}A$ is called a *performative equilibrium* of this game.

We have the following possibilities:

- both $K_i^{n+1}A$ and $K_j^{n+1}A$ are a performative equilibrium – this means that agents i and j are on the same *n*-th level of reflexion, simultaneously;
- only $K_i^{n+1}A$ is a performative equilibrium (then we can take $K_j^{n+1}A = K_j^n A$) – this means that agent i stays on the *n*-th level of reflexion, but agent j stays on the $(n-1)$-th level of reflexion;
- only $K_j^{n+1}A$ is a performative equilibrium (then we can take $K_i^{n+1}A = K_i^n A$) – this means that agent j stays on the *n*-th level of reflexion, but agent i stays on the $(n-1)$-th level of reflexion.

In reflexive game at level *n* for agent i it is important that it is $K_i^n A \subseteq K_j^{n+1} A$, i.e. that agent i really corresponded to level n. Correctly to define reflexion level *n* may mean a victory in game.

Now let us define $K_i^{n+1} A$ on p-adic probabilities. Assume we have $p \in \mathbf{N}$ reflexive gamers *i*, *j*. Then all possible combinations $K_{\alpha_n} \dots K_{\alpha_1} K_{\alpha_0} A$, where $\alpha_k \in \{i, j, \dots\}$, can be presented by finite p-adic integers

$$\dots 00\beta_n \dots \beta_2 \beta_1 \beta_0 = \sum_{k=0}^{n} \beta_k p^k ,$$

where $\beta_k \in \{0, \dots, p-1\}$ for each $k = 0, \dots, n$ and there is a bijection between the sets $\{0, \dots, p-1\}$ and $\{i, j, \dots\}$.

Let Ω be a finite set of possible states of the world and $A \subseteq \Omega$. Then a *finite* p *-adic probability measure* P_i^{n+1} is defined on sets $A, B \subseteq \Omega$ as follows:

$$P_i^{n+1}(\varnothing) = 0 \quad and \quad P_i^{n+1}(\Omega) = 1;$$

$$if \quad P_j^n(A) > 0, \quad then \quad P_i^{n+1}(A) > 0;$$

$$P_i^n(A) > 0 \quad iff \quad P_i^{n+1}(A) > 0;$$

$$if \quad P_j^{n+1}(A) = 1, \quad then \quad P_i^n(A) = 1;$$

$$P_i^{n+1}(A) = \frac{\sum_{k=0}^{n} \alpha_k p^k}{\sum_{k=0}^{n} (p-1) p^k} \quad and \quad P_i^{n+1}(B) = \frac{\sum_{k=0}^{n} \beta_k p^k}{\sum_{k=0}^{n} (p-1) p^k},$$

where α_k, $\beta_k \in \{0, \dots, p-1\}$ for each $k = 0, \dots, n$;

$$P_i^{n+1}(A \cup B) = P_i^{n+1}(A) + P_i^{n+1}(B) \; if \; A \cap B = \varnothing; \; P_i^{n+1}(\neg A) = 1 - P_i^{n+1}(A);$$

$$P_i^{n+1}(A|B) = \frac{\inf(\sum_{k=0}^{n} \alpha_k p^k, \sum_{k=0}^{n} \beta_k p^k)}{\sum_{k=0}^{n} \beta_k p^k},$$

where $\inf$ is defined digit by digit. For instance, if we have just two agents, then on the zero level of reflexion we have only two probability values: either 0 or 1 (meaning e.g. that an agent either does not follow the content $A \subseteq \Omega$ or does). On the first level of reflexion we have already the following four probability values: 0, 1/3, 2/3, 1 (meaning e.g. that both agents do not follow the content $A \subseteq \Omega$, one of them does not follow, another does, and both of them follow), etc.

Now we can define $K_i^{n+1}A$ in the following way:

$$K_i^{n+1}A = \{\omega : A \subseteq \mathbf{Q}_i(\omega) = \{a : P_i^{n+1}(a \mid \omega) > 0\}\}.$$

Notice that according to this definition, taking into account our assumption that if $P_j^n(a \mid \omega) > 0$, then $P_i^{n+1}(a \mid \omega) > 0$, we have $K_j^n A \subseteq K_i^{n+1}A$ for each agent j participating in the reflexive game.

Let us suppose that there are just three reflexive gamers k, l, m at reflexion level n = 2. Then $P_i^2(A \subseteq \Omega) \in \{0, 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8, 1\}$ for each $i \in \{k, l, m\}$. On the infinite level of reflexion we have the following *p*-adic probabilities:

$$P_i^\infty(A) = \lim_{n \to \infty} P_i^n(A).$$

The knowledge operators $K_i^{n+1}A$ satisfy the following relations:

$$(K_i^{n+1}A \cap K_i^{n+1}B) \Rightarrow K_i^{n+1}(A \cap B);$$

$$K_i^{n+1}(A \cup B) \Rightarrow (K_i^{n+1}A \cup K_i^{n+1}B);$$

$$K_i^{n+1}(A \cup B) = (K_i^{n+1}A \cap K_i^{n+1}B);$$

$$A \subseteq B \Rightarrow K_i^{n+1}A \subseteq K_i^{n+1}B;$$

$$A \subseteq K_i^{n+1}A;$$

$$K_i^{n+1}K_i^n A = K_i^n A.$$

Using (finite) p-adic probabilities, we understand reflexion levels discretely. For any finite number of agents we can always define a reflexive level *n* such that probabilities are distributed on an appropriate finite set of p-adic numbers. The larger *n* (or the larger the number of reflexive agents), the more finite p-adic probabilities. Thereby between *n* and $n+1$ there are no other reflexive levels.

Conclusion

The Reflexion Disagreement Theorem opens the door for new mathematics in game theory and decision theory, in particular it shows that it has sense to use stream calculus, non-Archimedean mathematics, and p-adic analysis there. Within this mathematics we can formalize reflexive games of different reflexive levels (up to the the infinite reflexive level). These results can be implemented in new mathematical tools of behavioral finance.

References

[1] Abreu, D., Pearce, D., Stacchetti, E., *Toward a theory of discounted repeated games with imperfect monitoring,* Econometrica, vol. 58, no. 5, pp. 1041-1063, 1990.

[2] Aczel, P., *Non-Well-Founded Sets,* Stanford, 1988.

[3] Aumann, R.J., *Agreeing to Disagree: The Annals of Statistics,* vol. 4, no. 6, pp. 1236-1239, 1976.

[4] Aumann, R.J., *Notes on Interactive Epistemology, Mimeo,* Hebrew University of Jerusalem, Jerusalem, 1989.

[5] Berger, S., *The Foundations of Non-Equilibrium Economics: The Principle of Circular Cumulative Causation,* Routledge Chapman & Hall, 2009.

[6] Brandenburger, A., *Epistemic game theory: an overview,* 2008.

[7] Brandenburger, A., Friedenberg, A., Keisler, H.J., *Fixed points in epistemic game theory,* [in:] Abramsky, S., Mislove, M. (ed.), *Mathematical Foundations of Information Flow,* American Mathematical Society, 2012.

[8] Brandenburger, A., Keisler, H.J., *An impossibility theorem on beliefs in games,* Studia Logica, vol. 84, no. 2, pp. 211-240, 2006.

[9] Capretta, V., *Common Knowledge as a Coinductive Modality,* 2007.

[10] Chevalley, C., Belzung, C., *Emotional behaviour as the result of stochastic interactions: a process crucial for cognition*, Behavioral Processes, vol. 60, no. 2, pp. 115-132, 2002.

[11] Chkhartishvili, A.G., *Bayes–Nash Equilibrium: Infinite-Depth Point Information Structures,* Automation and Remote Control, vol. 64, no. 12, pp. 1922-1927, 2003.

[12] Chkhartishvili, A.G., *Concordant Informational Control, Automation and Remote Control,* vol. 73, no. 8, pp. 1401-1409, 2012.

[13] Chkhartishvili, A.G., Novikov, D.A., *Models of Reflexive Decision-Making, Systems Science,* vol. 30, no. 2, pp. 45-59, 2004.

[14] Chkhartishvili, A.G., *Reflexive Games: Transformation of Awareness Structure,* Automation and Remote Control, vol. 71, no. 6, pp. 1208 – 1216, 2010.

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

[16] Heifetz, A., *Non-well-founded-type spaces, Games and Economic Behavior,* vol. 16, no. 2, pp. 202-217, 1996.
[17] Howard, N., *General Metagames: an Extension of the Metagame Concept / Game Theory as a Theory of Conflict Resolution,* Reidel, Dordrecht, 1974.
[18] Jacobs, B., Rutten, J., *A tutorial on (co)algebras and (co)induction,* EATCS Bulletin, vol. 62, pp. 222-259, 1997.
[19] Khrennikov, A.Yu., *Interpretations of Probability,* VSP Int. Sc. Publishers, Utrecht/Tokyo, 1999.
[20] Khrennikov, A.Yu., *Modeling of processes of thinking in p-adic coordinates*, Nauka, Fizmatlit, Moscow, 2004 (in Russian).
[21] Khrennikov, A.Yu., *p-adic quantum mechanics with p-adic valued functions,* Journmal of Mathematical Physics, vol. 32, no. 4, pp. 932-937, 1991.
[22] Khrennikov, A.Yu., *p-adic valued distributions in mathematical physics,* Kluwer Academic Publishers, Dordrecht, 1994.
[23] Khrennikov, A.Yu., Schumann, A., *Logical Approach to p-adic Probabilities,* Bulletin of the Section of Logic, vol. 35, no. 1, pp. 49-57, 2006.
[24] Knudsen, Ch., *Equilibrium, perfect rationality and the problem of self-reference in economics,* [in:] Maki, U. (ed.), *Rationality, Institutions and Economic Methodology,* Routledge Chapman & Hall, 1993.
[25] Lefebvre, V., *The Structure of Awareness: Toward a Symbolic Language of Human Reflexion,* Sage Publications, New York, 1977.
[26] Lefebvre, V.A., *Algebra of Conscience,* D. Reidel, 1982.
[27] Lefebvre, V.A., *Lectures on Reflexive Game Theory,* Leaf & Oaks, Los Angeles, 2010.
[28] Lefebvre, V.A., *The basic ideas of reflexive game's logic,* [in:] *Problems of systems and structures research,* pp. 73-79, 1965 (in Russian).
[29] Lescanne, P., Perrinel, M., *Backward coinduction, Nash equilibrium and the rationality of escalation,* Acta Informatica, vol. 49, no. 3, pp. 117-137, 2012.
[30] Mailath, G.J., Samuelson, L., *Repeated Games and Reputations: Long-Run Relationships,* Oxford University Press, 2006.
[31] Milner, R., *Communication and Concurrency,* Prentice-Hall, 1989.
[32] Moss, L.S., *Coalgebraic logic,* Ann. Pure & Appl. Logic, vol. 96, no. 1-3, pp. 77-317, 1999; Erratum in Ann. Pure & Appl. Logic, vol. 99, no. 1-3, pp. 241-259, 1999.
[33] Novikov, D.A., Chkhartishvili, A.G., *Stability of Information Equilibrium in Reflexive Games, Automation and Remote Control,* vol. 66, no. 3, pp. 441-448, 2005.
[34] Novikov, D.A., Ckhartishvili, A.G., *Reflexive Games. Sinteg,* Moscow, 2003 (in Russian).
[35] Pavlovic, D., Escardó, M.H., *Calculus in coinductive form,* [in:] *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science,* pp. 408-417, 1998.
[36] Reichel, H., Rutten, J. (ed.), *Coalgebraic Methods in Computer Science,* Elect. Notes in TCS 11, 1998.
[37] Robinson, A., *Non-Standard Analysis. Studies in Logic and the Foundations of Mathematics,* North-Holland, 1966.
[38] Rutten, J.M., *A coinductive calculus of streams,* Math. Struct. in Comp. Sci., vol. 15, no. 1, pp. 93-147, 2005.

[39] Rutten, J.M., *Universal coalgebra: a theory of systems,* Theor. Comput. Sci., vol. 249, no. 1, pp. 3-80, 2000.

[40] Rutten, J., *Behavioral differential equations: a coinductive calculus of streams, automata, and power series,* Theor. Comput. Sci., vol. 308, pp. 1-53, 2003.

[41] Schumann, A., *Non-Archimedean Fuzzy and Probability Logic,* Journal of Applied Non-Classical Logics, vol. 18, no. 1, pp. 29-48, 2008.

[42] Schumann, A., *Non-Archimedean Valued and p-Adic Valued Fuzzy Cellular Automata,* Journal of Cellular Automata, vol. 3, no. 4, pp. 337-354, 2008.

[43] Schumann, A., *Non-Archimedean Valued Extension of Logic LΠ and p-Adic Valued Extension of Logic BL,* Journal of Uncertain Systems, vol. 4, no. 2, pp. 99-115, 2010.

[44] Schumann, A., *Non-Archimedean Valued Predicate Logic,* Bulletin of the Section of Logic, vol. 36, no. 1-2, pp. 67-78, 2007.

[45] Schumann, A., *Non-Archimedean Valued Sequent Logic,* [in:] *Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (SYNASC'06), IEEE Press, pp. 89-92, 2006.

[46] Schumann, A., *Non-well-founded Probabilities and Coinductive Probability Logic,* [in:] *Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (SYNASC'08), IEEE Computer Society Press, pp. 54-57, 2008.

[47] Schumann, A., *Non-well-founded Probabilities on Streams,* [in:] Debois, D. et al. (eds), *Soft Methods for Hand. Var. and Impresion*, ASC 48, Springer-Verlag, pp. 59-65, 2008.

[48] Schumann, A., *p-Adic Multiple-Validity and p-Adic Valued Logical Calculi,* Journal of Multiple-Valued Logic and Soft Computing, vol. 13, no. 1-2, pp. 29-60, 2007.

[49] Schumann, A., *Proof-Theoretic Cellular Automata as Logic of Unconventional Computing,* Journal of Unconventional Computing, vol. 8, no. 3, pp. 263-280, 2012.

[50] Searle, J.R., *Expression and meaning: studies in the theory of speech* acts, Cambridge University Press, Cambridge, 1979.

[51] Searle, J.R., *Speech acts; an essay in the philosophy of language,* Cambridge University Press, Cambridge, 1969.

[52] Searle, J.R., Vanderveken, D., *Foundations of Illocutionary Logic,* Cambridge University Press, Cambridge, 1984.

[53] Winrich, J.S., *Self-reference and the incomplete structure of neoclassical economics,* Journal of Economic Issues, vol. 18, no. 4, pp. 987-1005, 1984.

Originally published: Schumann, A., Akimova, L., *Syllogistic System for the Propagation of Parasites. The Case of Schistosomatidae (Trematoda: Digenea)*, Studies in Logic, Grammar and Rhetoric, vol. 40, iss. 1, pp. 303-319, 2015, http://dx.doi.org/10.1515/slgr-2015-0015.

IV
Syllogistic System for the Propagation of Parasites. The Case of *Schistosomatidae* (Trematoda: Digenea)

Andrew Schumann, Ludmila Akimova

Abstract

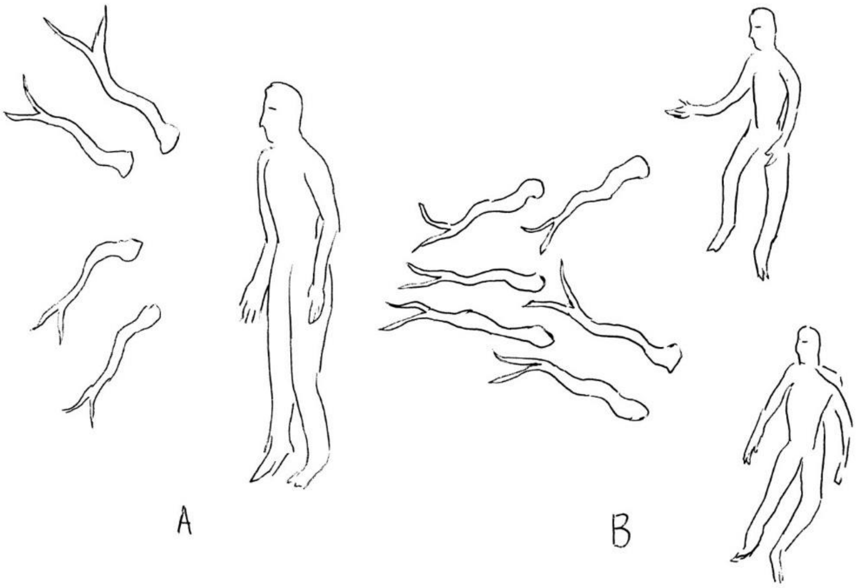
In the paper, the new syllogistic system is built up. This system simulates a massive-parallel behavior in the propagation of collectives of parasites. In particular, this system simulates the behavior of collectives of trematode larvae (miracidia and cercariae).

Introduction

Now we are working on designing a biological computer on the medium of *Physarum polycepalum* [2]. This project is supported by FP7 (FP7-ICT-2011-8). The point is that the plasmodium of *Physarum polycepalum* behaves and moves as a giant amoeba and its behavior can be considered a computer processor [1]. This allows us to use the plasmodium of *Physarum p*. as a spatial computing programmed by different configurations of attractants (substances with potentially high nutritional value) and repellents (some illumination-, thermo- and salt-based conditions). In [24], we showed that the behavior of plasmodium of *Physarum polycephalum* has an own spatial logic which is one of the natural implementations of process calculus. This logic, called *Physarum spatial logic*, can be used as a programming language for the biological computer based on plasmodia. Also, this logic can simulate the behavior of *Schistosomatidae* (Trematoda: Digenea) [25].

The basic acts of plasmodium are as follows: (i) *Direction*: the plasmodium moves towards the attractant; (ii) *Fuse*: the two plasmodia fuse after meeting the same attractant; (iii) *Split*: the plasmodium splits in front of many attractants. The same acts are observed in the behavior of *Schistosomatidae* cercariae (see Fig. 1) and many other parasites. The difference is just in different attractants: plasmodia are attracted by nutrients and cercariae are attracted by fatty acids of bird or human skin.

Figure 1. The stimulation of the following operations in cercariae motions: (A) the fusion of cercariae collectives, (B) the multiplication of cercariae collective, where the human beings are attractants.



We are constructing an object-oriented programming language [26] for modelling the behavior of plasmodia by thousands attractants on the basis of the basic acts mentioned above. This complex behavior with many and many attractants, on the one hand, can implement complex computer algorithms and, on the other hand, show, how intelligent the *Physarum* plasmodium is. Within this object-oriented language we can model the behavior of collectives of the genus *Trichobilharzia* Skrjabin & Zakharov, 1920 (*Schistosomatidae* Stiles, Hassall 1898) and the behavior of many other parasites, as well.

Schistosomatidae have been studied recently because of cercarial dermatitis which they cause in humans by the secretion of penetration glands. Notice that the cercaria begins the incorporation into the human skin approximately in 8 seconds (the range from 0 to 80 seconds) just after first contacts [9]. The process of full penetration into the skin takes about 4 minutes (the range from 83 seconds till 13 minutes 37 seconds). In human kids, the cercariae can be brought by the venous blood into the lungs, causing hemorrhages and inflammations [3]. Many other complications can come because of repeated infestations [3].

The meaning of the behavior of parasites such as cercariae consists in the propagation of their collectives in all possible directions in looking for hosts and then their further propagation according to their life cycle. That behavior is concurrent and not linear (it continuously splits and fuses). In this paper, we propose a syllogistic system for a computer modelling of this complex behavior within our language [26]. In this system, attractants for parasites are considered syllogistic letters. Any two neighbor attractants which can be immediately reached by the one collective of parasites are considered a syllogistic string that is true if both attractants are occupied by parasites and false otherwise. The good science-fiction example of the propagation of parasites is presented by the movie *World War Z* (2013) directed by Marc Forster. Zombies in this movie are attracted by the chemotaxis from able-bodied humans, sounds and motions, which causes the propagation of zombies in all possible directions. Cercariae are attracted by the chemotaxis from a skin of potential hosts, turbulence of water and lights. Notice that the syllogistic system for modelling the propagation of parasites is proposed in this paper for the first time.

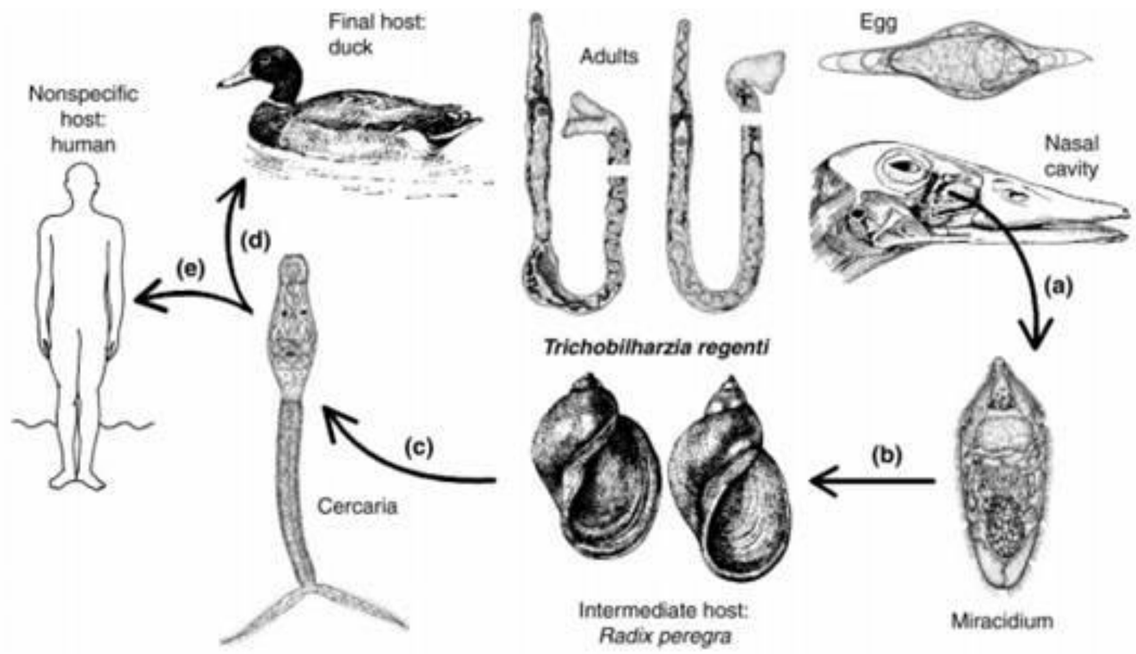
4.1. *Trichobilharzia szidati* (Diginea: Schistosomatidae) and their life cycle

All representatives of subclass *Digenea* Carus, 1863 (Platyhelminthes: Trematoda) including *Trichobilharzia szidati* are endoparasites of animals. Their life cycle has the form of heterogony: there are amphimictic and parthenogenetic stages. At these stages, digeneans have different outward, different ways of reproduction and different adaptation to different specific and incompetent hosts. The majority of them have the life cycle with participation of three hosts: intermediate, additional (metacercarial) and final. Molluscs are always the first intermediate hosts, while different classes of vertebrate animals are final hosts.

Among digeneans, there are parasites belonging to the family of *Schistosomatidae*, which have adapted to parasitizing in the circulatory system of vertebrate animals. This family includes the following three subfamilies: *Schistosomatinae* Stiles and Hassall, 1898, they parasitize a variety of birds and mammals, including human beings; *Bilharziellinae* Price, 1929 and *Gigantobilharziinae* Mehra, 1940, they parasitize birds. *Schistosomatidae* can penetrate the human skin, where they perish and, therefore, invoke allergic dermatitis.

The life cycle of all representatives of the family *Schistosomatidae* is identical (see Fig.2). Its members have the following two free-swimming stages: miracidia and cercariae, which actively search for their hosts (intermediate and final, respectively). Miracidial and cercarial host-finding is initiated mainly by response to some gravitational, light and chemical attractants.

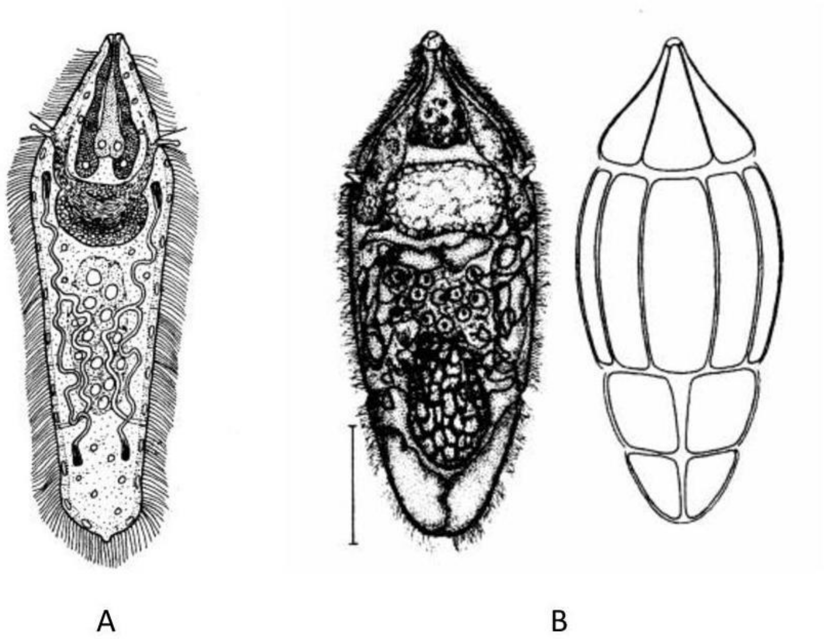
Figure 2. The life cycle of *Trichobilharzia regenti*: (a) mature worms lay the eggs in the nasal mucosa of ducks, then the eggs get to water and from these eggs miracidia hatch; (b) miracidia infect the intermediate hosts; (c) from these hosts cercariae are released; (d) they penetrate the skin of a specific host or (e) the skin of an incompetent mammalian host such as human being (from [16]).



4.1.1. Miracidia (Genus *Trichobilharzia*): morphology and behavior

Miracidia are free-swimming larval stage (Fig. 3). Their body surface is covered with four rows of epithelial plates that carry a multitude of cilia involved in active motions of miracidia (Fig. 3, B). The anterior part of the body contains terminal openings of the following penetration unicellular glands: two cells of apical glands and two cells of lateral glands. The posterior part of the body contains clusters of germ cells from which daughter sporosysts are formed subsequently. Miracidia lack any digestive system and they cannot feed.

Figure 3. The morphology of miracidia. A. the general morphology of *Trichobilharzia szidati* (from [23]); B. the general morphology of *Trichobilharzia regenti* (left) and the arrangement of ciliated plates (right), scale bar = 25 mm (from [16]).



Miracidia of *Trichobilharzia szidati* [23] hatch from eggs within a short time 5-10 min. In order to survive, miracidia must infect a snail host within 20h at 24°C before they die [23]. The positive photo- and negative geotactic orientation is an adaptation to reach the preferred habitats of their host-snail species [11], [12], [14], [33], [34]. Also, it is now confirmed that miracidia are attracted by some chemical host signals [19], [23], [31], [32]. Snail-hosts release various secreted/excreted products into water. Miracidia are activated by the macromolecular components of these products, which consist of glycoproteins larger than 30 kDa, termed *"miracidia attracting glycoproteins"* [17]. These glycoproteins produced by different species of snails have similar peptide-based, but differ by saccharide component. Miracidia can differentiate between these glycosylation patterns in order to find and infect mainly specific snail species [17], [18]. Thus, the chemical cues secreted by the freshwater gastropod *Lymnaea stagnalis* stimulate behavioral responses of *Trichobilharzia szidati* miracidia. In other words, a miracidium moves towards an appropriate chemical signal. Other kinds of attractants for miracidia are presented by light (there is a positive phototaxis) and gravitation (negative geotaxis). We will designate all miracidian attractants by syllogistic letters S_{m_1}, $S_{m_2}, \dots, S_{m_n}$, P_{m_1}, $P_{m_2}, \dots, P_{m_n}$, M_{m_1}, $M_{m_2}, \dots, M_{m_n}$. They can differ by their power and intensity.

4.1.2. Cercariae (Genus *Trichobilharzia*): morphology and behavior

Cercariae are free-swimming larvae of pubertal generation parasitizing vertebrate animals. Their length is about 1.0 mm. They are capable to insinuate into the skin of human being who is for them an incompetent host. As a result, they cause an allergic reaction, the so-called *cercarial dermatitis*. This term was proposed by V.V. Cort [4], who for the first time correlated this disease with molluscs of certain kinds, and then with cercariae.

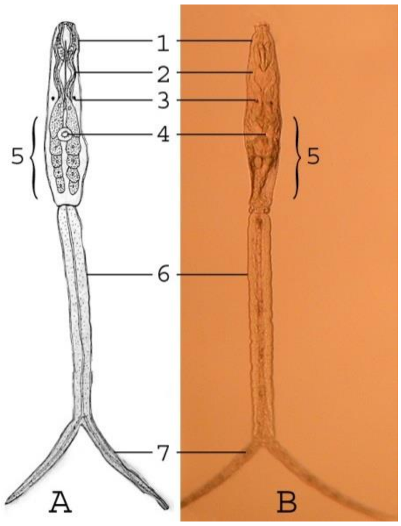
Cercariae of the genus *Trichobilharzia* belong to the bunch of furcocercariae, their posterior tail part consists of two branches (furcae), and the length of furca is approximately a half of length of tail. Even in a small zoom its pigmented eye-spots are well visible. The cercarial body is translucent. During motion it is strongly reduced, receiving various forms (Fig. 4).

Figure 4. The appearance of *Trichobilharzia szidati*. A detection place: the lake of Naroch (the Minsk region, the Mjadelsky destrict). *Lymnaea stagnalis* is an intermediate host.

All European species of *Trichobilharzia* possess the five pairs of penetration glands (Fig. 5). The two pairs are presented by circumacetabular glands located round the ventral sucker, and the three other by postacetabular glands located sequentially one after another below. The secreta of penetration glands helps larvae to break a dermal barrier of vertebrate hosts.

Cercariae of the genus *Trichobilharzia* after leaving a mollusc actively swimming in the water for an hour. Such an active behavior of larvae after leaving a mollusc provides a cercarial distribution in the water space. Then cercariae pass to a passive behavior. Free-swimming cercariae need to insinuate into a final host during the limited time interval (1-1.5 days at temperature 24°C) [23]. In a resting state, cercariae are attached to a vascular wall or on a water film by means of acetabulum. Active motions are characteristic only by the strong shaking of pot or by the water interfusion.

Figure 5. The constitution of *Trichobilharzia szidati*. A. The schematic structure of cercaria (from [5]), B. The photo of cercaria. 1. Penetration organ, 2. Penetration gland ducts, 3 Pigmented eye-spots, 4. Ventral sucker, 5. Penetration glands (5 pairs), 6. Tail stem, 7. Furcae.



The cercarial behavior of bird schistosomes (family *Schistosomatidae*) is well studied due to representatives of the genus *Trichobilharzia* [16]. Their behavior is characterized by the specific taxis implying their looking for specific hosts, their affixion to a surface of host body as well as their incorporation into a host cutaneous covering and their penetration into a circulatory system, where a parasite reaches sexual maturity. Thus, taxis is presented by an enough large family of attractants: larvae of digeneans of *Trichobilharzia* possess a positive phototaxis, negative geotaxis, chemotaxis, and also actively react to turbulence of water [8].

Larvae have chemoreceptors which receive appropriate chemical signals proceeding from a skin of potential host. The similarity of compound of fatty acids of bird and human skin leads to that cercariae equally react to the bird and human appearance in water: they move in their direction, and then they are attached to the skin and begin penetration into it [10]. In experimental researches, it has been shown that any attachment of cercariae of *Trichobilharzia* to the skin is stimulated by cholesterol and ceramides, and incorporation into the skin by linoleic and linolenic acids, all these materials are present on the skin of both the bird and the human being [9], [20]. Thereby surface lipids of human skin invoke higher frequency of cercarial incorporations into the skin, than surface lipids of birds [10].

Hence, the chemotaxis from a skin of potential hosts, the positive phototaxis, the negative geotaxis and the water turbulence should be considered cercarial attractants, which will be designated by S_{c_1}, S_{c_2}, … , S_{c_n}, P_{c_1}, P_{c_2}, … , P_{c_n}, M_{c_1}, M_{c_2}, … , M_{c_n}.

4.2. Syllogistic system for the propagation of *Schistosomatidae*

The alphabet of syllogistic system for modelling the propagation of *Schistosomatidae* contains:

- as descriptive signs the syllogistic letters S_x, P_x, M_x, …, where $x \in \{m,c\}$;
- as logical-semantic signs the syllogistic connectives:
  - $\mathbf{a}_t$, $\mathbf{e}_t$, i_t, $\mathbf{o}_t$ at time $t = 0,1,\ldots,n,\ldots$,
  - $\mathbf{a}_\infty$, $\mathbf{e}_\infty$, $\mathbf{i}_\infty$, $\mathbf{o}_\infty$ for infinite time;
- and the propositional connectives $\neg$, $\vee$, $\wedge$, $\Rightarrow$.

Simple propositions are defined as follows: S_xåP_x, where å $\in \{\mathbf{a}_t, \mathbf{e}_t, \mathrm{i}_t, \mathbf{o}_t, \mathbf{a}_\infty, \mathbf{e}_\infty, \mathbf{i}_\infty, \mathbf{o}_\infty\}$ All other propositions are defined thus: (i) each simple proposition is a proposition, (ii) if X, Y are propositions, then $\neg X$, $\neg Y$, XåY, where å $\in \{\vee, \wedge, \Rightarrow\}$, are propositions, too.

Syllogistic letters S_m, P_m, M_m, … are interpreted as attractants for miracidia as follows: a data point S_m is evaluated as empty if and only if an appropriate attractant for miracidia denoted by S_m is not occupied by any miracidium. Syllogistic letters S_c, P_c, M_c, … are interpreted as attractants for cercariae in the following manner: an item S_c is evaluated as empty if and only if an appropriate attractant for cercariae denoted by S_c is not occupied by any cercaria.

Let us define syllogistic strings of the form $S_m P_m$ and $S_c P_c$ at time t with the following notation: '$S_m \mathrm{is}_t P_m$' and '$S_c \mathrm{is}_t P_c$,' and with the following meaning:

- $S_m P_m$ at time $t = 0,1,2,\ldots$ is true if and only if S_m and P_m are cells occupied by miracidia at t and there is a path between S_m and P_m and this path consists of cells also occupied by miracidia at t (i.e. both S_m and P_m are not empty at t and between them there is a path of non-empty cells), otherwise $S_m P_m$ is false (e.g. in Fig.6 there are paths $S_4 S_5$, $S_4 S_{10}$ at $t = 0$);

- $S_c P_c$ at t is true if and only if both S_c and P_c are not empty cells at t and between them there is a path of non-empty cells at t, otherwise $S_c P_c$ is false.

Now, we define syllogistic propositions 'All ... are ...', 'No ... are ...', 'Some ... are ...', 'Some ... are not ...' in the non-Aristotelian way to construct a syllogistic of propagation. The meaning of the proposition 'All S are P' is that between points S and P we observe a propagation in all possible directions. The meaning of the proposition 'Some S are P' is that between points S and P we cannot observe a propagation in all possible directions and the propagation is just contingent and casual. So, while in the Aristotelian syllogistic the propositions 'All ... are ...' and 'No ... are ...' are contrary, in the syllogistic of propagation the propositions 'All ... are ...' and 'Some ... are ...' are contrary. Other propositions are understood conventionally: 'No ... are ...' := It is false that 'Some ... are ...'; 'Some ... are not ...' := It is false that 'All ... are ...'. The point of this non-Aristotelian interpretation is that we cannot exclude propagation at all. There are always some forms of propagation from massive-parallel (when 'All ... are ...' is true) to casual (when 'Some ... are ...' is true). For more details about this non-Aristotelian system, please see [27], [28].

Thus, using the definition of syllogistic strings, we can define simple syllogistic propositions as follows:

- 'All S_m are P_m at time t' ($S_m \mathbf{a}_t P_m$): there is a string $A_m S_m$ at time t and for any A_m which is a neighbor for S_m or P_m, there are strings $A_m S_m$ and $A_m P_m$ at t. This means that we have a massive-parallel occupation of the region at t, where the cells S_m and P_m are located, i.e. the propagation holds in all possible directions.
- 'All S_c are P_c at time t' ($S_c \mathbf{a}_t P_c$): there is a string $A_c S_c$ at time t and for any A_c which is a neighbor for S_c or P_c, there are strings $A_c S_c$ and $A_c P_c$ at t.
- 'Some S_m are P_m at time t' ($S_m \mathbf{i}_t P_m$): for any A_m which is a neighbor for S_m or P_m at t, there are no strings $A_m S_m$ and $A_m P_m$. This means that the collective of miracidia cannot reach S_m from P_m or P_m from S_m immediately at t, but it does not mean that there are no propagating miracidia. Some forms of their propagation ever exist.
- 'Some S_c are P_c at time t' ($S_c \mathbf{i}_t P_c$): for any A_c which is a neighbor for S_c or P_c at t, there are no strings $A_c S_c$ and $A_c P_c$.
- 'No S_m are P_m at time t' ($S_m \mathbf{e}_t P_m$): there exists A_m at time t which is a neighbor for S_m or P_m such that there is a string $A_m S_m$ or there is a string $A_m P_m$. This means that the collective of miracidia occupies S_m or P_m, but surely not the whole region at time t, where the cells S_m and P_m are located.

- ‘No S_c are P_c at time t’ ($S_c\mathbf{e}_tP_c$): there exists A_c at time t which is a neighbor for S_c or P_c such that there is a string A_cS_c or there is a string A_cP_c.
- ‘Some S_m are not P_m at time t’ ($S_m\mathbf{o}_tP_m$): for any A_m which is a neighbor for S_m or P_m at time t there is no string A_mS_m or there exists A_m which is a neighbor for S_m and P_m such that there is no string A_mS_m or there is no string A_mP_m. This means that at time t the collective of miracidia does not occupy S_m or there is a neighbor cell which is not connected with S_m or P_m by the same propagated collective of miracidia.
- ‘Some S_c are not P_c at time t’ ($S_c\mathbf{o}_tP_c$): for any A_c which is a neighbor for S_c or P_c at time t there is no string A_cS_c or there exists A_c which is a neighbor for S_c or P_c such that there is no string A_cS_c or there is no string A_cP_c.

Formally:

$$S_m\mathbf{a}_tP_m := (\exists A_m(A_m \mathrm{is}_t S_m) \wedge (\forall A_m(A_m \mathrm{is}_t S_m \wedge A_m \mathrm{is}_t P_m))); \tag{1}$$

$$S_c\mathbf{a}_tP_c := (\exists A_c(A_c \mathrm{is}_t S_c) \wedge (\forall A_c(A_c \mathrm{is}_t S_c \wedge A_c \mathrm{is}_t P_c))); \tag{2}$$

$$S_m\mathbf{i}_tP_m := \forall A_m(\neg(A_m \mathrm{is}_t S_m) \wedge \neg(A_m \mathrm{is}_t P_m)); \tag{3}$$

$$S_c\mathbf{i}_tP_c := \forall A_c(\neg(A_c \mathrm{is}_t S_c) \wedge \neg(A_c \mathrm{is}_t P_c)); \tag{4}$$

$$\begin{gathered} S_m\mathbf{e}_tP_m := \neg\forall A_m(\neg(A_m \mathrm{is}_t S_m) \wedge \neg(A_m \mathrm{is}_t P_m)), i.e. \\ \exists A_m(A_m \mathrm{is}_t S_m \vee A_m \mathrm{is}_t P_m). \end{gathered} \tag{5}$$

$$\begin{gathered} S_c\mathbf{e}_tP_c := \neg\forall A_c(\neg(A_c \mathrm{is}_t S_c) \wedge \neg(A_c \mathrm{is}_t P_c)), i.e. \\ \exists A_c(A_c \mathrm{is}_t S_c \vee A_c \mathrm{is}_t P_c). \end{gathered} \tag{6}$$

$$\begin{gathered} S_m\mathbf{o}_tP_m := \neg(\exists A_m(A_m \mathrm{is}_t S_m) \vee (\forall A_m(A_m \mathrm{is}_t P_m \wedge A_m \mathrm{is}_t S_m))), \\ i.e. (\forall A_m \neg(A_m \mathrm{is}_t S_m) \wedge \exists A_m(\neg(A_m \mathrm{is}_t P_m) \vee \neg(A_m \mathrm{is}_t S_m))); \end{gathered} \tag{7}$$

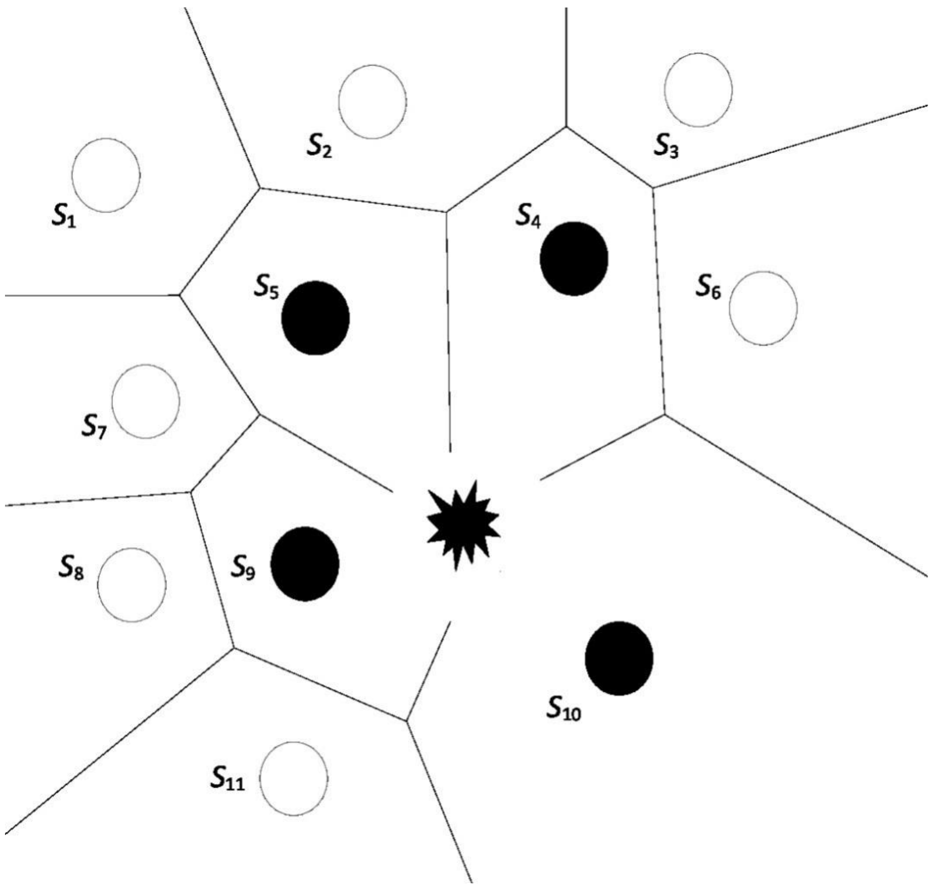
$$\begin{gathered} S_c\mathbf{o}_tP_c := \neg(\exists A_c(A_c \mathrm{is}_t S_c) \vee (\forall A_c(A_c \mathrm{is}_t P_c \wedge A_c \mathrm{is}_t S_c))), \\ i.e. (\forall A_c \neg(A_c \mathrm{is}_t S_c) \wedge \exists A_c(\neg(A_c \mathrm{is}_t P_c) \vee \neg(A_c \mathrm{is}_t S_c))); \end{gathered} \tag{8}$$

Notably, this system is essentially non-Aristotelian, in particular

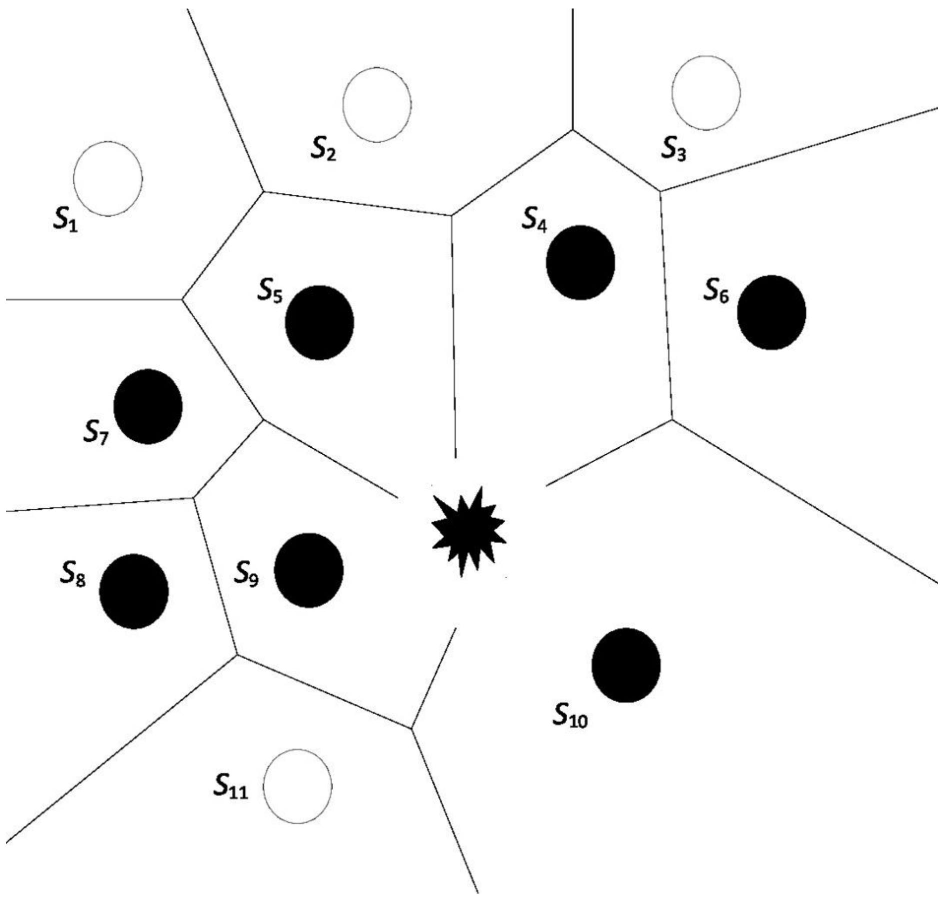
$S_m \mathbf{a}_t P_m \Rightarrow S_m \mathbf{e}_t P_m$; $S_m \mathbf{i}_t P_m \Rightarrow S_m \mathbf{o}_t P_m$; $S_m \mathbf{a}_t P_m \Rightarrow P_m \mathbf{a}_t S_m$; $S_m \mathbf{e}_t P_m \Rightarrow P_m \mathbf{e}_t S_m$; $S_m \mathbf{i}_t P_m \Rightarrow P_m \mathbf{i}_t S_m$; $S_m \mathbf{o}_t P_m \Rightarrow P_m \mathbf{o}_t S_m$.

The topology of attractants changes permanently (see Fig. 6), because different attractants become occupied in due course. Therefore for different time $t = 0,1,2,\ldots$, we observe different true syllogistic propositions (see Fig. 6).

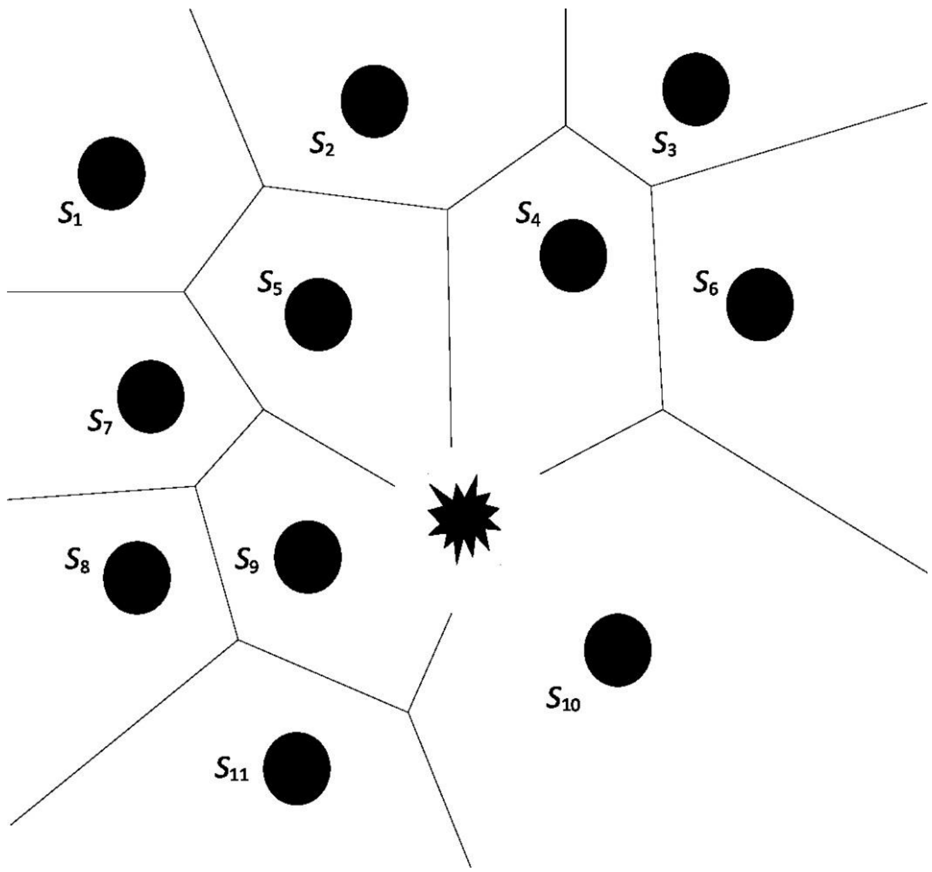
Figure 6. Example of propagation of parasites. First of all, parasites are located in the center of pictures (a), (b), (c). Attractants which are not occupied by parasites are denoted by white circles. Attractants which are occupied by parasites are denoted by black circles. Each attractant is a center of suitable Voronoi cell. The radius of Voronoi cell is a maximal distance of power to attract parasites in one direction. At time $t = 0,1,2$, different syllogistic propositions are true: (a) $S_7 \mathbf{i}_0 S_{11}$, $S_{11} \mathbf{i}_0 S_7$, $S_1 \mathbf{i}_0 S_3$, $S_3 \mathbf{i}_0 S_1$, $S_1 \mathbf{e}_0 S_{10}$, $S_{10} \mathbf{e}_0 S_1$, $S_1 \mathbf{e}_0 S_9$, $S_9 \mathbf{e}_0 S_1$, $S_1 \mathbf{e}_0 S_4$, $S_4 \mathbf{e}_0 S_1$, etc.; (b) $S_1 \mathbf{i}_1 S_{11}$, $S_{11} \mathbf{i}_1 S_1$, $S_2 \mathbf{o}_1 S_6$, $S_6 \mathbf{o}_1 S_2$, etc.; (c) $S_7 \mathbf{a}_2 S_{11}$, $S_{11} \mathbf{a}_2 S_7$, $S_1 \mathbf{a}_2 S_3$, $S_3 \mathbf{a}_2 S_1$, $S_1 \mathbf{a}_2 S_{10}$, $S_{10} \mathbf{a}_2 S_1$, $S_1 \mathbf{a}_2 S_9$, $S_9 \mathbf{a}_2 S_1$, $S_1 \mathbf{a}_2 S_4$, $S_4 \mathbf{a}_2 S_1$, etc.



(a) $t = 0$; the four attractants denoted by S_4, S_5, S_9, S_{10} are occupied by the parasites.



(b) $t = 1$; the seven attractants denoted by S_4, S_5, S_6, S_7, S_8, S_9, S_{10} are occupied by the parasites.



(c) $t = 2$; the eleven attractants denoted by S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11} are occupied by the parasites.

Also, we can assume that the attractants can move at $t = 0,1,2\ldots$ Let us postulate that their motions at different time are limited by the following rules, where $t+1$ means the next step of life cycle:

$$S_m \mathbf{a}_t P_m \Rightarrow S_m \mathbf{e}_{t+1} P_m. \tag{9}$$

$$S_c \mathbf{a}_t P_c \Rightarrow S_c \mathbf{e}_{t+1} P_c. \tag{10}$$

From this we have

$$S_m \mathbf{i}_{t+1} P_m \Rightarrow S_m \mathbf{o}_t P_m. \tag{11}$$

$$S_c \mathbf{i}_{t+1} P_c \Rightarrow S_c \mathbf{o}_t P_c. \tag{12}$$

Attractants for miracidia and cercariae are different, but they live in the same lake. We can postulate that if the whole region is occupied by miracidia (cercariae) at time t, then surely the part of this region is occupied by cercariae (miracidia) at time $t+1$. So, the propagation of collectives of miracidia at t causes the propagation of collectives of cercariae at $t+1$:

$$S_m \mathbf{a}_t P_m \Rightarrow S_c \mathbf{e}_{t+1} P_c. \quad (13)$$

$$S_m \mathbf{i}_{t+1} P_m \Rightarrow S_c \mathbf{o}_t P_c. \quad (14)$$

Furthermore, the propagation of collectives of cercariae at t causes the propagation of collectives of miracidia at $t+1$:

$$S_c \mathbf{a}_t P_c \Rightarrow S_m \mathbf{e}_{t+1} P_m. \quad (15)$$

$$S_c \mathbf{i}_{t+1} P_c \Rightarrow S_m \mathbf{o}_t P_m. \quad (16)$$

Now, we can define simple syllogistic propositions for infinite time: $S\mathbf{x}_\infty P := \lim_{t\to\infty} S\mathbf{x}_t P$, where $\mathbf{x} \in \{\mathbf{a}, \mathbf{e}, \mathbf{i}, \mathbf{o}\}$. The informal meaning of these propositions is that we put forward syllogistic propositions about the whole one life cycle of *Schistosomatidae* in the same lake. So, we may generalize equations (9) – (16) in the following way:

$$S_m \mathbf{a}_\infty P_m \Rightarrow \exists t. S_m \mathbf{a}_t P_m; \quad (17)$$

$$S_c \mathbf{a}_\infty P_c \Rightarrow \exists t. S_c \mathbf{a}_t P_c; \quad (18)$$

$$S_m \mathbf{a}_\infty P_m \Rightarrow \exists t. S_c \mathbf{a}_t P_c; \quad (19)$$

$$S_c \mathbf{a}_\infty P_c \Rightarrow \exists t. S_m \mathbf{a}_t P_m; \quad (20)$$

$$S_m \mathbf{i}_\infty P_m \Rightarrow \forall t. S_m \mathbf{i}_t P_m; \quad (21)$$

$$S_c \mathbf{i}_\infty P_c \Rightarrow \forall t. S_c \mathbf{i}_t P_c; \quad (22)$$

$$S_m \mathbf{i}_\infty P_m \Rightarrow \forall t. S_c \mathbf{i}_t P_c; \quad (23)$$

$$S_c \mathbf{i}_\infty P_c \Rightarrow \forall t. S_m \mathbf{i}_t P_m. \quad (24)$$

Formula (17) means that if the whole region is occupied by miracidia for one life cycle of *Schistosomatidae*, then there exists time t such that the whole region is occupied by miracidia. Formula (18) means that if the whole region is occupied by cercariae for one life cycle of *Schistosomatidae*, then there exists time t such that the whole region is occupied by cercariae. Formula (19) means that if the whole region is occupied by miracidia for one life cycle of *Schistosomatidae*, then there exists time t such that the whole region is occupied by cercariae. Formula (20) means that if the whole region is occupied by cercariae for one life cycle of *Schistosomatidae*, then there exists time t such that the whole region is occupied by miracidia.

Formula (21) means that if there is just a casual propagation of miracidia for one life cycle of *Schistosomatidae,* then for all time t of this life cycle there is just a casual propagation of miracidia. Formula (22) means that if there is just a casual propagation of cercariae for one life cycle of *Schistosomatidae,* then for all time t of this life cycle there is just a casual propagation of cercariae. Formula (23) means that if there is just a casual propagation of miracidia for one life cycle of *Schistosomatidae,* then for all time t of this life cycle there is just a casual propagation of cercariae. Formula (24) means that if there is just a casual propagation of cercariae for one life cycle of *Schistosomatidae*, then for all time t of this life cycle there is just a casual propagation of miracidia.

Other axioms are as follows (for $x \in \{m, c\}$):

$$S_x \mathbf{a}_t P_x \Rightarrow S_x \mathbf{e}_t P_x;\ S_x \mathbf{a}_\infty P_x \Rightarrow S_x \mathbf{e}_\infty P_x; \tag{25}$$

$$S_x \mathbf{a}_t P_x \Rightarrow P_x \mathbf{a}_t S_x;\ S_x \mathbf{a}_\infty P_x \Rightarrow P_x \mathbf{a}_\infty S_x; \tag{26}$$

$$S_x \mathbf{i}_t P_x \Rightarrow P_x \mathbf{i}_t S_x;\ S_x \mathbf{i}_\infty P_x \Rightarrow P_x \mathbf{i}_\infty S_x; \tag{27}$$

$$S_x \mathbf{a}_t M_x \Rightarrow S_x \mathbf{e}_t P_x;\ S_x \mathbf{a}_\infty M_x \Rightarrow S_x \mathbf{e}_\infty P_x; \tag{28}$$

$$M_x \mathbf{a}_t P_x \Rightarrow S_x \mathbf{e}_t P_x;\ M_x \mathbf{a}_\infty P_x \Rightarrow S_x \mathbf{e}_\infty P_x; \tag{29}$$

$$(M_x \mathbf{a}_t P_x \wedge S_x \mathbf{a}_t M_x) \Rightarrow S_x \mathbf{a}_t P_x; \tag{30}$$

$$(M_x \mathbf{i}_t P_x \wedge S_x \mathbf{i}_t M_x) \Rightarrow S_x \mathbf{i}_t P_x; \tag{31}$$

$$(M_x \mathbf{a}_\infty P_x \wedge S_x \mathbf{a}_\infty M_x) \Rightarrow S_x \mathbf{a}_\infty P_x; \tag{32}$$

$$(M_x \mathbf{i}_\infty P_x \wedge S_x \mathbf{i}_\infty M_x) \Rightarrow S_x \mathbf{i}_\infty P_x. \tag{33}$$

Some basic formal properties of that axiomatic system closed over axioms (1) – (8) and (25) – (33), where $\mathbf{a}$, $\mathbf{e}$, $\mathbf{i}$, $\mathbf{o}$ do not depend on time, are considered in [27], [28]. Probabilistic semantics for the time-depended version of that system for the slime mould behavior is proposed in [29].

Notice that formulas (31) and (33) are axioms of the syllogistic system of propagation, while they are not valid in the Aristotelian system. Indeed, $(M_x \mathbf{i}_t P_x \wedge S_x \mathbf{i}_t M_x) \Rightarrow S_x \mathbf{i}_t P_x$ means that

$$[\forall A_x (\neg(A_x \mathrm{is}_t M_x) \wedge \neg(A_x \mathrm{is}_t P_x)) \wedge \forall A_x (\neg(A_x \mathrm{is}_t S_x) \wedge \neg(A_x \mathrm{is}_t M_x))]$$

$$\Rightarrow \forall A_x(\neg(A_x \mathrm{is}_t S_x) \wedge \neg(A_x \mathrm{is}_t P_x)),$$

the latter formula is valid.

Conclusion

In this paper, we have constructed the syllogistic system for the propagation of *Schistosomatidae*. This system contains all tautologies of the syllogistic system for the propagation of *Physarum polycephalum* proposed in [29], but not vice versa. And in return, the syllogistic system for the propagation of *Physarum polycephalum* contains all tautologies of performative syllogistic proposed in [27], [28], but not vice versa. This means that as more complex intelligent behavior is observed in propagations as more axioms are contained in an appropriate syllogistic system.

In syllogistics for the explication of propagations, the inverse relations hold true for all syllogistic connectives:

$$S\mathbf{a}P \Rightarrow P\mathbf{a}S;$$

$$S\mathbf{e}P \Rightarrow P\mathbf{e}S;$$

$$S\mathbf{i}P \Rightarrow P\mathbf{i}S;$$

$$S\mathbf{o}P \Rightarrow P\mathbf{o}S.$$

It is inferred from definitions (1) – (8). These properties fix propagations in all possible directions.

Acknowledgment

This research is being fulfilled by the support of FP7-ICT-2011-8 and UMO-2012/07/B/HS1/00263.

References

[1] Adamatzky, A., *Physarum Machines: Computers from Slime Mould (World Scientific Series on Nonlinear Science, Series A),* World Scientific Publishing Company, 2010.

[2] Adamatzky, A., Erokhin, V., Grube, M., Schubert, Th., Schumann, A., *Physarum Chip Project: Growing Computers From Slime Mould,* International Journal of Unconventional Computing, 8 (4), pp. 319–323, 2012.

[3] Bear, S.A., Voronin, M.V., *Cercariae in Urbanized Ecosystems,* Nauka, Moskow, 2007 (in Russian).

[4] Cort, W.W., *Schistosome dermatitis in the United States (Michigan),* Journal of the American Medical Association, vol. 90, pp. 1027–1029, 1928.

[5] Ginetsinskaya, T.A., *Trematodes, their Life Cycles, Biology and Evolution,* Nauka, Leningrad, 1968 (in Russian).

[6] Feiler, W., Haas, W., *Trichobilharzia ocellata*: *chemical stimuli of duck skin for cercarial attachment,* Parasitology, vol. 96, pp. 507–517, 1988.

[7] Haas, W., *Host finding mechanisms – a physiological effect,* [in:] Mehlhorn, Y. (ed.), *Biology, Structure, Function: Encyclopedic Reference of Parasitology,* edn 2, pp. 382–383, 1988.

[8] Haas, W., *Physiological analysis of cercarial behavior,* Journal of Parasitology, vol. 78, pp. 243–255, 1992.

[9] Haas, W., Haeberlein, S., *Penetration of cercariae into the living human skin: Schistosoma mansoni* vs. *Trichobilharzia szidati, Parasitology Research*, vol. 105, no. 4, pp. 1061–1066, 2009.

[10] Haas, W., Roemer, A., *Invasion of the vertebrate skin by cercariae of Trichobilharzia ocellata*: *penetration processes and stimulating host signals, Parasitology Research*, vol. 84, no. 10, pp. 787–795, 1998.

[11] Haas, W., *Parasitic worms: strategies of host finding, recognition and invasion,* Zoology, vol. 106, pp. 349–364, 2003.

[12] Haas, W., Haberl, B., Kalbe, M., Körner, M., *Snail-host-finding by Miracidia and Cercariae: Chemical Host Cues, Parasitology Today,* vol. 11, pp. 468–472, 1995.

[13] Haas, W., Haberl, B., *Host recognition by trematode miracidia and cercariae,* Advances in Trematode Biology, CRC Press, Boca Raton, Florida, pp. 197–227, 1997.

[14] Hertel, J., Holweg, A., Haberl, B., Kalbe, M., Haas, W., *Snail odour-clouds: spreading and contribution to the transmission success of Trichobilharzia ocellata (Trematoda, Digenea) miracidia,* Oecologia, vol. 147, pp. 173–180, 2006.

[15] Horák, P., Kolárová, L., *Bird schistosomes: do they die in mammalian skin? Trends in Parasitology*, vol. 17, no. 2, pp. 66–69, 2001.

[16] Horák, P., Kolárová, L., Adema, C.M., *Biology of the schistosome genus Trichobilharzia, Advances in Parasitology*, 52, pp. 155–233, 2002.

[17] Kalbe, M., Haberl, B., Haas, W., *Finding of the snail host by Fasciola hepatica and Trichobilharzia ocellata: compound analysis of 'Miracidia attracting glycoprotein',* Experimental Parasitology, vol. 96, pp. 231–242, 2000.

[18] Kock, S., *Investigations on intermediate host specificity help to elucidate the taxonomic status of Trichobilharzia ocellata (Digenea: Schistosomatidae*), Parasitology, vol. 123, pp. 67–70, 2001.

[19] MacInnis, A.J., *How parasites find their hosts: some thoughts on the inception of host-parasite integration,* Ecological Aspects of Parasitology (Cennedy, C.R. ed.), Amsterdam: North-Holand, pp. 3–20, 1976.

[20] Mikes, L., Zìdková, L., Kasný, M., Dvorák, J., Horák, P., *In vitro stimulation of penetration gland emptying by Trichobilharzia szidati and T. regenti (Schistosomatidae) cercariae. Quantitative collection and partial characterization of the products,* Parasitology Research, vol. 96, no. 4, pp. 230–241, 2005.

[21] Molyneux, D.H., *Control of human parasitic diseases: context and overview,* Advances in Parasitology, vol. 61, pp. 1–45, 2006.

[22] Neuhaus, W., *Biologie und Entwicklund von Trichobilharzia szidati n. sp. (Trematoda, Schistosomatidae), einem Erreger von Dermatitis beim Menschen,* Zeitschrift fuer Parasitenkunde, vol. 15, pp. 203–266, 1952.

[23] Saladin, K.S., *Behavioral parasitology and perspectives on miracidial host-finding,* Zeitschrift fuer Parasitenkunde, vol. 60, no. 3, pp. 197–210, 1979.

[24] Schumann, A., Adamatzky, A., *Physarum Spatial Logic, New Math. and Nat. Computation*, vol. 7, no. 3, pp. 483–498, 2011.

[25] Schumann, A., Akimova, L., *Simulating of Schistosomatidae (Trematoda: Digenea) Behavior by Physarum Spatial Logic, Annals of Computer Science and Information Systems, vol. 1., Proceedings of the 2013 Federated Conference on Computer Science and Information Systems,* IEEE Xplore, pp. 225–230, 2013.

[26] Schumann, A., Pancerz, K., *Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing,* [in:] Szczuka, M., Czaja, L., Kacprzak, M. (eds.), *Proceedings of the Workshop on Concurrency, Specification and Programming* (CS&P'2013), Warsaw, Poland, September 25-27, pp. 389–397, 2013.

[27] Schumann, A., *On Two Squares of Opposition: the Lesniewski's Style Formalization of Synthetic Propositions, Acta Analytica*, vol. 28, pp. 71–93, 2013.

[28] Schumann, A., *Two Squares of Opposition: for Analytic and Synthetic Propositions, Bulletin of the Section of Logic*, vol. 40(3/4), pp. 165–178, 2011.

[29] Schumann, A., *p-Adic Valued Fuzzyness and Experiments with Physarum Polycephalum, The 11th International Conference on Fuzzy Systems and Knowledge Discovery,* IEEE Xplore, 2014 (accepted).

[30] Smyth, J.D., Halton, D.W., *The Physiology of Trematodes,* 2nd edition, Cambridge: Cambridge University Press, 1983.

[31] Sukhdeo, M.V.K., Mettrick, D.F., *Parasite behaviour: understanding platyhelminth responses,* Advances Parasitology, vol. 26, pp. 73–144, 1987.

[32] Sukhdeo, M.V.K., Sukhdeo, S.C., *Trematode behaviours and the perceptual worlds of parasites, Canadian Journal of Zoology*, vol. 82, pp. 292–315, 2004.

[33] Takahashi, T., Mori, K., Shigeta, Y., *Phototactic, thermotactic and geotactic responses of miracidia of Schistosoma japonicum, Japanese Journal of Parasitology,* vol. 110, pp. 686–691, 1961.

[34] Podhorsky, M., Huzova, Z., Mikes, L., Horak, P., *Cercarial dimensions and surface structures as a tool for species determination of Trichobilharzi*a spp. *Acta Parasitologica*, vol. 50, pp. 343–365, 2009.

Originally published: Schumann, A., Adamatzky, A., *Physarum Polycephalum Diagrams for Syllogistic Systems*, IfCoLog Journal of Logics and their Applications , 2 (1), pp. 35-68, 2015.

V

Physarum Polycephalum Diagrams for Syllogistic Systems

Andrew Schumann, Andrew Adamatzky

Abstract

In this paper, we propose syllogistic circuits on the medium of *Physarum polycephalum* plasmodium. These circuits are designed by chemical signals stimulating the plasmodium propagation. As a result, the circuits are built up on continuously growing plasmodia and any their motions are considered syllogistic conclusions. We show how we can implement different syllogistic systems on the medium of *Physarum polycephalum* plasmodia within these circuits, namely we implement the following systems: the Aristotelian syllogistic, the performative syllogistic and Talmudic reasoning by *qal wa-homer*. In this way, we can design biological devices, where inputs are represented not by electrical signals, but by chemical signals in accordance with the plasmodium chemotaxis, and outputs are represented by the plasmodium behavior.

Introduction

One of the first logicians who proposed a spatial implementation of Aristotelian syllogistic reasoning was Lewis Carroll [3], [4]. He used three kinds of syllogistic propositions: (i) the universal affirmative ('all members of its subject are members of its predicate'), (ii) the universal negative ('no members of its subject are members of its predicate'), (iii) and the particular affirmative ('some members of its subject are members of its predicate'). His examples are as follows: 'All red apples are ripe', 'No red apples are ripe', 'Some red apples are ripe'. For verifying syllogistic propositions he proposed the following biliteral diagram:

xy	xy'
$x'y$	$x'y'$

that plays the role of 'universe of discourse' for all syllogistic propositions over adjuncts x, y, non-x (which is denoted by x'), non-y (which is denoted by y'). For example, let x mean 'old,' so that x' will mean 'new'. Let y mean 'English,' so that y' will mean 'foreign'. Assume that ‘books’ are an appropriate universe of discourse. Then we can divide this universe into the following four classes: *xy* (‘old English books’), *xy'* (‘old foreign books’), *x'y* (‘new English books’), *x'y'* (‘new foreign books’).

Now let us take two kinds of counters: grey and black. If a black counter is placed within a cell, this means that “this cell is occupied” (i.e. “there is at least one thing in it”). If a grey counter is placed within a cell, this means that “this cell is empty” (i.e. “there is nothing in it”). Thus, using grey and black counters we can verify all the basic syllogistic propositions.

In this paper, we develop Carroll's ideas, but our diagrams will be represented by living organisms, namely by *Physarum* behaviors. In *Physarum Chip Project: Growing Computers From Slime Mould* supported by FP7 we are going to build a programmable amorphous biological computer. In this computer, logic circuits are represented by programmable behaviors of *Physarum polycephalum* plasmodium – the one-call organism that behaves according to different stimuli called attractants and repellents and propagates networks connecting all reachable food sources. When designing an object-oriented programming language for the simulation of plasmodium behaviors [16], we detect a possibility on the media of *Physarum* to construct logic circuits for different symbolic-logical systems, including syllogistic systems. The circuits for various syllogistics may be built in the way of Carroll's diagrams. One of the most interesting outcomes of this research is that, first, the universe of discourse is considered a continuously growing living organism (*Physarum* plasmodium), second, any motions of this organism are considered as inferring syllogistic conclusions. Carroll's like diagrams allow us to define all possible directions in any motion of *Physarum*.

In this paper, we show how we can implement the Aristotelian syllogistic (Section 5.3), the performative syllogistic (Section 5.4), and Talmudic reasoning (Section 5.5) via *Physarum* plasmodium behavior.

5.1. Biological computations by chemical signals

The plasmodium propagation can be interpreted as a kind of computation which can solve different tasks which assume concurrency: maze-solving [9], solving the Steiner Problem [20], minimum-risk path finding [10], [11], solving the traffic optimization problem [24], associative learning [18], memorizing and anticipating repeated events [12], etc. The feature of these computations is that a plasmodium black box always has much more outputs than inputs. We can generalize this property and claim that this feature concerns any computation on behavioral systems of living organisms. It is related to the problem of free will, where living organisms face a choice among several outputs given just one input: they can choose just one or simultaneously many outputs according to their will. Evidently, this is not typical for the electronic devices.

Hence, we can assume that it is possible to design a biological device, where instead of electrical signals the calculation process is performed by using the plasmodium *chemotaxis* [2] (about other approaches to information processing by chemotaxis see [6]), i.e. knowledge of the typical reactions of plasmodium to chemical signals attracting or repelling the plasmodium behavior. In this device, the number of outputs usually is much larger than the number of inputs, where inputs are chemical signals attracting or repelling the behavior and outputs are behavioral responses. We can artificially delete some outputs by using additional repellents. Nevertheless, as more repellents in complex gates as lower accuracy of computations in this device.

Notably, the accuracy of implementing classical logics and Turing-complete machines, such as Kolmogorov-Uspensky machines [22], [1], in the plasmodium behavior is low. For instance, let computing circuits be implemented using a collision-based computing approach, where the plasmodium propagates on non-nutrient substrate in the form of a compact wave-fragment of protoplasm. Logic gates can be constructed in the laboratory conditions as follows: $\langle x, y\rangle \to \langle x \text{ AND } y, x \text{ OR } y\rangle$ and $\langle x, y\rangle \to \langle x, \text{ NOT } x \text{ AND } y\rangle$. Then the accuracy of experimental laboratory prototypes of the gate $\langle x, y\rangle \to \langle x \text{ AND } y, x \text{ OR } y\rangle$ was over 69% and of the gate $\langle x, y\rangle \to \langle x, \text{ NOT } x \text{ AND } y\rangle$ was over 59%. In the frequency-based Boolean logical gates implemented with *Physarum* based on frequencies of oscillations, the accuracy is as follows: 90% for OR/NOR, 77.8% for AND/NAND, 91.7% for NOT, 70.8% for XOR/XNOR. Notice that as more complex circuits, as lower accuracy of their implementation.

This problem with accuracy is linked to the fact that the plasmodium as living organism wants to be propagated in all possible directions (to possess more outputs than inputs). This means that the plasmodium does not follow the induction principle to satisfy the minimal conditions in funding and occupying the food sources, e.g. by shortest distances. In other words, the plasmodium behavior is context-dependent and it is not a kind of spatial Turing-complete machines such as Kolmogorov-Uspensky machines. We can implement these machines only with some accuracy using repellents, because to follow these machines and classical logic gates is not natural for behavioral systems.

Usually, computations on *Physarum* are studied to approximate transportation systems and hierarchies of planar proximity graphs, e.g. to approximate concurrent phenomena. In our research, we are going to obtain an abstract chip based on plasmodium and we need to find out logical systems which are natural for the plasmodium behavior and, therefore, they could be implemented with a high accuracy.

In this paper, we propose the implementation of Aristotelian syllogistic of [8] as a system whose semantics can be expressed by Kolmogorov-Uspensky machines. In this implementation we need many repellents and, thus, the behavior of plasmodium is not so natural here. Then we show how we can implement the performative syllogistic of [15]. For this implementation we do not need repellents at all. The performative syllogistic, on the one hand, is more expressive than the Aristotelian syllogistic (it contains all tautologies of the latter, see [15]), on the other hand, its semantics is context-based and depends on neighbors for plasmodium active zones. Due to this semantics, the performative syllogistic is sound and complete on plasmodium propagations which are performed without repellents. This semantics cannot be expressed by Kolmogorov-Uspensky machines.

Logic circuits constructed on the basis of the performative syllogistic seem to be natural for behavioral systems and these circuits have very high accuracy in implementing. Our general motivation in designing logic circuits in behavioral systems without repellents is as follows:

- in this way, we can present behavioral systems as a calculation process more naturally;
- we can design devices, where there are much more outputs than inputs, for performing massive-parallel computations in the bio-inspired way;
- we can obtain unconventional (co)algorithms by programming behavioral systems.

Biological and bio-inspired computations allow us to perform calculations through behaviors of living organisms. The main disadvantage of these computations is that biological devices are much slower than digital computers, but they can effectively solve complex massive-parallel tasks.

5.2. Strings in the *Physarum* growing universe

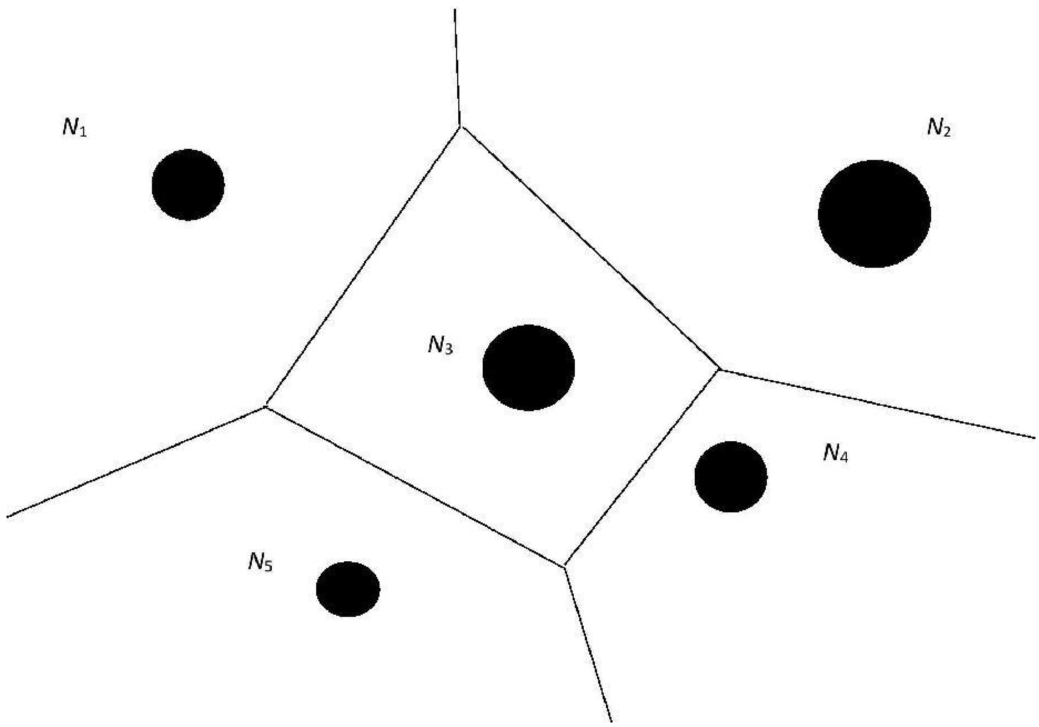
Physarum polycephalum behaves by plasmodia which can have the form either waves or protoplasmic tubes. Plasmodia grow from active zones concurrently. At these active zones, three basic operations stimulated by nutrients (attractants) and some other conditions can be observed: fusion, multiplication, direction, and repelling operations. (i) The *fusion*, $Fuse$, means that two active zones A_1 and A_2 produce a new active zone A_3. (ii) The *multiplication*, $Mult$, means that the active zone A_1 splits into two independent active zones A_2 and A_3 propagating along their own trajectories. (iii) The *direction*, $Direct$, means that the active zone A moves to a source of nutrients. (iv) The *repelling*, $Repel$, means that the active zone A avoids repellents. These operations, $Fuse$, $Mult$, $Direct$, $Repel$ can be determined by the following stimuli: (1) the set of attractants $\{N_1, N_2, \ldots\}$, sources of nutrients, on which the plasmodium feeds; (2) the set of repellents $\{R_1, R_2, \ldots\}$, light and some thermo- and salt-based conditions.

The universe, where *Physarum* lives, consists of cells possessing different topological properties according to the intensity of chemo-attractants and chemo-repellents. The intensity entails the natural or geographical neighborhood of the set's elements in accordance with the spreading of attractants or repellents. As a result, we obtain Voronoi cells. Let us define what they are mathematically. Let P be a nonempty finite set of planar points and $|P| = n$. For points $p = (p_1, p_2)$ and $x = (x_1, x_2)$ let $d(p,x) = \sqrt{(p_1 - x_1)^2 + (p_2 - x_2)^2}$ denote their Euclidean distance. A planar Voronoi diagram of the set P is a partition of the plane into cells, such that for any element of P, a cell corresponding to a unique point p contains all those points of the plane which are closer to p in respect to the distance d than to any other node of P. A unique region:

$$vor(p) = \bigcap_{m \in P, m \neq p} \{z \in \mathbf{R}^2 : d(p,z) < d(m,z)\}$$

assigned to the point p is called the *Voronoi cell* of the point p. Within one Voronoi cell a reagent has the full power to attract or repel the plasmodium. The distance d is defined by intensity of reagent spreading like in other chemical reactions simulated by Voronoi diagrams. When two spreading wave fronts of two reagents meet, this means that the plasmodium cannot choose any further direction, and splits (see Fig. 1). Within the same Voronoi cell two active zones will fuse.

Figure 1. The Voronoi diagram for *Physarum*, where different attractants have different intensity and power.

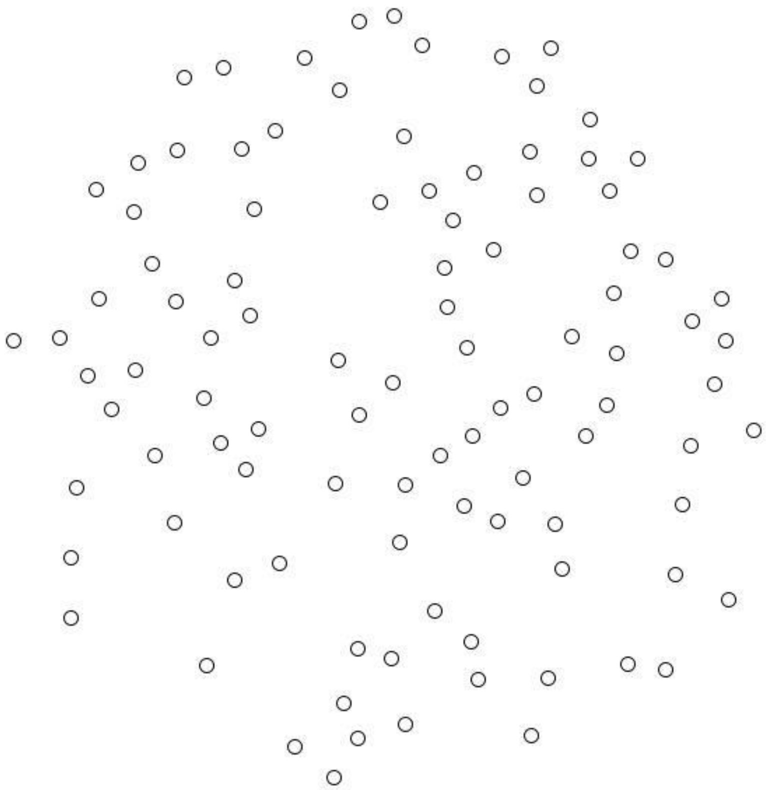


If a Voronoi center is presented by an attractant a that is activated and occupied by the plasmodium, this means that there exists a string a. This string has the meaning "a exists". If a Voronoi center is presented by a repellent $[a]$ that is activated and avoided by the plasmodium, this means that there exists a string $[a]$. This string has the meaning "a does not exist". If two neighbor Voronoi cells contain activated attractants a and b, which are occupied by the plasmodium, and between both centers there are protoplasmic tubes, then we say that there exists a string ab and a string ba. The meaning of those strings is equal and it is as follows: "ab exist", "ba exist", "some a is b", "some b is a".

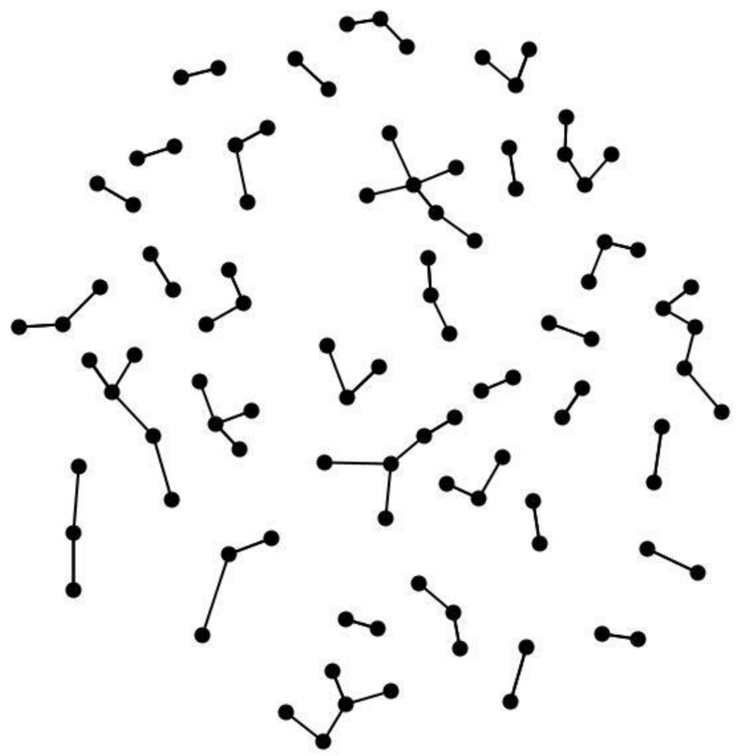
If one neighbor Voronoi cell contains an activated attractant a which is occupied by the plasmodium and another neighbor Voronoi cell contains an activated repellent $[b]$ which is avoided by the plasmodium, then we say that there exists a string $a[b]$ and a string $[b]a$. The meaning of those strings is equal and it is as follows: "ab do not exist, but a exists without b", "there exists a and no a is b", "no b is a and there exists a", "a exists and b does not exist".

If two neighbor Voronoi cells contain activated repellents $[a]$ and $[b]$ which are avoided by the plasmodium, then there exists a string $[ab]$ and a string $[ba]$. The meaning of those strings is equal and it is as follows: "ab do not exist together", "there are no a and there are no b", "no b is a", "no a is b".

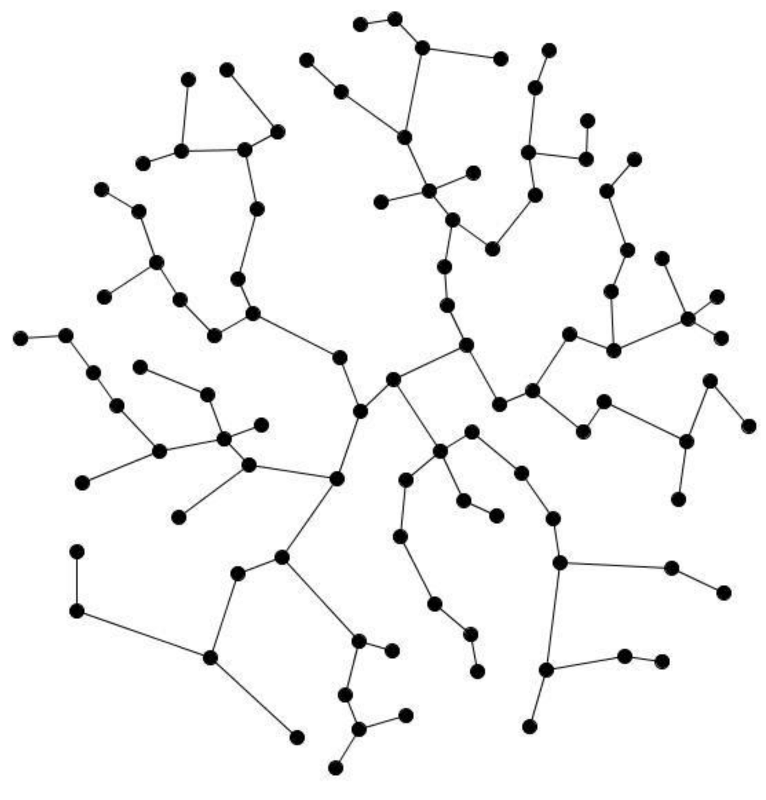
Figure 2. Examples of proximity graphs.



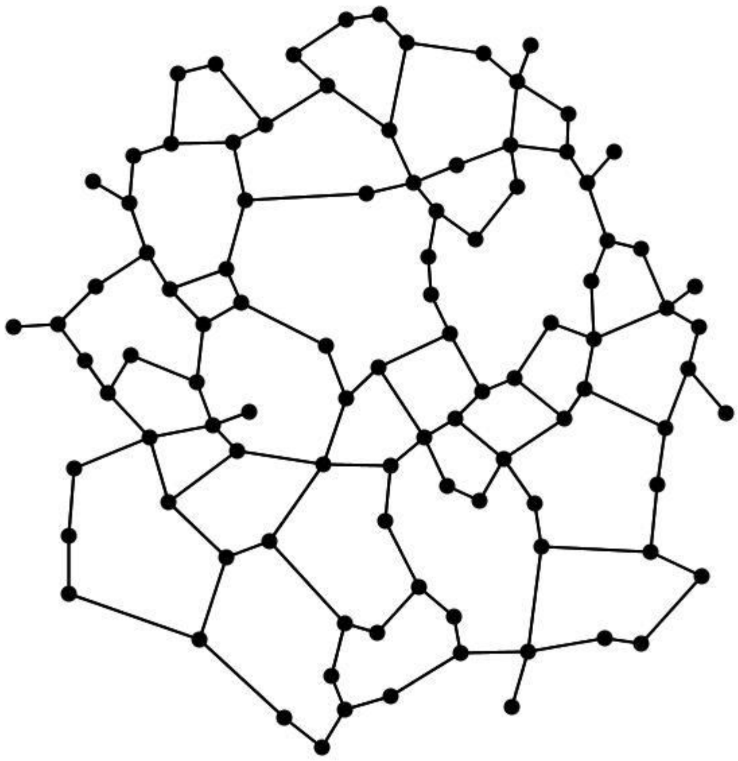
(a) Data points



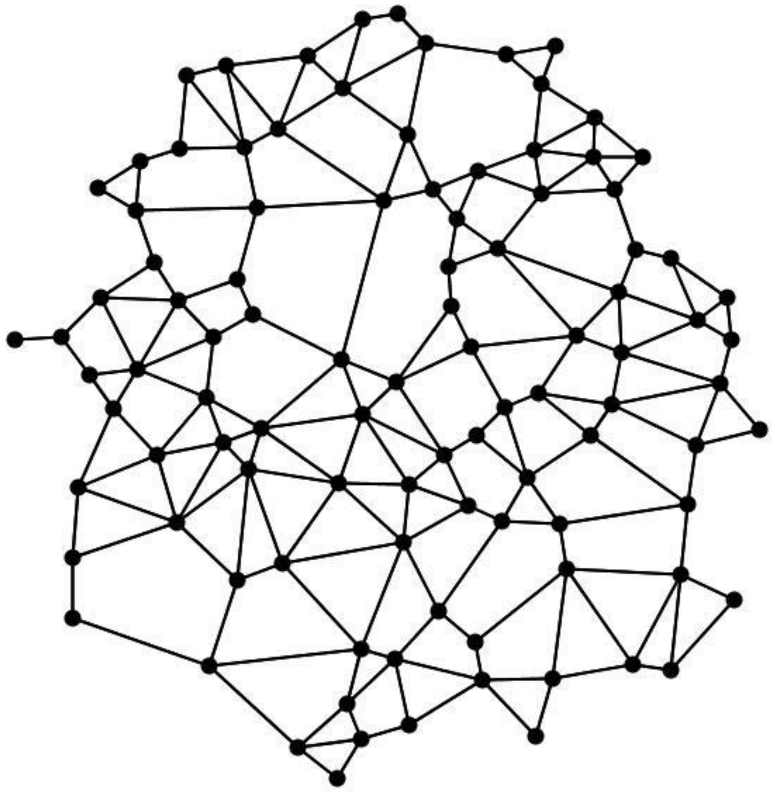
(b) NNG



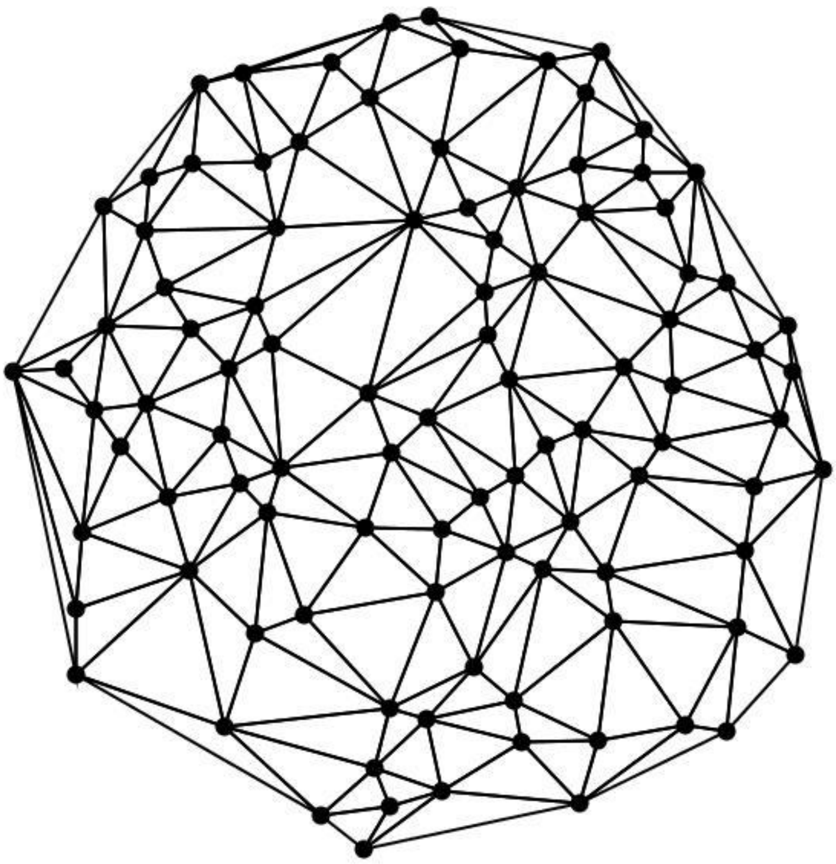
(c) MST



(d) RNG



(e) GG



(f) DT

Thus, the family of strings between nearest attractants presents a proximity graph which continuously grows from one attractant to another. This expansion could be demonstrated as a *Toussaint hierarchy* [1], where the family of strings starts from a nearest-neighborhood graph and any next graph in the hierarchy is produced from previous graph by adding some edges between non-adjacent nodes, see fig. 2:

$$\mathrm{NNG} \to \mathrm{MST} \to \mathrm{RNG} \to \mathrm{GG} \to \mathrm{DT},$$

where:

- NNG is a *nearest-neighborhood graph*. It is the simplest and possibly most natural of proximity graphs. A point in the graph is connected by an edge to its nearest neighbor. Given planar set V we can define the graph as follows: $\mathrm{NNG}(V) = \langle V, E \rangle$, where for $a, b \in V$ we have $(ab) \in E$ iff $|ab| = \min_{c \in V - \{a\}} |ac|$. In general case NNG is a disconnected directed graph.
- MST is a *minimal spanning tree*. It is a connected acyclic graph which has minimal possible sum of edges' lengths.

- RNG is a *relative neighborhood graph*. It is a graph, where any two points (a,b) are connected by an edge if the intersection of open disks of radius $|ab|$ centered at *a* and *b* is empty: $(ab) \in E$ iff $|ab| \leq \max_{c \in V - \{a,b\}}\{|ac|,|bc|\}$.
- GG is a *Gabriel graph*. It is a graph, where points *a* and *b* are connected by an edge if the closed disk having the segment (ab) as its diameter is empty: $(ab) \in E$ iff $|ab| \geq \min_{c \in V - \{a,b\}}\{|\frac{a+b}{2}c|\}$.
- DT is a *Delaunay triangulation*. It is a graph subdividing the space onto triangles with vertices in V and edges in E where the circumcircle of any triangle contains no points of V other then its vertices.

Each string in the Toussaint hierarchy could be interpreted as a syllogistic proposition. At the beginning, we have just data points without strings (Fig. 2 (a)). Then some first strings have grown up (Fig. 2 (b)) and in some cases we see the first syllogistic conclusions, when three or more points are connected by protoplasmic tubes. At the end, we observe all possible syllogistic conclusions (Fig. 2 (f)) in the topology of attractants and repellents we have set up for *Physarum*. As a result, the growth of plasmodia is considered syllogistic conclusions. Due to different stimuli, we can manage this growth in directions we want, therefore we can fore-see all possible syllogistic conclusions, which can be implemented within the certain topology of attractants and repellents. Moreover, we can deal with different syllogistic systems in man-aging the *Physarum* behavior. In particular, we can implement the Aristotelian syllogistic (Section 5.3), performative syllogistic (Section 5.4), and Talmudic reasoning by *qal wa-homer* (Section 5.5). For denoting all possible logic circuits of syllogistic systems implement-ed in the *Physarum* behavior, we will use the so-called *Physarum diagrams* which are a modi-fication of the well-known Lewis Carroll's diagrams [3], [4].

5.3. *Physarum* Aristotelian syllogistic

5.3.1. Plasmodium without repellents

The Aristotelian syllogistic is the first formal system, it was created in Ancient time. Its axiomatization was laid first by Łukasiewicz [8]. In his axiomatization, the alphabet consists of the syllogistic letters S, P, M, …, the syllogistic connectives a, e, i, o, and the propositional connectives $\neg$, $\vee$, $\wedge$, $\Rightarrow$. Atomic propositions are defined as follows: *SxP*, where $x \in \{a, e, i, o\}$. All other propositions are defined in the following way: (i) each atomic proposition is a proposition, (ii) if X, Y are propositions, then $\neg X$, $\neg Y$, XåY, where å $\in \{\vee, \wedge, \Rightarrow\}$, are propositions, also. The axioms proposed by Łukasiewicz are as follows:

$$SaP := (\exists A(A \text{ is } S) \wedge \forall A(A \text{ is } S \Rightarrow A \text{ is } P)); \tag{1}$$

$$SiP := \exists A(A \text{ is } S \wedge A \text{ is } P); \tag{2}$$

$$SeP := \neg(SiP); \tag{3}$$

$$SoP := \neg(SaP); \tag{4}$$

$$SaS; \tag{5}$$

$$SiS; \tag{6}$$

$$(MaP \wedge SaM) \Rightarrow SaP; \tag{7}$$

$$(MaP \wedge MiS) \Rightarrow SiP. \tag{8}$$

In the *Physarum* implementation of Aristotelian syllogistic, all data points are denoted by appropriate syllogistic letters as attractants. A data point S is considered empty if and only if an appropriate attractant denoted by S is not occupied by plasmodium. We have syllogistic strings of the form SP with the following interpretation: ‘ S is P ’, and with the following meaning: SP is true if and only if S and P are neighbors and both S and P are not empty, otherwise SP is false. By this definition of syllogistic strings, we can define atomic syllogistic propositions as follows:

‘All *S* are *P*’ (*SaP*): *In the formal syllogistic*: there exists A such that A is S and for any A, if A is S, then A is P. *In the Physarum model*: there is a plasmodium A and for any A, if A is located at S, then A is located at P.

'Some *S* **are** *P***' (***SiP***):** *In the formal syllogistic*: there exists A such that both 'A is S' is true and 'A is P' is true. *In the Physarum model*: there exists a plasmodium A such that A is located at S and A is located at P.

'No *S* **are** *P***' (***SeP***):** *In the formal syllogistic*: for all A, 'A is S' is false or 'A is P' is false. *In the Physarum model*: for all plasmodia A, A is not located at S or A is not located at P.

'Some *S* **are not** *P***' (***SoP***):** *In the formal syllogistic*: for any A, 'A is S' is false or there exists A such that 'A is S' is true and 'A is P' is false. *In the Physarum model*: for any plasmodia A, A is not located at S or there exists A such that A is located at S and A is not located at P.

Formally, this semantics is defined as follows. Let M be a set of attractants. Take a subset $|X| \subseteq M$ of attractants occupied by the plasmodium as a meaning for each syllogistic variable X. Next, define an ordering relation $\subseteq$ on subsets $|S|, |P| \subseteq M$ as: $|S| \subseteq |P|$ iff all attractants from $|P|$ are reachable for the plasmodium located at the attractants from $|S|$. Hence, $|S| \cap |P| \neq \varnothing$ means that some attractants from $|P|$ are reachable for the plasmodium located at the attractants from $|S|$ and $|S| \cap |P| = \varnothing$ means that no attractants from $|P|$ are reachable for the plasmodium located at the attractants from $|S|$. This gives rise to models $\mathsf{M} = \langle M, |\cdot| \rangle$ such that:

$$\mathsf{M} \models SaP \text{ iff } |S| \subseteq |P|;$$

$$\mathsf{M} \models SiP \text{ iff } |S| \cap |P| \neq \varnothing;$$

$$\mathsf{M} \models SeP \text{ iff } |S| \cap |P| = \varnothing;$$

$$\mathsf{M} \models p \wedge q \text{ iff } \mathsf{M} \models p \text{ and } \mathsf{M} \models q;$$

$$\mathsf{M} \models p \vee q \text{ iff } \mathsf{M} \models p \text{ or } \mathsf{M} \models q;$$

$$\mathsf{M} \models \neg p \text{ iff it is false that } \mathsf{M} \models p.$$

Proposition 1. *The Aristotelian syllogistic is sound and complete relatively to* M *if we understand* $\subseteq$ *as an inclusion relation (it is a well-known result* [19]*).*

However, relatively to all possible plasmodium behaviors the Aristotelian syllogistic is not complete. Indeed, the relation $\subseteq$ can have the following verification on *Physarum* according to our definitions: $|S| \subseteq |P|$ and $|S| \subseteq |P'|$, where $|P| \cap |P'| = \varnothing$, i.e. all attractants from $|P|$ are reachable for the plasmodium located at the attractants from $|S|$ and all attractants from $|P'|$ are reachable for the plasmodium located at the attractants from $|S|$, but between $|P|$ and $|P'|$ there are no paths. In this case $\subseteq$ is not an inclusion relation and proposition 1 does not hold. Hence, we need repellents to make $\subseteq$ the inclusion relations in all cases.

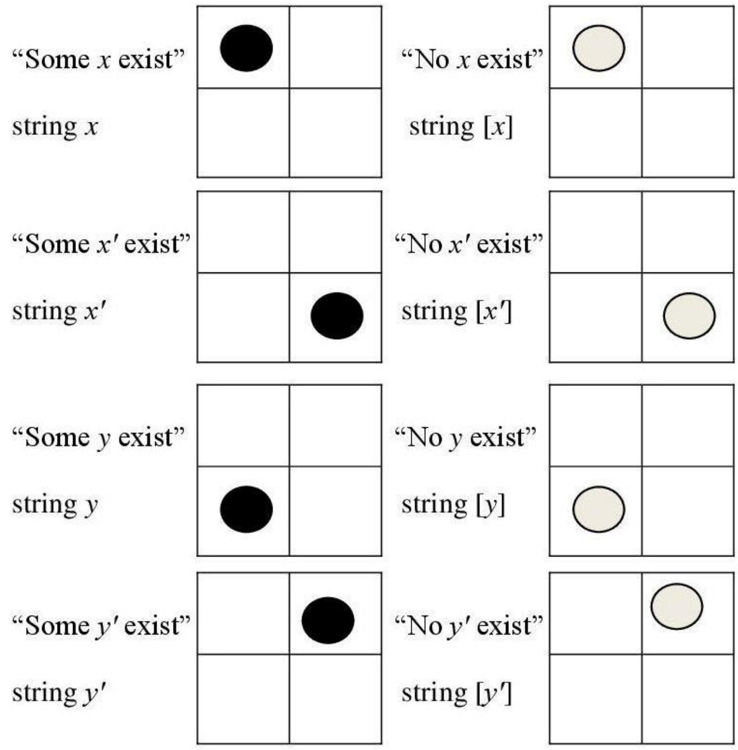
5.3.2. Plasmodium with repellents

In the *Physarum* diagrams for verifying all the basic syllogistic propositions, we will use the following four cells: x, y, x', y', where x' means all cells which differ from x, but they are neighbors for y, and y' means all cells which differ from y and are neighbors for x. These cells express appropriate meanings of syllogistic letters. The corresponding universe of discourse will be denoted by means of the following diagram:

x	y'
y	x'

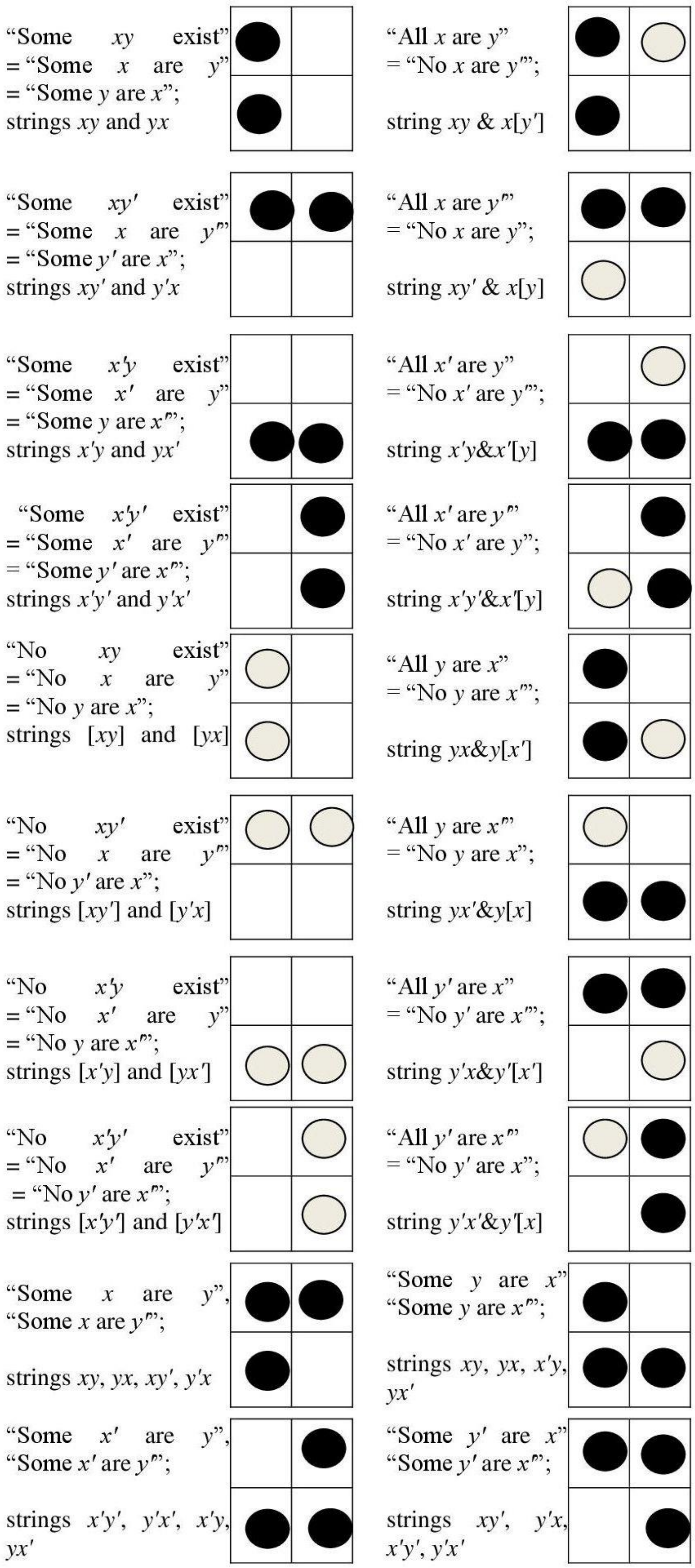
Assume that a black counter denotes an attractant and if it is placed within a cell x, this means that "this Voronoi cell contains an attractant N_x activated and occupied by the plasmodium." It is a verification of the syllogistic letter S_x at cell x. A grey counter denotes a repellent and if it is placed within a cell x, this means that this Voronoi cell contains a repellent R_x activated and there is no plasmodium in it". It is a verification of a new syllogistic letter $[S_x]$. For the sake of convenience, we will denote S_x by x and $[S_x]$ by $[x]$. Using these counters, we can verify all the basic existence syllogistic propositions in a way analogous, though different to Carroll's diagrams (see Fig. 3).

Figure 3. The *Physarum* diagrams for the basic existence strings.



Physarum strings of the form xy, yx are interpreted as particular affirmative propositions “Some x are y” and “Some y are x” respectively, strings of the form $[xy]$, $[yx]$, $x[y]$, $y[x]$ are interpreted as universal negative propositions “No x are y” and “No y are x.” A universal affirmative proposition “All x are y” are presented by a complex string $xy \& x[y']$. The sign & means that we have strings xy and $x[y']$ simultaneously and they are considered the one complex string. All these strings are verified on the basis of the diagrams of fig. 4.

Figure 4. The *Physarum* diagrams for syllogistic propositions.



For verifying syllogisms, we will use the following diagrams symbolizing some neighbor cells:

	m	m'	
m'	x	y'	m
m	y	x'	m'
	m'	m	

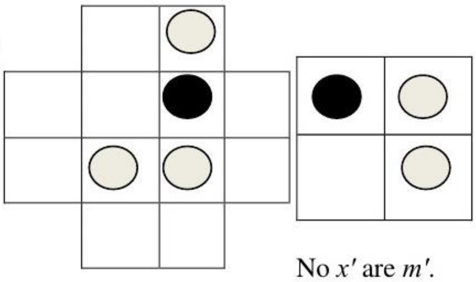
The motion of plasmodium starts from one of the central cells (x, y, x', y') and goes towards one of the four directions (northwest, southwest, northeast, southeast). The syllogism shows a connection between two not-neighbor cells on the basis of its joint neighbor and says if there was either multiplication or fusion. As a syllogistic conclusion, we obtain another diagram:

x	m'
m	x'

Different syllogistic conclusions derived show directions of plasmodium's propagation. Some examples are provided in fig. 5–7:

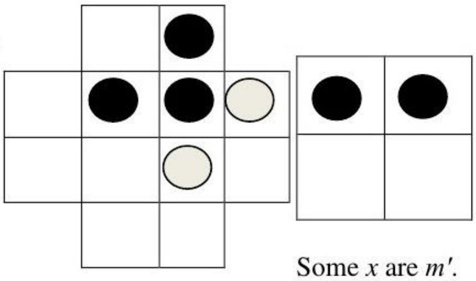
Figure 5. *Physarum* diagrams for syllogisms (part 1).

1\. No y are x';
All y' are m.



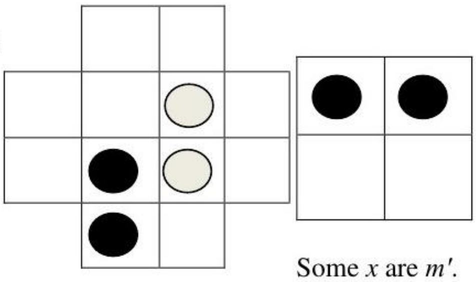
No x' are m'.

2\. All y' are x;
All y' are m'.



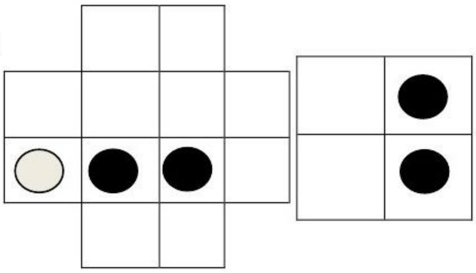
Some x are m'.

3\. No x' are y';
Some y are m'.



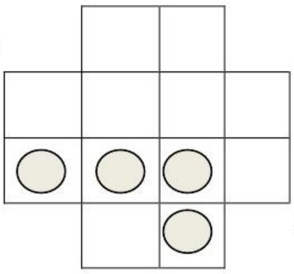
Some x are m'.

5\. Some y are x';
No m are y.



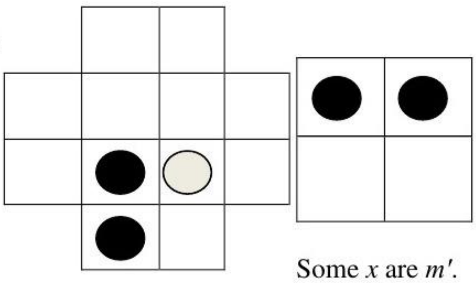
Some x' are m'.

6\. No x' are m;
No m are y.



There is no conclus

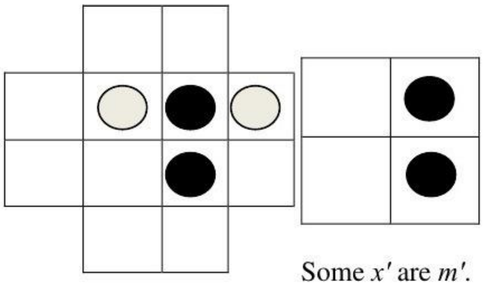
7\. No y are x';
Some m' are y.



Some x are m'.

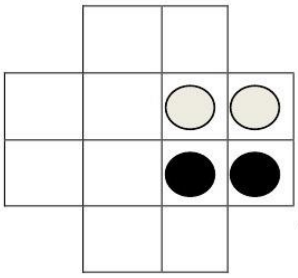
Figure 6. *Physarum* diagrams for syllogisms (part 2).

8. All y' are x'; No y' are m.



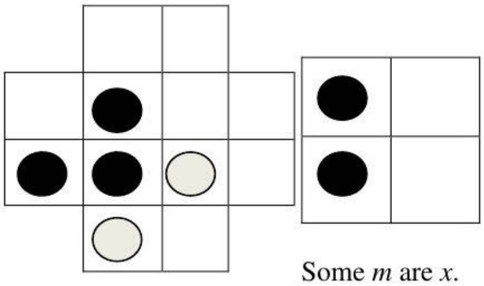
Some x' are m'.

9. Some x' are m'; No m are y'.



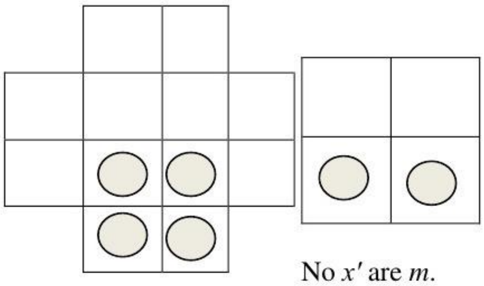
There is no conclusion.

10. All y are x; All y are m.



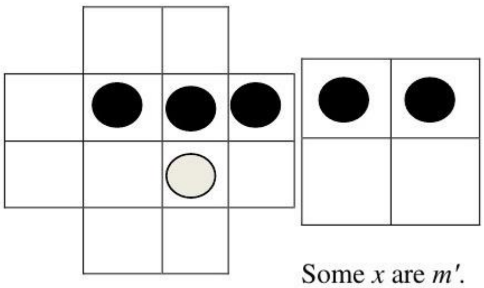
Some m are x.

11. No x' are m; No m' are y.



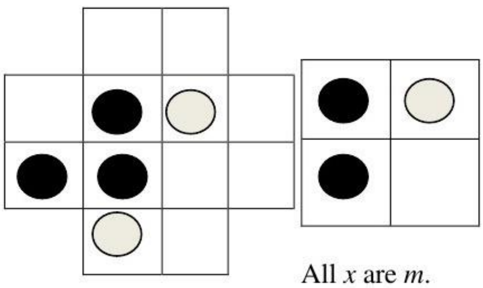
No x' are m.

12. All y' are x; Some m are y'.



Some x are m'.

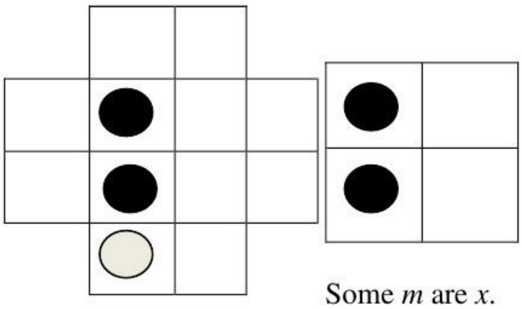
13. All y are m; All x are y.



All x are m.

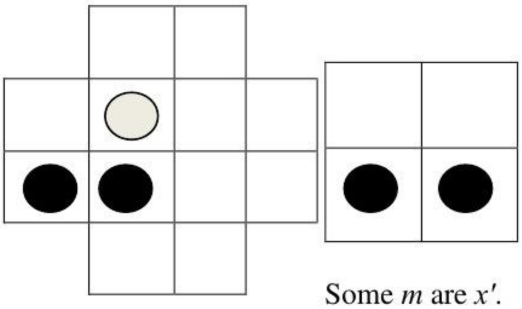
Figure 7. *Physarum* diagrams for syllogisms (part 3).

14. Some y are x;
No m' are y.



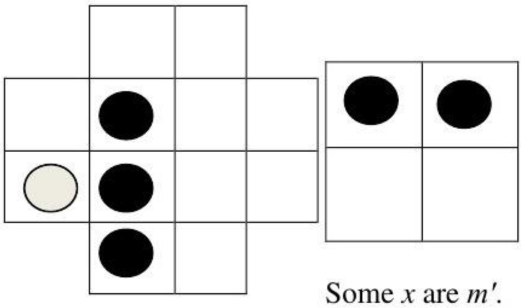
Some m are x.

15. No x are y;
Some m are y.



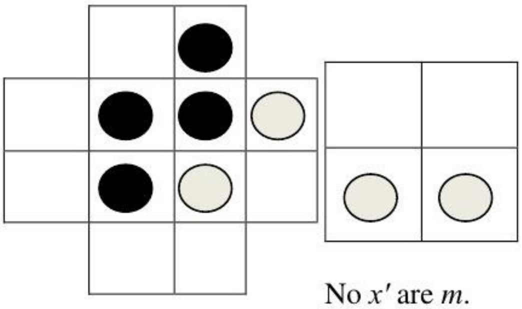
Some m are x'.

16. Some y are x;
All y are m'.



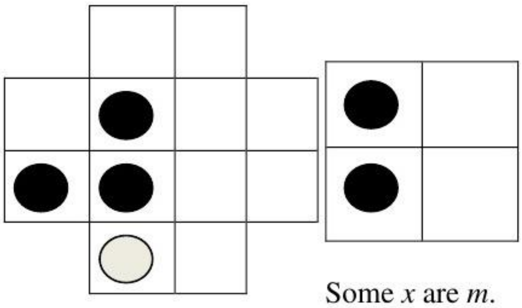
Some x are m'.

17. All y are x;
All y' are m'.



No x' are m.

18. Some x are y;
All y are m.



Some x are m.

Continuing in the same way, we can construct a syllogistic system, where conclusions are derived from three premises. The motion of plasmodium starts from one of the central cells (x, y, x', y') and goes towards one of the four directions (northwest, southwest, northeast, southeast), then towards one of the eight directions (north-northwest, west-northwest, south-southwest, west-southwest, north-northeast, east-northeast, south-southeast, east-southeast), etc.

Hence, a spatial expansion of plasmodium is interpreted as a set of syllogistic propositions. The universal affirmative proposition $xy \& x[y']$ means that the plasmodium at the place x goes only to y and all other directions are excluded. The universal negative proposition $x[y]$ or $[xy]$ means that the plasmodium at the place x cannot go to y and we know nothing about other directions. The particular affirmative proposition xy means that the plasmodium at the place x goes to y and we know nothing about other directions. Syllogistic conclusions allow us to mentally reduce the number of syllogistic propositions showing plasmodium's propagation.

For the implementation of Aristotelian syllogistic we appeal to repellents to delete some possibilities in the plasmodium propagation. So, model M defined above should be understood as follows:

- $\mathsf{M} \models$ *All* x *are* y iff $xy \& x[y']$, i.e. the plasmodium is located at x and can move only to y and cannot move towards all other directions;
- $\mathsf{M} \models$ *Some* x *are* y iff xy, i.e. the plasmodium is located at x and can move to y;
- $\mathsf{M} \models$ *No* x *are* y iff $x[y]$ or $[xy]$, i.e. the plasmodium cannot move to y in any case.

It is evident in this formulation that the Aristotelian syllogistic is so unnatural for plasmodia. Without repellents, this syllogistic system cannot be verified in the medium of plasmodium propagations. In other words, we can prove the next proposition:

Proposition 2. *The Aristotelian syllogistic is not sound and complete on the plasmodium without repellents.*

5.4. *Physarum* non-Aristotelian syllogistic

While in Aristotelian syllogisms we are concentrating on one direction of many *Physarum* motions, and dealing with acyclic directed graphs with fusions of many protoplasmic tubes toward one data point, in most cases of *Physarum* behavior, not limited by repellents, we observe a spatial expansion of *Physarum* protoplasm in all directions with many cycles, see fig. 8. Under these circumstances it is more natural to define all the basic syllogistic propositions SaP, SiP, SeP, SoP in a way they satisfies the inverse relationship when all converses are valid: $SaP \Rightarrow PaS$, $SiP \Rightarrow PiS$, $SeP \Rightarrow PeS$, $SoP \Rightarrow PoS$. In other words, we can draw more natural conclusions for protoplasmic tubes which are decentralized and have some cycles. The formal syllogistic system over propositions with such properties is constructed in [15]. This system is called the *performative syllogistic*. The alphabet of this system contains as descriptive signs the syllogistic letters S, P, M, …, as logical-semantic signs the syllogistic connectives a, e, i, o and the propositional connectives $\neg$, $\vee$, $\wedge$, $\Rightarrow$. Atomic propositions are defined as follows: SxP, where $x \in \{a, e, i, o\}$. All other propositions are defined thus: (i) each atomic proposition is a proposition, (ii) if X, Y are propositions, then $\neg X$, $\neg Y$, XåY, where å $\in \{\vee, \wedge, \Rightarrow\}$, are propositions, too.

Figure 8. Development of protoplasmic network by the plasmodium. Snapshots (a)–(d) are recorded with 10 hour intervals. In this experiment we observe cycles and decentralization of *Physarum* motions.

(a)

(b)

(c)

(d)

In order to implement the performative syllogistic in the behavior of *Physarum* plasmodium, we will interpret all data points denoted by appropriate syllogistic letters as attractants. A data point S is considered empty if and only if an appropriate attractant denoted by S is not occupied by plasmodium. Let us define syllogistic strings of the form SP with the following interpretation: 'S is P', and with the following meaning: SP is true if and only if S and P are reachable for each other by the plasmodium and both S and P are not empty, otherwise SP is false. Using this definition of syllogistic strings, we can define atomic syllogistic propositions as follows:

'All *S* **are** *P*' **(***SaP***):** *In the formal performative syllogistic*: there exists A such that A is S and for any A, A is S and A is P. *In the Physarum model*: there is a string AS and for any A which is a neighbor for S and P, there are strings AS and AP. This means that we have a massive-parallel occupation of the region where the cells S and P are located.

'Some *S* **are** *P*' **(***SiP***):** *In the formal performative syllogistic*: for any A, both 'A is S' is false and 'A is P' is false. *In the Physarum model*: for any A which is a neighbor for S and P, there are no strings AS and AP. This means that the plasmodium cannot reach S from P or P from S immediately.

'No *S* **are** *P*' **(***SeP***):** *In the formal performative syllogistic*: there exists A such that if 'A is S' is false, then 'A is P' is true. *In the Physarum model*: there exists A which is a neighbor for S and P such that there is a string AS or there is a string AP. This means that the plasmodium occupies S or P, but not the whole region where the cells S and P are located.

'Some *S* **are not** *P***' (***SoP***):** *In the formal performative syllogistic*: for any A, 'A is S' is false or there exists A such that 'A is S' is false or 'A is P' is false. *In the Physarum model*: for any A which is a neighbor for S and P there is no string AS or there exists A which is a neighbor for S and P such that there is no string AS or there is no string AP. This means that the plasmodium does not occupy S or there is a neighboring cell which is not connected to S or P by a protoplasmic tube.

Composite propositions are defined in the standard way.

In the performative syllogistic we have the following axioms:

$$SaP := (\exists A(A \mathrm{is} S) \wedge (\forall A(A \mathrm{is} S \wedge A \mathrm{is} P))); \tag{9}$$

$$SiP := \forall A(\neg(A \mathrm{is} S) \wedge \neg(A \mathrm{is} P)); \tag{10}$$

$$\begin{aligned} &SoP := \neg(\exists A(A \mathrm{is} S) \vee (\forall A(A \mathrm{is} P \wedge A \mathrm{is} S))), i.e. \\ &(\forall A \neg(A \mathrm{is} S) \wedge \exists A(\neg(A \mathrm{is} P) \vee \neg(A \mathrm{is} S))); \end{aligned} \tag{11}$$

$$\begin{aligned} &SeP := \neg \forall A(\neg(A \mathrm{is} S) \wedge \neg(A \mathrm{is} P)), i.e. \\ &\exists A(A \mathrm{is} S \vee A \mathrm{is} P). \end{aligned} \tag{12}$$

$$SaP \Rightarrow SeP; \tag{13}$$

$$SaP \Rightarrow PaS; \tag{14}$$

$$SiP \Rightarrow PiS; \tag{15}$$

$$SaM \Rightarrow SeP; \tag{16}$$

$$MaP \Rightarrow SeP; \tag{17}$$

$$(MaP \wedge SaM) \Rightarrow SaP; \tag{18}$$

$$(MiP \wedge SiM) \Rightarrow SiP. \tag{19}$$

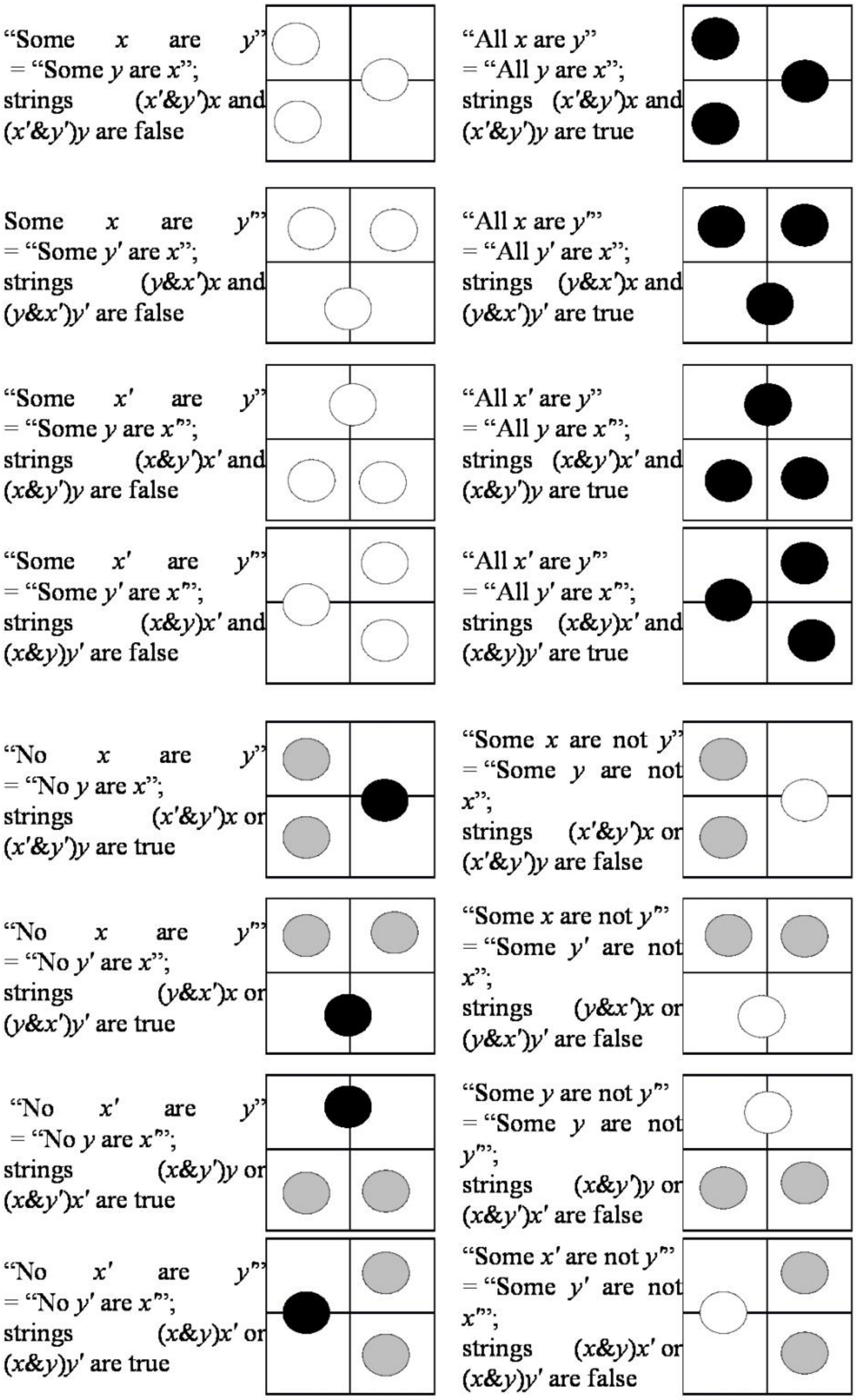
The formal properties of this axiomatic system are considered in [15]. In the performative syllogistic we can analyze the collective dimension of behavior. Within this system we can study how the plasmodium occupies all possible attractants in any direction if it can see them. So, this system shows logical properties of a massive-parallel behavior (i.e. the collective dimension of behavior). One of the most significant notions involved in this implementation of the performative syllogistic in *Physarum* topology is a *neighborhood*. We can define a distance for the neighborhood differently, i.e. we can make it broader or narrower. So, from different neighborhoods it will follow that we deal with different 'universes of discourse'.

In the *Physarum* diagrams for the performative syllogistic, the 'universe of discourse' cover cells x, y, non-x (which be denoted by x'), non-y (which he denoted by y'):

x	y'
y	x'

where x, y, x', y' are neighbor cells containing attractants for *Physarum*, x' are all neighbors for y which differ from x, and y' are all neighbors for x which differ from y. Suppose that we have black, white, and grey counters and (i) if a black counter is placed within a cell, this means that "this cell is occupied" (i.e. "there is at least one thing in it"), (ii) if a white counter is placed within a cell, this means that "this cell is not occupied" (i.e. "there is not thing in it"), (ii) if a grey counter is placed within a cell, this means that "it is not known if this cell is occupied". All possible combinations of *Physarum* diagrams for atomic propositions within our universe of discourse are pictured in fig. 9.

Figure 9. The *Physarum* diagrams for premises of performative syllogisms. Strings of the form $(x' \,\&\, y')x$ mean that in cells x' and y' there are neighbors A for x such that Ax, i.e. $(x' \,\&\, y')$ is a metavariable in $(x' \,\&\, y')x$ that is used to denote all attractants of x' and y' which are neighbors for the attractant of x.



The universe of discourse for simulating performative syllogisms by means of *Physarum* behaviors covers cells x, y, m, x', y', m' in the following manner:

y'	m	m'	x'
m'	x	y'	m
m	y	x'	m'
x	m'	m	y

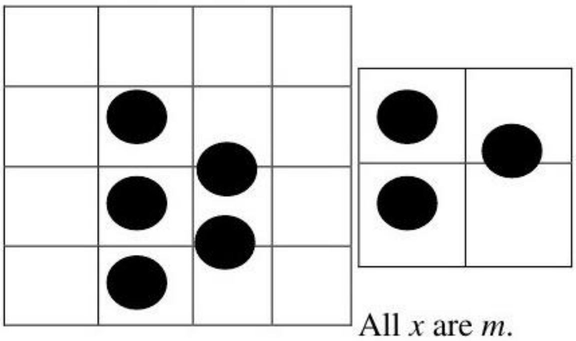
The motion of plasmodium starts from one of the central cells (x, y, x', y') and goes towards one of the four directions (northwest, southwest, northeast, southeast). The *Physarum* diagram for syllogistic conclusions is as follows:

x	m'
m	x'

Some examples of performative syllogistic conclusions are regarded in fig. 10. A zone of true universal affirmative propositions is pictured in fig. 11.

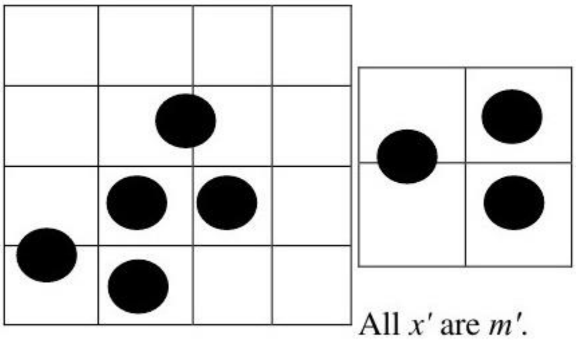
Figure 10. The *Physarum* diagrams for performative syllogisms with true conclusions.

All x are y;
All y are m.



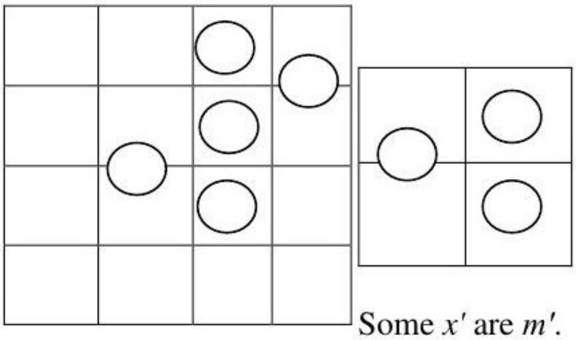
All x are m.

All x' are y;
All y are m'.



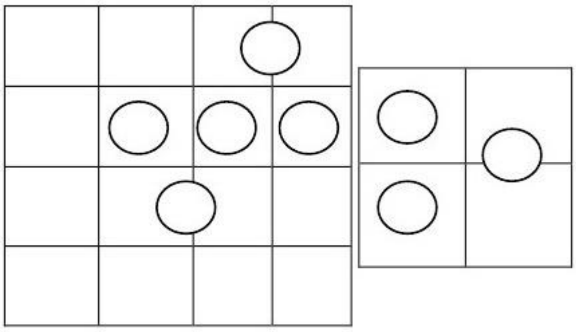
All x' are m'.

Some x' are y';
Some y' are m'.



Some x' are m'.

Some x are y';
Some y' are m.

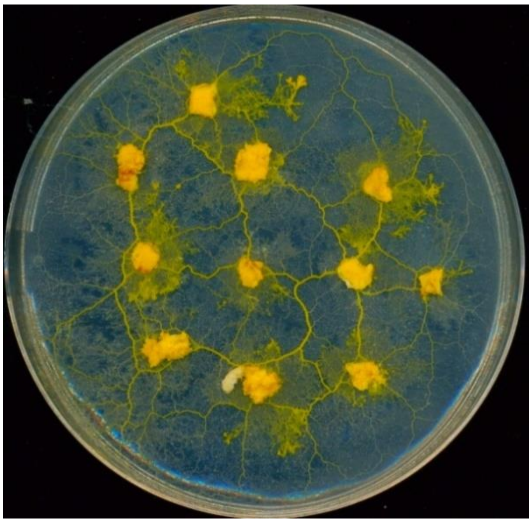


Some x are m.

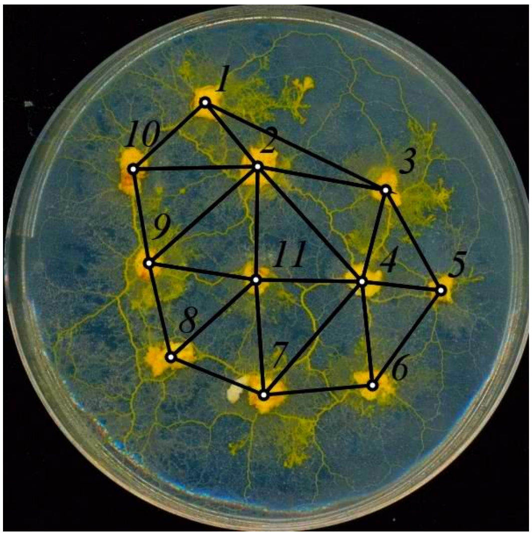
Figure 11. The active zone, where all cells containing attractants are connected by protoplasmic tubes, therefore the following syllogistic propositions are valid: S_1aS_{10}, S_2aS_3, S_9aS_{11}, S_7aS_6, etc. and the following syllogistic conclusions are true:

$$((S_{11}aS_9 \wedge S_9aS_{10}) \Rightarrow S_{11}aS_{10}),\ ((S_1aS_2 \wedge S_2aS_4) \Rightarrow S_1aS_4), \text{etc.}$$

(a)



(b)



Thus, the performative syllogistic allows us to study different zones containing attractants for *Physarum* if they are connected by protoplasmic tubes homogenously as in fig. 11.

A model $\mathrm{M}' = \langle M', |\cdot|_x \rangle$ for the performative syllogistic, where M' is the set of attractants and $|X|_x \subseteq M'$ is a meaning of syllogistic letter X which is understood as all attractants reachable for the plasmodium from the point x, is defined as follows:

$\mathsf{M'} \models$ *All* x *are* y iff $|X|_x \neq \varnothing$, $|X|_y \neq \varnothing$, and $|X|_x \cap |X|_y \neq \varnothing$, more precisely both $(x' \& y')x$ and $(x' \& y')y$ hold in $\mathsf{M'}$, i.e. the plasmodium can move from neighbors of y to x and it can move from neighbors of x to y;

$\mathsf{M'} \models$ *Some* x *are* y iff $y \notin |X|_x$ and $x \notin |X|_y$, more precisely neither $(x' \& y')x$ nor $(x' \& y')y$ hold in $\mathsf{M'}$, i.e. the plasmodium cannot move from neighbors of y to x and it cannot move from neighbors of x to y;

$\mathsf{M'} \models$ *No* x *are* y iff $y \in |X|_x$ or $x \in |X|_y$, more precisely $(x' \& y')x$ or $(x' \& y')y$ hold in $\mathsf{M'}$, i.e. the plasmodium can move from neighbors of y to x or it can move from neighbors of x to y;

$\mathsf{M'} \models$ *Some* x *are not* y iff $y \notin |X|_x$ or $x \notin |X|_y$, more precisely $(x' \& y')x$ or $(x' \& y')y$ do not hold in $\mathsf{M'}$, i.e. the plasmodium cannot move from neighbors of y to x or it cannot move from neighbors of x to y;

$\mathsf{M'} \models p \wedge q$ iff $\mathsf{M'} \models p$ and $\mathsf{M'} \models q$;

$\mathsf{M'} \models p \vee q$ iff $\mathsf{M'} \models p$ or $\mathsf{M'} \models q$;

$\mathsf{M'} \models \neg p$ iff it is false that $\mathsf{M'} \models p$.

Proposition 3. *The performative syllogistic is sound and complete in* $\mathsf{M'}$.

For more details on formal properties of performative syllogistic, please see [15]. This syllogistic describes the logic of plasmodium propagation in all possible directions. For the implementation of this syllogistic we do not need repellents. It is a natural system.

5.5. *Physarum* Talmudic reasoning

Deductions in Talmudic reasoning are constructed by using different inference rules. The oldest family of these rules consists of the thirteen rules of Rabbi Ishmael. The most important rule among them is called *qal wa-homer*. The direct meaning of the word *qal* is ‘light in weight’. From a logical point of view, *qal* is regarded as *minor*, i.e. as being less important or less significant. The direct meaning of *homer* is ‘heaviness’. It is major, i.e. more important, more significant. Hence, *qal wa-homer* is an inference from minor to major, and vice versa, from major to minor. For example, the Sabbath day is in some respects regarded as minor of a common holiday or festival (*yom tov*). Therefore, if a certain kind of work is permitted on the Sabbath we can infer that such a work is the more permissible on a festival; and vice versa, if a certain work is forbidden on festival, it has to be forbidden on the Sabbath also. Thus, *qal wa-homer* concerns actions when they are permitted to be performed.

Notice that in the Talmud it is claimed that it is sufficient to derive from an inference a result that is equivalent to the law from which it is made, i.e. the law transferred to the major must never surpass in severity the original law in the minor, from which the inference was drawn. This way of thinking is said to be the *dayo* principle.

Let us consider a Talmudic example of *qal wa-homer*. In the *Baba Qama* (one of the Talmudic books), different kinds of damages (*nezeqin*) are analyzed, among which three genera are examined: foot action (*regel*), tooth action (*shen*) and horn action (*qeren*). These three are damages that could be caused by an ox (he can trample (foot), eat (tooth) and gore (horn)). Due to the Torah it is known that tooth damage (as well as foot damage) by an ox at a public place needs to pay zero compensation. Horn damage at a public place entails payment of 50% the damage costs as compensation. In a private area foot/tooth damage must be paid in full. What can we say now about payments for horn actions of private places?

Damages (nezeqin)	Public place	Private place
Horn action (*qeren*)	50%	?
Foot action (*regel*)	0	100%
Tooth action (*shen*)	0	100%

In order to draw up a conclusion by *qal wa-homer*, we should define a two-dimensional ordering relation on the set of data: (i) on the one hand, according to the *dayo* principle, we know that payment for horn action in a private area cannot be greater than the same in a public area, (ii) on the other hand, payment for horn action at a private place cannot be greater than foot/tooth action at the same place. Hence, we infer that payment of compensation for horn action at a private place is equal to 50% of the damage costs.

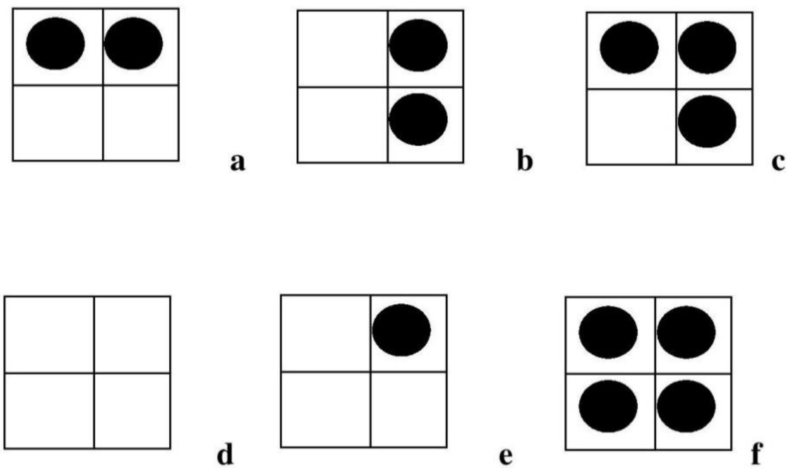
The table above shows us that *qal wa-homer* can be interpreted spatially, too. Already Yisrael Ury [23] proposed to use the Carroll's bilateral diagrams for modelling conclusions by *qal wa-homer*. Let us take now the diagram:

xy'	xy
$x'y'$	$x'y$

that plays the role of 'universe of discourse' for Talmudic reasoning over adjuncts x, y, non-x in the neighborhood (that is denoted by x'), non-y in the neighborhood (that is denoted by y'). Assume that we have only black counters and if a black counter is placed within a cell, this means that "this cell is occupied" (i.e. "there is at least one thing in it", "an appropriate Talmudic rule should be obeyed"). Thus, the cell that does not contain a black counter indicates a situation in which the obligation is not fulfilled, whereas the cell containing a black counter indicates a situation in which the obligation is fulfilled.

Hence, if we have two rows and two columns, there are sixteen possible ways to cover such a diagram by means of black counters, in connect to Yisrael Ury who accepts only six of them (Fig. 12).

Figure 12. Ury's diagrams for conclusions by *qal wa-homer*.



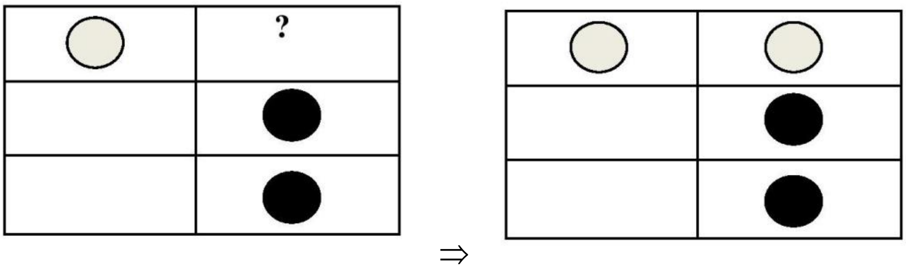
Let x mean proposition 1 and y mean proposition 2. Then the above mentioned accepted diagrams (Fig. 12) have the following sense: (a) It is necessary and sufficient to obey x; (b) It is necessary and sufficient to obey y; (c) It is sufficient to obey either x or y; (d) It is not sufficient to obey x and/or y; (e) It is necessary and sufficient to obey both x and y; (f) It is not necessary to obey either x or y.

Let us return to our example. Let x_3 be a 'foot action', x_2 a 'tooth action', x_1 a 'horn action', y' 'at public place', y 'at private place'. So, we obtain the following diagram:

x_1y'	x_1y
x_2y'	x_2y
x_3y'	x_3y

Assume that a black counter means an to pay 100% of the damage costs as compensation and a grey counter means an obligation to pay any 50%. We can cover this diagram by counters as shown in fig. 13.

Figure 13. Ury's diagrams for inferring whether we should pay a horn action at a private place.



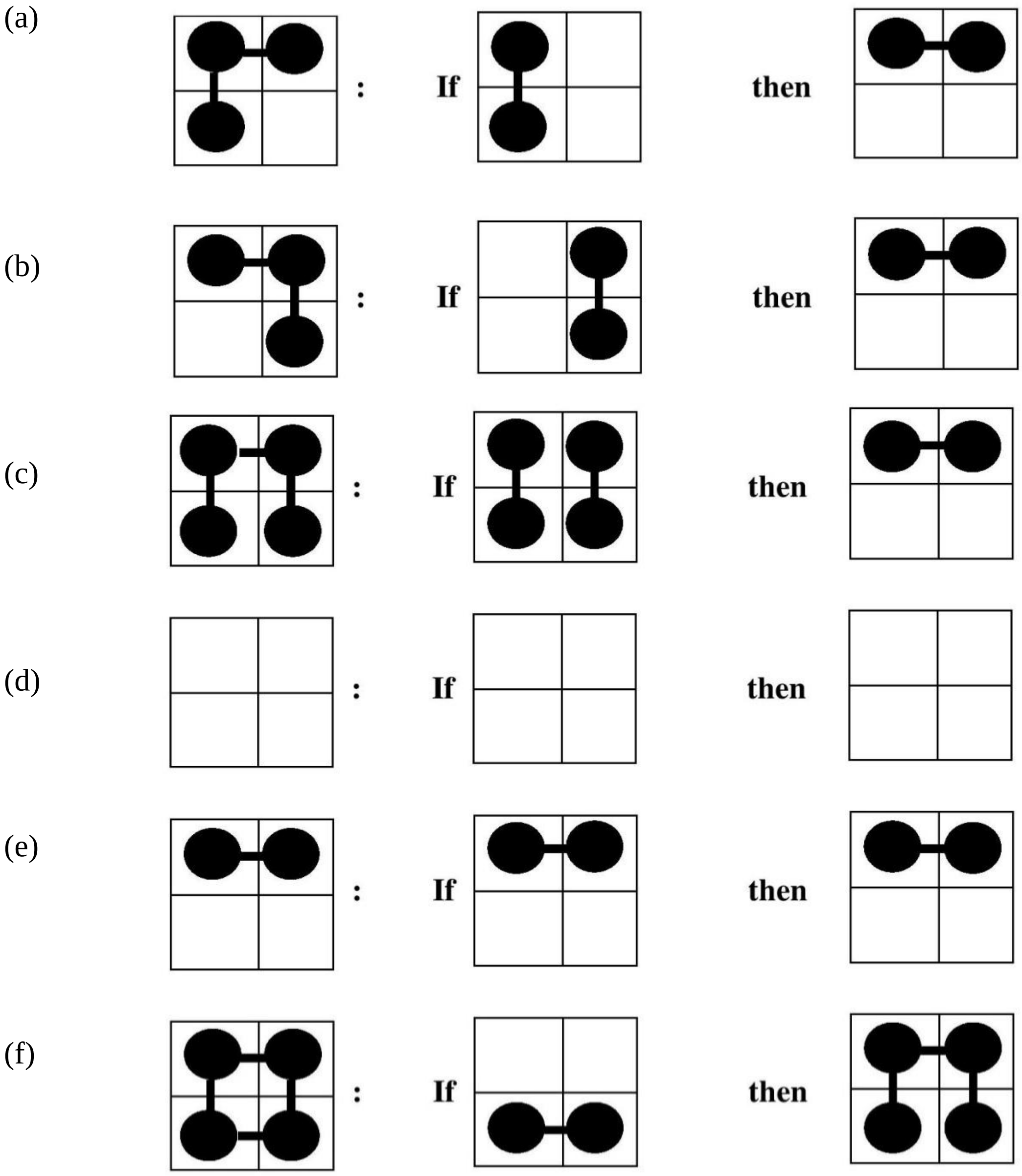
In so doing, we have supposed that there is a different power of intensity in obligation. In this case our rule for inferring by *qal wa-homer* is formulated thus: *if a cell contains a black or grey counter, all cells above it and to its right also contain a black or grey counter and the color of that counter has the minimal hardness of black and grey in counters of the neighbor cells; if a cell does not contain any counter, all cells below it and to its left are also without counters*.

Talmudic diagrams defined above are closer to Carroll's diagrams and are more natural for *Physarum* behavior than Ury's diagrams (cf. [23]). In the *Physarum* topology, Talmudic diagrams are built on syllogistic strings of the form xy, yx, $x'y$, yx', xy', $y'x$, $x'y'$, $y'x'$, where *x* and *y* in xy are interpreted as two neighbor attractants connected by protoplasmic tubes, x' is understood as all attractants which differ from *x*, but are neighbors of *y*, and y' is understood as all attractants which differ from *y* and are neighbors of *x*. The Talmudic diagrams for the *Physarum* simulation have the following form:

x	y
y'	x'

where x' is a non-empty class of neighbor attractants for *y* and y' is a non-empty class of neighbor attractants for *x*. Then *qal wa-homer* tells us whether a multiplication took place during the plasmodium's propagation at points *x* and/or *y*. In fig. 14, all the possible conclusions inferred by *qal wa-homer* in relation to *x* and *y* are considered, and they are defined if we have a multiplication at those points.

Figure 14. *The Physarum diagrams for qal wa-homer syllogisms*: (a) If the string xy' is verified, then the string xy is verified, too (i.e. if xy' is verified, then x has a multiplication of plasmodium). (b) If the string yx' is verified, then the string xy is verified, too (i.e. if yx' is verified, then y has a multiplication of plasmodium). (c) If the strings xy' and yx' are verified, then the string xy is verified, too (i.e. if both xy' and yx' are verified, then both and x y have multiplications of plasmodium). (d) If no string is verified, then there is no multiplication of plasmodium. (e) If the string xy is verified, then the there is no multiplication of plasmodium. (f) If the strings $x'y'$ is verified, then the strings xy', $x'y$, xy are verified, too (i.e. if $x'y'$ is verified, then both x and y have multiplications of plasmodium).



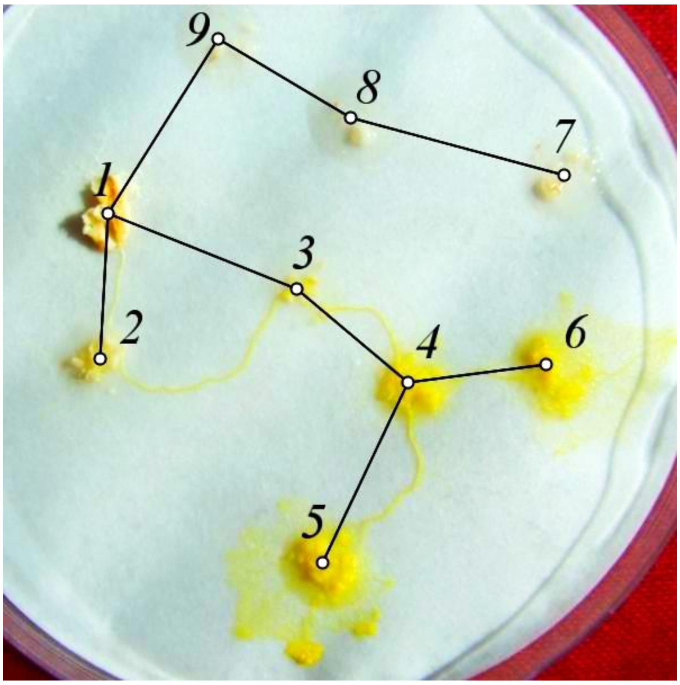
Hence, the main difference between the Aristotelian syllogistic and Talmudic reasoning is that, on the one hand, we are concentrating on fusions of plasmodium in the case of the Aristotelian syllogistic in the *Physarum* topology and, on the other hand, we deal with multiplications of plasmodia in the case of Talmudic reasoning. An example of an experiment with

Physarum that satisfies the *qal wa-homer* rule is shown in fig. 15. Talmudic reasoning can describe only fragments of plasmodium behaviors while the Aristotelian syllogistic describes some fragments of plasmodium propagations. Only the performative syllogistic is sound and complete on plasmodium interactions. For more details on Talmudic reasoning and its modern formalizations see [17].

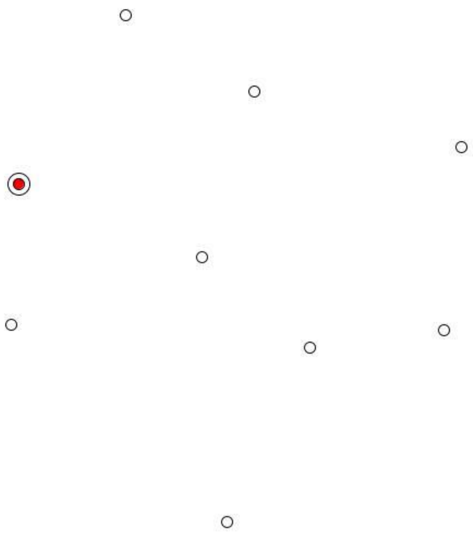
Figure 15. The *qal wa-homer* with plasmodium: (a) position of oat flakes representing data points, (b) the plasmodium tree corresponding to *qal wa-homer*, where $x = 1$, $y = 9$, $x' = 8$, $y' = 2 \cup 3$, etc., (c)–(g) dynamics of *qal wa-homer* by the growing plasmodium.

(a)

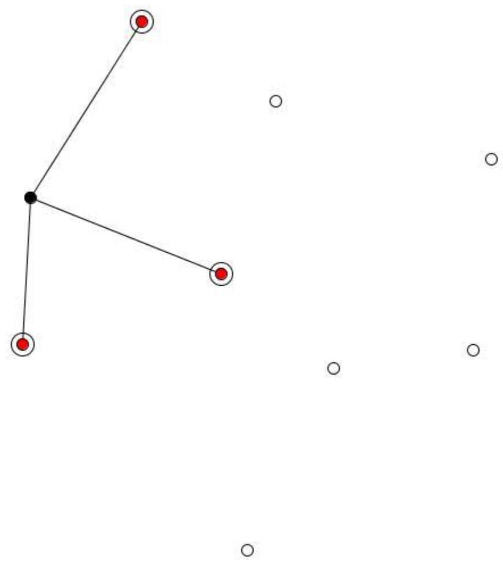
(b)



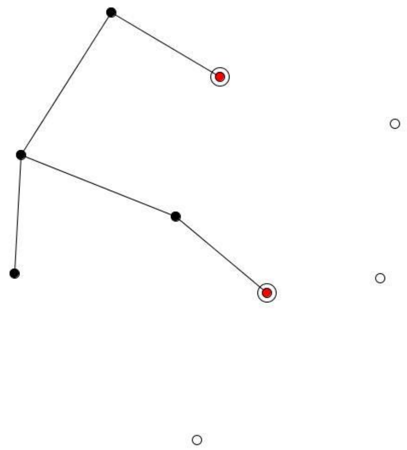
(c)



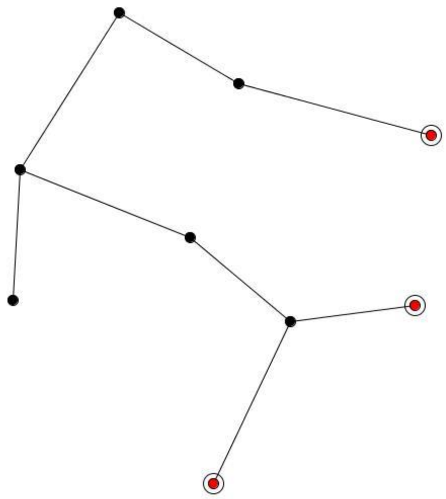
(d)



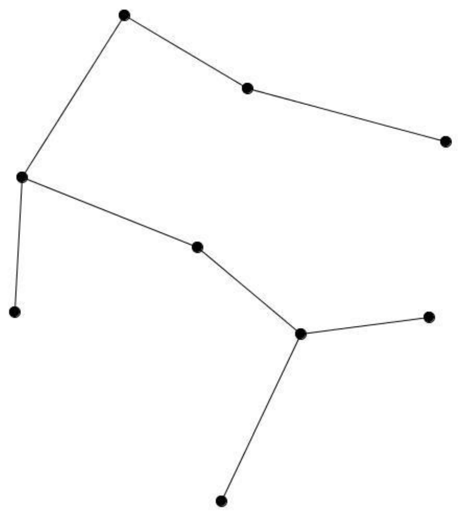
(e)



(f)



(g)



Conclusion

In this paper, we have considered bio-inspired implementations of several spatial syllogistic calculi. The medium of implementations chosen is the plasmodium of *Physarum Polycephalum*. These implementations are called *Physarum diagrams for syllogistic*. One of the most interesting results is that the Aristotelian syllogistic is quite unnatural in the sense that this system assumes fusions and concentrations of all plasmodium motions in one conclusion. This is difficult for the plasmodium, as it aims to be propagated in all possible directions.

The main theoretical result of our paper is to demonstrate that the performative syllogistic of [15] can simulate massive-parallel behaviors of living organisms such as plasmodia. This result can find many applications in behavioral sciences, because the plasmodium behavior can be considered the simplest natural intelligent behavior solving complex tasks. Thus, our result may have an impact on computational models.

Acknowledgments

This research is supported by FP7-ICT-2011-8.

References

[1] Adamatzky, A., *Physarum machine: implementation of a Kolmogorov-Uspensky machine on a biological substrate,* Parallel Processing Letters, 17(4):455–467, 2007.

[2] Adamatzky, A., *Physarum Machines: Computers from Slime Mould (World Scientific Series on Nonlinear Science, Series A),* World Scientific Publishing Company, 2010.

[3] Carroll, L., *The Game of Logic*, London: Macmillan and co., 1886.

[4] Carroll, L., *Symbolic Logic. Part I. Elementary,* Macmillan and CO., London, 1897.

[5] Chopard, B., Droz, M., *Cellular Automata Modeling of Physical Systems,* Cambridge University Press, 2005.

[6] Gutiérrez, A., Marco, S. (eds.), *Biologically Inspired Signal Processing for Chemical Sensing,* Series: Studies in Computational Intelligence, vol. 188, XIII, 2009.

[7] Kolmogorov, A.N., *On the concept of algorithm,* Uspekhi Mat Nauk, 8(4):175–176, 1953.

[8] Łukasiewicz, J., *Aristotle's Syllogistic From the Standpoint of Modern Formal Logic,* Oxford Clarendon Press, 2nd edition, 1957.

[9] Nakagaki, T., Yamada, H., Toth, A., *Maze-solving by an amoeboid organism, Nature* 407:470–470, 2000.

[10] Nakagaki, T., Yamada, H., Tothm, A., *Path finding by tube morphogenesis in an amoeboid organism*, Biophysical Chemistry, 92:47–52, 2001.

[11] Nakagaki, T., Iima, M., Ueda, T., Nishiura, Y., Saigusa, T., Tero, A., Kobayashi, R., Showalter, K., *Minimum-risk path finding by an adaptive amoeba network*, Physical Review Letters, 99:68–104, 2007.

[12] Saigusa, T., Tero, A., Nakagaki, T., Kuramoto, Y., *Amoebae Anticipate Periodic Events,* Phys. Rev. Lett, 100(1), 2008.

[13] Schumann, A., Akimova, L., *Simulating of Schistosomatidae (Trematoda: Digenea) Behavior by Physarum Spatial Logic*, *Annals of Computer Science and Information Systems,* vol. 1. *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems,* IEEE Xplore, pp. 225–230, 2013.

[14] Schumann, A., Khrennikov, A., *p-Adic Physics, Non-well-Founded Reality and Unconventional Computing*, P-Adic Numbers, Ultrametric Analysis and Applications, 1(4):297–306, 2009.

[15] Schumann, A., *On Two Squares of Opposition: the Lesniewski's Style Formalization of Synthetic Propositions,* Acta Analytica 28:71–93, 2013.

[16] Schumann, A., Pancerz, K., *Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing,* [in:] Szczuka, M., Czaja, L., Kacprzak, M. (eds.), *Proceedings of the Workshop on Concurrency, Specification and Programming* (CS&P'2013), Warsaw, Poland, September 25-27:389–397, 2013.

[17] Schumann, A. (ed.), *Modern Review of Judaic Logic,* Special issue of History and Philosophy of Logic, 2011.

[18] Shirakawa, T., Gunji, Y.P., Miyake, Y., *An associative learning experiment using the plasmodium of Physarum polycephalum,* Nano Communication Networks, 2:99–105, 2011.

[19] Smith, R., *Completeness of an ecthetic syllogistic*, Notre Dame J. Formal Logic, 24(2):224–232, 1983.

[20] Tero, A., T. Nakagaki, K., Toyabe, K., Yumiki, R., Kobayashi., A., *Method Inspired by Physarum for Solving the Steiner Problem, IJUC*, 6(2):109–123, 2010.

[21] Tsuda, S., Aono, M., Gunji, Y.P., *Robust and emergent Physarum-computing,* BioSystems, 73:45–55, 2004.

[22] Uspensky, V.U., *Kolmogorov and mathematical logic*, Symb Logic, 57:385–412, 1992.

[23] Ury, Yisrael, *Charting the Sea of Talmud: A Visual Method for Understanding the Talmud,* 2012.

[24] Watanabe, Sh., Tero, A., Takamatsu, A., Nakagaki, T., *Traffic optimization in railroad networks using an algorithm mimicking an amoeba-like organism,* Physarum plasmodium, Biosystems, 105(3):225–232, 2011.

Originally published: Schumann, A., *Physarum Syllogistic L-Systems*, FUTURE COMPUTING 2014, The Sixth International Conference on Future Computational Technologies and Applications, 2014.

VI
Physarum Syllogistic L-Systems

Andrew Schumann

Abstract

One of the best media for studying natural computing is presented by the behavior of *Physarum polycephalum* plasmodia. Plasmodium has active zones of growing pseudopodia and these zones interact concurrently and in a parallel manner. This behavior can be stimulated by attractants and repellents. In the paper, different syllogistic systems are proposed for simulating the plasmodium's behavior. While Aristotelian syllogistic may describe concrete directions of *Physarum* spatial expansions, pragmatic syllogistic proposed in this paper may describe *Physarum* simultaneous propagations in all directions. It is a more suitable system for applying syllogistic models in designing logic gates in plasmodia.

Introduction

There are many approaches to biological computing as a kind of unconventional computing; one of them is presented by systems invented by Aristid Lindenmayer [4]. They are called L-systems and allow us to simulate the growth of plants by formal grammars [6][13]. In the project [2], we are going to develop another approach to biological computing, assuming a massive parallelism of biological behavior. In this paper, we will show that we can implement two syllogistics in the biological behavior: the Aristotelian syllogistic [5] and a non-Aristotelian syllogistic constructed in [10][12]. The first is implementable within standard trees of appropriate L-systems. The second is massive-parallel and contain cycles and, therefore, can be implementable just within non-standard trees of some rigorous extensions of L-systems. This means that *Physarum polycephalum*, the medium of computations, which we have studied in the project, embodies complex extensions of L-systems.

Let us recall that *Physarum polycephalum* is a one-cell organism that behaves according to different stimuli and can be considered the basic medium of simple actions that are intelligent in the human meaning [1][2][7][8][9]. It behaves by plasmodia which can have the form either waves or protoplasmic tubes (arches). Hence, it is a system that is being spatially extended, as well as standard L-systems. This extension can be described as an extension of L-system called *Physarum* L-system (Section 2). Within this system we can implement (i) Aristotelian syllogistic in the *Physarum* media (Section 4), as well as non-Aristotelian syllogistic defined in [10][12] (Section 5).

In our project [2], we obtained a basis of new object-oriented programming language for *Physarum polycephalum* computing [11]. Within this language we are going to check possibilities of practical implementations of storage modification machines on plasmodia and their applications to behavioral science such as behavioral economics and game theory. The point is that experiments with plasmodia may show fundamental properties of any intelligent behavior. The language, proposed by us, can be used for developing programs for *Physarum polycephalum* by the spatial configuration of stationary nodes. Some preliminary results of computational models on plasmodia are obtained in [1]. In this paper, we consider possibilities to implement syllogistic models as logic gates for *Physarum polycephalum*, which can be programmable within our language. In section 6.1, we define *Physarum* L-systems. In section 6.2, we consider their particular case presented by Aristotelian trees. In section 6.3, we show how we can implement Aristotelian syllogistic in the *Physarum* behavior. In section 6.4, we show how we can implement non-Aristotelian syllogistic defined in [10][12].

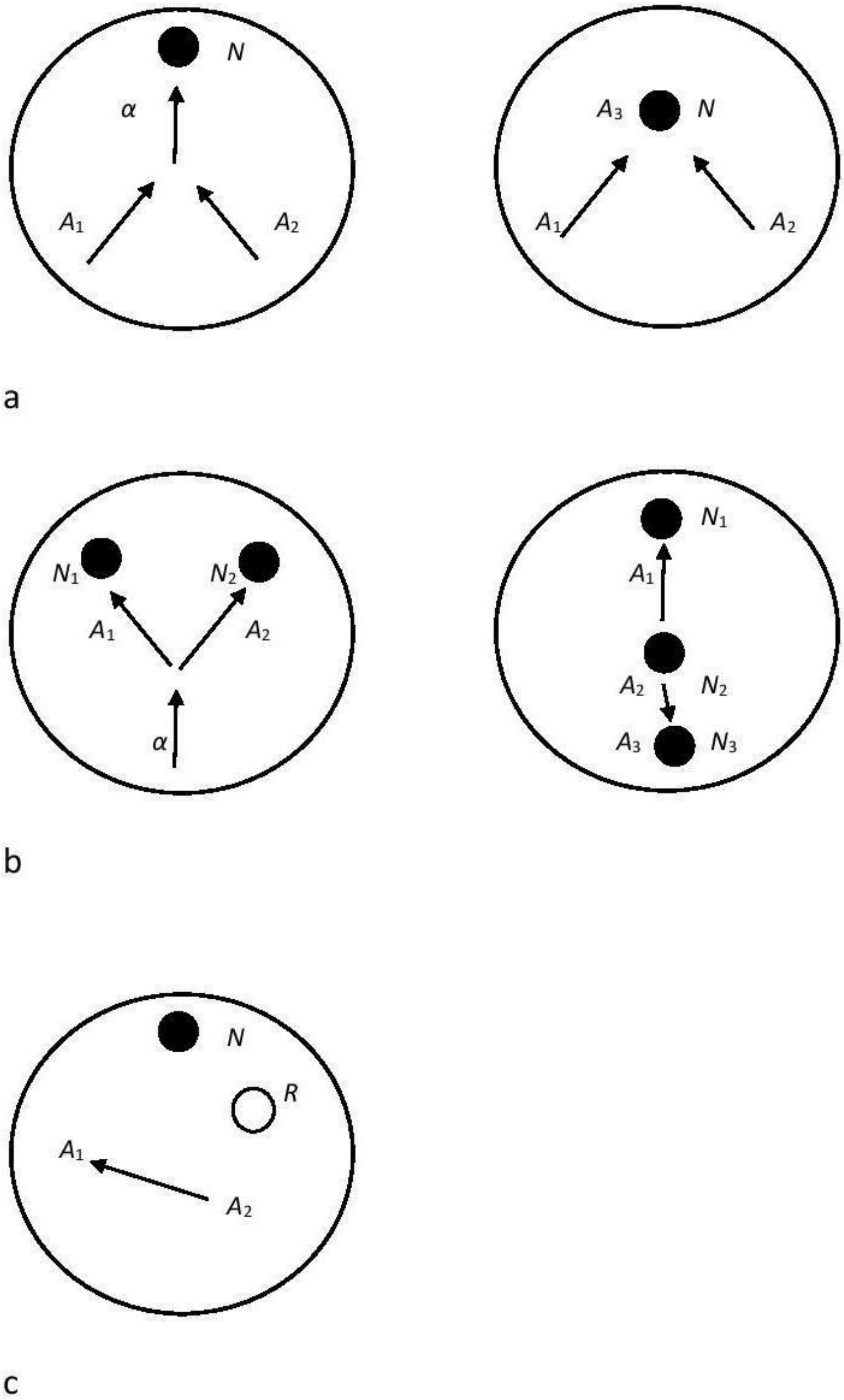
6.1. *Physarum* L-system

The behavior of *Physarum* plasmodia can be stimulated by attractants and repellents. We have the following entities which can be used in programming plasmodia:

- The set of *active zones* of *Physarum* $\{V_1, V_2, ...\}$, from which any behavior begin to carry out.
- The set of *attractants* $\{A_1, A_2, ...\}$; they are sources of nutrients, on which the plasmodium feeds, or pheromones which chemically attract the plasmodium. Any attractant is characterized by its position and intensity.
- The set of *repellents* $\{R_1, R_2, ...\}$. Plasmodium of *Physarum* avoids light and some thermo- and salt-based conditions. Thus, domains of high illumination (or high grade of salt) are repellents such that each repellent is characterized by its position and intensity, or force of repelling.
- The set of *protoplasmic tubes* $\{T_1, T_2, ...\}$. Typically, plasmodium spans sources of nutrients with protoplasmic tubes/veins. The plasmodium builds a planar graph, where nodes are sources of nutrients or pheromones, e.g., oat flakes, and edges are protoplasmic tubes.

Plasmodia grow from active zones. At these active zones, according to Adamatzky's experiments [2][3], the following three basic operations stimulated by nutrients (attractants) and some other conditions can be observed: fusion, multiplication, and direction operations (see Fig. 1):

Figure 1. The stimulation of the following operations in *Physarum* automata: (a) fusion, (b) multiplication, and (c) direction, where A_1, A_2, A_3 are active zones, N, N_1, N_2, N_3 are attractants, α is a protoplasmic tube, R is a repellent.



1. The *fusion*, denoted *Fuse*, means that two active zones A_1 and A_2 either produce new active zone A_3 (i.e., there is a collision of the active zones) or just a protoplasmic tube α: $Fuse(A_1, A_2) = A_3$ or $Fuse(A_1, A_2) = \alpha$.
2. The *multiplication*, *Mult*, means that the active zone A_1 splits into two independent active

zones A_2 and A_3, propagating along their own trajectories: $Mult(A_1) = \{A_2, A_3\}$ or $Mult(\alpha) = \{A_2, A_3\}$.

3. The *direction*, *Direct*, means that the active zone A is not translated to a source of nutrients but to a domain of an active space with certain initial velocity vector v: $Direct(A, v)$.

These operations, *Fuse, Mult*, *Direct*, can be determined by the attractants $\{A_1, A_2, ...\}$ and repellents $\{R_1, R_2, ...\}$.

On the basis of active zones $\{V_1, V_2, ...\}$, attractants $\{A_1, A_2, ...\}$, repellents $\{R_1, R_2, ...\}$, and protoplasmic tubes $\{T_1, T_2, ...\}$, we can define a *Physarum* L-system. Let us remember that an L-system consists of (i) an alphabet of symbols that can be used to make strings, (ii) a collection of production rules that expand each symbol into some larger or shorter string of symbols, and (iii) an initial string from which we move. These systems were introduced by Lindenmayer [4][6][13] to describe and simulate the behavior of plant cells.

The *Physarum* L-system is defined as follows: $\mathbf{G} = \langle G, \omega, Q\rangle$, where (i) G (the *alphabet*) is a set of symbols containing elements that can be replaced (*variables*), namely they are active zones $\{V_1, V_2, ...\}$, which can be propagated towards attractants $\{A_1, A_2, ...\}$ by protoplasmic tubes and avoid repellents $\{R_1, R_2, ...\}$, i.e., $G = \{V_1, V_2, ...\} \cup \{A_1, A_2, ...\} \cup \{R_1, R_2, ...\}$; (ii) ω (*start*, *axiom* or *initiator*) is a string of symbols from G defining the initial state of the system, i.e., ω always belongs to $\{V_1, V_2, ...\}$; (iii) Q is a set of *production rules* or *productions* defining the way variables can be replaced with combinations of constants and other variables, i.e., production rules show a propagation of active zones by protoplasmic tubes towards attractants with avoiding repellents.

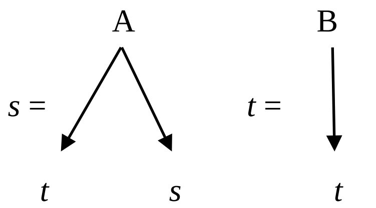
Let A, B, C are called primary strings, their meanings run over symbols V_1, V_2, ..., A_1, A_2, ... Production rules allow us to build composite strings from primary strings. So, a production A $\rightarrow_Q$ B consists of two strings, the *predecessor* A and the *successor* B. Some basic cases of productions are as follows: (i) the *fusion*, denoted AB $\rightarrow_Q$ C, means that two active zones A and B produce new active zone C at the place of an attractant denoted by C; (ii) the *multiplication*, A $\rightarrow_Q$ BC, means that the active zone A splits into two independent active zones B and C propagating along their own trajectories towards two different attractants denoted then by B and C; (iii) the *direction*, A $\rightarrow_Q$ B, means that the active zone A is translated to a source of nutrients B.

L-systems can generate infinite data structure. Therefore it is better to define some production rules, denoted by A $\rightarrow$ B, recursively like that: A $\rightarrow$ BA, producing an infinite sequence BABABABA… from A, or A $\rightarrow$ BCA, producing an infinite sequence BCABCABCABCA… from A. In the *Physarum* L-system, the rule A $\rightarrow$ BA means that we will fulfill the *direction*, A $\rightarrow_Q$ B, infinitely many time, the rule A $\rightarrow$ BCA means that we will fulfill the *multiplication*, A $\rightarrow_Q$ BC, infinitely many time. Let us consider an example of recursive production rules. Let G = {A, B} and let us start with the string A. Assume (A $\rightarrow$ BA) and (B $\rightarrow$ B). Thus, we obtain the following strings:

Generation $n = 0$: A
Generation $n = 1$: BA
Generation $n = 2$: BBA
Generation $n = 3$: BBBA
Generation $n = 4$: BBBBBA
Generation $n = 5$: BBBBBBA

In an appropriate *Physarum* L-system, these generations are represented as an infinite tree by permanent additions new attractants before the plasmodium propagation. In other words, we obtain the binary tree labeled with s and t, and whose interior nodes are either one unary node labeled with B or one binary node labeled with A (Fig. 2).

Figure 2. Example of labels for binary trees.

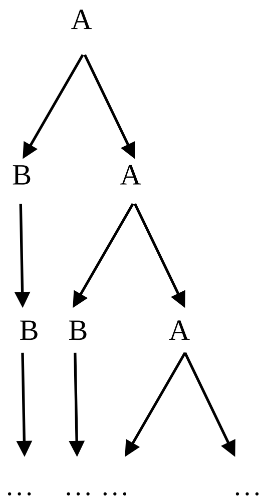


To sum up, we obtain the infinite binary tree of fig. 3.

If we are limited just by the *multiplication*, A $\to_Q$ BC, and the *direction*, A $\to_Q$ B, we can build up binary trees in *Physarum* L-systems using the following definition of binary trees labeled with x, y, …, whose interior nodes are either unary nodes labeled with u_1, u_2, … or binary nodes labeled with b_1, b_2, …:

1. the variables x, y, … are trees;
2. if t is a tree, then adding a single node labeled with one of u_1, u_2, … as a new root with t as its only subtree gives a tree;
3. if s and t are trees, then adding a single node labeled with one of b_1, b_2, … as a new root with s as the left subtree and t as the right subtree again gives a tree;
4. trees may go on forever.

Figure 3. Example of infinite binary tree.



Let Tr be the set of trees that we have been defined. Then our definition introduces a coalgebra [14]:

$$\mathrm{Tr} = \{x, y, \ldots\} \cup (\{u_1, u_2, \ldots\} \times \mathrm{Tr}) \cup (\{b_1, b_2, \ldots\} \times \mathrm{Tr} \times \mathrm{Tr}).$$

Thus, within L-systems, we can obtain complex structures including infinite structures defined coalgebraically. In some cases, it is better to deal with infinite structures (infinite trees), assuming that all strings are finite.

6.2. Aristotelian trees

Let us consider Aristotelian syllogistic trees, which can be large, but their strings are only of the length 1 or 2. An *Aristotelian syllogistic tree* is labeled with x, y, …, its interior nodes are n-ary nodes labeled with b_1, b_2, …, and it is defined as follows: (1) the variables x, y, … are Aristotelian syllogistic trees whose single descendants are underlying things (*hypokeimenon*, ὑποκείμενον) such that for each x, y, …, parents are supremums of descendants (notice that all underlying things are mutually disjoint); (2) if t_1, t_2, …, t_n are Aristotelian syllogistic trees such that their tops are concepts which are mutually disjoint and their supremum is $b_x \in \{b_1, b_2, \ldots\}$, then adding a single node labeled with b_x as a new root with t_1, t_2, …, t_n as its only subtrees gives an Aristotelian syllogistic tree; (3) an Aristotelian syllogistic tree is finite.

The idea of *hypokeimenon* allowed Aristotle to build up finite trees. He starts with underlying things as primary descendants of trees in constructing syllogistic databases. Now, let us define syllogistic strings of the length 1 or 2 by means a *Physarum* L-system. Let each $b_x \in \{b_1, b_2, \ldots\}$ be presented by an appropriate attractant and underlying things by initial active zones of Physarum. So, first trees x, y, …, whose single descendants are underlying things, are obtained by fusion or direction. Their supremums are denoted by attractants which were occupied by the first plasmodium propagation. These trees are considered subtrees for the next plasmodia propagation by fusion or direction. At the end, we can obtain just one supremum combining all subtrees. Let a_1, a_2, a_3,… be underlying things. Then they are initial strings, i.e., they can be identified with active zones of plasmodia. Their meanings are as follows: “there exists a_1”, “there exists a_2”, “there exists a_3”, … Assume that in the tree structure the supremum of a_1 and a_2 is b_1, the supremum of a_2 and a_3 is b_2, … These supremums are fusions of plasmodia. Then, we have the strings a_1b_1, a_2b_1, a_2b_2, a_3b_2, … Their meanings are as follows: “a_1 is b_1”, “a_2 is b_1”, “a_2 is b_2”, “a_3 is b_2”, … Further, let b_n be a supremum for b_1 and b_2. It denotes an attractant that was occupied by the plasmodium at the third step of the propagation. Our new strings are as follows: “b_1 is b_n”, “b_2 is b_n”, etc. Now we can appeal also to the following new production rule: if “x is y” and “y is z”, then “x is z”. Thus, we have the strings: a_1b_n, a_2b_n, a_2b_n, a_3b_n, …

6.3. Aristotelian syllogistic

The symbolic system of Aristotelian syllogistic can be implemented in the behavior of *Physarum* plasmodium. Let us design cells of *Physarum* syllogistic which will designate classes of terms. We can suppose that cells can possess different topological properties. This depends on intensity of chemo-attractants and chemo-repellents. The intensity entails the natural or geographical neighborhood of the set's elements in accordance with the spreading of attractants or repellents. As a result, we obtain Voronoi cells [3][11]. Let us define what they are mathematically. Let $\mathbf{P}$ be a nonempty finite set of planar points and $|\mathbf{P}| = n$. For points $p = (p_1, p_2)$ and $x = (x_1, x_2)$, let

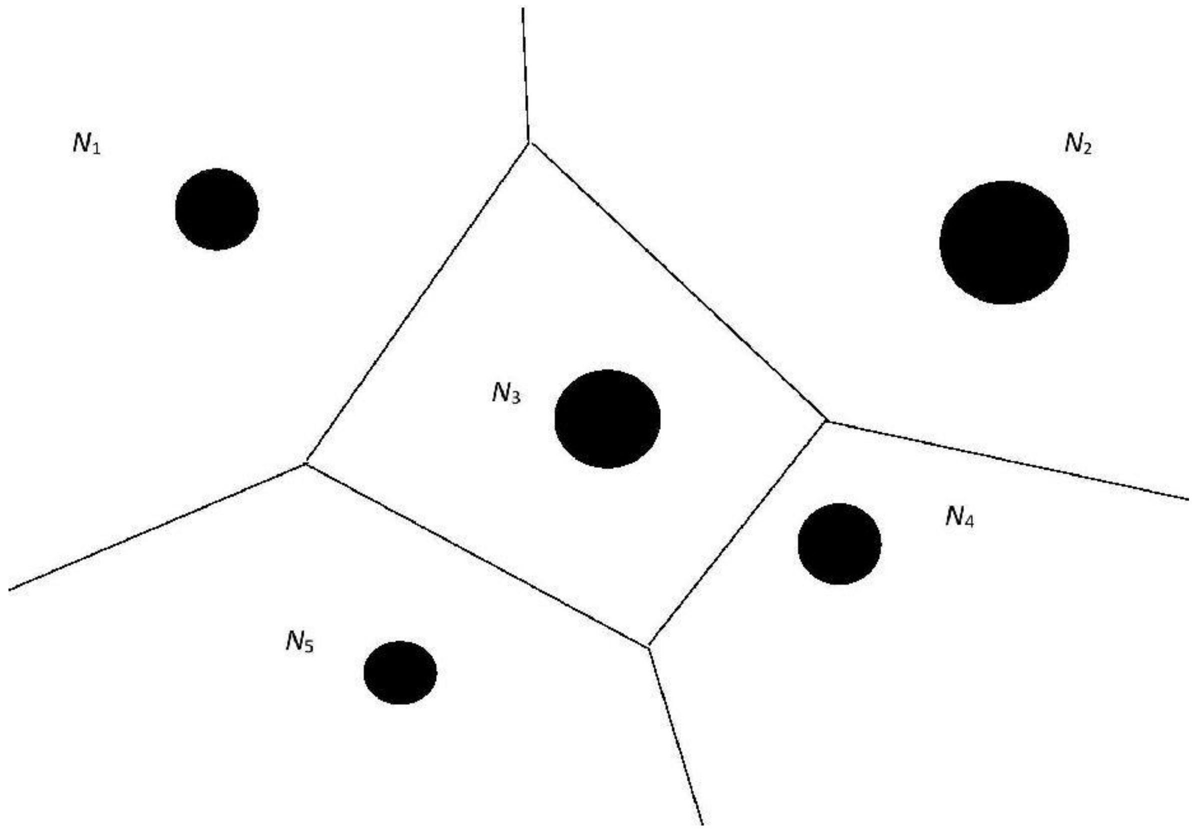
$$d(p,x) = \sqrt{(p_1 - x_1)^2 + (p_2 - x_2)^2}$$

denote their Euclidean distance. A planar Voronoi diagram of the set $\mathbf{P}$ is a partition of the plane into cells, such that for any element of $\mathbf{P}$, a cell corresponding to a unique point p contains all those points of the plane which are closer to p in respect to the distance d than to any other node of $\mathbf{P}$. A unique region

$$vor(p) = \bigcap_{m \in \mathbf{P}, m \neq p} \{z \in \mathbf{R}^2 : d(p,z) < d(m,z)\}$$

assigned to the point p is called a *Voronoi cell* of the point p. Within one Voronoi cell, a reagent has a full power to attract or repel the plasmodium. The distance d is defined by intensity of reagent spreading like in other chemical reactions simulated by Voronoi diagrams. A reagent attracts or repels the plasmodium and the distance on that it is possible corresponds to the elements of a given planar set $\mathbf{P}$. When two spreading wave fronts of two reagents meet, this means that on the board of meeting the plasmodium cannot choose its one further direction and splits (see Fig. 4). Within the same Voronoi cell, two active zones will fuse.

Figure 4. The Voronoi diagram for *Physarum*, where different attractants have different intensity and power.



Now, we can obtain coordinates $(x, y) \in \mathbf{Z}^2$ for each Voronoi center. The number (x, y) can be assigned to each concept as its character. If a Voronoi center with the coordinates (x_a, y_a) is presented by an attractant that is activated and occupied by the plasmodium, this means that in an appropriate *Physarum* syllogistic model there exists a string *a* with the coordinates (x_a, y_a). This string has the meaning "*a* exists". If a Voronoi center with the coordinates (x_a, y_a) is presented by a repellent that is activated and avoided by the plasmodium, this means that in an appropriate *Physarum* syllogistic model there exists a string [*a*] with the coordinates (x_a, y_a). This string has the meaning "*a* does not exist". If two neighbor Voronoi cells with the coordinates (x_a, y_a) and (x_b, y_b) of centers contain activated attractants which are occupied by the plasmodium and between both centers there are protoplasmic tubes, then in an appropriate *Physarum* syllogistic model there exists a string *ab* and a string *ba* where *a* has the coordinates (x_a, y_a) and *b* has the coordinates (x_b, y_b). The meaning of those strings is the same and it is as follows: "*ab* exist", "*ba* exist", "some *a* is *b*", "some *b* is *a*".

If one neighbor Voronoi cell with the coordinates (x_a, y_a) of its center contains an activated attractant which is occupied by the plasmodium and another neighbor Voronoi cell with the coordinates (x_b, y_b) of its centre contains an activated repellent which is avoided by the plasmodium, then in an appropriate *Physarum* L-system there exists a string *a*[*b*] and a string [*b*]*a* where *a* has the character (x_a, y_a) and [*b*] has the character (x_b, y_b). The meaning of those strings is the same and it is as follows: "*ab* do not exist, but *a* exists without *b*", "there exists *a* and no *a* is *b*", "no *b* is *a* and there exists *a*", "*a* exists and *b* does not exist".

If two neighbor Voronoi cells with the coordinates (x_a, y_a) and (x_b, y_b) of their centers contain activated repellents which are avoided by the plasmodium, then in an appropriate *Physarum* L-system there exists a string [*ab*] and a string [*ba*] where [*a*] has the character (x_a, y_a) and [*b*] has the character (x_b, y_b). The meaning of those strings is the same and it is as follows: "*ab* do not exist together", "there are no *a* and there are no *b*", "no *b* is *a*", "no *a* is *b*". Hence, existence propositions of Aristotelian syllogistic are spatially implemented in *Physarum* L-systems.

Let *y*' denote all neighbor Voronoi cells for *x* which differ from *y*. Now, let us consider a complex string *xy*&*x*[*y*']. The sign & means that we have strings *xy* and *x*[*y*'] simultaneously and they are considered the one complex string. The meaning of the string *xy*&*x*[*y*'] is a universal affirmative proposition "all *x* are *y*".

As a consequence, each *Physarum* L-system is considered a discourse universe verifying some propositions of Aristotelian syllogistic.

6.4. Non-Aristotelian syllogistic

Let us propose now the syllogistic system formalizing performative propositions of the form '*A* is *P*' (see [10][12]), i.e., propositions with context-based meanings. This system is said to be *synthetic* (*pragmatic*) *syllogistic*, while we are assuming that Aristotelian syllogistic is analytic (informative). The basic logical connectives of pragmatic syllogistic are as follows: a ('every + noun + is + adjective'), i ('some + noun + is + adjective'), e ('no + noun + is + adjective') and o ('some + noun + is not + adjective') that are defined in the following way:

$$S\mathrm{a}P := \exists A\ (A \text{ is } S) \wedge (\forall A(A \text{ is } S \wedge A \text{ is } P)). \tag{1}$$

$$S\mathrm{i}P := \forall A(\neg(A \text{ is } S) \wedge \neg(A \text{ is } P)). \tag{2}$$

$$S\mathrm{o}P := \neg(\exists A\ (A \text{ is } S) \wedge (\forall A(A \text{ is } S \wedge A \text{ is } P))), \text{ i.e.,} \\ \forall A \neg(A \text{ is } S) \vee (\exists A(\neg(A \text{ is } S) \vee \neg(A \text{ is } P))). \tag{3}$$

$$S\mathrm{e}P := \neg(\forall A(\neg(A \text{ is } S) \wedge \neg(A \text{ is } P))), \text{ i.e.,} \\ \exists A(A \text{ is } S \vee A \text{ is } P). \tag{4}$$

Now, let us formulate axioms of pragmatic syllogistic:

$$S\mathrm{a}P \Rightarrow S\mathrm{e}P. \tag{5}$$

$$S\mathrm{a}P \Rightarrow P\mathrm{a}S. \tag{6}$$

$$S\mathrm{i}P \Rightarrow P\mathrm{i}S. \tag{7}$$

$$S\mathrm{a}M \Rightarrow S\mathrm{e}P. \tag{8}$$

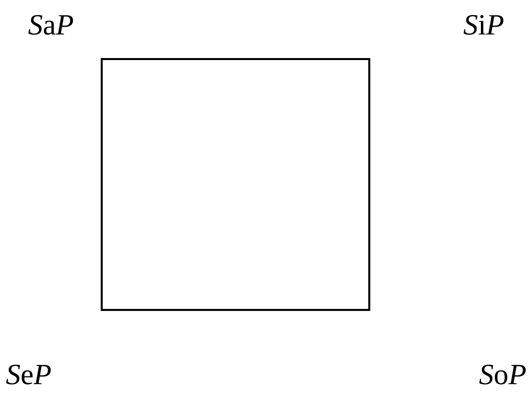
$$M\mathrm{a}P \Rightarrow S\mathrm{e}P. \tag{9}$$

$$(M\mathrm{a}P \wedge S\mathrm{a}M) \Rightarrow S\mathrm{a}P. \tag{10}$$

$$(M\mathrm{i}P \wedge S\mathrm{i}M) \Rightarrow S\mathrm{i}P. \tag{11}$$

In pragmatic syllogistic, we have a novel square of opposition that we call the *synthetic square of opposition* (see Fig. 5), where the following theorems are inferred from (1) – (11):

Figure 5. The synthetic square of opposition.



$S\mathrm{a}P \Rightarrow \neg(S\mathrm{o}P)$, $\neg(S\mathrm{o}P) \Rightarrow S\mathrm{a}P$, $S\mathrm{i}P \Rightarrow \neg(S\mathrm{e}P)$, $\neg(S\mathrm{e}P) \Rightarrow S\mathrm{i}P$, $S\mathrm{e}P \Rightarrow \neg(S\mathrm{i}P)$, $\neg(S\mathrm{i}P) \Rightarrow S\mathrm{e}P$, $S\mathrm{o}P \Rightarrow \neg(S\mathrm{a}P)$, $\neg(S\mathrm{a}P) \Rightarrow S\mathrm{o}P$, $S\mathrm{a}P \Rightarrow \neg(S\mathrm{i}P)$, $S\mathrm{i}P \Rightarrow \neg(S\mathrm{a}P)$, $\neg(S\mathrm{e}P) \Rightarrow S\mathrm{o}P$, $\neg(S\mathrm{o}P) \Rightarrow S\mathrm{e}P$, $S\mathrm{a}P \Rightarrow S\mathrm{e}P$, $S\mathrm{i}P \Rightarrow S\mathrm{o}P$, $S\mathrm{e}P \vee S\mathrm{i}P$, $\neg(S\mathrm{e}P \wedge S\mathrm{i}P)$, $S\mathrm{a}P \vee S\mathrm{o}P$, $\neg(S\mathrm{a}P \wedge S\mathrm{o}P)$, $\neg(S\mathrm{a}P \wedge S\mathrm{i}P)$, $S\mathrm{e}P \vee S\mathrm{o}P$.

For more details, see [10][12].

In the implementations within *Physarum* L-systems, the four basic syllogistic propositions of non-Aristotelian syllogistic defined above are understood as follows:

- ‘All *S* are *P*’: there is a string *AS* and for any *A* which is a neighbor for *S* and *P*, there are strings *AS* and *AP*. This means that we have a massive-parallel occupation of region, where the cells *S* and *P* are located.
- ‘Some *S* are *P*’: for any *A* which is a neighbor for *S* and *P*, there are no strings *AS* and *AP*. This means that the plasmodium cannot reach *S* from *P* or *P* from *S* immediately.
- ‘No *S* are *P*’: there exists *A* which is a neighbor for *S* and *P* such that there is a string *AS* or there is a string *AP*. This means that the plasmodium occupies *S* or *P*, but surely not the whole region, where the cells *S* and *P* are located.
- ‘Some *S* are not *P*’: for any *A* which is a neighbor for *S* and *P* there is no string *AS* or there exist *A* which is a neighbor for *S* and *P* such that there is no string *AS* or there is no string *AP*. This means that the plasmodium does not occupy *S* or there is a neighbor cell which is not connected with *S* or *P* by a protoplasmic tube.

Thus, the pragmatic syllogistic allows us to study different zones containing attractants for *Physarum* if they are connected by protoplasmic tubes homogenously.

Conclusion

We constructed two syllogistic versions of storage modification machine in *Physarum polycephalum*: Aristotelian syllogistic and pragmatic syllogistic (non-Aristotelian syllogistic of section 6.4). While Aristotelian syllogistic may describe concrete directions of *Physarum* spatial expansions, pragmatic syllogistic may describe *Physarum* simultaneous propagations in all directions. Therefore, while for the implementation of Aristotelian syllogistic we need repellents to avoid some possibilities in the *Physarum* propagations, for the implementation of pragmatic syllogistic we do not need them. Hence, the second syllogistic can simulate massive-parallel behaviors, including different form of propagations such as processes of public opinion formation.

In our opinion, the general purpose of *Physarum* computing covers many behavioural sciences, because the slime mould’s behaviour can be considered the simplest natural intelligent behaviour. Thus, our results may have an impact on computational models in behavioural sciences in general.

Acknowledgment

This research is supported by FP7-ICT-2011-8 and UMO-2012/07/B/HS1/00263.

References

[1] Adamatzky, A., Erokhin, V., Grube, M., Schubert, Th., Schumann, A., *Physarum Chip Project: Growing Computers From Slime Mould*, International Journal of Unconventional Computing, 8(4), pp. 319-323, 2012.

[2] Adamatzky, A., *Physarum machine: implementation of a Kolmogorov-Uspensky machine on a biological substrate,* Parallel Processing Letters, vol. 17, no. 4, pp. 455-467, 2007.

[3] Adamatzky, A., *Physarum Machines: Computers from Slime Mould,* World Scientific Series on Nonlinear Science, Series A, World Scientific Publishing Company, 2010.

[4] Lindenmayer, A., *Mathematical models for cellular interaction in development. Parts I and II,* Journal of Theoretical Biology, 18, pp. 280-299; 300-315, 1968.

[5] Łukasiewicz, J., *Aristotle's Syllogistic From the Standpoint of Modern Formal Logic,* Oxford Clarendon Press, 2nd edition, 1957.

[6] Niklas, K., *Computer Simulated Plant Evolution*, Scientific American, 1985.
[7] Schumann, A., Adamatzky, A., *Logical Modelling of Physarum Polycephalum*, Analele Universitatii Din Timisoara, seria Matemtica-Informatica 48 (3), pp. 175-190, 2010.
[8] Schumann, A., Adamatzky, A., *Physarum Spatial Logic*, New Mathematics and Natural Computation 7 (3), pp. 483-498, 2011.
[9] Schumann, A., Akimova, L., *Simulating of Schistosomatidae (Trematoda: Digenea) Behaviour by Physarum Spatial Logic*, Annals of Computer Science and Information Systems, vol. 1, Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, IEEE Xplore, pp. 225-230, 2013.
[10] Schumann, A., *On Two Squares of Opposition: the Leśniewski's Style Formalization of Synthetic Propositions*, Acta Analytica 28, pp. 71-93, 2013.
[11] Schumann, A., Pancerz, K., *Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing*, [in:] Szczuka, M., Czaja, L., Kacprzak, M., (eds.), *Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P'2013)*, Warsaw, Poland, September 25-27, pp. 389-397, 2013.
[12] Schumann, A., *Two Squares of Opposition: for Analytic and Synthetic Propositions*, Bulletin of the Section of Logic 40 (3/4), pp. 165-178, 2011.
[13] Prusinkiewicz, P., Lindenmayer, A., *The Algorithmic Beauty of Plants*, Springer-Verlag, 1990.
[14] Rutten, J.M., *Universal coalgebra: a theory of systems*, Theor. Comput. Sci., 249 (1), pp. 3-80, 2000.

Originally published: Schumann, A., *Rationality in the Behaviour of Slime Moulds and the Individual-Collective Duality,* AISB Convention, 2015.

VII
Rationality in the Behaviour of Slime Moulds and the Individual-Collective Duality

Andrew Schumann

Abstract

We introduce the notion of the so-called context-based games to describe rationality of the slime mould. In these games we assume that, first, strategies can change permanently, second, players cannot be defined as individuals performing just one action at each time step. They can perform many actions simultaneously. In other words, each player can behave as an individual or as a collective of individuals. This significant feature of context-based games is called individual-collective duality.

Introduction

In *Physarum Chip Project: Growing Computers From Slime Mould* [1] supported by FP7 we are going to design an unconventional computer on programmable behaviour of *Physarum polycephalum*, a one-cell organism that behaves by its plasmodium that is sensible to different stimuli called attractants, it looks for them and in case it finds them, it propagates protoplasmic tubes toward those attractants. These motions can be regarded as the basic medium of simple actions that are intelligent [1], [3], [4].

Notice that the *Physarum* motions are a kind of *natural transition systems*, $\langle$States, Edg$\rangle$, where States is a set of states presented by attractants and Edg $\subseteq$ States $\times$ States is a transition of plasmodium from one attractant to another. The point is that the plasmodium looks for attractants, propagates protoplasmic tubes towards them, feeds on them and goes on. As a result, a transition system is built up. Now, labelled transition systems have been used for defining the so-called *concurrent games*, a new semantics for games proposed by Samson Abramsky. Traditionally, a play of the game is formalized as a sequence of moves. This way assumes the polarization of two-person games, when in each position there is only one player's turn to move. In concurrent games, players can move concurrently.

On the medium of *Physarum polycephalum* we can, first, define concurrent games and, second, extend the notion of concurrent games strongly and introduce the so-called *context-based games*. In these games we assume that strategies can change permanently. Another feature of context-based games is that players cannot be defined as individuals who perform just one action at each time step. They can perform many actions simultaneously. So, each player can behave as an individual or as a collective of individuals. This significant feature of context-based games is called *individual-collective duality*.

In this paper, we will talk about the notion of rationality within context-based games.

7.1. Actions of plasmodia

Physarum polycephalum verifies the following three basic operations which transform one states to others in ⟨States, Edg⟩: fusion, multiplication, and direction. (i) The *fusion* means that two active zones (attractants occupied by the plasmodium) either produce new active zone (i.e. there is a collision of the active zones) or just a protoplasmic tube. (ii) The *multiplication* means that the active zone splits into two independent active zones propagating along their own trajectories. (iii) The *direction* means that the active zone is not translated to a source of nutrients but to a domain of an active space with certain initial velocity vector. These three operations can be examined as the most basic forms of intelligent behaviour of living organisms. For example, in the paper [4] we showed that the behaviour of collectives of the genus *Trichobilharzia* Skrjabin & Zakharov, 1920 (Schistosomatidae Stiles, Hassall 1898) can be simulated in the *Physarum* spatial logic. This means that, first, a local group of Schistosomatidae can behave as a programmable biological computer, second, a biologized kind of process calculus such as *Physarum* transition system can describe concurrent biological processes at all.

The main result of our research is that, on the one hand, the *Physarum* motions are intelligent, but, on the other hand, they do not verify the *induction principle* (when the minimal set satisfying appropriate properties is given). This means that they can implement Kolmogorov-Uspensky machines or other spatial algorithms only in a form of approximation, because *Physarum* performs much more, than just conventional calculations (the set realised is not minimal), i.e. it achieves goals (attractants) not only by “Caesarian” straight paths.

Let us consider the following thought experiment as counterexample showing that the set of actions for the plasmodium is infinite in principle, therefore we cannot implement Kolmogorov-Uspensky machines. Assume that the transition system for the plasmodium consists just of one action presented by one neighbour attractant. The plasmodium is expected to propagate a protoplasmic tube towards this attractant. Now, let us place a barrier with one slit in front of the plasmodium. Because of this slit, the plasmodium can be propagated according to the shortest distance between two points and in this case the plasmodium does not pay attention on the barrier. However, sometimes the plasmodium can evaluate the same barrier as a repellent for any case and it gets round the barrier to reach the attractant according to the longest distance. So, even if the environment conditions change a little bit, the behaviour changes, too. The plasmodium is very sensible to the environment.

Thus, simple actions of *Physarum* plasmodia cannot be regarded as atomic so that composite actions can be obtained over them inductively. In other words, it is ever possible to face a hybrid action which is singular, but it is not one of the basic simple actions. It is a hybrid of them.

In the transition system with only one stimulus presented by one attractant, a passable barrier can be evaluated as a repellent 'for any case'. Therefore, the transition system with only one stimulus and one passable barrier may have the following three simple actions: (i) pass trough, (ii) avoid from left, (iii) avoid from right. But in essence, we deal only with one stimulus and, therefore, with one action, although this action has the three modifications defined above.

Simple actions which have modifications depending on the environment are called *hybrid*. The problem is that the set of actions in any labelled transition systems must consist of the so-called atomic actions – simple actions that have no modifications.

7.2. Individual-collective duality and non-additivity

In context-based games, we cannot use conventional probability theory. The matter is that if we assume the existence of hybrid actions, then the entities of games are certain and, therefore, cannot be additive.

The double slit experiment with the plasmodium of *Physarum polycephalum* is the best example of that conventional probability theory is unapplied for *Physarum* acts. Let us take the first screen with two slits which are covered or opened and the second screen behind the first at which attractants are distributed evenly. Before the first screen there is an active zone of plasmodium. Then let us perform the following three experiments: (i) slit 1 is opened, slit 2 is covered; (ii) slit 1 is covered, slit 2 is opened; (iii) both slit 1 and 2 are opened. In the first (second) experiment protoplasmic tubes arrive at the screen at random in a region somewhere opposite the position of slit 1 (slit 2). Let us denote all tubes landing at the second screen by A, thereby all tubes that pass through slit 1 by A_1 and all tubes that pass through slit 2 by A_2. Now we can check in case of *Physarum* if there is a partition of set A into sets A_1 and A_2. We open both slits. Then we see that the plasmodium behaves like electrons, namely it can propagates just one tube passing through either slit 1 or slit 2 or it can propagates two tubes passing through both slits simultaneously. In the second case, these tubes split before the second screen and appear to occur randomly across the whole screen. Thus, the total probability $P(A)$, corresponding to the intensity of plasmodium reaching the screen, is not just the sum of the probabilities $P(A_1)$ and $P(A_2)$. This means that the plasmodium has the fundamental property of electrons, discovered in the double-slit experiment. It is the proof of non-additivity of probabilities.

Economics and conventional business intelligence tries to continue the empiricist tradition, where reality is measurable and additive, and in statistical and econometric tools they deal only with the measurable additive aspects of reality. They try to obtain additive measures in economics and studies of real intelligent behaviour, also. Nevertheless, there is always the possibility that there are important variables of economic systems which are unobservable and non-additive in principle. We should understand that statistical and econometric methods can be rigorously applied in economics just after the presupposition that the phenomena of our social world are ruled by stable causal relations between variables. However, let us assume that we have obtained a fixed parameter model with values estimated in specific spatio-temporal contexts. Can it be exportable to totally different contexts? Are real social systems governed by stable causal mechanisms with atomistic and additive features?

Hence, our study of context-based games on the medium of *Physarum polycephalum* can make impacts for many behavioural sciences: game theory, behavioural economics, behavioural finance, etc.

Non-additivity of phenomena does not mean that they cannot be studied mathematically. There are some rigorous approaches such as p-adic probability theory, which allow us to do it. The most significant feature of p-adic probabilities (or more generally, non-Archimedean probabilities or probabilities on infinite streams) is that they do not satisfy additivity. On the one hand, the p-adic analogies of the central limit theorem in real numbers face the problem that the normalized sums of independent and i.d. random variables do not converge to a unique distribution, there are many limit points, therefore there is no connection with the usual bell type curve. In other words, in p-adic distributions we cannot build up the Gauss curve as fundamental notion of statistics and econometrics. On the other hand, the powerset over infinite streams like p-adic numbers is not a Boolean algebra in general case. In particular, there is no additivity (we cannot obtain a partition for any set into disjoint subsets whose sum gives the whole set). Using p-adic (non-Archimedean) probabilities we can disprove Aumann's agreement theorem and develop new mathematical tools for game theory, in particular define context-based games by means of coalgebras or cellular automata. In these context-based games we can appeal just to non-Archimedean probabilities. These games can describe and formalize complex reflexive processes of behavioural finances (such as short selling or long buying).

Notice that the p-adic number system for any prime number p extends the ordinary arithmetic of the rational numbers in a way different from the extension of the rational number system to the real and complex number systems. The extension is achieved by an alternative interpretation of the concept of absolute value.

Let us suppose that the sample space of probability theory is not fixed, but changes continuously. It can grow, be expanded, decrease or just change in itself. In this case we will deal not with atoms as members of sample space, but with streams. The powerset of this growing set cannot be a Boolean algebra and probability measure is not additive.

We can consider *Physarum* behaviours within a certain topology of attractants and repellents as growing sample space. Assume that there are two neighbour attractants *a* and *b*. We say that there is a string *ab* or *ba* if both attractants *a* and *b* are occupied by the plasmodium. As a result, we observe a continuous expansion of the set of strings. It can be regarded as a sample space of probability theory. Its values will be presented by p-adic integers.

Let us show, how we can build up the sample space Ω^{ω} constructively. Suppose that Ω consists of $p - 1$ attractants and A, B, ... are subsets of Ω. Such A, B, ... are conditions (properties) of the experiment we are performing. For instance, let A := “Attractants accessible for the attractant N_1 by protoplasmic tubes” and B := “Neighbours for the attractant N_1”, etc. Some conditions of the experiment, fixed by subsets of Ω^{ω}, do not change for different time $t = 0, 1, 2, \ldots$ Some other conditions change for different time $t = 0, 1, 2, \ldots$ So, we can see that the property B is verified on the same number of members of Ω for any time $t = 0, 1, 2, \ldots$ Nevertheless, the property A is verified on a different number of members for different time $t = 0, 1, 2, \ldots$ Thus, describing the experiment, we deal not with properties A, B, etc., but with properties A^{ω}, B^{ω}, etc. Let us define the cardinality number of $X^{\omega} \subseteq \Omega^{\omega}$ as follows: $|X^{\omega}| := (|X|$ for $t = 0$; $|X|$ for $t = 1$; $|X|$ for $t = 2, \ldots)$, where $|X|$ means a cardinality number of X. Notice that if $|\Omega| = p - 1$, then $|A^{\omega}|$, $|B^{\omega}|$, and $|\Omega^{\omega}|$ cover p-adic integers.

The simplest way to define p-adic probabilities is as follows:

$$P(A^{\omega}) = |\, A^{\omega} \,| \text{ or } P(A^{\omega}) = |\, A^{\omega} \,| \,/\, |\, \Omega^{\omega} \,|$$

Notice that in p-adic metric, $|\, \Omega^{\omega} \,| = -1$

Agent i's knowledge structure is a function $\mathbf{P}_i$ which assigns to each $a \in \Omega^{\omega}$ a non-empty subset of Ω^{ω}, so that each world a belongs to one or more elements of each $\mathbf{P}_i$, i.e. Ω^{ω} is contained in a union of $\mathbf{P}_i$, but $\mathbf{P}_i$ are not mutually disjoint. The function $\mathbf{P}_i$ is interpreted on p-adic probabilities.

$$K_i A^{\omega} = \{a : A^{\omega} \subseteq P_i(\omega)\}$$

The double-slit experiment with *Physarum polycephalum* shows that, first, we cannot extract atomic actions from all the kinds of the plasmodium behaviour, second, probability measures used in describing this experiment are not additive. We can deal just with hybrid actions.

The informal meaning of hybrid actions (e.g. hybrid terms or hybrid formulas) is that any hybrid action is defined just on streams and we cannot say in accordance with which stream the hybrid action will be embodied in the given environment. It can behave like any stream it contains but there is an uncertainty how exactly.

7.3. Conclusions and future work

Thus, context-based games on the medium of *Physarum polycephalum* can have many impacts in the development of unconventional computing: from behavioural sciences to quantum computing and many other fields.

So, if we perform the *double-slit experiment* for *Physarum polycephalum*, we detect self-inconsistencies showing that we cannot approximate atomic individual acts of Physarum as well as it is impossible to approximate single photons. From the standpoint of measure theory, it means that we cannot define additive measures for *Physarum* actions. In our opinion, it is a fundamental result for many behavioural sciences. Non-additivity of actions can be expressed in different ways: (i) *natural transition systems, such as Physarum behaviour, cannot be reduced to Kolmogorov-Uspensky machines*, although their actions are intelligent, (ii) *there is an individual-collective duality, when we cannot approximate atomic individual acts* (an individual, such as plasmodium, can behaves like a collective and a collective, such as collective of plasmodia, can behaves like an individual).

Acknowledgements

This research is supported by FP7-ICT-2011-8.

References

[1] Adamatzky, A., Erokhin, V., Grube, M., Schubert, Th., Schumann, A., *Physarum Chip Project: Growing Computers From Slime Mould,* Int. J. of Unconventional Computing, 8(4): 319-323, 2012.

[2] Schumann, A., *Payoff Cellular Automata and Reflexive Games,* J. of Cellular Automata, 9(4): 287-313, 2015.

[3] Schumann, A., Akimova, L., *Simulating of Schistosomatidae (Trematoda: Digenea) Behavior by Physarum Spatial Logic, Annals of Computer Science and Information Systems, vol. 1, Proceedings of the 2013 Federated Conference on Computer Science and Information Systems,* IEEE Xplore, pp. 225-230, 2013.

[4] Schumann, A., Pancerz, K., *Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing,* [in:] Szczuka, M., Czaja, L., Kacprzak, M. (eds.), *Proceedings of the Workshop on Concurrency,* Specification and Programming (CS&P'2013), Warsaw, Poland, September 25-27, pp. 389-397, 2013.

Originally published: Schumann, A., Pancerz, K., *Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing*, [in:] Szczuka, M., Czaja, L., Kacprzak, M. (eds.), *Proceedings of the Workshop on Concurrency,* Specification and Programming (CS&P'2013), Warsaw, Poland, September 25-27, pp. 389-397, 2013.

VIII
Towards an Object-Oriented Programming Language for *Physarum Polycephalum* Computing

Andrew Schumann, Krzysztof Pancerz

Abstract

In the paper, we present foundations of a new object-oriented programming language for *Physarum polycephalum* computing. Both theoretical foundations and assumptions for a language specification are considered. *Physarum polycephalum* is a one-cell organism. In the phase of plasmodium, its behavior can be regarded as a biological substrate that implements the Kolmogorov-Uspensky machine which is the most generalized and nature-oriented version of a mathematical machine. The proposed language will be used for developing programs for *Physarum polycyphalum* by the spatial configuration of stationary nodes (inputs).

Introduction

Physarum polycephalum is a one-cell organism belonging to *Physarales*, subclass *Myxogastromycetidae*, class *Myxomycetes* and division *Myxostelida*. In the phase of plasmodium, it looks like an amorphous giant amoeba with networks of protoplasmic tubes. It feeds on bacteria, spores and other microbial creatures (substances with potentially high nutritional value) by propagating towards sources of food particles and occupying these sources. A network of protoplasmic tubes connects the masses of protoplasm. As a result, the plasmodium develops a planar graph, where the food sources or pheromones are considered as nodes and protoplasmic tubes as edges. This fact allows us to claim that plasmodium behavior can be regarded as a biological implementation of Kolmogorov-Uspensky machines [7]. The modification of locations of nutrients (food sources) causes a storage modification of plasmodium. Hence, the plasmodium may be used for developing a biological architecture of different abstract automata such as Kolmogorov-Uspensky machines [16, 22], Tarjan's reference machine [21], and Schönhage's storage modification machines [19, 22]. In *Physarum Chip Project: Growing Computers From Slime Mould* [2] supported by FP7 we are going to implement programmable amorphous biological computers in plasmodium of *Physarum*. This abstract computer is called *slime mould based computer*.

One of the paths of our research in this area concerns creating a new programming language that simulates plasmodium behavior. The following main tasks can be distinguished in the first step of this path:

- Constructing the programming language on the basis of storage machines. The static storage structure is represented by a two-dimensional configuration of point-wise sources of chemo-attractants and chemo-repellents.
- Constructing the programming language on the basis of the Kolmogorov-Uspensky machine (KUM), where edges are represented by protoplasmic strands.
- Developing programs represented by the spatial configuration of stationary nodes (treated as inputs of the programs). Outputs of the programs may be recorded optically.

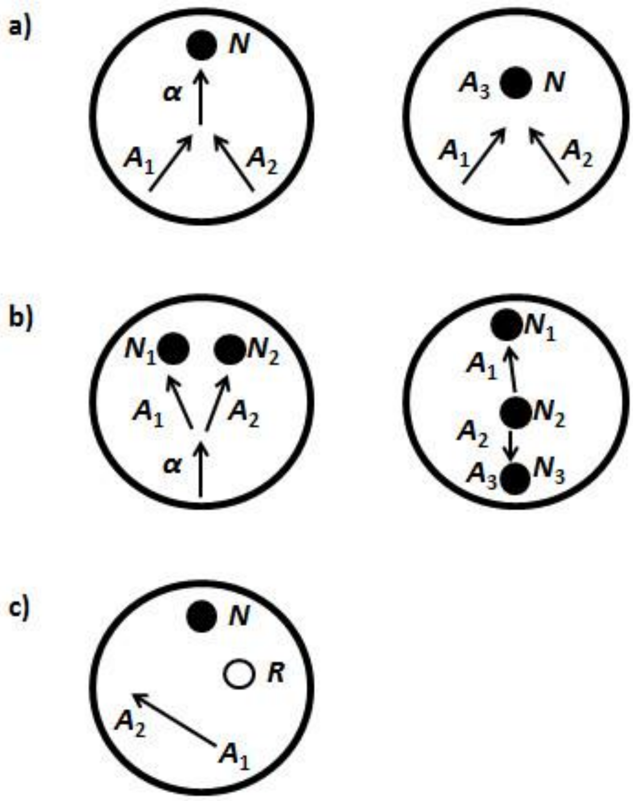
The rest of the paper is organized as follows. In section 8.2, we give foundations of specification of a new language. Assumptions of specification are preceded by a theoretical background of *Physarum* automata (see Section 8.1).

8.1. *Physarum* automata

Plasmodium's active zones of growing pseudopodia interact concurrently and in a parallel manner. At these active zones, three basic operations stimulated by nutrients and some other conditions can be observed: fusion, multiplication, and direction operations. The *fusion* $Fuse$ means that two active zones A_1 and A_2 both produce new active zone A_3 (i.e. there is a collision of the active zones). The *multiplication* $Mult$ means that the active zone A_1 splits into two independent active zones A_2 and A_3 propagating along their own trajectories. The *direction* $Direct$ means that the active zone A is not translated to a source of nutrients but to a domain of an active space with a certain initial velocity vector *v*. These operations, $Fuse$, $Mult$, $Direct$, can be determined by the following stimuli:

- The set of attractants $\{N_1, N_2, \ldots\}$. Attractants are sources of nutrients or pheromones, on which the plasmodium feeds. Each attractant N is characterized by its position and intensity. It is a function from one active zone to another.
- The set of repellents $\{R_1, R_2, \ldots\}$. Plasmodium of *Physarum* avoids light and some thermo- and salt-based conditions. Thus, domains of high illumination (or high grade of salt) are repellents such that each repellent R is characterized by its position and intensity, or force of repelling. In other words, each repellent R is a function from one active zone to another.

Figure 1. The stimulation of the following operations in *Physarum* automata: (a) fusion, (b) multiplication, and (c) direction, where A_1, A_2, A_3 are active zones, N, N_1, N_2, N_3 are attractants, α is a protoplasmic tube, R is a repellent.



Such plasmodium behavior can be presented as an implementation of some abstract automata.

8.1.1. *Physarum* cellular automata

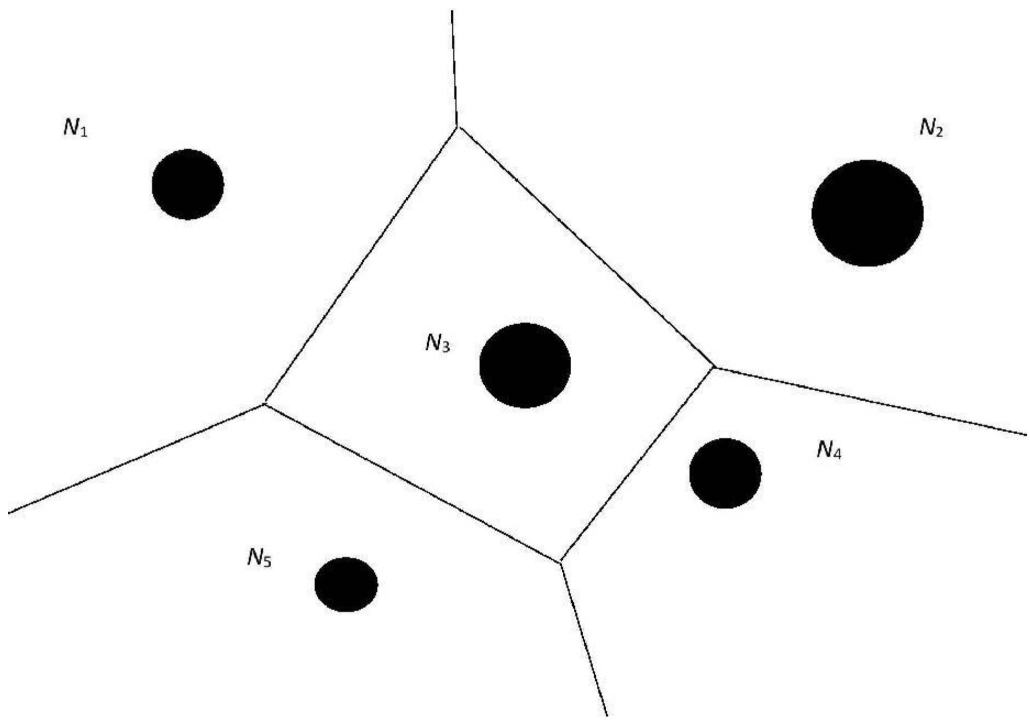
Recall that a cellular automaton is a 4-tuple $\mathrm{A} = \langle \mathbf{Z}^d, S, u, f \rangle$, where (1) $d \in \mathbf{N}$ is a number of dimensions, and the members of $\mathbf{Z}^d$ are referred to as cells, (2) S is a finite set of elements called the states of an automaton A, the members of $\mathbf{Z}^d$ take their values in S, (3) $u \subset \mathbf{Z}^d \setminus \{0\}^d$ is a finite ordered set of n elements, $u(x)$ is said to be a neighborhood for the cell x, (4) $f : S^{n+1} \to S$ that is f is the local transition function (or local rule). As we see an automaton is considered on the endless d-dimensional space of integers, i.e., on $\mathbf{Z}^d$. Discrete time is introduced for $t = 0,1,2,\ldots$ For instance, the cell x at time t is denoted by x^t. Each automaton calculates its next state depending on states of its closest neighbors. The cellular automata thus represent locality of physics of information and massive-parallelism in space-time dynamics of natural systems.

In abstract cellular automata, cells are physically identical. They can differ just by one of the possible states of S. In case of *Physarum*, cells can possess different topological properties. This depends on intensity of chemo-attractants and chemo-repellents. The intensity entails the natural or geographical neighborhood of the set's elements in accordance with the spreading of attractants or repellents. As a result, we obtain Voronoi cells. Let us define what they are mathematically. Let $\mathbf{P}$ be a nonempty finite set of planar points and $|\mathbf{P}| = n$. For points $p = (p_1, p_2)$ and $x = (x_1, x_2)$ let $d(p,x) = \sqrt{(p_1 - x_1)^2 + (p_2 - x_2)^2}$ denote their Euclidean distance. A planar Voronoi diagram of the set $\mathbf{P}$ is a partition of the plane into cells, such that for any element of $\mathbf{P}$, a cell corresponding to a unique point p contains all those points of the plane which are closer to p in respect to the distance d than to any other node of $\mathbf{P}$. A unique region

$$vor(p) = \bigcap_{m \in \mathbf{P}, m \neq p} \{z \in \mathbf{R}^2 : d(p,z) < d(m,z)\}$$

assigned to a point p is called a *Voronoi cell* of the point p. Within one Voronoi cell a reagent has a full power to attract or repel the plasmodium. The distance d is defined by the intensity of reagent spreading. A reagent attracts or repels the plasmodium and the distance, on which it is possible, corresponds to the elements of a given planar set P. When two spreading wave fronts of the two reagents meet, this means that on the board of meeting the plasmodium cannot choose its one further direction and splits (see Figure 2).

Figure 2. The Voronoi diagram for *Physarum,* where different attractants have different intensity and power.



The direction of protoplasmic tubes is defined by concentrations of chemo-attractants or chemo-repellents in Voronoi neighborhood. Each dynamics of protoplasmic tube can be characterized at time step t by its current position x_t and the angle α_t.

8.1.2. *Physarum* Kolmogorov-Uspensky machines

Let Γ be an alphabet, k a natural number. We say that a tree is (Γ, k)-tree, if one of nodes is designated and is called *root* and all edges are directed. Each node is labeled by one of the signs of Γ and each edge from the same node is labeled by different numbers $\{1,\ldots,k\}$ (so, each node has not more than k edges). We see that by this definition of (Γ, k)-tree, the pseudopodia growing from the one active zone, where all attractants are labeled by signs of Γ, and protoplasmic tubes are labeled by numbers of $\{1,\ldots,k\}$, is a (Γ, k)-tree.

Let r be the maximal possible path of $(\Gamma; k)$-tree. We can always design *Physarum* Voronoi diagrams (using attractants and repellents) for inducing different numbers r and appropriate local properties. The $(\Gamma; k)$-tree limited by r is called $(\Gamma; k)$-complex. Programming in Kolmogorov-Uspensky machines is considered as transforming one $(\Gamma; k)$-complex to another with the same r by changing nodes and edges using some rules. In case of *Physarum* implementation of Kolmogorov-Uspensky machines programming is presented as transforming one Voronoi diagram into another with the same r by dynamics of *Physarum* (e.g. when some attractants become eaten by *Physarum*).

The simpler version of the Kolmogorov-Uspensky machines is presented by Schönhage's storage modification machines.

8.1.3. *Physarum* Schönhage's storage modification machines

These machines consist of a fixed alphabet of input symbols, Γ, and a mutable directed graph with its arrows labeled by Γ. The set of nodes X, identified with attractants is finite, as well. One fixed node $a \in X$ is identified as a distinguished center node of the graph. It is the first active zone of growing pseudopodia. The distinguished node a has an edge x such that $x_\gamma(a) = a$ for all $\gamma \in \Gamma$. That is, all pointers from the distinguished center node point back to the center node. Each $\gamma \in \Gamma$ defines a mapping x_γ from X to X. Each word of symbols in the alphabet Γ is a pathway through the machine from the distinguished center node.

Schönhage's machine modifies storage by adding new elements and redirecting edges. Its basic instructions are as follows:

- Creating a new node: **new** W. The machine reads the word W, following the path represented by the symbols of W until the machine comes to the last symbol in the word.

It causes a new node y, associated with the last symbol of W, to be created and added to X. Adding a new node means adding a new attractant within a *Physarum* Voronoi diagram.

- A pointer redirection: **set** W **to** V. This instruction redirects an edge from the path represented by word W to a former node that represents word V. It means that we can remove some attractants within a *Physarum* Voronoi diagram.
- A conditional instruction: **if** $V = W$ **then instruction** Z. It compares two paths represented by words W and V and if they end at the same node, then we jump to instruction Z, otherwise we continue. This instruction serves to add edges between existing nodes. It corresponds to the splitting or fusion of *Physarum*.

8.2. Foundations of specification of an object-oriented programming language for *Physarum polycephalum*

The plasmodium of *Physarum polycephalum* functions as a parallel amorphous computer with parallel inputs and parallel outputs. Data are represented by spatial configurations of sources of nutrients. Therefore, we can generally assume that a program of computation is coded via configurations of repellents and attractants. The plasmodium of *Physarum polycephalum* is a computing substrate. In [2], Adamatzky underlined that *Physarum* does not compute. It obeys physical, chemical and biological laws. Its behavior can be translated to the language of computations.

In this section, we deal with foundations of specification of a new object-oriented programming language for *Physarum polycephalum* computing on the basis of using a Voronoi diagram for implementing Kolmogorov-Uspensky machines. In an object-oriented programming (OOP) paradigm, concepts are represented as objects that have data fields (properties describing objects) and associated procedures known as methods. The OOP approach assumes that properties describing objects are not directly accessible by the rest of the program. They are accessed by calling special methods, which are bundled in with the properties. This approach has been implemented in our new language. Moreover, we have referred to conventions used in the JavaBeans API [4], i.e., the object properties must be accessible using *get*, *set*, and *is* (used for Boolean properties instead of *get*). They are called accessor methods. For readable properties, there are getter methods reading the property values. For writable properties, there are setter methods allowing the property values to be set or updated.

Our new language has been proposed as a prototype-based programming language like, for example, Self [1], JavaScript and other ECMAScript implementations [2]. Unlike traditional class-based object-oriented languages, it is based on a style of object-oriented programming in which classes are not present. Behavior reuse is performed via a process of cloning existing objects that serve as prototypes. This model is also known as instance-based programming.

Table 1. Main objects identified in *Physarum polycephalum* computing.

Object	Properties
$Layer$	id, $size$, $elements$
$Physarum$	id, $position$, $intensity$
$Attractant$	id, $position$, $intensity$
$Repellent$	id, $position$, $intensity$

The main objects identified in *Physarum polycephalum* computing are collected in table 1. We assume that a computational space is divided into two-dimensional computational layers on which *Physarum polycephalum*, as well as attractants and repellents, can be scattered. Our approach allows interaction between elements placed on different layers. This property enables us to use, in the future, the multi-agent paradigm in *Physarum polycephalum* computing. The user can define, in the computational space, as many computational layers as needed. For each layer, its size can be determined individually. We apply the point-wise configuration of elements scattered on the layers. Therefore, for each element (*Physarum*, attractant, repellent), its position can be determined using two integers (coordinates). As it was mentioned in section 2, attractants and repellents are characterized by the property called intensity. This property plays an important role in creation of the Voronoi cells. For each attractant and repellent, the intensity is a fuzzy value from the interval $[0,1]$, where 1 denotes the maximal intensity, while 0 the minimal intensity, i.e., a total lack of impact of a given attractant or repellent on *Physarum polycephalum*. The force of attracting (repelling) of *Physarum* is a combination of intensity of attractants (repellents) and distances between plasmodium and attractants (repellents), respectively.

Let $p = (p_1, p_2)$ and $x = (x_1, x_2)$ be points on the layer where *Physarum* and attractant (repellent), respectively, are located. To create the Voronoi cells, we can use the following measure modyfying a distance, which is commonly used:

$$f(p,x) = \frac{1}{\varepsilon(x)}\sqrt{(p_1 - x_1)^2 + (p_2 - x_2)^2},$$

where $\varepsilon(x)$ is the intensity of attractant (repellent) placed at x. It means that the Voronoi cells cover the force of attracting (repelling) of plasmodium instead of simple distances between it and attractants (repellents). In the current version of the language, the Voronoi cells are built within layers only.

Analogously to layers, the user can create and scatter on layers as many elements as needed.

Below, we present an exemplary fragment of a code in our language responsible for creating the layer and elements, setting individual properties of elements and scattering elements on the layer.

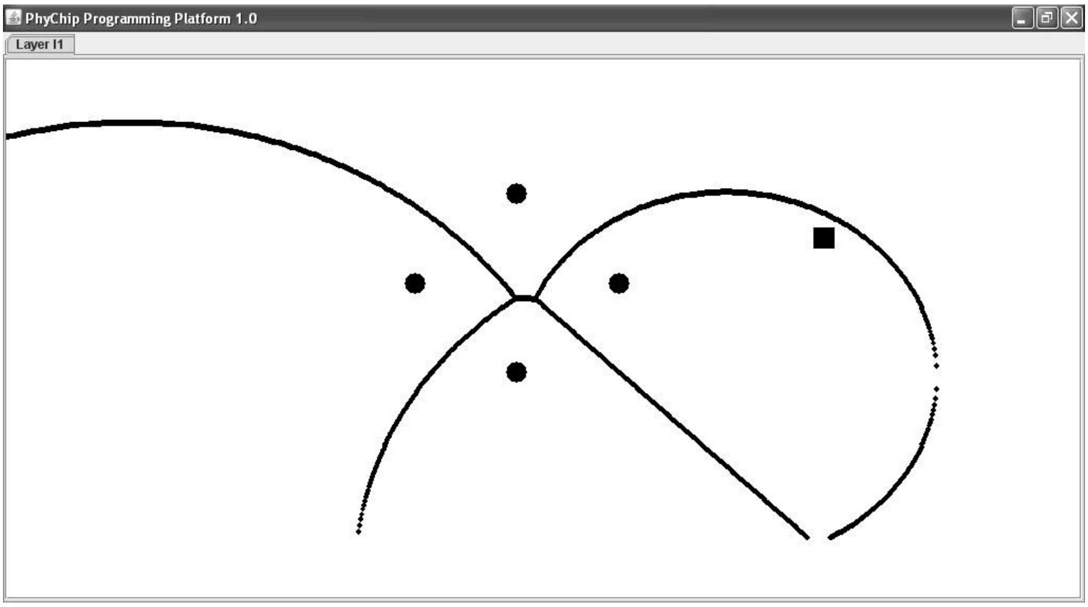
```
l1=new Layer;
```

```
p1=new Physarum;
a1=new Attractant;
a2=new Attractant;
a3=new Attractant;
a4=new Attractant;
l1.add(p1); p1.setPosition(800,200);
l1.add(a1);
a1.setPosition(500,150);
a1.setIntensity(0.7);
l1.add(a2);
a2.setPosition(500,350);
a2.setIntensity(0.5);
l1.add(a3);
a3.setPosition(400,250);
a3.setIntensity(0.6);
l1.add(a4);
a4.setPosition(600,250);
a4.setIntensity(0.5);
```

For experiments with *Physarum polycephalum* computing, a specialized computer tool (*PhyChip Programming Platform*) is being developed using the Java environment. The tool consists of two main modules:

- *Code creation and compilation module*. For generating the compiler of our language, the Java Compiler Compiler (JavaCC) tool [10] is used. JavaCC is the most popular parser generator for use with Java applications.
- *Simulation module*. It enables the user to perform time simulation of growing pseudopodia, i.e., to run the program.

Figure 3. The Voronoi cells for 4 attractants defined in the exemplary program generated in our tool (attractants are marked with dots whereas *Physarum* with a square).



In figure 3, we have shown the Voronoi cells generated in our computer tool for 4 attractants ($a1$, $a2$, $a3$, $a4$) with different intensity assigned to them, defined in the exemplary program. Attractants are marked with dots whereas *Physarum* with a square. The measure defined earlier has been used to create cells. It is easy to see that *Physarum* is attracted first of all by the most right attractant.

Summation

In the paper, we have outlined theoretical foundations as well as assumptions for a new object-oriented programming language for *Physarum polycephalum* computing. The next mile steps in our research are the following: implementation of operations based on the π -calculus model [17] of processes and extension of the programming platform to the agent-oriented programming language for computation with raw plasmodium.

Acknowledgments

This research is being fulfilled by the support of FP7-ICT-2011-8.

References

[1] Self, http://selflanguage.org/
[2] ECMAScript, http://www.ecmascript.org/
[3] JavaCC, http://java.net/projects/javacc/
[4] JavaBeans, Tech. rep., Sun Microsystems, 1997.
[5] Adamatzky, A., *Reaction-diffusion algorithm for constructing discrete generalized voronoi diagram*, Neural Network World 6, pp. 635–643, 1994.
[6] Adamatzky, A., *Computing in Nonlinear Media and Automata Collectives,* Institute of Physics Publishing, 2001.
[7] Adamatzky, A., *Physarum machine: implementation of a kolmogorov-uspensky machine on a biological substrate*, Parallel Processing Letters 17(4), pp. 455–467, 2007.
[8] Adamatzky, A., *Growing spanning trees in plasmodium machines,* Kybernetes 37(2), pp. 258–264, 2008.
[9] Adamatzky, A., De Lacy Costello, B.T., *Universal computation with limited resources: Belousov-zhabotinsky and physarum computers,* International Journal of Bifurcation and Chaos 18(8), pp. 2373–2389, 2008.
[10] Adamatzky, A., *Physarum Machines: Computers from Slime Mould,* World Scientific, 2010.
[11] Adamatzky, A., Erokhin, V., Grube, M., Schubert, T., Schumann, A., *Physarum chip project: Growing computers from slime mould,* International Journal of Unconventional

Computing 8(4), pp. 319–323, 2012.

[12] Adamatzky, A., Costello, B.D.L., Asai, T., *Reaction-Diffusion Computers*, Elsevier Science, Amsterdam, 2005.

[13] Chopard, B., Droz, M., *Cellular Automata Modeling of Physical Systems*, Cambridge University Press, 2005.

[14] De Lacy Costello, B., Ratcliffe, N., Adamatzky, A., Zanin, A.L., Liehr, A.W., Purwins, H.G., *The formation of voronoi diagrams in chemical and physical systems: Experimental findings and theoretical models*, International Journal of Bifurcation and Chaos 14(7), pp. 2187–2210, 2004.

[15] Ilachinski, A., *Cellular Automata: A Discrete Universe*, World Scientific, 2001.

[16] Kolmogorov, A., *On the concept of algorithm*, Uspekhi Mat. Nauk 8(4), pp. 175–176, 1953.

[17] Milner, R., Parrow, J., Walker, D., *A calculus of mobile processes*, Information and Computation 100(1), pp. 1–40, 1992.

[18] Pavlović, D., Escardó, M., *Calculus in coinductive form*, [in:] *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science*, pp. 408–417, 1998.

[19] Schönhage, A., *Real-time simulation of multi-dimensional turing machines by storage modification machines*, Project MAC Technical Memorandum 37, MIT, 1973.

[20] Schönhage, A., *Storage modification machines*, SIAM Journal on Computing 9(3), pp. 490–508, 1980.

[21] Tarjan, R., *Reference machines require non-linear time to maintain disjoint sets*, Tech. Rep. STAN-CS-77-603, 1977.

[22] Uspensky, V., *Kolmogorov and mathematical logic*, The Journal of Symbolic Logic 57, pp. 385–412, 1992.

Originally published: Schumann, A., Akimova, L., *Simulating of Schistosomatidae (Trematoda: Digenea) Behaviour by Physarum Spatial Logic,* Annals of Computer Science and Information Systems, vol. 1, Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, IEEE Xplore, pp. 225–231, 2013.

IX
Simulating of *Schistosomatidae* (Trematoda: Digenea) Behavior by *Physarum* Spatial Logic

Andrew Schumann, Ludmila Akimova

Abstract

In this paper, we consider possibilities of simulating behavior of the group of miracidiae and cercariae by the slime mould based computer that is programmed by attractants and repellents. For describing this simulation, we appeal to the language which is a kind of π-calculus called *Physarum* spatial logic. This language contains labels for attractants and repellents. Taking into account the fact that the behavior of miracidiae and cercariae can be programmed only by attractants (repellents for miracidiae and cercariae are not known still), we can claim that the behavior of miracidiae and cercariae is a restricted (poorer) form of *Physarum* spatial logic.

Introduction

In *Physarum Chip Project: Growing Computers From Slime Mould* [3] supported by FP7 we are going to implement programmable amorphous biological computers in plasmodium of *Physarum*. This abstract computer we are going to obtain is called *slime mould based computer*. The plasmodium behaves and moves as a giant amoeba and its behavior can be considered as a biological implementation of Kolmogorov-Uspensky machines [2]. This allows us to use the plasmodium of *Physarum* for solving different tasks that can be solved in Kolmogorov-Uspensky machines as well. The slime mould based computer is programmed using attractants and repellents (Fig. 1). On the one hand, it was experimentally proved that the slime mould prefers substances with potentially high nutritional value, e.g. it is attracted by peptones, aminoacids phenylalanine, leucine, serine, asparagine, glycine, alanine, aspartate, glutamate, and threonine. On the other hand, repellents for *Physarum polycephalum* can be presented by some illumination-, thermo- and salt-based conditions. Usually the plasmodium forms a congregation of protoplasm in food sources to surround them, secret enzymes and digest the food. Slime mould based computer can be regarded as a parallel computing substrate complementary to existing massive-parallel reaction-diffusion chemical processors [1].

In papers [15; 16] we showed that the behavior of plasmodium of *Physarum polycephalum* has an own spatial logic which is one of the natural implementations of π-calculus. This logic called *Physarum spatial logic* can be used as a programming language for the slime mould based computer. Taking into account the fact that within π-calculus we can formalize and describe different concurrent processes, within *Physarum* spatial logic we can do the same as well.

Figure 1: An 'ideal' example of plasmodium attracted by source of nutrients. Initially an oat flake colonized by plasmodium is placed in the southern part of Petri dish, and a group of intact oat flakes in the northern part. Plasmodium propagates towards the intact flakes and occupies them. (a) Snapshot the experimental Petri dish with the plasmodium. (b) Scheme of the plasmodium attraction: initial position of the plasmodium is shown by circle, newly occupied oat flakes (attractants) by solid discs, trajectory of the plasmodium by curve. *Courtesy of Andy Adamatzky.*

(a) (b)

In this paper we will show that the behavior of local group of the genus *Trichobilharzia* Skrjabin & Zakharov, 1920 (*Schistosomatidae* Stiles, Hassall 1898) can be simulated by *Physarum* logic. This means that, first, a local group of *schistosomatidae* can behave as a programmable biological computer, second, a biologized kind of π-calculus such as *Physarum* spatial logic can describe concurrent biological processes at all.

9.1. *Physarum* spatial logic

In this section we will consider some basics of *Physarum* spatial logic.

The behavior of *Physarum* plasmodium can be divided into the following elementary processes: inaction, fusion, cooperation, and choice, which could be interpreted as unconventional (spatial) falsity, conjunction, weak and strong disjunction respectively, denoted by *Nil*, &, |, and +. These operations differ from conventional ones, because they cannot have a denotational semantics in the standard way. However, they may be described as special spatial transitions over states of *Physarum* machine: inaction (*Nil*) means that pseudopodia has just stopped to behave; fusion (&) means that two pseudopodia come in contact one with another and then merge; cooperation (|) means that two pseudopodia behave concurrently; choice (+) means a competition between two pseudopodia in their behaviors.

A π-calculus for describing the dynamics of *Physarum* machine is presented as a labeled transition system with some logical relations.

Assume that there are N active species or growing pseudopodia and the state of species i is denoted by $p_i \in \mathrm{L}$. These states are time dependent and they are changed by plasmodium's active zones interacting with each other and affected by attractants or repellents. Plasmodium's active zones interact concurrently and in a parallel manner. Foraging plasmodium can be represented as a set of the following abstract entities.

- The set of actions (growing pseudopodia), $\mathrm{T}' = \{\alpha, \beta, ...\}$, localized in *active zones*. The actions from T' are called the *simplest transitions*, the latter are defined as $\{p \xrightarrow{\alpha} q : p, q \in \mathrm{L}, \alpha \in \mathrm{T}'\}$. Notice that we also have transitions that do not belong to T'. Assume that the set of all transitions is denoted by T.

On a nutrient-rich substrate plasmodium propagates as a typical circular, target wave, while on the nutrient-poor substrates localized wave-fragments are formed. Each action $\alpha \in \mathrm{T}'$, starts on a state p_i, which is its current position, and says about a transition (propagation) of a state p_i to another state of the same or another computation cell. Part of plasmodium feeding on a source of nutrients may not propagate, so its transition may be *Nil*, but this part can always start moving.

- The set of *attractants* $\{A_1, A_2, ...\} \subset \mathrm{T} \setminus \mathrm{T}'$ are sources of nutrients, on which the plasmodium feeds. It is still subject of discussion how exactly plasmodium feels presence of attractants. Each attractant A is a function from T' to T'.
- The set of *repellents* $\{R_1, R_2, ...\} \subset \mathrm{T} \setminus \mathrm{T}'$. Plasmodium of *Physarum* avoids light and some thermo- and salt-based conditions. Thus, domains of high illumination are repellents such that each repellent R is characterized by its position and intensity of illumination, or force of repelling. In other words, each repellent R is a function from T' to T'.
- The set of *protoplasmic tubes* $\{C_1, C_2, ...\} \subset \mathrm{T} \setminus \mathrm{T}'$. Typically plasmodium spans sources of nutrients with protoplasmic tubes/veins. The plasmodium builds a planar graph, where

nodes are sources of nutrients, e.g. oat flakes, and edges are protoplasmic tubes. $C(\alpha)$ means a diffusion of growing pseudopodia $\alpha \in$ T'.

Hence, T = T' $\cup$ $\{A_1, A_2, ...\} \cup \{R_1, R_2, ...\} \cup \{C_1, C_2, ...\}$.

Our process calculus contains the following basic operators: *Nil* (inaction), · (prefix), | (cooperation), \ (hiding), & (reaction/fusion), + (choice), *a* (constant or restriction to a stable state), $A(\cdot)$ (attraction), $R(\cdot)$ (repelling), $C(\cdot)$ (spreading/diffusion). Let T = $\{a, b, ...\}$, the set of all actions (evidently, this set is finite), be considered as a set of names. A name refers to a communication link or channel. With every $a \in$ T we associate a complementary action $\bar{a}$. Let us suppose that a designates an input port and $\bar{a}$ designates an output port. Any behavior of *Physarum* will be considered as outputs and any form of outside control and stimuli as appropriate inputs. For instance, T' is to be regarded just as the set of output ports and thereby T \ T' contains ports that can be interpreted under different conditions as input ports or output ports. So, for each $X \in \{A_1, A_2, ...\} \cup \{R_1, R_2, ...\}$, we take that $X(\gamma) = \delta$ is a function from T' to T' such that X is a stimulus for γ and X makes an output $\delta \in$ T' along γ. Evidently that $X(Nil) = Nil$. Let $X(\gamma)$ denote an input and $\overline{X(\gamma)}$ an output.

Define **L** be the set of labels built on T (under this interpretation, $a = \bar{\bar{a}}$). Suppose that an action a communicates with its complement $\bar{a}$ to produce the internal action τ and τ belongs to **L**, too.

We use now the symbols γ, δ, … , etc., to range over labels (actions), with $a = \bar{\bar{a}}$, and the symbols P, Q, etc., to range over processes on states p_i. The processes are given by the syntax:

$$P, Q ::= Nil \mid \gamma.P \mid A(\gamma).P \mid \overline{A(\gamma)}.P \mid R(\gamma).P \mid \overline{R(\gamma)}.P \mid C(\gamma).P \mid (P \mid Q) \mid P \setminus Q \mid P \,\&\, Q \mid P + Q \mid a$$

Each label is a process, but not vice versa, because a process may consists of many labels combined by the basic operators.

An operational semantics for this syntax is defined as follows: $\gamma ::= p_i$, where $p_i \in$ L.

Prefix: $\dfrac{}{\gamma.P \xrightarrow{\gamma} P}$,

$$\frac{}{A(\gamma).P \xrightarrow{\delta} P}\ (A(\gamma) = \delta), \quad \frac{}{\overline{A(\gamma)}.P \xrightarrow{\delta} P}\ (\overline{A(\gamma)} = \delta),$$

$$\frac{}{R(\gamma).P \xrightarrow{\delta} P}\ (R(\gamma) = \delta), \quad \frac{}{\overline{R(\gamma)}.P \xrightarrow{\delta} P}\ (\overline{R(\gamma)} = \delta),$$

(the conclusion states that the process of the form $\gamma.P$ (resp. $A(\gamma).P$, $\overline{A(\gamma)}.P$, $R(\gamma).P$, $\overline{R(\gamma)}.P$) may engage in γ (resp. $A(\gamma)$, $\overline{A(\gamma)}$, $R(\gamma)$, $\overline{R(\gamma)}$) and thereafter they behave like P; in the presentations of behaviors as trees, $\gamma.P$ (resp. $A(\gamma).P$, $\overline{A(\gamma)}.P$, $R(\gamma).P$, $\overline{R(\gamma)}.P$) is understood as an edge with two nodes: γ (resp. $A(\gamma)$, $\overline{A(\gamma)}$, $R(\gamma)$, $\overline{R(\gamma)}$) and the first action of P),

$$\text{Diffusion}: \ \frac{P \xrightarrow{\gamma} P'}{P \xrightarrow{\gamma} C(\gamma)} \ \ (C(\gamma) = P'),$$

$$\text{Constant}: \ \frac{P \xrightarrow{\gamma} P'}{a \xrightarrow{\gamma} P'} \ \ (a = P, a \in L),$$

$$\text{Choice}: \ \frac{P \xrightarrow{\gamma} P'}{P + Q \xrightarrow{\gamma} P'}, \ \frac{Q \xrightarrow{\gamma} Q'}{P + Q \xrightarrow{\gamma} Q'},$$

(these both rules state that a system of the form $P + Q$ saves the transitions of its subsystems P and Q),

$$\text{Cooperation}: \ \frac{P \xrightarrow{\gamma} P'}{P \,|\, Q \xrightarrow{\gamma} P' \,|\, Q}, \ \frac{Q \xrightarrow{\gamma} Q'}{P \,|\, Q \xrightarrow{\gamma} P \,|\, Q'},$$

(according to these rules, the cooperation | interleaves the transitions of its subsystems),

$$\frac{P \xrightarrow{\gamma} P' \qquad Q \xrightarrow{\bar{\gamma}} Q'}{P \,|\, Q \xrightarrow{\tau} P' \,|\, Q'},$$

(i.e. subsystems may synchronize in the internal action τ on complementary actions γ and $\bar{\gamma}$),

$$\text{Hiding}: \ \frac{P \xrightarrow{\gamma} P'}{P \setminus Q \xrightarrow{\gamma} P' \setminus Q} \ \ (\gamma \notin Q, Q \subseteq \mathbf{L}),$$

(this rule allows actions not mentioned in Q to be performed by $P \setminus Q$),

$$\text{Fusion}: \ \frac{}{\gamma.P \,\&\, \breve{P} \xrightarrow{\gamma} Nil}$$

(the fusion of dual processes are to be performed into the inaction, e.g. a fusion of an attractant/repellent P and appropriate repellent/attractant $\breve{P}$),

$$\frac{P \xrightarrow{\gamma} P' \qquad Q \xrightarrow{\gamma} P'}{P \,\&\, Q \xrightarrow{\gamma} P'}, \qquad \frac{P \xrightarrow{\gamma} P' \qquad Q \xrightarrow{\gamma} P'}{Q \,\&\, P \xrightarrow{\gamma} P'}$$

(this means that if we obtain the same result P' that is produced by the same action γ and evaluates from two different processes P and Q, then P' may be obtained by that action γ started from the fusion P & Q or Q & P),

$$\frac{P \stackrel{\gamma}{\rightarrow} P'}{P \,\&\, Q \stackrel{\gamma}{\rightarrow} Nil + C(\gamma) + P'}, \qquad \frac{P \stackrel{\gamma}{\rightarrow} P'}{Q \,\&\, P \stackrel{\gamma}{\rightarrow} Nil + C(\gamma) + P'}$$

(these rules state that if the result P' is produced by the action γ from the processes P, then a fusion P & Q (or Q & P) is transformed by that same γ either into the inaction or diffusion or process P').

These are inference rules for basic operations. The ternary relation $P \stackrel{\gamma}{\rightarrow} P'$ means that the initial action P is capable of engaging in action γ and then behaving like P'.

In this process calculus we have two kinds of logical connectives.

- The group of connectives defined by coinduction. They are derivable from the hiding. Indeed, let U be a set of growing pseudopodia, then the following equalities hold:

$$\neg P ::= U \setminus P$$

$$P \wedge Q ::= P \setminus (U \setminus Q)$$

$$P \vee Q ::= U \setminus ((U \setminus P) \setminus Q)$$

$$P \supset Q ::= U \setminus (P \setminus Q)$$

These connectives satisfy all properties of Boolean algebra.

- The group of connectives defined as transitions. It consists of three operations: inaction, fusion and choice. Their basic properties:

$$Nil \setminus P \cong Nil,$$

$$P \,\&\, \breve{P} \cong Nil,$$

$$P \,\&\, P \cong P,$$

$$P \,\&\, Nil \cong Nil,$$

$$(P + Q) \setminus P' \cong P \setminus P' + Q \setminus P',$$

$$(P \,\&\, Q) \setminus P' \cong P \setminus P' \,\&\, Q \setminus P',$$
$$P \,\&\, Q \cong Q \,\&\, P,$$

$$P \,\&\, (Q \,\&\, R) \cong (P \,\&\, Q) \,\&\, R,$$

$$P + P \cong P,$$

$$P + Nil \cong P,$$

$$P + Q \cong Q + P,$$

$$P + (Q + R) \cong (P + Q) + R,$$

$$P \,\&\, (Q + R) \cong (P \,\&\, Q) + (P \,\&\, R),$$

$$P + (Q \,\&\, R) \cong (P + Q) \,\&\, (P + R),$$

where $\cong$ is a congruence relation defined on the set of processes.

Now we can show that in the behavior of any local group of *schistosomatidae* we can observe the same elementary processes: inaction, fusion, cooperation, and choice, which are defined in the same way.

9.2. Life cycle of *Schistosomatidae* (Trematoda : Digenea)

All representatives of subclass *Digenea* Carus, 1863 (*Platyhelminthes: Trematoda*) are exclusively endoparasites of animals. The digenean life cycle has the form of heterogony, i.e. there is a natural alternation of amphimictic (usually synarmophytous) and parthenogenetic stages. At these stages digeneae have different outward, different means of reproduction and different adaptation to different hosts. The interchange of hosts is necessary for a successful realization of digenean flukes life cycle. The majority of representatives of this subclass have a complete life cycle with participation of three hosts: mediate, additional (metacercaria) and definitive. Molluscs are always the first mediate hosts, while different classes of vertebrate animals are definitive hosts.

Among digeneae there is a bunch of parasites, namely the family of *schistosomatidae*, which represents an isolated bunch which has adapted to parasitizing in the circulatory system of vertebrate animals. Puberal representatives of this family are diecious individuals (in other digenean families maritas are hermaphrodites). The family includes the following three sub-families: *Schistosomatinae* Stiles and Hassall, 1898, they parasitize a variety of birds and mammals, including human being; *Bilharziellinae* Price, 1929 and *Gigantobilharziinae* Mehra, 1940, they parasitize birds. Representatives of the first subfamily (in particular the genus *Schistosoma* Weinland, 1858) parasitize mammals, including human being. In the tropical and subtropical countries, about 200 million persons are infected by them from which 11 thousand persons annually die because of the given infestation [13]. Representatives of the latter two subfamilies, the so-called avian schistosomatidae, have been observed on all continents, including Europe. In a puberal state they parasitize birds, however they are capable to incorporate into a human organism as nonspecific host. After they penetrate human skin, where they perish, they invoke thereby allergodermia. The fact of incorporation of these larvae into nonspecific hosts invokes interest to avian schistosomatidae. Therefore the simulation of behavior of their local groups can be interesting from a medicine view, because it allows to perceive better features of digenean behavior. Simulating their behavior is possible by means of *Physarum* spatial logic as we will show.

The life cycle of all representatives of the family *Schistosomatidae* is identical, it passes with participation of two hosts.

From an egg which got to water from an organism of definitive host, a miracidia pips. It is a settle free-floating larval stage of parthenogenetic generation of schistosomatidae. Miracidia for a short span should find a mollusk of a certain kind to insinuate into it, otherwise it perishes. Mollusks, thus, are attractants for miracidiae. More precisely, in respect to miracidiae the chemotaxis as attractant holds (miracidia moves towards a chemical signal proceeding from a mollusk). Other kinds of attractants for miracidiae are presented by light (there is a positive phototaxis) and gravitation (negative geotaxis). We will designate all miracidian attractants by $A^m{}_1, A^m{}_2, \ldots, A^m{}_n$.

Miracidian repellents have not been detected still, i.e. $\{R^m{}_1, R^m{}_2, \ldots, R^m{}_n\} = \varnothing$.

The continuation of digenean life cycle will take place just in case a miracidia detects a mollusk for which a certain kind of digeneas has a hostal specificity [10]. Otherwise the miracidia dies. Miracidia *Trichobilharzia szidati* Neuhaus, 1952 can look for a mediate host only for the period of 20 h at temperature 20 °C [14].

In a body of mollusk, a miracidia undergoes metamorphoses and it is transformed into a maternal sporocyst in which filial sporocysts educe. In the latter then cercariae start to be formed. This state can be called the miracidian diffusion. We will designate these diffusions by $C^m{}_1, C^m{}_2, \ldots, C^m{}_n$.

Now we can construct a version of *Physarum* spatial logic for simulating the behavior of local groups of miracidiae. The processes have the following syntax which is defined in the way of *Physarum* logic:

$$P, Q ::= Nil \mid \gamma.P \mid A^m(\gamma).P \mid \overline{A^m(\gamma)}.P \mid$$
$$C^m(\gamma).P \mid (P \mid \mathrm{Q}) \mid P \setminus Q \mid P \,\&\, Q \mid P + Q \mid a$$

For the simulation we need also to have two sets of actions T and T^m, where T contains actions of *Physarum* plasmodium, T^m includes actions of local group of miracidiae. These sets should have the same number of members (the same cardinality), namely we should have the same number of active zones (growing pseudopodia and active miracidiae), the same number of attractants, and the same number of diffusions (motions of protoplasmic tubes towards food and miracidian motions towards chemical signals of eventual hosts to transform into a maternal sporocyst). For instance, if we have five molluscs in one experimental dish with water and a suspension of miracidiae, then we can try to simulate the miracidian processes by *Physarum* spatial logic for stimuli of five nutrient sources with similar localizations as that for mollusks.

9.3. Morphology of the cercariae *Trichobilharzia* spp.

Cercariae are free-floating larvae of pubertal generation parasitizing vertebrate animals. They are capable to insinuate into skin of human being who is for them a casual host, invoking at him/her an allergic reaction, the so-called cercarial dermatitis. For the first time this term (‘cercarial dermatitis’) was used by V.V. Cort [4]. This author first correlated this disease with mollusks of certain kinds, and then with cercariae (free-floating larvae of pubertal maritas) excreted by them.

In the human body these parasites cannot reach a full growth, perishing soon after penetration [10]. Any identification of specific difference of cercariae of the genus *Trichobilharzia* on morphological characters is extremely complicated because of the identity of their constitution and close dimensional characteristics [17], as well as the similarity of their behavior. They belongs to the bunch of furcocercariae, their tail is doubled in the form of fork, the length of forks is approximately a half of length of tail caulis. Even in a small zoom its pigmentary eyes are well discernible. The cercarial body is translucent. During motion it is strongly reduced, receiving various forms (Fig. 2).

Figure 2. The outward of *Trichobilharzia szidati*. A detection place: the lake of Naroch (the Minsk region, the Mjadelsky destrict). *Lymnaea stagnalis* L is a mediate host.

T. szidati has the length about 1.0–1.3 mm and it is easily distinguished by human eye. Its body is oblong, a bit more thickened in the middle. The tail is without appreciable thickenings, its length always is more than the length of body (Fig. 3). The two pigmentary eyes are located dorsally (their distance from the frontal end is about 2/5 of the body length). The distance of abdominal sucker is about 2/3 of the body length from the frontal side.

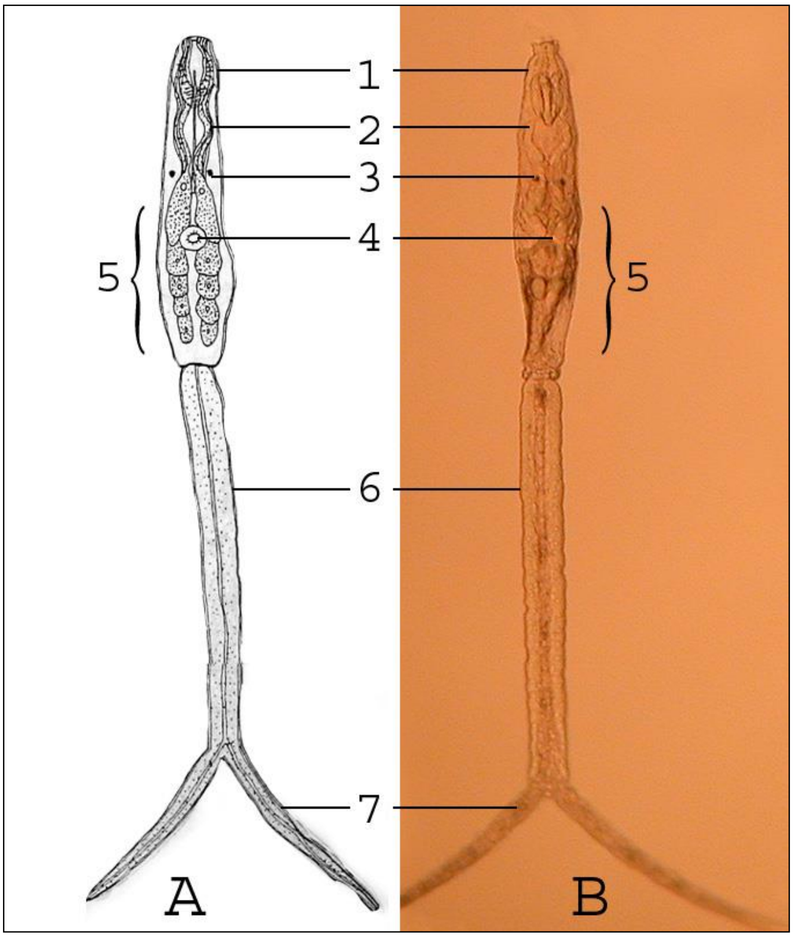
The frontal part of body is occupied by a penetration organ (frontal organ) which represents the transformed oral sucker. This pear-shaped formation penetrated by oral fissure and penetration gland ducts, consisting of a back sinewy part and a lobby part including the secreta suc.

All species of *Trichobilharzia* possess the five pairs of penetration glands. The two pairs are presented by circumacetabular glands located round the abdominal sucker, and the three other by postacetabular glands located sequentially one after another below. The secreta of penetration glands helps larvae to break a dermal barrier of backboned host.

The somatotype of excretory system is called "mesostoma" which is characterized by that the main collective canals do not come forward further than the level of abdominal sucker, divided into the frontal and back canals of the second order.

In a resting state (Fig. 4), cercariae are attached to a vascular wall or on a water film by means of acetabulum. Active motions are characteristic only by the strong shaking of pot or by the water interfusion. At a weak rotation of pot it is visible that the cercarial body and its tail follow the water stream, while their acetabulums keep cercariae on the pot wall. Any continuous active motion is not observed.

Figure 3. The constitution of *Trichobilharzia szidati*. A. The schematic cercarial image, B. The photo of cercaria. 1. Penetration organ, 2. Penetration gland ducts, 3. Pigmentary eyes, 4. Abdominal sucker, 5. Penetration glands (5 pairs), 6. Caudal fulcrum, 7. Furcas with mobile membranes.



9.4. The behavior of cercariae of bird schistosomes (Genus *Trichobilharzia*)

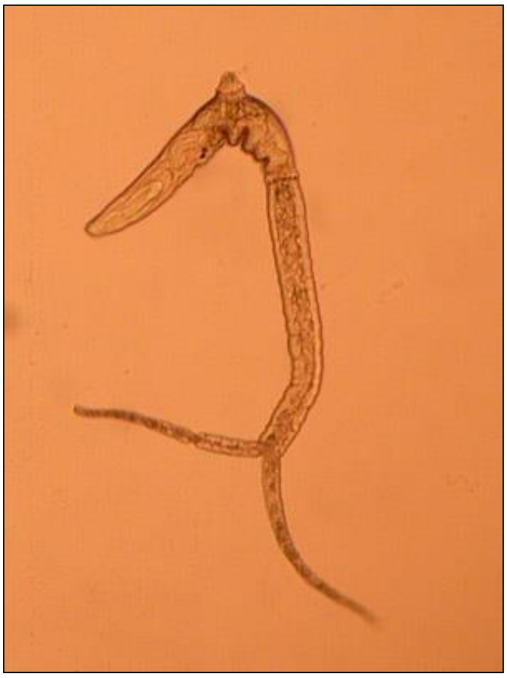
The cercarial behavior of bird schistosomes (class *Trematoda*, family *Schistosomatidae*) is well studied due to representatives of the genus *Trichobilharzia*. Their behavior is characterized by specific taxises which are referred to an effective search of necessary definitive hosts. These taxises developed by evolution of larvae of bird schistosomes allow their looking for specific hosts to be successful, forward their affixion to a surface of host body as well as their incorporation into a host cutaneous covering and their penetration into a circulatory system, where a parasite reaches sexual maturity. Thus, taxises form an enough large family of attractants for cercariae.

Mature cercariae of the genus *Trichobilharzia* after leaving a mollusk actively float in the depth of water for a period (about 1 hour). Such an active behavior of larvae after leaving a mollusk provides a cercarial allocation in water space. Then cercariae pass to a passive behavior. They are attached by an abdominal sucker to a surface film of water or to various subjects near a water surface, getting thus a characteristic resting state (Fig. 4). The resting state allows cercariae in absence of specific to them stimulants to stop the search of host and to conserve their energy.

Free-floating cercariae need to insinuate into a definitive host during the limited time interval (1–1.5 days at temperature 24°C) since otherwise larvae perish [14].

For successful search of hosts, larvae of digeneae of *Trichobilharzia* have developed by evolution a behavior facilitating this problem. They possess a positive phototaxis, negative geotaxis, chemotaxis, and also actively react to turbulence of water [7]. It means that for cercariae there are already many other attractants.

Figure 4. The resting state of cercaria *Trichobilharzia szidati* (zoom 100).



The light sensitivity of cercariae of *Trichobilharzia* is very high. As experiments show, cercariae always move towards a light source, and then take a resting state on the lighted side of capacity in which they are. The given taxis, and also negative geotaxis allow cercariae to be kept in the nature at a surface of water in expectation of suitable hosts.

Cercariae actively react to changes in intensity of illumination (shadings) and to turbulence of water [5]. These external factors, corresponding to possible appearances of definitive hosts in water, stimulate the cercarial transition from a resting state into actions that enlarge their chances to meet hosts.

Cercariae possess a chemotaxis in relation to specific hosts. On body surfaces larvae of the genus *Trichobilharzia* have chemoceptors which receive appropriate chemical signals proceeding from a skin of potential host. The similarity of compound of fatty acids of bird and human skin leads to that cercariae equally react to the bird and human appearance in water: they move in their direction, and then they are attached to skin and begin penetration into it [9]. So, the chemotaxis from a skin of potential hosts (surface lipids of skin of human being and swimming bird), the positive phototaxis, the negative geotaxis and the water turbulence present cercarial attractants of different degree of appeal. We will designate these attractants by $A^c{}_1$, $A^c{}_2$, …, $A^c{}_n$.

In experimental researches it has been shown that any affixion of cercariae of *Trichobilharzia* to skin is stimulated by cholesterol and ceramides, and incorporation into skin by linoleic and linolenic acids, all these materials are present on skin of both bird and human being [12; 8]. Thereby surface lipids of human skin invoke higher frequency of cercarial incorporations into skin, than surface lipids of birds [9]. One more reason that cercariae of *Trichobilharzia* successfully insinuate into human skin is the fact of that the skin of duck foots has thicker keratinized surface which, possibly, is more difficult for overcoming, than that of human being [8].

On the basis of experiments the rate of penetration of larvae of schistosomes *Trichobilharzia szidati* into human skin [8] has been defined. The larva begins incorporation into human skin approximately in 8 seconds (range from 0 to 80 seconds) after first contacts. The process of full penetration into skin takes about 4 minutes (range from 83 till 13 minutes 37 seconds). The given numerals testify that it is enough if the person has even a short-term contact to water where there are cercariae of bird schistosomes to give them possibility to incorporate into skin.

In some cases, for example children, cercarial larvae can “chip” skin and be brought by venous blood to lungs, invoking there hemorrhages and inflammation. If cercariae are lucky to insinuate into blood and then to lungs, the disease can get harder by the pulmonary syndrome from small cough to symptoms of bronchial obstruction [18].

At the same time, repellents for cercariae have not been found yet. For example, Ludmila Akimova’s experience shows that cercarial motions towards a smaller concentration of material which invokes a destruction of larvae are not observed at all. The experience principle consists in that in a small cavity with the length of 10 cm, the width and depth of 0.5 cm there is water with a suspension of cercariae. Then a thin essential oil is added in one of the side of this small cavity. Cercariae, which are nearby, quickly perish, although other cercariae do not move aside where the reacting material is absent. Cercariae simply freely float and as soon as they appear in that part where there is the reacting material they perish. Thus, $\{R^c{}_1, R^c{}_2, \ldots, R^c{}_n\} = \varnothing$.

In definitive hosts cercariae reach diffusion states. We will designate these diffusions by $C^c{}_1, C^c{}_2, \ldots, C^c{}_n$.

The behavior of local groups of cercariae can be simulated within a version of *Physarum* spatial logic, where the processes have the following syntax defined in section II:

$$P, Q ::= Nil \mid \gamma.P \mid A^c(\gamma).P \mid \overline{A^c(\gamma)}.P \mid$$
$$C^c(\gamma).P \mid (P \mid \mathrm{Q}) \mid P \setminus Q \mid P \,\&\, Q \mid P + Q \mid a$$

The sets of actions T and T^c, where T consists of actions of *Physarum* plasmodium, T^c contains actions of local group of cercariae, should have the same number of members. For example, if we have three human beings in one lake with cercariae, then we can simulate the cercarial processes by *Physarum* spatial logic where three nutrient sources with similar localizations as that for human beings act as stimuli. Hence, the behavior of local groups of cercariae is another biological implementation of Kolmogorov-Uspensky machines. It can build planar graphs as well.

9.5. Arithmetic Operations in Physarum Spatial Logic and In Schistosomatidae Behavioral Logic

We know that within π-calculus we can convert expressions from λ-calculus. In particular, it means that we can consider arithmetic operations as processes. *Physarum* spatial logic as well as its modification in the form of behavioral logic for local groups of miracidiae (cercariae) is a biologized version of π-calculus. Therefore we can convert arithmetic operations into processes of either *Physarum* spatial logic or schistosomatidae behavioral logic.

Indeed, growing pseudopodia may represent a natural number n by the following parametric process:

$$\underline{n}(x,z) ::= \underbrace{\bar{x}.\bar{x}.\cdots.\bar{x}}_{n}.\bar{z}.Nil$$

The process $\underline{n}(x, z)$ proceeds n times on an output port called the successor channel $\bar{x} \in \{A_1, A_2, \ldots\} \cup \{R_1, R_2, \ldots\}$ (e.g. it is the same output of attractant) and once on the zero output port $\bar{z} \in \{A_1, A_2, \ldots\} \cup \{R_1, R_2, \ldots\}$ before becoming inactive *Nil*. Recall that it is a "Church-like" encoding of numerals used first in λ-calculus. Notice that in case of miracidiae or cercariae $\bar{x} \in \{A_1, A_2, \ldots\}$ and $\bar{z} \in \{A_1, A_2, \ldots\}$.

An addition process takes two natural numbers i and j represented using the channels $x[i]$, $z[i]$ and $x[j]$, $z[j]$ and returns their sum as a natural number represented using channels $x[i+j]$, $z[i+j]$:

$$Add(x[i], z[i], x[j], z[j], x[i+j], z[i+j]) ::=$$
$$(x[i].\ \overline{x}[i+j].\ Add(x[i], z[i], x[j], z[j], x[i+j], z[i+j])) +$$
$$z[i].Copy(x[j], z[j], x[i+j], z[i+j])).$$

A multiplication process takes two natural numbers i and j represented using the channels $x[i]$, $z[i]$ and $x[j]$, $z[j]$ and returns their multiplication as a natural number represented using channels $x[\underbrace{i+\ldots+i}_{j}]$, $z[\underbrace{i+\ldots+i}_{j}]$:

$$Mult(x[i], z[i], x[j], z[j], x[i*j], z[i*j]) ::= Add(x[i], z[i], x[j], z[j], x[i+...+i], z[i+...+i]).$$

The *Copy* process replicates the signal pattern on channels x and y on to channels u and v. It is defined as follows:

$$Copy(x, y, u, v) ::= (x.\ \overline{u}.\ Copy(x, y, u, v) + y.\ \overline{v}\ .\ Nil)$$

As we see, within *Physarum* spatial logic and its poorer version in the form of schistosomatidae behavioral logic we can consider some processes as arithmetic operations. Also, we can combine several arithmetic operations within one process. Let us regard the following expression:

$$(10 + 20) * (30 + 40)$$

An appropriate process is as follows:

$$Mult(Add(x[10], z[10], x[20], z[20], x[10+20], z[10+20]), z[10+20], Add(x[30], z[30], x[40], z[40], x[30+40], z[30+40]), z[30+40], Add(x[30], z[30], x[70], z[70], x[2100], z[2100])).$$

Conclusion

We show that many biologized versions of π-calculus are possible: *Physarum* spatial logic, schistosomatidae behavioral logic, etc. One of its basic versions, *Physarum* spatial logic, can be used for constructing slime mould based computer. This logic is richer than schistosomatidae behavioral logic and may be involved for simulations of the latter. The fact that we can formalize biological behaviors as kind of logic confirms that biological processes can be considered as forms of concurrent and parallel computations.

References

[1] Adamatzky, A., De Lacy Costello, B., Asai, T., *Reaction–Diffusion Computers,* Amsterdam, Elsevier, 2005.

[2] Adamatzky, A., *Physarum machine: implementation of a Kolmogorov-Uspensky machine on a biological substrate*, Parallel Processing Letters, vol. 17, no. 4, pp. 455–467, December, 2007.

[3] Adamatzky, A., Erokhin, V., Grube, M., Schubert, Th., Schumann, A., *Physarum Chip Project: Growing Computers From Slime Mould*, International Journal of Unconventional Computing, 8(4), pp. 319–323, 2012.

[4] Cort, W.W., *Schistosome dermatitis in the United States (Michigan),* Journal of the American Medical Association, vol. 90, pp. 1027–1029, 1928.

[5] Feiler, W., Haas, W., *Trichobilharzia ocellata*: *chemical stimuli of duck skin for cercarial attachment*, Parasitology, vol. 96, pp. 507–517, 1988.

[6] Haas, W., *Host finding mechanisms – a physiological effect,* [in:] Mehlhorn, Y. (ed.), *Biology, Structure, Function: Encyclopedic Reference of Parasitology,* edn 2, pp. 382–383, 1988.

[7] Haas, W., *Physiological analysis of cercarial behavior*, Journal of Parasitology, vol. 78, pp. 243–255, 1992.

[8] Haas, W., Haeberlein, S., *Penetration of cercariae into the living human skin: Schistosoma mansoni* vs. *Trichobilharzia szidati,* Parasitology Research, vol. 105, no. 4, pp. 1061–1066, 2009.

[9] Haas, W., Roemer, A., *Invasion of the vertebrate skin by cercariae of Trichobilharzia ocellata: penetration processes and stimulating host signals,* Parasitology Research, vol. 84, no. 10, pp. 787–795, 1998.

[10] Horák, P., Kolárová, L., *Bird schistosomes: do they die in mammalian skin?* Trends in Parasitology, vol. 17, no. 2, pp. 66–69, 2001.

[11] Horák, P., Kolárová, L., Adema, C.M., *Biology of the schistosome genus Trichobilharzia,* Advances in Parasitology, 52, pp. 155–233, 2002.

[12] Mikes, L., Zìdková, L., Kasný, M., Dvorák, J., Horák, P., *In vitro stimulation of penetration gland emptying by Trichobilharzia szidati and T. regenti (Schistosomatidae) cercariae. Quantitative collection and partial characterization of the products,* Parasitology Research, vol. 96, no. 4, pp. 230–241, 2005.

[13] Molyneux, D.H., *Control of human parasitic diseases: context and overview*, Advances in Parasitology, vol. 61, pp. 1–45, 2006.

[14] Neuhaus, W., *Biologie und Entwicklund von Trichobilharzia szidati n. sp. (Trematoda, Schistosomatidae), einem Erreger von Dermatitis beim Menschen,* Zeitschrift für Parasitenkunde, vol. 15, pp. 203–266, 1952.

[15] Schumann, A., Adamatzky, A., *Logical Modelling of Physarum Polycephalum,* Analele Universitatii de Vest, Timisoara, Seria Matematica – Informatica XLVIII, 3, pp. 175–190, 2010.

[16] Schumann, A., Adamatzky, A., *Physarum Spatial Logic*, New Math. and Nat. Computation, vol. 7, no. 3, pp. 483–498, 2011.

[17] Podhorsky, M., Huzova, Z., Mikes, L., Horak, P., *Cercarial dimensions and surface structures as a tool for species determination of Trichobilharzi*a spp. *Acta Parasitologica*, vol. 50, pp. 343–365, 2009.

[18] Беэр, С.А., Воронин, М.В., *Церкариозы в урбанизированных экосистемах,* Москва: Наука, 2007.

Originally published: Schumann, A., Pancerz, K., Jones, J., *Towards Logic Circuits Based on Physarum Polycephalum Machines: The Ladder Diagram Approach*, [in:] Cliquet, A., Plantier, Jr., G., Schultz, T., Fred, A., Gamboa, H. (eds.), *Proceedings of the International Conference on Biomedical Electronics and Devices (BIODEVICES'2014),* Angers, France, March 3-6, pp. 165-170, 2014.

X
Towards Logic Circuits Based on *Physarum Polycephalum* Machines: the Ladder Diagram Approach

Andrew Schumann, Krzysztof Pancerz, Jeff Jones

Abstract

In the paper, we present foundations of logic circuits based on *Physarum polycephalum* machines. We propose to apply the ladder diagram approach for constructing topological structures of such circuits. Relationships between basic ladder diagram elements and topological constructions present in *Physarum polycephalum* machines are emphasized. At the beginning, basic logic gates (AND, OR, NOT) are considered. Such a set of gates constitutes a functionally complete system. This fact is important for building computationally universal devices.

Introduction

Physarum polycephalum is a one-cell organism belonging to the species of order *Physarales*, subclass *Myxogastromycetidae*, class *Myxomycetes*, and division *Myxostelida*. In the phase of plasmodium, it looks like an amorphous giant amoeba with networks of protoplasmic tubes. It feeds on bacteria, spores and other microbial creatures (substances with a potentially high nutritional value) by propagating towards sources of food particles and occupying these sources. A network of protoplasmic tubes connects the masses of protoplasm. As a result, the plasmodium develops a planar graph, where the food sources or pheromones are considered as nodes and protoplasmic tubes as edges. The plasmodium may be used for developing a biological architecture of different abstract devices, among others, digital. Plasmodium's ability to perform useful computational tasks, in its propagating and foraging behavior, was firstly emphasized by T. Nakagaki et al. (cf. [9]). In *Physarum Chip Project: Growing Computers From Slime Mould* [2] supported by FP7, we are going to implement programmable amorphous biological computers in plasmodium of *Physarum*. This abstract computer we are going to obtain is called *slime mould based computer*. One of the tracks in the project is to develop a new object-oriented programming language for *Physarum polycephalum* computing [10].

The problem of constructing logic gates in chemical media or on biological substrates has been considered earlier in the literature. Different approaches have been proposed. One of them is to constrain the substrate into channels and allow disturbances to propagate along the channels and interact with other disturbances at the junctions between the channels. For example, this approach has been implemented in a geometrically constrained Belousov-Zhabotinsky medium, cf. [4], [6], [10], [11]. Also, non-excitable chemical implementation of logic gates has been proposed [2]. A wider discussion of *Physarum polycephalum* gates is included in [2].

Our approach, presented in this paper, is a little different. We propose to construct logic gates through the proper geometrical distribution of stimuli for *Physarum polycephalum*. This distribution is determined according to ladder diagrams [8] representing basic logic gates (AND, OR, NOT). Rungs of the ladder can consist of serial or parallel connected paths of *Physarum* propagation. A kind of connection depends on the arrangement of regions of influences of individual stimuli. If both stimuli influence *Physarum*, we obtain alternative paths for its propagation. It corresponds to a parallel connection (i.e., the OR gate). If the stimuli influence *Physarum* sequentially, at the beginning only the first one, then the second one, we obtain a serial connection (i.e., the AND gate). The NOT gate is imitated by the repellent avoiding *Physarum* propagation.

The rest of the paper is organized as follows. In section 10.1, we recall basics of *Physarum polycephalum* machines with a special focus on stimuli. Section 10.2 mentions a basic idea of ladder diagrams. This idea is used in section 10.3 for constructing logic gates based on *Physarum* propagation. Section 10.4 summarizes the presented approach and suggests directions for further work.

10.1. Basics of *Physarum polycephalum* machines

In *Physarum polycephalum* machines, we can distinguish the following stimuli constituting their data nodes:

- Attractants that are sources of nutrients or pheromones, on which the plasmodium feeds. Each attractant A is characterized by its position and intensity.
- Repellents. Plasmodium of *Physarum* avoids light and some thermo- and salt-based conditions. Thus, domains of high illumination (or high grade of salt) are repellents such that each repellent R is characterized by its position and intensity, or force of repelling.

Plasmodium of *Physarum polycephalum* functions as a parallel amorphous computer with parallel inputs and parallel outputs. Data are represented by spatial (topological) configurations of attractants and repellents. Plasmodium of *Physarum polycephalum* is a computing substrate. In [2], Adamatzky underlined that *Physarum* does not compute. It obeys physical, chemical and biological laws. Its behavior can be translated to the language of computations. At the beginning of computation, data nodes are distributed in a computational space, and plasmodium is placed at given points in the space. Plasmodium proceeds computation even if the solution has been reached and halts only when physical resources are exhausted. Typically, plasmodium spans attractants (sources of nutrients or pheromones) with protoplasmic tubes (veins). Plasmodium builds a planar graph, where nodes are attractants and edges are protoplasmic tubes.

It is a subject of discussion how plasmodium feels attractants. Experiments show that plasmodium can locate and colonize the nearby sources of nutrients or pheromones (attractants). In our approach, we assume that each attractant (repellent) is characterized by its region of influence (ROI) in the form of a circle surrounding the location point of the attractant (repellent), i.e., its center point. The intensity determining the force of attracting (repelling) decreases as the distance from it increases. A radius of the circle can be set assuming some threshold value of the force.

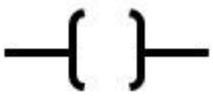
10.2. Basics of ladder diagrams

Ladder logic is the most popular programming language used to program Programmable Logic Controllers (PLCs), cf. [8]. This language was developed from the electromechanical relay system-wiring diagrams. Programs in the ladder logic language are written graphically in the form of the so-called ladder diagrams. Basically, this notation assumes that contacts are controlled by discrete (binary) inputs and coils control discrete (binary) outputs. We can distinguish three main types of elements of ladder diagrams:

- *Normally open contact* (NOC). It passes power (i.e., it is *on*) if the binary input assigned to it has value 1. Otherwise, it does not pass power (i.e., it is *off*).
- *Normally closed contact* (NCC). It passes power (*on*) if the binary input assigned to it has value 0. Otherwise, it does not pass power (*off*).
- *Coil* (C). If it is passing power (i.e., it is *on*), a value of the binary output assigned to it is set to 1. Otherwise (i.e., it is *off*), a value of the binary output assigned to it is set to 0.

The symbols of ladder diagram elements mentioned earlier are collected in table 1.

Table 1. Symbols of main types of elements in ladder diagrams.

Symbol	Meaning
	Normally open contact
	Normally closed contact
	Coil

Using three main types of elements of ladder diagrams we can build basic logic gates shown in figures 1 (AND), 2 (OR), and 3 (NOT).

Figure 1. The ladder diagram AND gate.

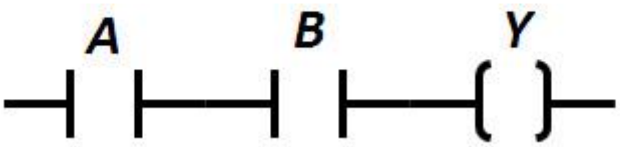


Figure 2. The ladder diagram OR gate.

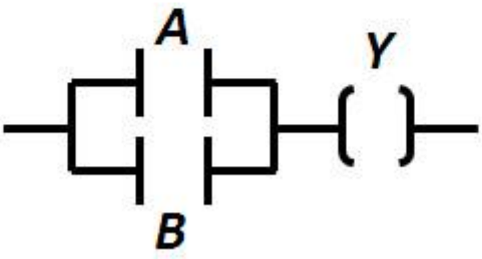
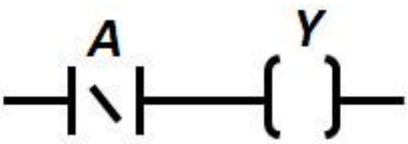


Figure 3. The ladder diagram NOT gate.



The gates mentioned work as follows:

- AND: The coil is *on* ($Y = 1$) if and only if both contacts are *on*. It is satisfied if $A = 1$ and $B = 1$. Otherwise, the coil is *off*.
- OR: The coil is *on* ($Y = 1$) if at least one contact is *on*. It is satisfied if $A = 1$ or $B = 1$. Otherwise, the coil is *off*.
- NOT: The coil is *on* ($Y = 1$) if a contact is *on*. $A = 0$ causes the contact to be switched on. The coil is *off* ($Y = 0$) if a contact is *off*. $A = 1$ causes the contact to be switched off.

Using structers of basic logic gates we can build, in ladder diagrams, more complex digital systems. Now, it is out of scope of this paper. We will consider this problem in the future.

10.3. Logic circuits based on *Physarum polycephalum* propagation

Ladder diagrams implement an idea of flowing power from left to right. The output for the rung in the ladder diagram occurs on the extreme right side of the rung and power is assumed to flow from left to right if and only if there exists at least one closed path from left to right making the flow possible. We apply this idea to build logic gates in *Physarum polycephalum* machines. Flowing power is replaced with propagation of plasmodium of *Physarum polycephalum*. Plasmodium propagation is stimulated by attractants and repellents (see Section 10.1). In our approach, stimuli (attractants and repellents) are treated as data nodes. We assume that plasmodium must occur in a proper region to be influenced by a given stimulus. This region is determined by the radius depending on the intensity of the stimulus. Using the analogy to flowing power in rungs of ladder diagrams, we can build logic gates in *Physarum polycephalum* machines by the proper geometrical distribution of stimuli (attractants and repellents) on the substrate. Controlling the power flow in rungs by opening/closing contacts is replaced with controlling the plasmodium propagation by activating/deactivating stimuli. Relationships between elements of ladder diagrams and stimuli of *Physarum polycephalum* computing are collected in table 2.

Table 2. Relationships between elements of ladder diagrams and stimuli of Physarum polycephalum computing.

Ladder diagram element	Physarum polycephalum computing stimulus
Normally open contact	Attractant controlled by input
Normally closed contact	Repellent controlled by input
Coil	Attractant controlling output

Table 3 shows interpretation of logic values (0 and 1) for inputs in terms of states of stimuli. Input values cause activation/deactivation of stimuli. Value 1 activates the stimuli whereas value 0 deactivates the stimuli. Analogously, table 4 shows interpretation of logic values (0 and 1) for outputs in terms of states of stimuli. In our approach, the output represented by the coil in ladder diagrams is replaced with the attractant. We assume that the output attractant is always activated. If plasmodium is attracted by it and occupies it, then we interpret this state as 1. Otherwise, if there is no plasmodium occupying the attractant, i.e., plasmodium is not attracted, the state is interpreted as 0.

Table 3. Representation of input logic values.

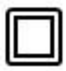
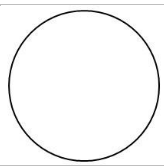
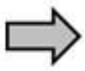
Boolean value	Representation
0	Attractant/repellent deactivated
1	Attractant/repellent activated

Table 4. Representation of output logic values.

Boolean value	Representation
0	Absence of Physarum at the attractant
1	Presence of Physarum at the attractant

Our idea of paths of plasmodium propagation is further presented graphically. In table 5, we have collected symbols used by us in diagrams.

Table 5. Symbols of elements used in figures.

Symbol	Meaning
	Physarum
	Attractant deactivated
	Attractant activated
	Repellent deactivated
	Repellent activated
	Region of influence
	Direction of plasmodium propagation

As it was mentioned earlier, the idea of ladder diagrams has been applied in our logic gates constructed in *Physarum polycephalum* machines. Figure 4 shows distribution of stimuli for the AND gate. This distribution simulates a serial connection of contacts. Plasmodium of *Physarum polycephalum* P can be propagated to the output attractant A_y if and only if both attractants A_{x1} and A_{x2} are activated. First, plasmodium is attracted to A_{x1} (because it is placed only in its region of influence). After the achievement of this goal, it is in the region of influence of A_{x2} and it is attracted by it. The achievement of A_{x2} causes that plasmodium is in the region of influence of A_y and it is attracted by it. Finally, plasmodium achieves A_y. It is interpreted as a logic output with value 1. Deactivation of either the attractant A_{x1} or A_{x2} causes that the path of propagation becomes broken, i.e., there is a place where plasmodium is not attracted by any attractant. Figure 7 shows paths of plasmodium propagation for all possible combinations of input values for A_{x1} and A_{x2}.

Figure 4. The Physarum AND gate.

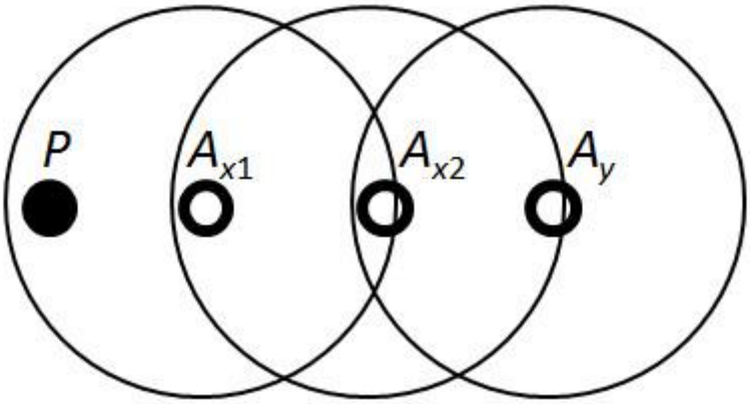
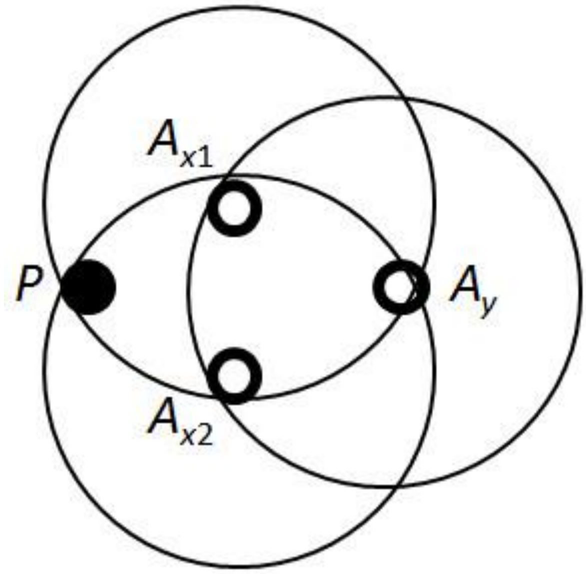


Figure 5 shows distribution of stimuli for the OR gate. This distribution simulates a parallel connection of contacts. Plasmodium of Physarum polycephalum P can be propagated to the output attractant A_y if one of the attractants A_{x1} or A_{x2} is activated. First, plasmodium is attracted to A_{x1} or A_{x2} (because it is placed in regions of influences). After the achievement of one or both of them, it is in the region of influence of A_y and it is attracted by it. Finally, plasmodium achieves A_y. It is interpreted as a logic output with value 1. Deactivation of both attractants A_{x1} and A_{x2} causes that the path of propagation becomes broken, i.e., plasmodium is not attracted by any attractant and cannot start from the initial position. Figure 8 shows paths of plasmodium propagation for all possible combinations of input values for A_{x1} and A_{x2}.

Figure 5. The Physarum OR gate.



The NOT gate behavior is simulated by the repellent as it is shown in figure 6. If the repellent R_x is activated (i.e., the input value is 1), then it avoids plasmodium to be attracted by the output attractant A_y. Therefore, Physarum is not present at A_y, i.e., the output value is 0. Otherwise, plasmodium is not avoided and achieves A_y. Figure 9 shows paths of plasmodium propagation for all possible combinations of input values for R_x.

Figure 6. The Physarum NOT gate.

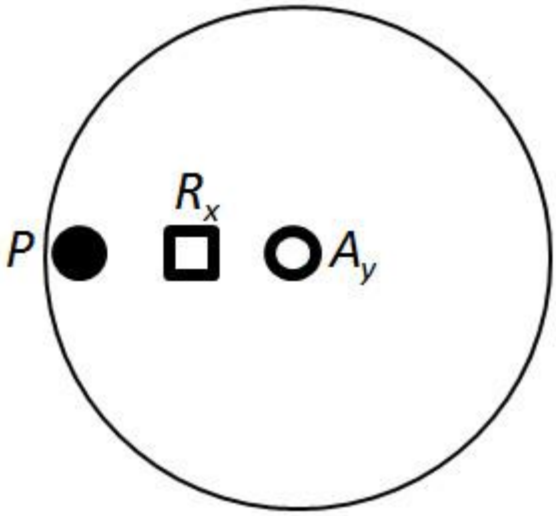


Figure 7. States of the AND gate for all input combinations.

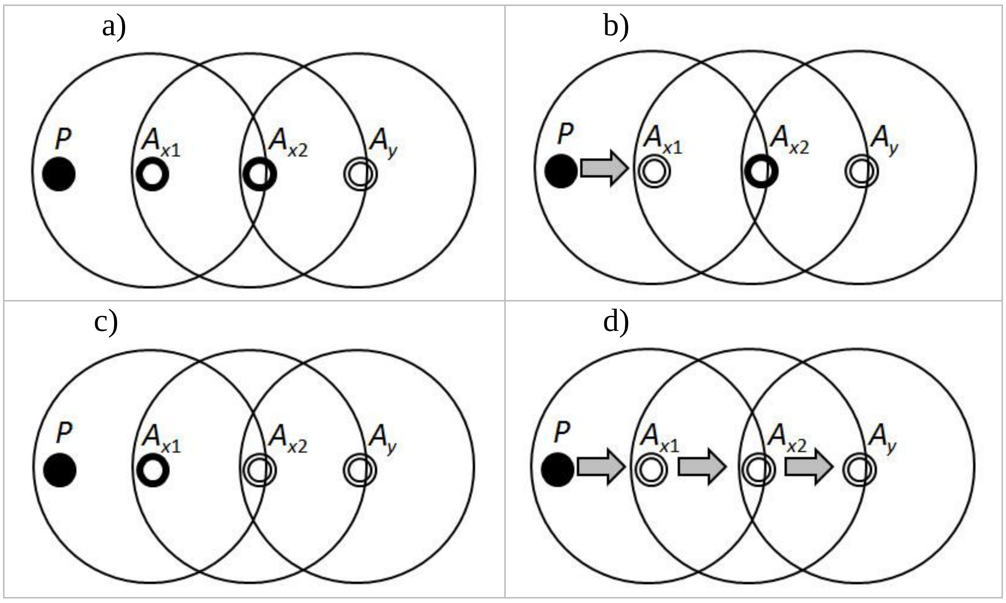


Figure 8. States of the OR gate for all input combinations.

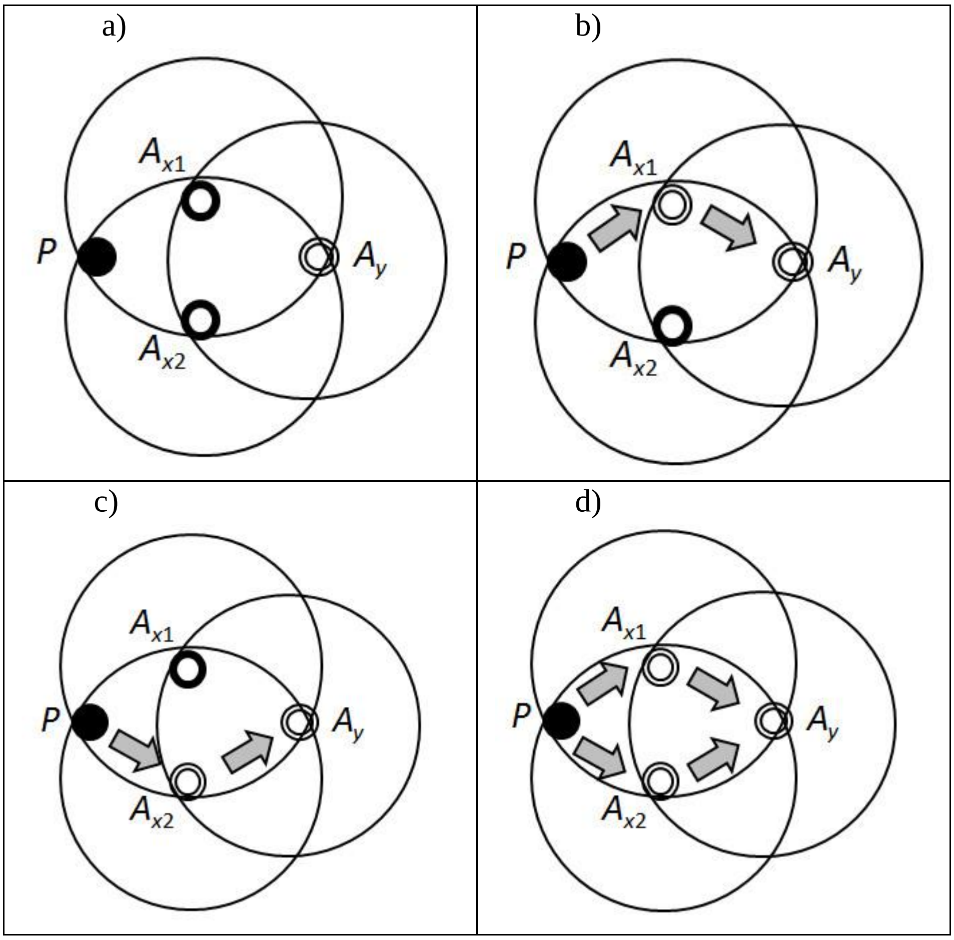
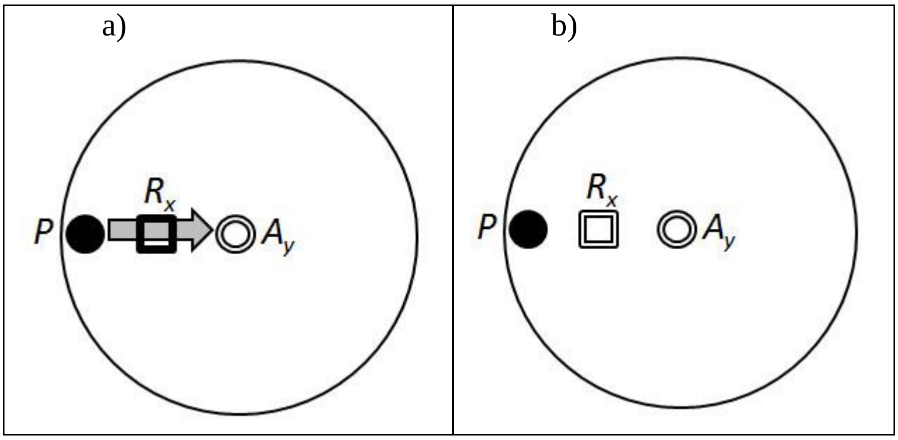


Figure 9. States of the NOT gate for all input combinations.



Experiment

In the experiment, we have built a *Physarum polycephalum* demultiplexer based on the ladder diagram structure. A demultiplexer is a device taking a single input signal and selecting one of many data-output-lines, which is connected to the single input. In figure 10, a schematic symbol of the 1-to-2 demultiplexer (a) and its implementation (b) using a particle model of *Physarum polycephalum* [5] are shown. In the schematic symbol: d is a data input, s is a select input, and y_0, y_1 are outputs. The operation of the demultiplexer can be described as follows:

- if $s = 0$, then $y_0 = d$,
- if $s = 1$, then $y_1 = d$.

The functional specification can be written as $y_0 = \bar{s}d$ and $y_1 = sd$. In the *Physarum polycephalum* implementation of the demultiplexer, one can see: *Physarum polycephalum* (*P*), attractants (A_d, A_s, A_{y0}, A_{y1}), repellent (R_s). Let *ROI* denote the region of influence. For the topological distribution of *Physarum polycephalum*, attractants and repellents, we assume that:

- P belongs to $ROI(A_d)$,
- A_d belongs to $ROI(A_{y0})$, $ROI(R_s)$, $ROI(A_s)$,
- A_s belongs to $ROI(A_{y1})$.

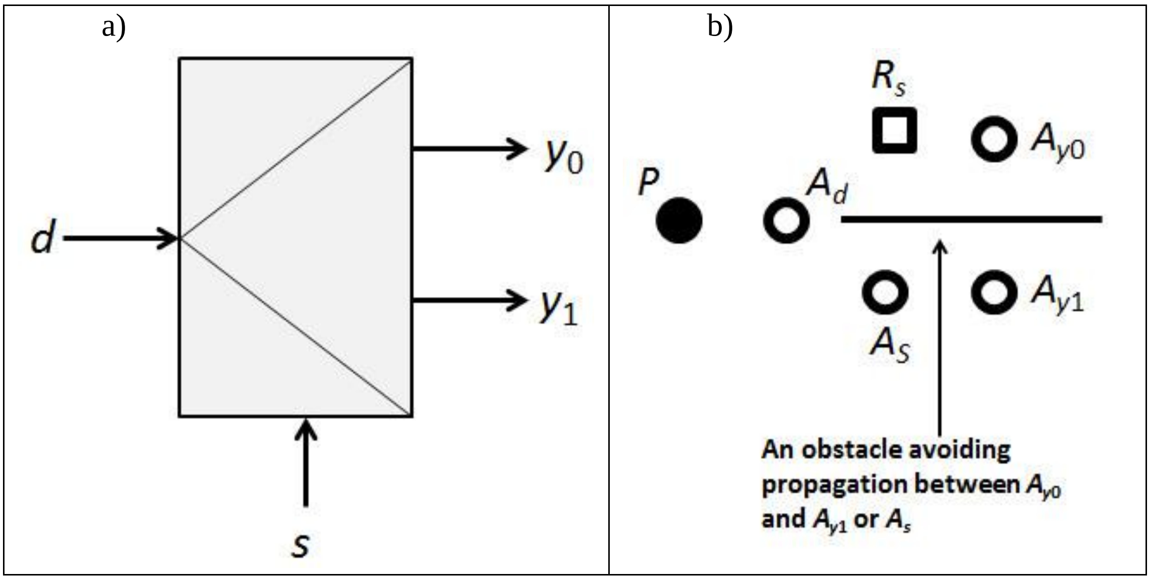
Logical states are implemented in the following way:

- $s = 0$ means R_s and A_s are deactivated, $s = 1$ means R_s and A_s are activated,

- $d = 0$ means A_d is deactivated, $d = 1$ means A_d is activated.

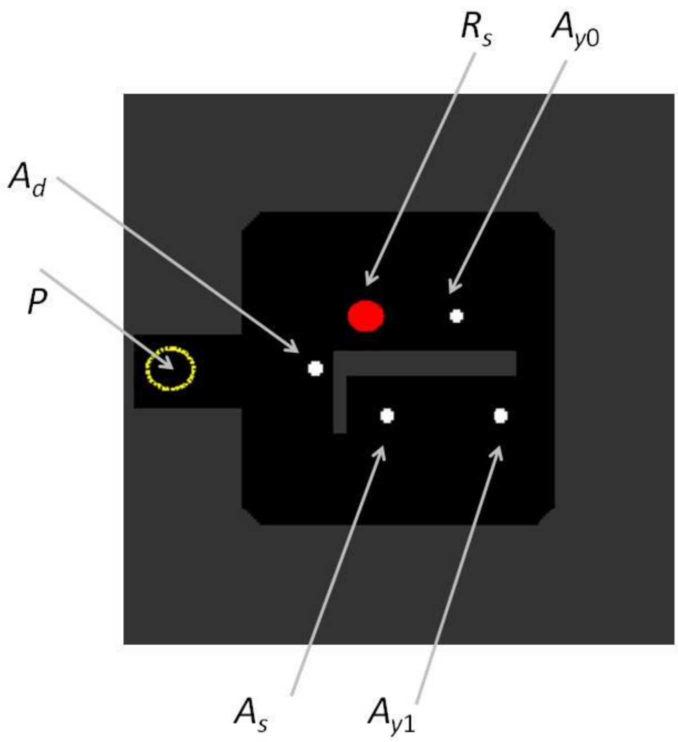
It is worth noting that A_{y0} and A_{y1} are always activated.

Figure 10. 1-to-2 demultiplexer: (a) a schematic symbol, (b) distribution of stimuli.



In figure 11, the experimental environment for a particle model of *Physarum polycephalum* is shown.

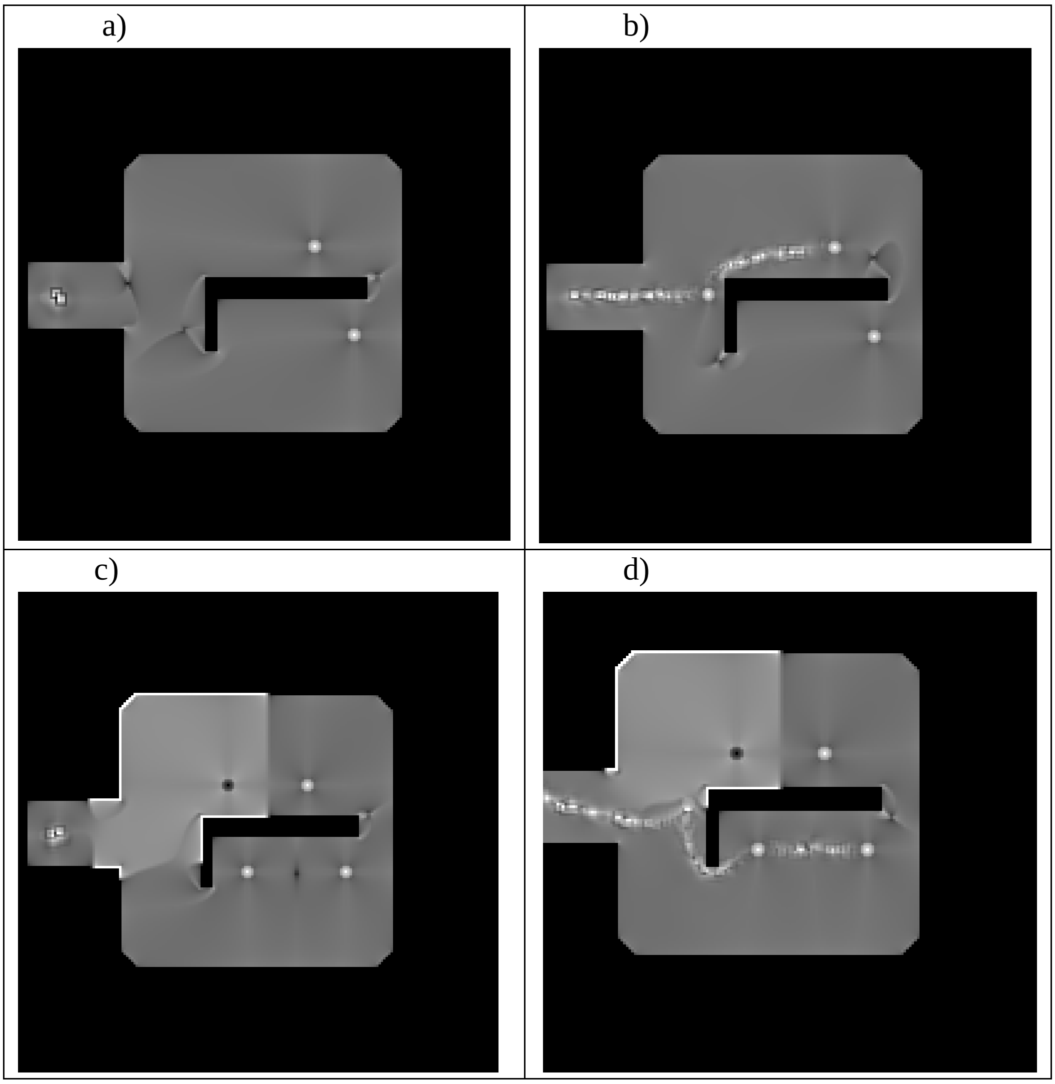
Figure 11. The experimental environment for a particle model of *Physarum polycephalum*.



In figure 12, results of experiments are presented. Pictures taken by us show how *Physarum polycephalum* was propagated in each situation.

Figure 12. Results of experiments: (a) for $s = 0$ and $d = 0$, (b) for $s = 0$ and $d = 1$, (c) for $s = 1$ and $d = 0$, (d) for $s = 1$ and $d = 1$. One can see the following cases:

- $s = 0$ and $d = 0$: uneventful, because there is no data regardless of switch position,
- $s = 0$ and $d = 1$: no repellent causes the stream to go to A_{y0}, the model does not grow down because it is outside the region of influence of A_{y1},
- $s = 1$ and $d = 0$: uneventful, because there is no data regardless of switch position,
- $s = 1$ and $d = 1$: the repellent causes selection of the lower path to A_{y1}.



It means that the *Physarum polycephalum* behaves as intended.

Summation

In the paper, we have shown how to construct basic logic gates in *Physarum polycephalum* machines using the idea of ladder diagrams. Proper relationships between ladder diagrams and *Physarum polycephalum* computing have been pointed out. The paper consists, in the first step, in research connected to developing a biological architecture of different abstract digital devices based on the ladder diagram principle. This principle is very popular in programming Programmable Logic Controllers (PLCs). However, in case of PLCs, the ladder diagram principle is used only at the abstract level as a high-level programming language. The program is executed by silicon microprocessors based on the standard architectures not reflected in the direct flow of power. Our approach could allow a direct hardware implementation of this principle in different controllers. In our case, it is a biological hardware implementation.

The approach presented in this paper may be used in different constructions of logic gates in chemical media or on biological substrates which are based on the flow or propagation of some medium. An important thing is to find the mechanism of controlling the flow or propagation in the restricted regions by some elements which can be activated or deactivated. The main problem for the further work is to search for mechanisms of constructing complex digital systems. For example, in ladder diagrams, negations of complex expressions must be realized using some internal variables enabling us to carry states of coils to states of contacts.

Another task for the further work is to implement the presented idea in the experimental environment for more complex circuits. In this case, an important thing is the proper control over states of stimuli, i.e., their rapid activation or deactivation. Moreover, the construction requires adjusting proper regions of influences of individual stimuli to model serial or parallel connections.

Acknowledgments

This research is being fulfilled by the support of FP7-ICT-2011-8.

References

[1] Adamatzky, A., *Physarum Machines: Computers from Slime Mould,* World Scientific, 2010.

[2] Adamatzky, A., De Lacy Costello, B., *Experimental logical gates in a reaction-diffusion medium: The xor gate and beyond,* Physical Review E, 66:046112, 2002.

[3] Adamatzky, A., Erokhin, V., Grube, M., Schubert, T., Schumann, A., *Physarum chip*

project: Growing computers from slime mould, International Journal of Unconventional Computing, 8(4):319–323, 2012.

[4] Górecki, J., Górecka, J., Igarashi, Y., *Information processing with structured excitable medium,* Natural Computing, 8(3):473–492, 2009.

[5] Jones, J., Adamatzky, A., *Towards physarum binary adders,* Biosystems, 101(1):51–58, 2010.

[6] Motoike, I.N., Yoshikawa, K., *Information operations with multiple pulses on an excitable field,* Chaos, Solitons & Fractals, 17(2–3):455 – 461, 2003.

[7] Nakagaki, T., Yamada, H., Toth, *A., Maze-solving by an amoeboid organism*, Nature, 407:470–470, 2000.

[8] Rosandich, R.G., *Fundamentals of Ladder Diagrams Programming*, EC&M Books, 1999.

[9] Schumann, A., Pancerz, K., *Towards an object-oriented programming language for physarum polycephalum computing*, [in:] Szczuka, M., Czaja, L., Kacprzak, M. (ed.), *Proceedings of the Workshop on Concurrency,* Specification and Programming (CS&P'2013), Warsaw, Poland, pp. 389–397, 2013.

[10] Sielewiesiuk, J., Górecki, J., *Logical functions of a cross junction of excitable chemical media*, The Journal of Physical Chemistry A, 105(35):8189–8195, 2001.

[11] Steinbock, O., Kettunen, P., Showalter, K., *Chemical wave logic gates*, The Journal of Physical Chemistry, 100(49):18970–18975, 1996.

Originally published: Schumann, A., Pancerz, K., *Timed Transition System Models for Programming Physarum Machines: Extended Abstract,* [in:] Popova-Zeugmann, L. (ed.), *Proceedings of the Workshop on Concurrency*, Specification and Programming (CS&P'2014), Chemnitz, Germany, September 29 - October 1, pp. 180-183, 2014.

XI
Timed Transition System Models for Programming *Physarum* Machines. Extended Abstract

Andrew Schumann, Krzysztof Pancerz

Abstract

In the paper, we show that timed transition system models can be used as a high-level model of behavior of *Physarum* machines. A *Physarum* machine is a programmable amorphous biological computer experimentally implemented in the vegetative state of *Physarum polycephalum*. Timed transition system models have been used in our new object-oriented programming language for *Physarum polycephalum* computing.

11.1. Timed transition system models

A *Physarum* machine is a programmable amorphous biological computer, experimentally implemented in the vegetative state of *Physarum polycephalum* (also called slime mould) [2] that is a one-cell organism belonging to the species of order *Physarales*, subclass *Myxogastromycetidae*, class *Myxomycetes*, and division *Myxostelida*. The plasmodium of *Physarum polycephalum* spread by networks can be programmable. The ability to perform useful computational tasks, in propagating and foraging behaviour of the plasmodium, was firstly emphasized by T. Nakagaki et al. (cf. [3]). In *Physarum Chip Project: Growing Computers from Slime Mould* [2] funded by the Seventh Framework Programme (FP7), we are going to construct an unconventional computer on programmable behavior of *Physarum polycephalum*. The *Physarum* machine comprises an amorphous yellowish mass with networks of protoplasmic veins, programmed by spatial configurations of attracting and repelling stimuli. The plasmodium looks for attractants, propagates protoplasmic veins towards them, feeds on them and goes on. As a result, a transition system is built up. Therefore, *Physarum* motions can be treated as a kind of natural transition systems with states presented by attractants and events presented by plasmodium transitions between attractants [2]. We can define the following three basic forms of *Physarum* transitions (motions): *direct* (direction: a movement from one place, where the plasmodium is located, towards another place, where there is a neighbouring attractant), $fuse$ (fusion of two plasmodia at the place, where they meet the same attractant), $split$ (splitting plasmodium from one active place into two active places, where two neighbouring attractants with a similar power of intensity are located).

Formally, a transition system is a quadruple $TS = (S, E, T, s_0)$ [14], where S is the non-empty set of states, E is the set of events, $T \subseteq S \times E \times S$ is the transition relation, s_0 is the initial state. Usually transition systems are based on actions which may be viewed as labelled events. If $(s, e, s') \in T$ then the idea is that TS can go from s to s' as a result of the event e occurring at s. A transition system can be presented as a graph structure with nodes corresponding to states and edges corresponding to transitions.

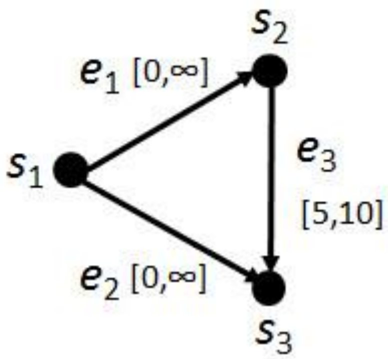
To program computation tasks for amorphous biological computers, we are developing a new object-oriented programming language (cf. [10]). The proposed language can be used for developing programs for *Physarum polycephalum* by the spatial configuration of stimuli. This configuration of stimuli can be identified with a low-level programming language for *Physarum* machines. To program behavior of *Physarum polycephalum*, we have also proposed to use some high-level models, e.g., ladder diagrams, Petri nets, and transition systems [9].

We can consider some other operations (instructions) in *Physarum* machines like: *add node*, *remove node*, *add edge*, *remove edge* [2]. Adding and removing nodes can be implemented through activation and deactivation of attractants, respectively. Adding and removing edges can be implemented by means of repellents put in proper places in the space. An activated repellent can avoid a plasmodium transition between attractants. Adding and removing edges can change dynamically over time. To model such behavior, we propose to add another high-level model, based on timed transition systems [4], to our language. It is assumed, in transition systems mentioned earlier, that all events happen instantaneously. In timed transition systems, timing constraints restrict the times at which events may occur. The timing constraints are classified into two categories: lower-bound and upper-bound requirements.

Let N be a set of nonnegative integers. Formally, a timed transition system $TTS = (S, E, T, s_0, l, u)$ consists of an underlying transition system $TS = (S, E, T, s_0)$ as well as a minimal delay function (a lower bound) $l : E \to N$ assigning a nonnegative integer to each event and a maximal delay function (an upper bound) $u : E \to N \cup \{\infty\}$ assigning a nonnegative integer or infinity to each event.

Let us consider a simple timed transition system shown as a graph structure in figure 1 with the following timing constraints: $l(e_1) = 0$, $u(e_1) = \infty$, $l(e_2) = 0$, $u(e_2) = \infty$, $l(e_3) = 5$, $u(e_3) = 10$.

Figure 1. A timed transition system.



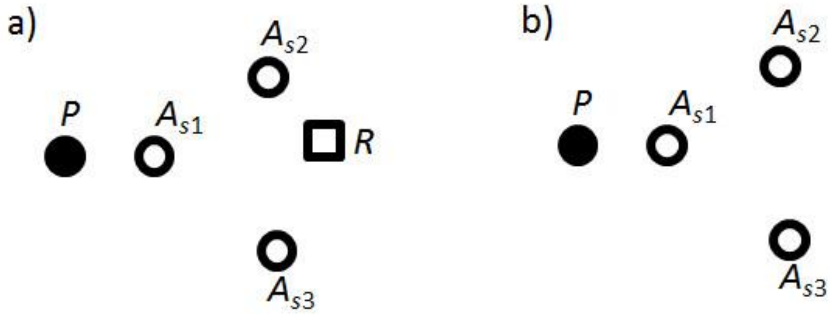
The code in our created language has the following form:

```
#TRANSITION_SYSTEM
s1=new TS.State("s1");
s1.setAsInitial;
s2=new TS.State("s2");
s3=new TS.State("s3");
e1=new TS.Event("e1");
t1=new TS.Transition(s1,e1,s2);
e2=new TS.Event("e2");
t2=new TS.Transition(s1,e2,s3);
e3=new TS.Event("e3");
e3.setTimingConstraints(5,10);
t3=new TS.Transition(s2,e3,s3);
```

The default timing constraints are 0 as a lower bound and ∞ as an upper bound.

As a result of programming the *Physarum* machine, we obtain spatial configurations of stimuli presented in figure 2, (a) for the time instant $\tau = 4$, (b) for the time instant $\tau = 8$, where P is *Physarum*, A_{s_1}, A_{s_2}, A_{s_3} are attractants, and R is a repellent. It is easy to see that the event e_3 is allowed only if actual time $t \in \{5,6,\ldots,10\}$. Therefore, in the model in figure 2 (a), a repellent, avoiding the transition between states s_2 and s_3 as a result of the event e_3, is present, i.e., it is activated.

Figure 2. Spatial configurations of stimuli for the *Physarum* machine.



In a real-life implementation of *Physarum* machines, attractants are sources of nutrients or pheromones, on which the plasmodium feeds whereas in case of repellents the fact that plasmodium of *Physarum* avoids light and some thermo- and salt-based conditions is used. Repellents can be activated or deactivated for proper time periods, especially in case of light.

Modelling dynamically changed structures over time is important in some cases of abstract machines used in non-classical computations. One of them is a Kolmogorov-Uspensky machine. In [1], there was shown that the plasmodium of *Physarum polycephalum* is a biological substrate that implements a Kolmogorov-Uspensky machine (at the beginning called a Kolmogorov complex) – a concept of an abstract machine defined on a dynamically changing graph-based structure outlined by A.N. Kolmogorov and V.A. Uspensky [5], [6]. The Kolmogorov-Uspensky machine is a process on a finite undirected connected graph with distinctly labelled nodes. A computational process travels on the graph, activates nodes and removes and adds edges. There is only one active node at any step of development.

Acknowledgments

This research is being fulfilled by the support of FP7-ICT-2011-8.

References

[1] Adamatzky, A., *Physarum machine: Implementation of a Kolmogorov-Uspensky machine on a biological substrate,* Parallel Processing Letters 17(4), pp. 455–467, 2007.

[2] Adamatzky, A., *Physarum Machines: Computers from Slime Mould,* World Scientific, 2010.

[3] Adamatzky, A., Erokhin, V., Grube, M., Schubert, T., Schumann, A., *Physarum Chip Project: Growing computers from slime mould,* International Journal of Unconventional Computing 8(4), pp. 319–323, 2012.

[4] Henzinger, T., Manna, Z., Pnueli, A., *Timed transition systems,* [in:] de Bakker, J., Huizing, C., de Roever, W., Rozenberg, G. (eds.), *Real-Time: Theory in Practice,* Lecture Notes in Computer Science, vol. 600, Springer Berlin Heidelberg, pp. 226–251, 1992.

[5] Kolmogorov, A., Uspensky, V., *On the definition of an algorithm,* Uspekhi Mat. Nauk 13(4), 3–28, 1958 (in Russian).

[6] Kolmogorov, A.N., *On the concept of algorithm,* Uspekhi Mat. Nauk 8(4), pp. 175–176, 1953 (in Russian).

[7] Nakagaki, T., Yamada, H., Toth, A., *Maze-solving by an amoeboid organism,* Nature 407, pp. 470–470, 2000.

[8] Nielsen, M., Rozenberg, G., Thiagarajan, P., *Elementary transition systems*, Theoretical Computer Science 96(1), pp. 3–33, 1992.

[9] Pancerz, K., Schumann, A., *Principles of an object-oriented programming language for Physarum polycephalum computing,* [in:] *Proceedings of the 10th International Conference on Digital Technologies (DT'2014)*, Zilina, Slovak Republic, pp. 273–280, 2014.

[10] Schumann, A., Pancerz, K., *Towards an object-oriented programming language for Physarum polycephalum computing*, [in:] Szczuka, M., Czaja, L., Kacprzak, M. (eds.), *Proceedings of the Workshop on Concurrency,* Specification and Programming (CS&P'2013), Warsaw, Poland, pp. 389–397, 2013.

Originally published: Schumann, A., Pancerz, K., *PhysarumSoft – a Software Tool for Programming Physarum Machines and Simulating Physarum Games*, [in:] Ganzha, M., Maciaszek, L., Paprzycki, M. (eds.), *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, Lodz, Poland, September 13-16, Annals of Computer Science and Information Systems, vol. 5, PTI, Warsaw, pp. 607-614, 2015.

XII
PhysarumSoft – a Software Tool for Programming *Physarum* Machines and Simulating *Physarum* Games

Andrew Schumann, Krzysztof Pancerz

Abstract

In the paper, we describe selected functionality of the current version of a new software tool, called *PhysarumSoft*, developed for programming *Physarum* machines and simulating *Physarum* games. The tool was designed for the Java platform. A *Physarum* machine is a biological computing device implemented in the plasmodium of *Physarum polycephalum* or *Badhamia utricularis* that are one-cell organisms able to build programmable complex networks. The plasmodial stage of such organisms is a natural transition system that can be used as a medium for solving different computational tasks as well as creating bio-inspired strategy games.

Introduction

A *Physarum* machine is a programmable amorphous biological computing device, experimentally implemented in the plasmodium of *Physarum polycephalum*, also called true slime mould [1]. *Physarum polycephalum* is a single cell organism belonging to the species of order *Physarales*. In the considered case, the term of *Physarum* machine covers, in general, a hybrid device implemented in two plasmodia (cf. [2]), namely the plasmodium of *Physarum polycephalum* as well as the plasmodium of *Badhamia utricularis*. *Badhamia utricularis* is also the species of order *Physarales*. The plasmodium of *Physarum polycephalum* or *Badhamia utricularis*, spread by networks, can be programmable. In propagating and foraging behavior of the plasmodium, we can perform useful computational tasks. This ability was firstly discerned by T. Nakagaki et al. [3]. The *Physarum* machine comprises an amorphous yellowish mass with networks of protoplasmic veins, programmed by spatial configurations of attracting and repelling stimuli. When several attractants are scattered in the plasmodium range, the plasmodium looks for attractants, propagates protoplasmic veins towards them, feeds on them and goes on. Repellents play the role of elements blocking propagation of protoplasmic veins.

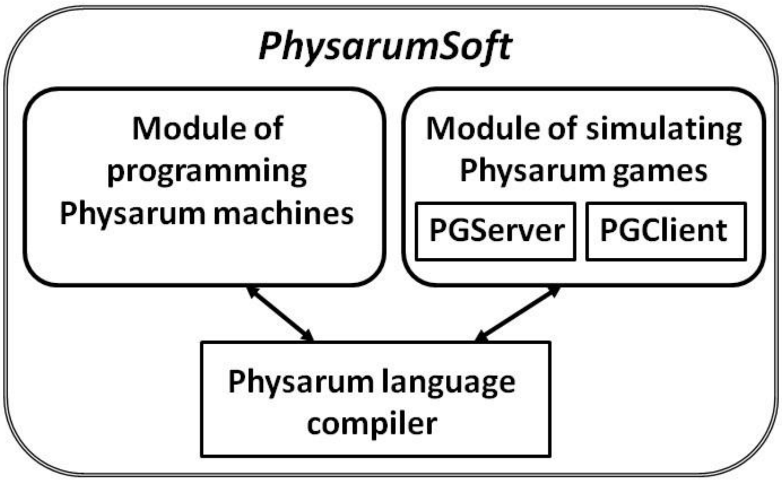
Solving computational tasks by means of *Physarum* machines is one of the main goals of the *Physarum Chip Project: Growing Computers from Slime Mould* [4] funded by the Seventh Framework Programme (FP7). In this project, we are going to construct an unconventional computer on programmable behaviour of *Physarum polycephalum*.

To program computational tasks for *Physarum* machines, we are developing a new object-oriented programming language [6], [9], [15], called a *Physarum* language. The *Physarum* language is a prototype-based language [3] consisting of inbuilt sets of prototypes corresponding to both the high-level models used for describing behaviour of *Physarum polycephalum* (e.g., ladder diagrams, transition systems, timed transition systems, Petri nets) and the low-level model (distribution of stimuli). More information is given in section 12.1.

Another task that can be performed in *Physarum* machine environments concerns bio-inspired strategy games. Fundamental topics of the research area related to bio-inspired games on *Physarum* machines were earlier considered, for example in [2] and [9]. Simulating *Physarum* games is considered in section 12.2.

To support reserach on programming *Physarum* machines and simulating *Physarum* games, we are developing a specialized software tool, called the *Physarum* software system, shortly *PhysarumSoft*. The tool was designed for the Java platform. In the paper, we describe selected functionality of the current version of *PhysarumSoft*. A general structure of this system is shown in figure 1.

Figure 1. A general structure of *PhysarumSoft.*



We can distinguish three main parts of *PhysarumSoft*:

- *Physarum* language compiler. The *Physarum* language is an object-oriented high-level programming language. For generating the compiler of the language, the Java Compiler Compiler (JavaCC) tool [10] was used. JavaCC is the most popular parser generator for use with Java applications. For the programming purpose, a compiler embodied in our tool translates the high-level code describing a model of the *Physarum* machine into the spatial distribution (configuration) of stimuli (attractants, repellents) controlling propagation of protoplasmic veins of the plasmodium. A grammar of our language was described in [9].
- Module of programming *Physarum* machines described in section 12.1.
- Module of simulating *Physarum* games described in section 12.2.

The main features of *PhysarumSoft* are the following:

- Portability. Thanks to the Java technology, the created tool can be run on various software and hardware platforms. In the future, the tool will be adapted for platforms available in mobile devices and as a service in the cloud.
- User-friendly interface (see some screenshots shown in sections 2 and 3).
- Modularity. The project of *PhysarumSoft* and its implementation covers modularity. It makes the tool extend in the future eaisly.

12.1. Programming *Physarum* machines

To program *Physarum* machines (i.e., to set the spatial distribution (configuration) of stimuli (attractants, repellents) controlling propagation of protoplasmic veins of the plasmodium), we are developing a new object-oriented programming language [6], [9], [15], called the *Physarum* language. Our language is based on the prototype-based approach (cf. [3]) that is less common than the class-based one, although, it has a great deal to offer. This approach is also called class-less or instance-based programming because prototype-based languages are based upon the idea that objects, representing individuals, can be created without reference to class-defining. In this approach, the objects, that are manipulated at runtime, are prototypes. In our language, there are inbuilt sets of prototypes corresponding to both the high-level models used for describing behaviour of *Physarum polycephalum* (e.g., ladder diagrams, transition systems, timed transition systems, Petri nets) and the low-level model (distribution of stimuli). According to the prototype-based approach, objects are created by means of a copy operation, called cloning, which is applied to a given prototype. Objects can be instantiated (cloned) via the keyword *new* using defined constructors. Different methods are used to manipulate features of the objects and create relationships between objects.

In our approach, the starting point, in programming the behaviour of the *Physarum* machine, is a high-level model describing propagation of protoplasmic veins of *Physarum*. We have proposed several high-level models used in programming *Physarum* machines, i.e.:

- ladder diagrams (see [17]),
- transition systems ([6]) and timed transition systems (see [14]),
- Petri nets (see [15]).

In the remaining part of this section, we recall basic definitions concerning transition systems, timed transition systems and Petri nets. They are general purpose tools that can be used to model dynamic systems with distinguished states and transitions between states. Application of ladder diagrams is restricted to modelling digital circuits. The recalled definitions are illustrated with some examples.

Transition systems are a simple and powerful tool for explaining the operational behaviour of models of concurrency. Formally, a transition system is a quadruple $TS = (S, E, T, I)$, cf. [14], where:

- S is the non-empty set of states,

- E is the set of events,
- $T \subseteq S \times E \times S$ is the transition relation,
- I is the set of initial states.

Usually transition systems are based on actions which may be viewed as labelled events. If $(s, e, s') \in T$, then the idea is that TS can go from s to s' as a result of the event e occurring at s. Any transition system $TS = (S, E, T, I)$ can be presented in the form of a labelled graph with nodes corresponding to states from S, edges representing the transition relation T, and labels of edges corresponding to events from E.

The behaviour of *Physarum* machines is often dynamically changed in time. It is assumed, in the transition systems mentioned earlier, that all events happen instantaneously. Therefore, in [14], we proposed to use another high-level model, based on timed transition systems [15]. In the timed transition systems, timing constraints restrict the times at which events may occur. The timing constraints are classified into two categories: lower-bound and upper-bound requirements.

Let N be a set of nonnegative integers. Formally, a timed transition system $TTS = (S, E, T, I, l, u)$ consists of:

- an underlying transition system $TS = (S, E, T, I)$,
- a minimal delay function (a lower bound) $l : E \rightarrow N$ assigning a nonnegative integer to each event,
- a maximal delay function (an upper bound) $u : E \rightarrow N \cup \infty$ assigning a nonnegative integer or infinity to each event.

In *Physarum* machines, timing constraints can be implemented through activation and deactivation of stimuli (attractants and/or repellents). Each state corresponds to either an original point of the plasmodium or an attractant. Especially, initial states of transition systems can be presented by original points, where protoplasmic veins originate from. Edges represent plasmodium transitions between attractants as well as the original points of the plasmodium.

In case of transition system and timed transition system models, the main prototypes defined in the *Physarum* language and their selected methods are collected in table 1. Main prototypes, corresponding to transition system and timed transition system models, defined in the Physarum language, and their selected methods

Prototype	Selected methods
TS.State	*setDescription, setAsInitial*
TS.Event	*setDescription*, *setTimingConstraints*
TS.Transition	

Let us consider an exemplary timed transition system shown in figure 2. Formally, we have $TTS = (S, E, T, I, l, u)$, where:

- $S = \{s_1, s_2, s_3, s_4\}$,

- $E = \{e_1, e_2, e_3\}$,
- $T = \{(s_1, e_1, s_2), (s_2, e_2, s_3), (s_3, e_3, s_4)\}$,
- $I = \{s_1, s_2\}$,
- $l(e_1) = l(e_2) = l(e_3) = 0$,
- and $u(e_1) = u(e_2) = \infty$, $u(e_3) = 3$.

Figure 2. An exemplary timed transition system TTS .

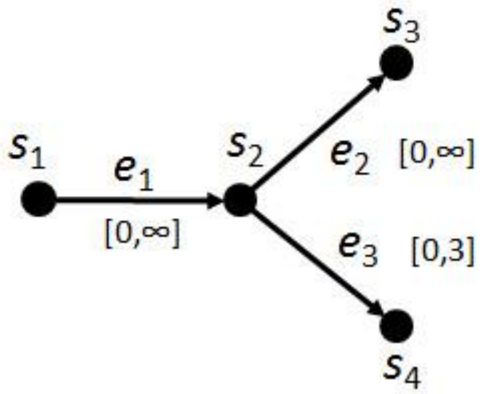
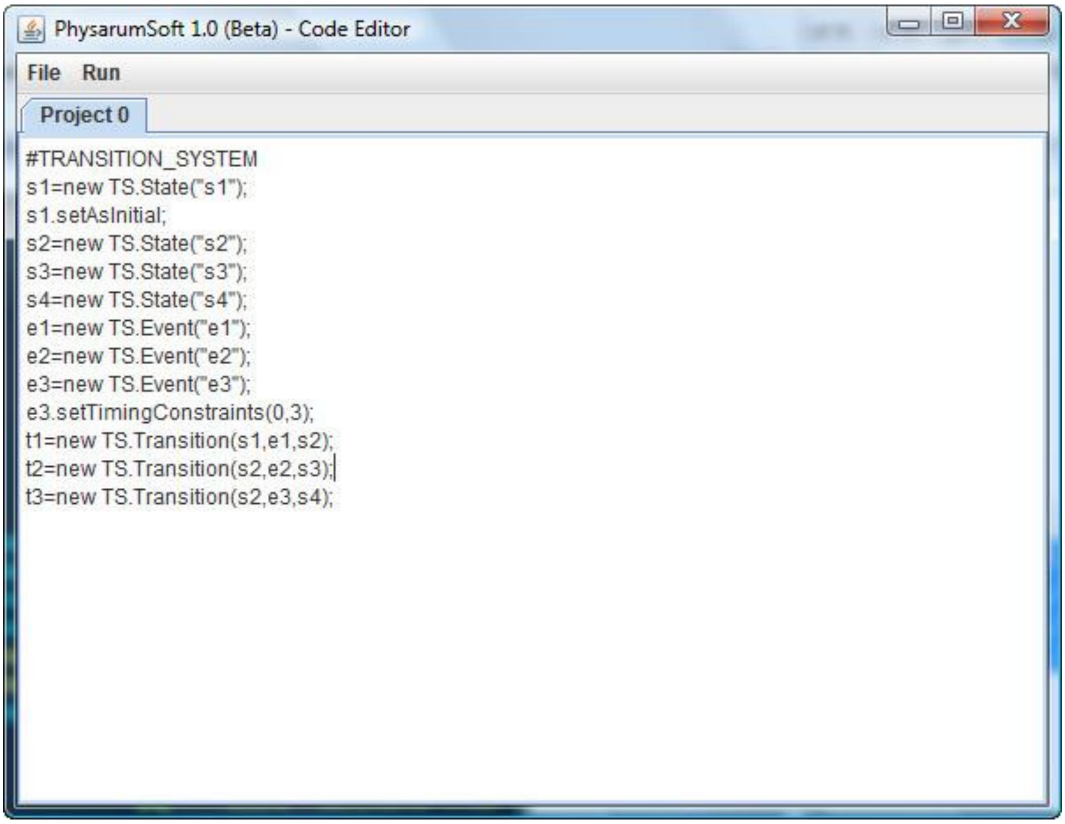


Figure 3. The code for the model in the form of TTS .

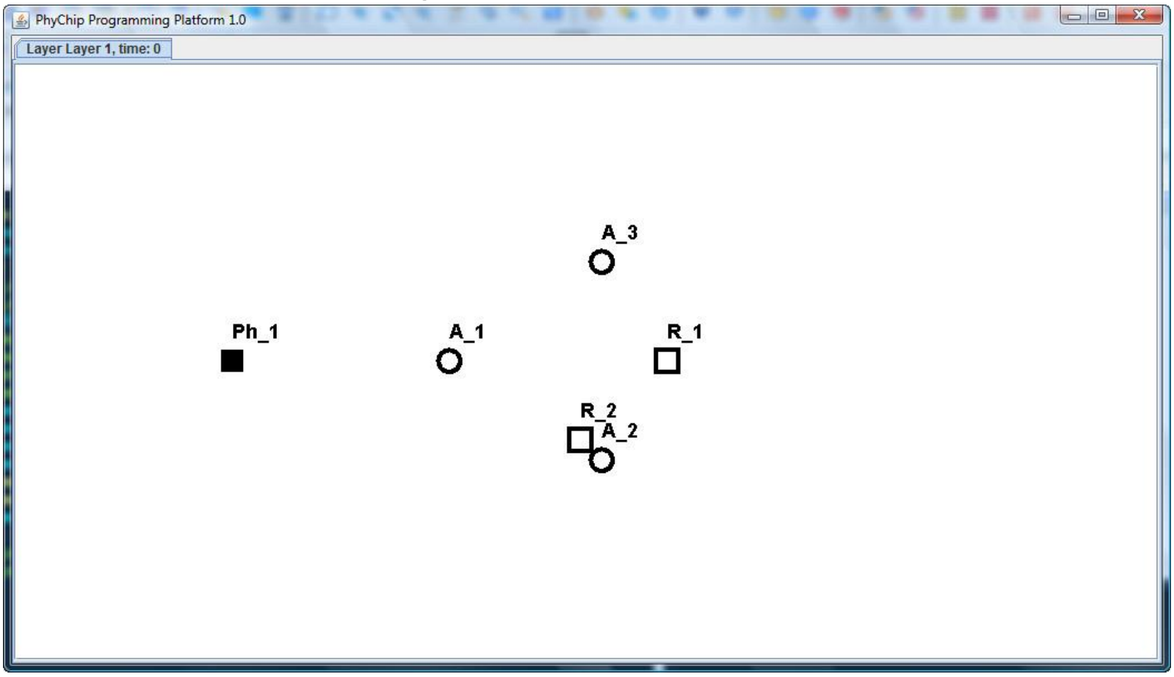


The code, for the model in the form of TTS , in the *Physarum* language written in the editor of the module of programming *Physarum* machines, is shown in figure 3. The result of compilation, i.e., the spatial distribution of stimuli (attractants, repellents) controlling propagation of protoplasmic veins of the plasmodium, is shown in figure 4.

One can see that:

- *Physarum* Ph_1 represents initial state s_1, attractants A_1, A_2, A_3 represent states s_2, s_4, s_3, respectively.
- Splitting the plasmodium at A_1 is supported by repellent R_1.
- Repellent R_2 is placed next to attractant A_2 because timing constraints are set for event e_3.
- For $t > 3$, R_2 must be activated to annihilate the vein of the plasmodium between A_1 and A_2.

Figure 4. The result of compilation.



Petri nets introduced by C.A. Petri [12] are a formal tool used to model discrete event systems. In [15], we proposed to use Petri nets with inhibitor arcs (cf. [3]) to model behaviour of *Physarum polycephalum*. The inhibitor arcs test the absence of tokens in a place and they can be used to disable transitions. This fact can model repellents in *Physarum* machines. A transition can only fire if all its places connected through inhibitor arcs are empty (cf. [20]).

Formally, a marked Petri net with inhibitor arcs is a five-tuple

$$MPN = (Pl, Tr, Ar, w, m),$$

where:

- Pl is the finite set of places (marked graphically with circles),
- Tr is the finite set of transitions (marked graphically with rectangles),

- $Ar = Ar_O \cup Ar_I$ such that $Ar_O \subseteq (Pl \times Tr) \cup (Tr \times Pl)$ is the set of ordinary arcs (marked graphically with arrows) from places to transitions and from transitions to places whereas $Ar_I \subseteq Pl \times Tr$ is the set of inhibitor arcs (marked graphically with lines ended with small circles) from places to transitions,
- $w : Ar \rightarrow \{1,2,3,\ldots\}$ is the weight function on the arcs,
- $m : Pl \rightarrow \{0,1,2,\ldots\}$ is the initial marking function on the places.

In describing the Petri net behaviour, it is convenient to use for any $t \in Tr$:

- $I_O(t) = \{p \in Pl : (p,t) \in Ar_O\}$ – a set of input places connected through ordinary arcs to the transition t,
- $I_I(t) = \{p \in Pl : (p,t) \in Ar_I\}$ – a set of input places connected through inhibitor arcs to the transition t,
- $O(t) = \{p \in Pl : (t,p) \in Ar_O\}$ – a set of output places connected through ordinary arcs from the transition t.

In the proposed approach, we have additionally assumed the following limits for the Petri net:

- $w(a) = 1$ for each $a \in Ar$,
- $m(p) \leq 1$ for each $p \in Pl$ (the capacity limit).

If $m(p) = 1$, then a token (i.e., a black dot) is drawn in the graphical representation of the place p. Assuming limits as the ones above, a transition $t \in Tr$ is said to be enabled if and only if $m(p) = 1$ for all $p \in I_O(t)$, i.e., the token is present in all input places p connected with the transition t through the ordinary arcs, and $m(p) = 0$ for all $p \in I_I(t)$, i.e., the token is absent in all input places p connected with the transition t through the inhibitor arcs, and $m(p) = 0$ for all $p \in O(t)$, i.e., the token is absent in all output places p of the transition t. If the transition t is enabled, we say that it can fire. A new marking function $m' : Pl \rightarrow \{0,1,2,\ldots\}$ defines the next state of the Petri net after firing the transition t:

$$m'(p) = \begin{cases} m(p) - 1 & if p \in I_O(t) and p \notin O(t), \\ m(p) + 1 & if p \in O(t) and p \notin I_O(t), \\ m(p) & otherwise \end{cases}$$

It is worth noting that in all figures including Petri net models, to simplify them, we have used bidirectional arcs between input places and transitions instead of arcs from input places to transitions and from transitions to input places. A bidirectional arc causes that the token is not consumed (removed) from the input place after firing a transition. This fact has a natural justification, i.e., firing a transition does not cause deactivation of the attractants and disappearance of the plasmodium from the original point. The plasmodium grows to build a dendritic network of veins.

In the proposed Petri net models of *Physarum* machines, we can distinguish three kinds of places:

- Places representing *Physarum polycephalum*.
- Places representing control stimuli (repellents).
- Places representing output stimuli (attractants).

In the *Physarum* language, the kind of a place is determined by the role played by it.

For each kind of places, we adopt different meaning (interpretation) of tokens. The meaning of tokens in places representing *Physarum polycephalum* is natural, i.e., the token in a given place corresponds to the presence of the plasmodium of *Physarum polycephalum* in an original point, where it starts to grow. The meaning of tokens in places representing control stimuli is shown in table 2, whereas the meaning of tokens in places representing output stimuli is shown in table 3. In case of control stimuli, we are interested in whether a given stimulus is activated or not. In case of output stimuli (attractants), we are interested in whether a given attractant is occupied by the plasmodium of *Physarum polycephalum*. Transitions in Petri net models represent the flow (propagation) of the plasmodium from the original points to attractants as well as between attractants.

Table 2. The meaning of tokens in places representing control stimuli.

Token	Meaning
Present	Stimulus activated
Absent	Stimulus deactivated

Table 3. The meaning of tokens in places representing output stimuli.

Token	Meaning
Present	Stimulus occupied by plasmodium of *Physarum polycephalum*
Absent	Stimulus not occupied by plasmodium of *Physarum polycephalum*

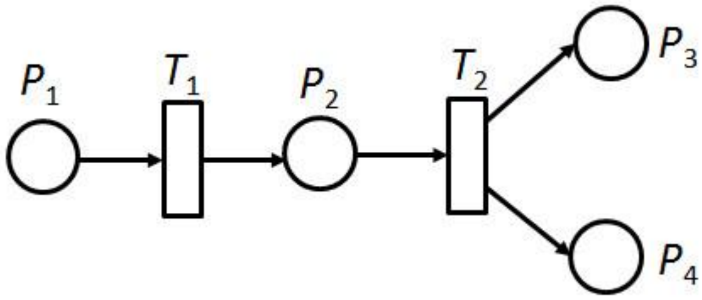
In case of Petri net models, the main prototypes defined in the *Physarum* language and their selected methods are collected in table 4. Main prototypes, corresponding to Petri net models, defined in the *Physarum* language, and their selected methods.

Prototype	Selected methods
PN.Place	*setDescription, setRole*
PN.Transition	*setDescription*
PN.Arc	*setAsInhibitor, setAsBidirectional*

Let us consider an exemplary Petri net shown in figure 5. Formally, we have $MPN = (Pl, Tr, Ar, w, m)$, where:

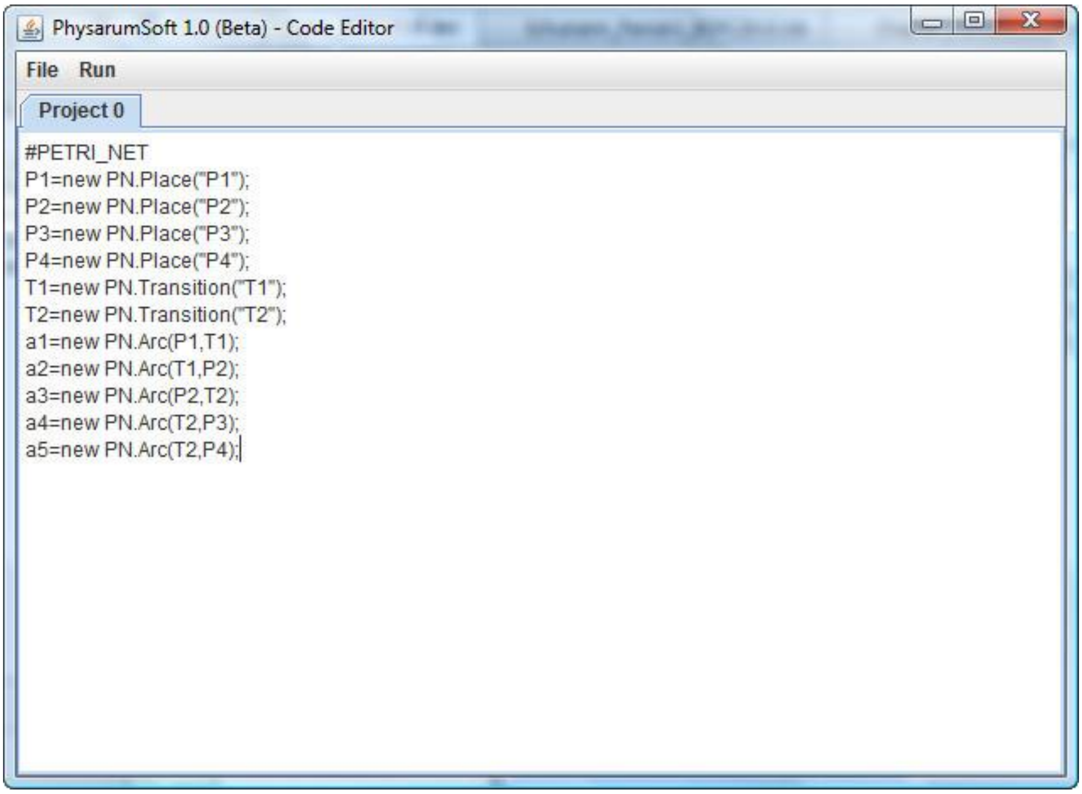
- $Pl = \{P_1, P_2, P_3, P_4\}$,
- $Tr = \{T_1, T_2\}$,
- $Ar = Ar_O \cup Ar_I$, such that
  - $Ar_O = \{(P_1, T_1), (T_1, P_2), (P_2, T_2), (T_2, P_3), (T_2, P_4)\}$,
  - and $Ar_I = \varnothing$,
- $w(a) = 1$ for all $a \in Ar$,
- and $m(p) = 0$ for each $p \in Pl$,

Figure 5. An exemplary Petri net MPN.



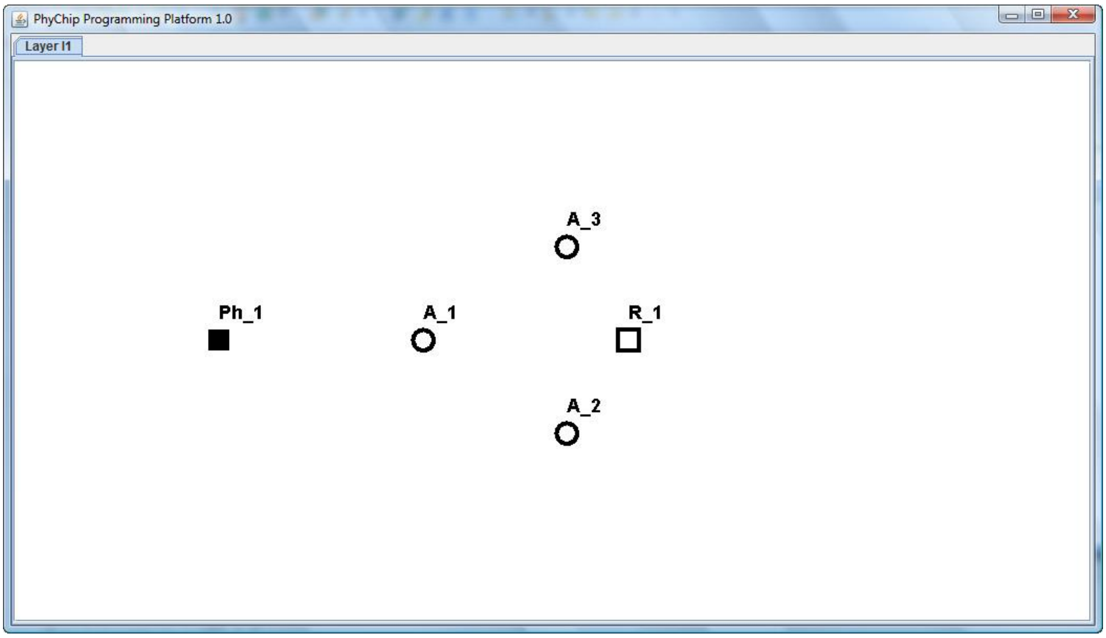
The code, for the model in the form of MPN, in the *Physarum* language written in the editor of the module of programming *Physarum* machines, is shown in figure 6.

Figure 6. The code for the model in the form of MPN.



The result of compilation is shown in figure 7.

Figure 7. The code for the model in the form of MPN .



One can see that *Physarum* Ph_1 represents place P_1, attractants A_1, A_2, A_3 represent places P_2, P_4, P_3, respectively. Splitting the plasmodium at A_2 is supported by repellent R_1.

In [9], we showed how to use high-level models (transition systems and Petri nets) to describe four basic forms of *Physarum* motions:

- *direct* - direction, i.e., a movement from one point, where the plasmodium is located, towards another point, where there is a neighbouring attractant,
- *fuse* - fusion of two plasmodia at the point, where they meet the same attractant,
- *split* - splitting the plasmodium from one active point into two active points, where two neighbouring attractants with a similar power of intensity are located,
- *repel* - repelling of the plasmodium or inaction.

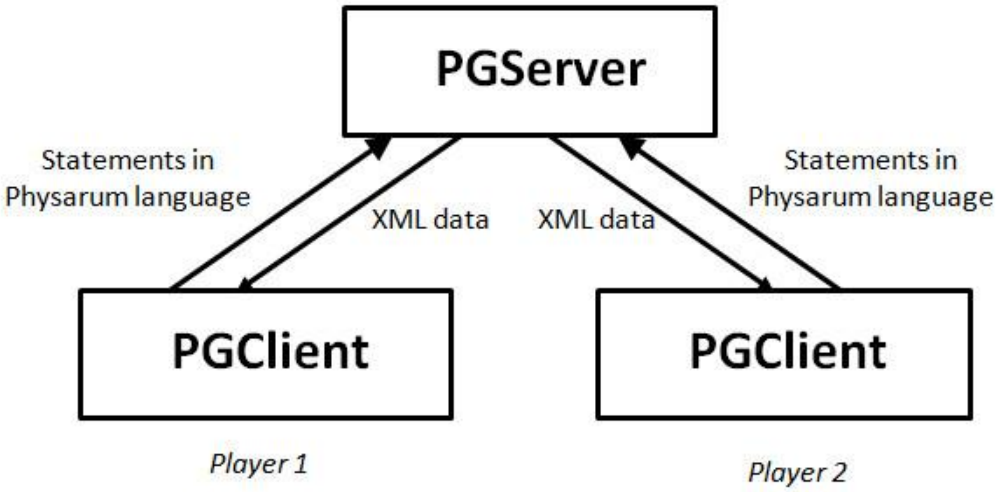
It is worth noting that four basic forms are fundamental components used to build or describe more complex systems.

12.2. Simulating *Physarum* games

In [2], we showed that the slime mould (*Physarum polycephalum*) is a natural transition system which can be considered a biological model for strategic games. General assumptions for such games were presented both in [2] and [9]. They are based on the experiments performed by A. Adamatsky and M. Grube. If there are only two agents of the plasmodium game, where the first agent is presented by a usual *Physarum polycephalum* plasmodium and the second agent by its modification, a *Badhamia utricularis* plasmodium, then both start to compete with each other.

To simulate games on *Physarum* machines, we are developing a special module of *PhysarumSoft* called the *Physarum* game simulator. This module works under the client-server paradigm. A general structure of the *Physarum* game simulator is shown in figure 8.

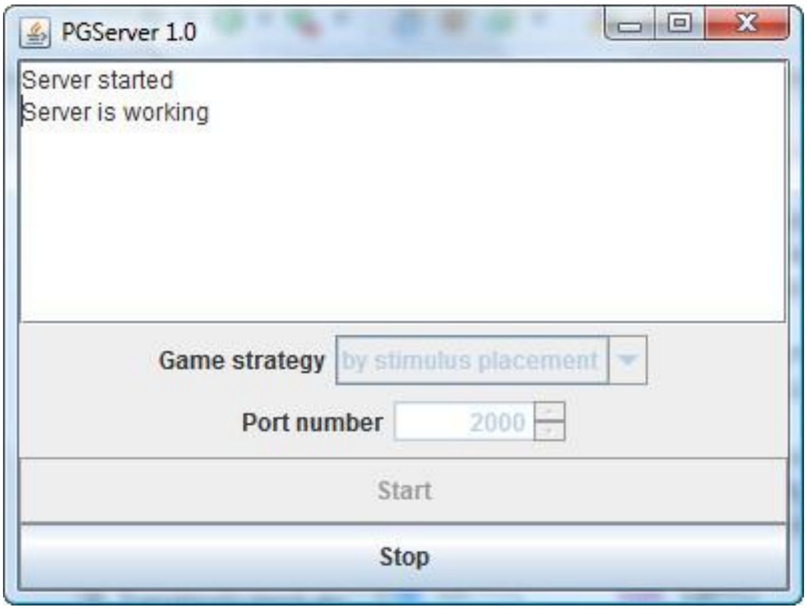
Figure 8. A general structure of the *Physarum* game simulator.



The server-side application of the *Physarum* game simulator is called *PGServer*. The main window of *PGServer* is shown in figure 9. In this window, the user can:

- select the port number on which the server listens for connections,
- start and stop the server,
- set the game strategy:
    - strategy by stimulus placement,
    - strategy by stimulus activation,
- shadow information about actions undertaken.

Figure 9. The main window of *PGServer.*



The client-side application of the *Physarum* game simulator is called *PGClient*. The main window of *PGClient* is shown in figures 10 and 11. In this window, the user can:

- set the server IP address and its port number,
- start the participation in the game,
- manipulate stimuli (place or activate them) during the game,
- monitor the current result.

In the *Physarum* game simulator, we have two players:

- the first one plays for the *Physarum polycephalum* plasmodia,
- the second one plays for the *Badhamia utricularis* plasmodia.

Locations of the original points of both plasmodia are randomly generated. The players can control motions of plasmodia via attracting or repelling stimuli. There are two strategies which can be defined for the game:

- Locations of attractants and repellents are *a priori* generated in a random way. During the game, each player can activate one stimulus (attractant or repellent) at each step.
- Locations of attractants and repellents are determined by the players during the game.

At each step, each player can put one stimulus (attractant or repellent) at any location and this stimulus becomes automatically activated.

The client-side main window for the first strategy (locations of attractants and repellents are *a priori* generated in a random way) is shown in figure 12. At the beginning, the original points of *Physarum polycephalum* and *Badhamia utricularis,* as well as stimuli, are scattered randomly on the plane. The window after several player's movements is shown in figure 13.

Figure 10. The main window of *PGClient* for the first strategy.

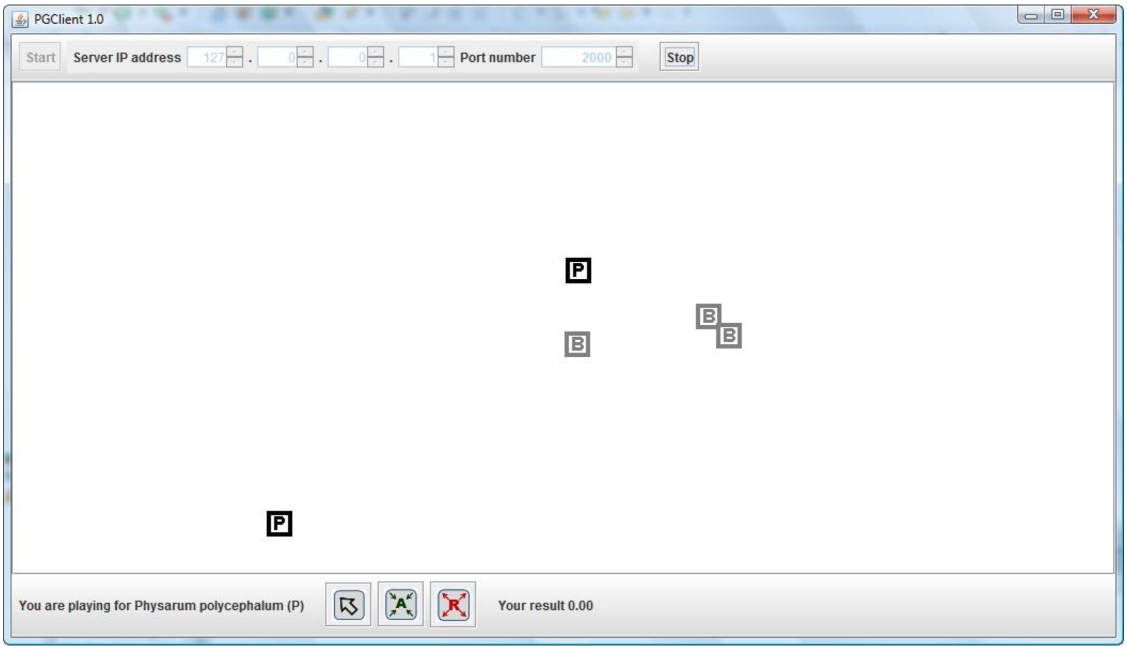
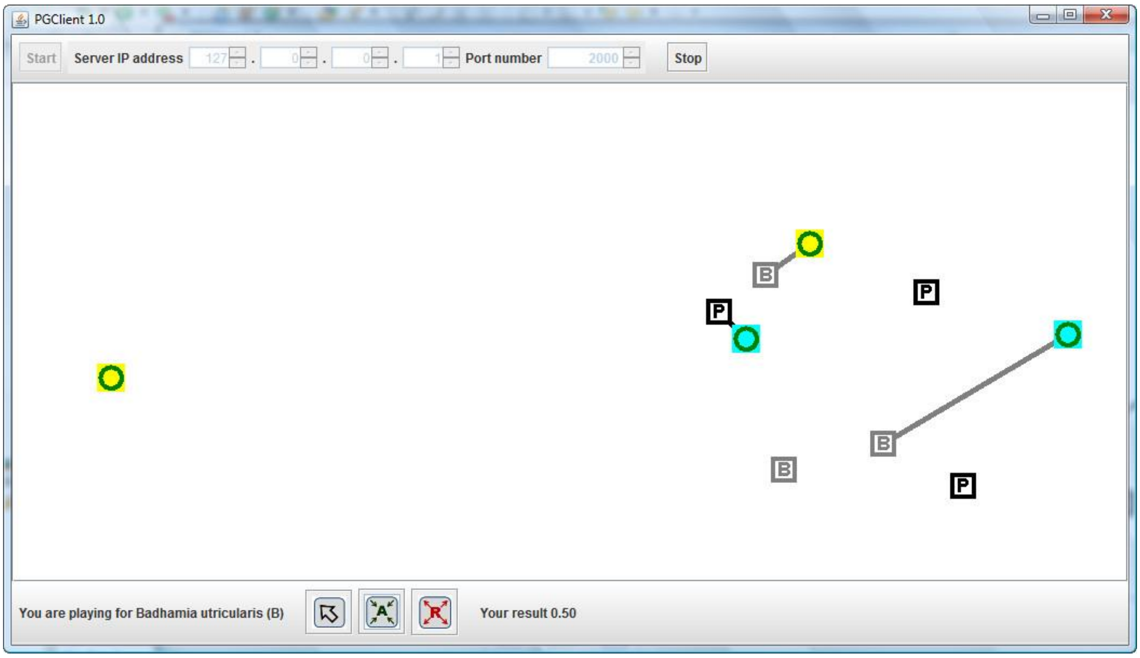


Figure 11. The main window of *PGClient* for the first strategy after several player's movements.



A box labelled by P represents an original point of *Physarum polycephalum*. A box labelled by B represents an original point of *Badhamia utricularis*. A single circle denotes an attractant whereas a double circle denotes repellent. Different background colors of stimuli differentiate between players.

The client-side main window for the second strategy (locations of attractants and repellents are determined by the players during the game) is shown in figure 12. At the beginning, the original points of *Physarum polycephalum* and *Badhamia utricularis* are scattered randomly on the plane. During the game, players can place stimuli. New veins of plasmodia are created. The window after several player's movements is shown in figure 13.

Figure 12. The main window of *PGClient* for the second strategy.

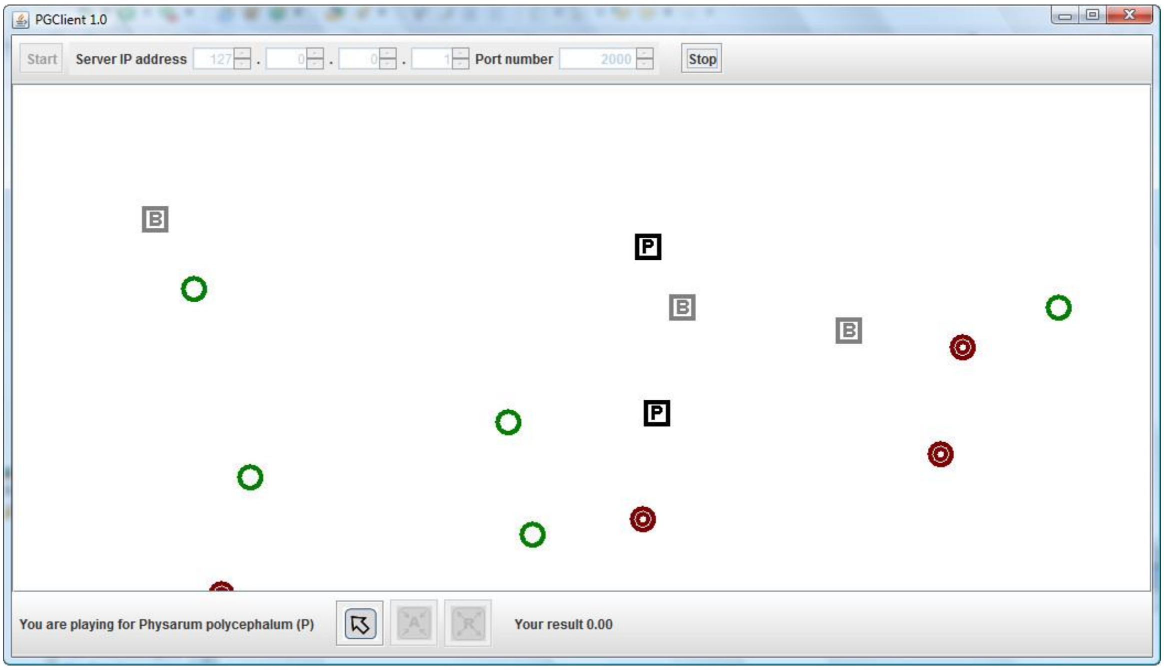
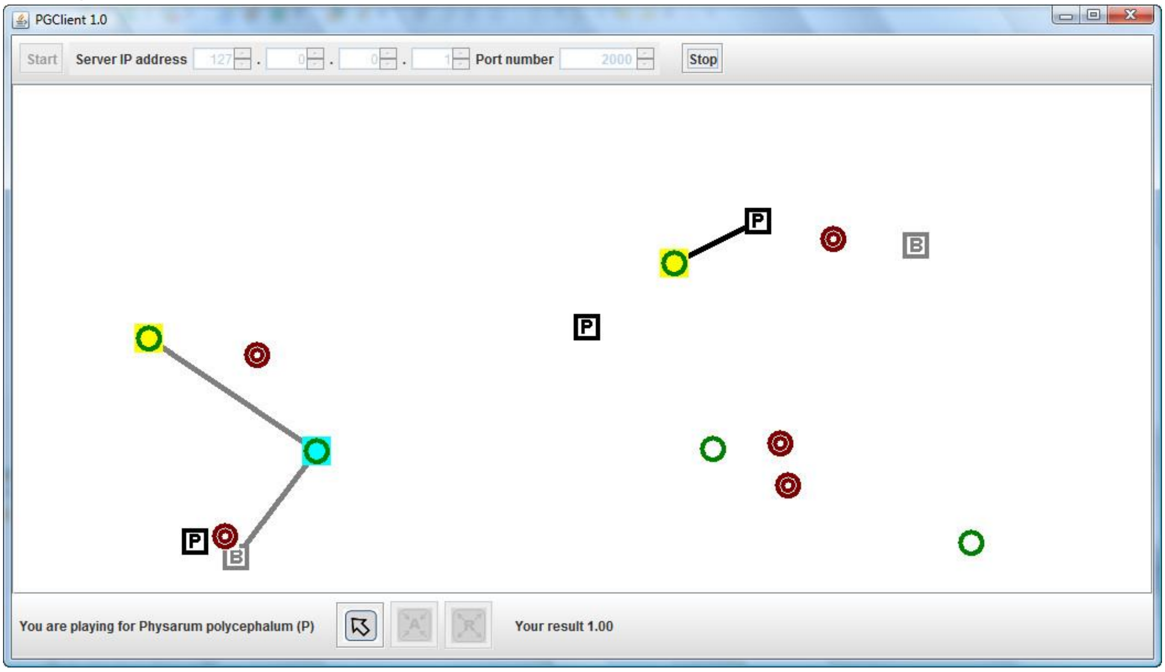


Figure 13. The main window of *PGClient* for the second strategy after several player's movements.



Communication between clients and the server is realized through text messages containing statements of the *Physarum* language. The exemplary code responsible for creation of stimuli has the form:

```
p1_a1=new Attractant(195,224,1);
p1_a2=new Attractant(541,310,1);
p1_a1=new Attractant(580,92,2);
p2_r1=new Repellent(452,130,2);
p2_r1=new Repellent(659,327,1);
```

The first two parameters of stimulus constructors determine the location whereas the last parameter is the player's ID.

The server sends to clients information about the current configuration of the *Physarum* machine (localization of the original points of *Physarum polycephalum* and *Badhamia utricularis*, localization of stimuli, as well as a list of edges, corresponding to veins of plasmodia, between active points) through the XML file. The exemplary XML file has the form:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<network>
<elements>
<element id="0" player="0" type="0" x="96" y="310"/>
<element id="1" player="0" type="0" x="766" y="178"/>
<element id="2" player="0" type="0" x="566" y="248"/>
<element id="3" player="0" type="3" x="550" y="53"/>
<element id="4" player="0" type="3" x="374" y="534"/>
<element id="5" player="0" type="3" x="746" y="217"/>
<element id="6" player="1" type="1" x="195" y="224"/>
<element id="7" player="1" type="1" x="541" y="310"/>
<element id="8" player="2" type="1" x="580" y="92"/>
<element id="9" player="2" type="2" x="452" y="130"/>
<element id="10" player="1" type="2" x="659" y="327"/>
</elements>
<veins>
<vein createdBy="0" firstNodeID="2" secondNodeID="7"/>
<vein createdBy="3" firstNodeID="3" secondNodeID="8"/>
</veins>
</network>
```

The attribute “player” equal to 0 means that elements are created by the system, in this case, the original points of plasmodia (*Physarum polycephalum* or *Badhamia utricularis*).

As payoffs for the created bio-inspired games on *Physarum* machines, we may define a variety of tasks, including simple ones like achieving as many attractants as possible, occupied by plasmodia of organisms for which we play or constructing the longest path consisting of attractants occupied by plasmodia. Determining different payoffs for *Physarum* games appears to be an interesting field of research due to a huge number of different methodologies and paradigms which can be applied.

The activated attractant A^* causes that the plasmodia propagate protoplasmic veins towards it and feed on it. It means that new transitions are created between the current active points of plasmodia and a new one on the attractant A^*. Propagating protoplasmic veins is possible if the current active points are located in the region of influence (ROI) of A^*. It means that a proper neighborhood of A^* is taken into consideration. From that moment, the activated attractant A^* is occupied by plasmodia. It is worth noting that, as the experiments showed, the attractant occupied by the plasmodium of *Physarum polycephalum* cannot be simultaneously occupied by the plasmodium of *Badhamia utricularis* and vice versa. Moreover, the *Physarum polycephalum* plasmodium grows faster and could grow into branches of *Badhamia utricularis*, while the *Badhamia utricularis* plasmodium could grow over *Physarum polycephalum* veins.

The activated repellent R^* can change the direction of plasmodium motions or can avoid propagating plasmodium protoplasmic veins towards activated attractants. Such influences are possible if plasmodia are in the region of influence (ROI) of R^*.

The control capabilities presented above enable the players to choose, at each step, one of the possible tactics:

- The attractant or repellent activated by the player can help propagation of his/her plasmodia (of either *Physarum polycephalum* or *Badhamia utricularis*).
- The attractant or repellent activated by the player can disturb propagation of the second player's plasmodia.

The second possibility is worth considering if we adopt the payoff approximations based on the rough set model described in [19]. It will be the main direction of further investigations. During the game, the players can switch between two possible tactics according to the current game configuration. At the end of the game, we determine who wins.

12.3. Conclusions and further work

In the paper, we have described selected functionality of the current version of a new software tool called *PhysarumSoft*. This tool is intended for programming *Physarum* machines and simulating *Physarum* games. For over the last two years, we have designed a new object-oriented programming language, called the *Physarum* language, that constitutes the basis for modelling behaviour of *Physarum* maachines. This language is used in *PhysarumSoft*. In the nearest future, we plan to extend *PhysarumSoft* to other high-level models, e.g., π -calculus and cellular automata. Moreover, we are developing a new model, based on rough sets [10], to approximate payoffs of strategy games created on *Physarum* machines.

Acknowledgment

This research is being fulfilled by the support of FP7-ICT-2011-8.

References

[1] Adamatzky, A., *Physarum Machines: Computers from Slime Mould*, World Scientific, 2010.

[2] Schumann, A., Pancerz, K., Adamatzky, A., Grube, M., *Bio-inspired game theory: The case of Physarum polycephalum*, [in:] *Proceedings of the 8th International Conference on Bio-inspired Information and Communications Technologies (BICT'2014)*, Boston, Massachusetts, USA, 2014, doi: 10.4108/icst.bict.2014.257869.

[3] Nakagaki, T., Yamada, H., Toth, A., *Maze-solving by an amoeboid organism*, Nature, vol. 407, pp. 470–470, 2000, doi: 10.1038/35035159.

[4] Adamatzky, A., Erokhin, V., Grube, M., Schubert, T., Schumann, A., *Physarum Chip Project: Growing computers from slime mould*, International Journal of Unconventional Computing, vol. 8, no. 4, pp. 319–323, 2012.

[5] Pancerz, K., Schumann, A., *Principles of an object-oriented programming language for Physarum polycephalum computing*, [in:] *Proceedings of the 10th International Conference on Digital Technologies (DT'2014)*, Zilina, Slovak Republic, pp. 273–280, 2014, doi: 10.1109/DT.2014.6868725

[6] *Some issues on an object-oriented programming language for Physarum machines*, [in:] Bris, R., Majernik, J., Pancerz, K., Zaitseva, E. (eds.), *Applications of Computational Intelligence in Biomedical Technology*, ser. Studies in Computational Intelligence, Springer International Publishing, Switzerland, vol. 606, pp. 185–199, 2016.

[7] Schumann, A., Pancerz, K., *Towards an object-oriented programming language for Physarum polycephalum computing*, [in:] Szczuka, M., Czaja, L., Kacprzak, M. (eds.), *Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P'2013)*, Warsaw, Poland, pp. 389–397, 2013.

[8] Craig, I., *Object-Oriented Programming Languages: Interpretation*, London, Springer-Verlag, 2007.

[9] Schumann, A., Pancerz, K., *Interfaces in a game-theoretic setting for controlling the plasmodium motions*, [in:] *Proceedings of the 8th International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS'2015)*, Lisbon, Portugal, pp. 338–343, 2015, doi: 10.5220/0005285203380343.

[10] JavaCC, http://java.net/projects/javacc/

[11] Schumann, A., Pancerz, K., Jones, J., *Towards logic circuits based on physarum polycephalum machines: The ladder diagram approach*, [in:] Plantier, G., Schultz, T., Fred, A., Gamboa, H. (eds.), *Proceedings of the International Conference on Biomedical Electronics and Devices (BIODEVICES'2014)*, A. Cliquet Jr., Angers, France, pp. 165–170, 2014, doi: 10.5220/0004839301650170.

[12] Schumann, A., Pancerz, K., *Timed transition system models for programming Physarum machines: Extended abstract*, [in:] Popova-Zeugmann, L. (ed.), *Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P'2014)*, Chemnitz, Germany, pp. 180–183, 2014.

[13] *Towards an object-oriented programming language for Physarum polycephalum com-*

puting: A Petri net model approach, Fundamenta Informaticae, vol. 133, no. 2-3, pp. 271–285, 2014, doi: 10.3233/FI-2014-1076.

[14] Nielsen, M., Rozenberg, G., Thiagarajan, P., *Elementary transition systems,* Theoretical Computer Science, vol. 96, no. 1, pp. 3–33, 1992, doi: 10.1016/0304-3975(92)90180-N.

[15] Henzinger, T.A., Manna, Z., Pnueli, A., *Timed transition systems,* [in:] de Bakker, J., Huizing, C., de Roever, W., Rozenberg, G. (eds.), *Real-Time: Theory in Practice,* ser. Lecture Notes in Computer Science, Berlin Heidelberg, Springer, vol. 600, pp. 226–251, 1992.

[16] Petri, A., *Kommunikation mit automaten,* Institut für Instrumentelle Mathematik, Bonn, Schriften des IIM Nr. 2, 1962.

[17] Agerwala, T., Flynn, M., *Comments on capabilities, limitations and 'correctness' of Petri nets,* [in:] *Proceedings of the 1st Annual Symposium on Computer Architecture (ISCA'1973),* Atlanta, USA, pp. 81–86, 1973.

[18] Verbeek, H., Wynn, M., van der Aalst, W., ter Hofstede, A., *Reduction rules for reset/inhibitor nets,* Journal of Computer and System Sciences, vol. 76, no. 2, pp. 125–143, 2010. doi: 10.1016/j.jcss.2009.06.003.

[19] Pancerz, K., Schumann, A., *Rough set models of Physarum machines,* International Journal of General Systems, vol. 44, no. 3, pp. 314–325, 2015, doi: 10.1080/03081079.2014.997529.

[20] Pawlak, Z., *Rough Sets. Theoretical Aspects of Reasoning about Data,* Dordrecht, Kluwer Academic Publishers, 1991.

Originally published: Schumann, A., Pancerz, K., *Petri Net Models of Simple Rule-Based Systems for Programming Physarum Machines: Extended Abstract*, [in:] Suraj, Z., Czaja, L. (eds.), *Proceedings of the 24th International Workshop on Concurrency,* Specification and Programming (CS&P'2015), Rzeszow, Poland, September 28-30, vol. 2, pp. 155-160, 2015.

XIII
Petri Net Models of Simple Rule-Based Systems for Programming *Physarum* Machines. Extended Abstract

Andrew Schumann, Krzysztof Pancerz

Abstract

In the paper, we show that biological substrate in the form of *Physarum polycephalum* can be used to simulate simple rule-based systems. To extort a proper behavior from the substrate, appropriate distribution of stimuli (attractants and/or repellents) is required. To model behavior of the substrate and then program *Physarum* machine (a biological computing device experimentally implemented in the plasmodium of *Physarum polycephalum*), we propose to use Petri net models that can be treated as a high-level description. Petri net models enable us to reflect propagation of protoplasmic veins of the plasmodium in consecutive time instants (step by step).

13.1. Models of simple rule-based systems

There are various knowledge representation methods (cf. [4]) that have been developed to make real-world knowledge suitable for being processed by computers. One of the most popular knowledge representation systems are the rule-based ones. Rules can be easily interpreted by humans. Formally, rules can be presented in the framework of propositional logics. Propositional logics is concerned with the study of propositions, whether they are true or false. Propositions are formed by other propositions with the use of logical connectives. A production rule in rule-based systems is a rule which describes the relation between two propositions d_i and d_j, i.e., a production rule points out to us an antecedent-consequence relationship from proposition d_i to proposition d_j, where $d_i \neq d_j$. The general formulation of a production rule has the form $IF\ d_i, THEN\ d_j$, where d_i and d_j are propositions that can be evaluated as true or false with respect to any circumstance. If the antecedent part or consequence part of a production rule contains *AND* or *OR* connectives, then it is called a composite production rule. Four types of the composite production rules can be distinguished [8]:

- Type 1: IF d_{i_1} AND d_{i_2} AND … AND d_{i_k}, THEN d_j.
- Type 2: IF d_i, THEN d_{j_1} AND d_{j_2} AND … AND d_{j_k}.
- Type 3: IF d_{i_1} OR d_{i_2} OR … OR d_{i_k}, THEN d_j.
- Type 4: IF d_i, THEN d_{j_1} OR d_{j_2} OR … OR d_{j_k}.

Further, we will take into consideration types of rules 1 and 3, only.

Unconventional computing becomes an interdisciplinary field of science, where computer scientists, physicists and mathematicians apply principles of information processing in natural systems to design novel computing devices and architectures. In *Physarum Chip Project: Growing Computers from Slime Mould* [2] supported by FP7, we are going to implement programmable amorphous biological computers in plasmodium of *Physarum polycephalum*. *Physarum polycephalum* is a one-cell organism manifesting some primitive intelligence in its propagating and foraging behavior (cf. [9]). A biological computing device implemented in the plasmodium of *Physarum polycephalum* is said to be a *Physarum* machine. A comprehensive information on *Physarum* machines can be found in [1]. The *Physarum* machine comprises an amorphous yellowish mass with networks of protoplasmic veins, programmed by spatial configurations of attracting and repelling stimuli.

To program *Physarum* machines, i.e., to set the spatial distribution of stimuli, we are designing a new object-oriented programming language [6], [9], [15], called the *Physarum* language. Moreover, to support research on programming *Physarum* machines, we are developing a specialized software tool, called the *Physarum* software system, shortly *PhysarumSoft* (see [13]). Our language is based on the prototype-based approach (cf. [3]). According to this approach, there are inbuilt sets of prototypes, implemented in the language, that correspond to both the high-level models used for describing behaviour of *Physarum polycephalum* (e.g., ladder diagrams, transition systems, timed transition systems, Petri nets) and the low-level model (distribution of stimuli). In [15], we proposed to use Petri nets with inhibitor arcs (cf. [3]) as one of the high-level models to describe behaviour of *Physarum polycephalum*. The inhibitor arcs test the absence of tokens in a place and they can be used to disable transitions. This fact can model repellents in *Physarum* machines. A transition can only fire if all its places connected through inhibitor arcs are empty (cf. [20]). Each high-level model (including a Petri net one) is translated into the low-level language, i.e., spatial distribution of stimuli (attractants and/or repellents). Such distribution can be treated as a program for the *Physarum* machine.

In the literature, one can find a lot of approaches using Petri nets as models of rule-based systems (e.g., [5], [7], [18], [19]). First of all, structures of Petri nets reflect structures of rule-based systems. Various options of structures have been considered, according to respective approaches. Moreover, the proposed approaches differ in the dynamics that models reasoning processes. In our research, we propose another approach in order to reflect dynamics of *Physarum* machines, i.e., propagation of protoplasmic veins of the plasmodium according to activation/deactivation of stimuli. In the proposed Petri net models of *Physarum* machines, we can distinguish several kinds of places:

- Places representing *Physarum polycephalum*.
- Places representing control stimuli (attractants or repellents) corresponding to propositions in antecedent parts of rules.
- Places representing auxiliary stimuli (attractants) corresponding to partial results of evaluation of logical expressions in antecedent parts of composite production rules.
- Places representing output stimuli (attractants) corresponding to propositions in consequence parts of rules.

For each kind of places, we adopt different meaning (interpretation) of tokens (see, for example, tables 1, 2 and 3 for places representing control stimuli and places representing output stimuli, respectively). Each token corresponds to proper evaluation of the proposition according to the role played by a given stimulus.

Table 1. The meaning of tokens in places representing control stimuli.

Token	Meaning	Evaluation of proposition
Present	Stimulus activated	*true* (for attractants), $false$ (for repellents)
Absent	Stimulus deactivated	$false$ (for attractants), *true* (for repellents)

Table 2. The meaning of tokens in places representing auxiliary stimuli.

Token	Meaning	Partial evaluation of expression
Present	Stimulus occupied by plasmodium	*true*
Absent	Stimulus not occupied by plasmodium	*false*

Table 3. The meaning of tokens in places representing output stimuli.

Token	Meaning	Evaluation of proposition
Present	Stimulus occupied by plasmodium	*true*
Absent	Stimulus not occupied by plasmodium	*false*

In our models of simple rule-based systems, we have implemented an idea of flowing power used in ladder diagrams to model digital circuits. The same idea was used by us to construct logic gates through the proper geometrical distribution of stimuli in *Physarum* machines (see [17]). Flowing power is replaced with propagation of plasmodium of *Physarum polycephalum*. Therefore, in each Petri net model of a rule, a place representing *Physarum polycephalum* is present. Petri net models are a useful tool to reflect dynamics of *Physarum* machines, i.e., propagation of protoplasmic veins of the plasmodium in consecutive time instants. Tokens present in places representing output stimuli show which attractants of *Physarum* machines are occupied by the plasmodium at given time instants.

In general, we can distinguish two techniques to control behavior of *Physarum polycephalum*: repellent-based and attractant-based [1]. Attractants are sources of nutrients or pheromones, on which the plasmodium feeds. In case of repellents, the fact that plasmodium of *Physarum* avoids light and some thermo- and salt-based conditions is used. These possibilities are reflected in the cretaed Petri net models. Technically, the second approach is easier to implement. In case of repellent-based control approach, Petri net models of production rules of type 1 and 3 have the form as in figures 1 and 2, respectively.

Figure 1. A Petri net model of a rule of type 1: the repellent-based control approach.

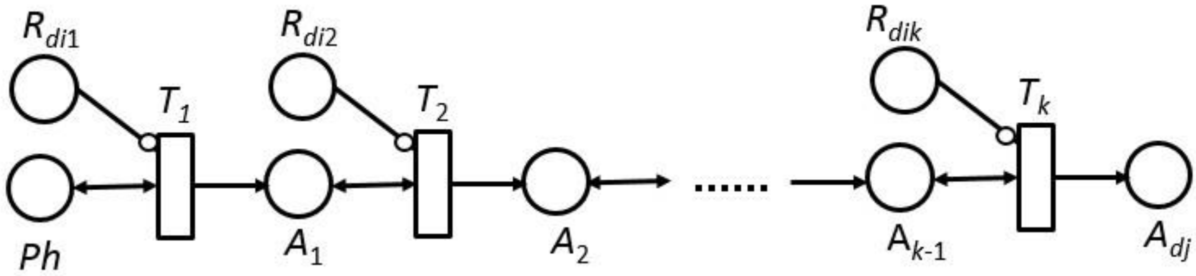
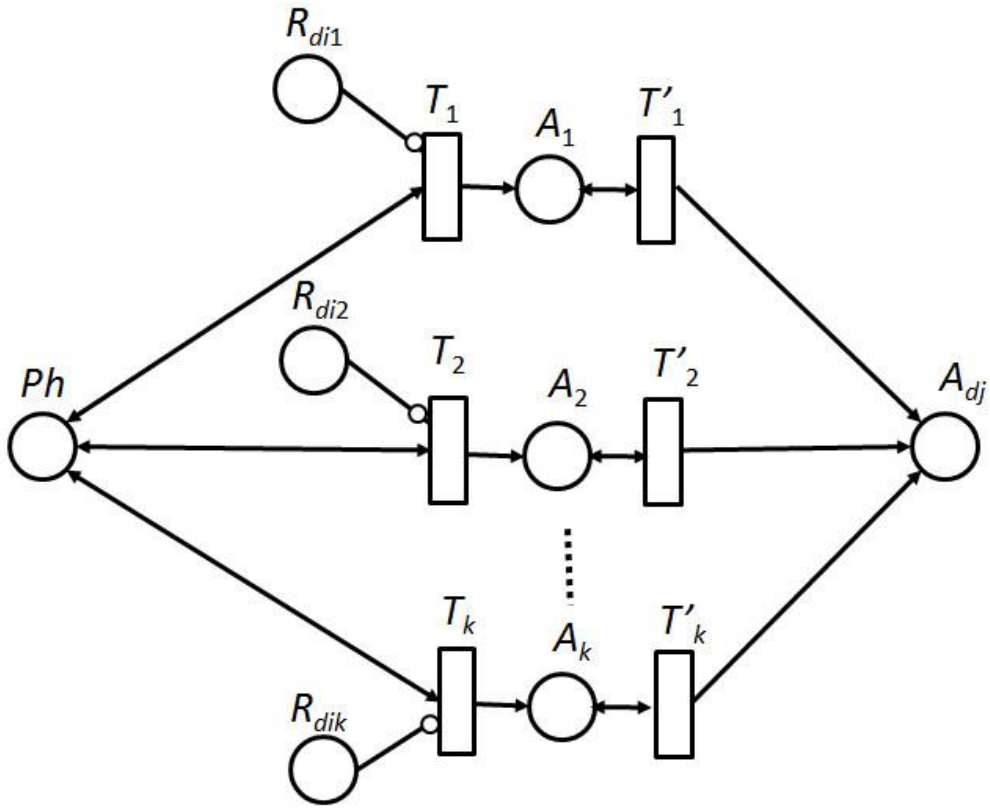
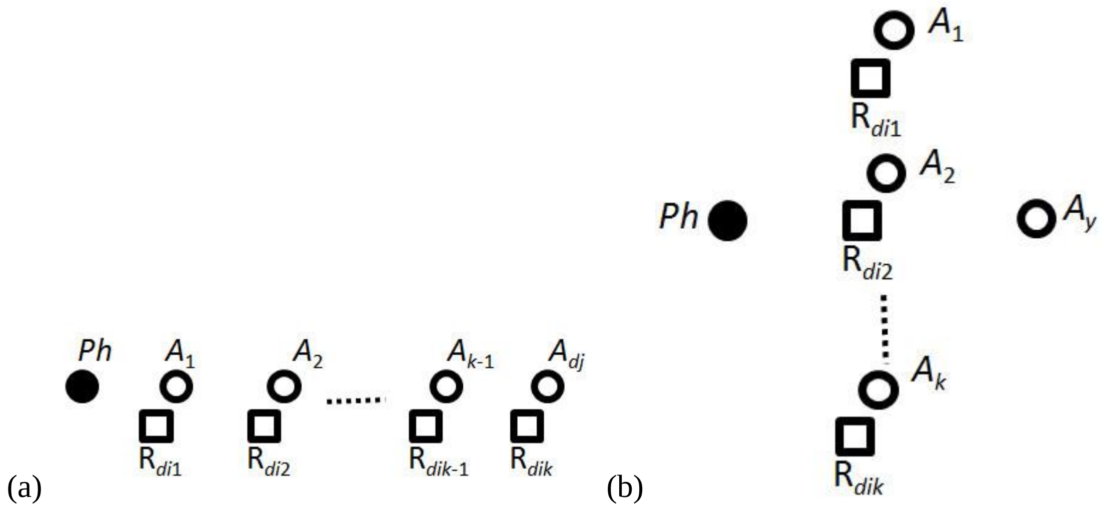


Figure 2. A Petri net model of a rule of type 3: the repellent-based control approach.



In these models, the places R_{di1}, R_{di2}, ..., R_{di3} correspond to propositions in the antecedent parts of the rules. The relationship between meaning of tokens and evaluation of propositions is shown in table 1. These places are translated into repellents in the low-level model (distribution of stimuli). The places A_1, A_2, ..., A_{k-1} correspond to auxiliary stimuli. The relationship between the meaning of tokens and the evaluation of propositions is shown in table 2. The place A_{dj} correspond to the output stimulus. The relationship between meaning of tokens and evaluation of propositions is shown in table 3. It is easy to see that, in case of type 1 of production rules, the token is present in A_{dj} (the proposition in the consequence part is true), if all places R_{di1}, R_{di2}, ..., R_{di3} do not hold tokens (the propositions in the antecedent part are true). In case of type 3 of production rules, the token is present in A_{dj} (the proposition in the consequence part is true), if at least one of the places R_{di1}, R_{di2}, ..., R_{di3} does not hold a token (at least one of the propositions in the antecedent part is true). The structures of *Physarum* machines for production rules of type 1 and 3 are shown in figure 3 (a) and (b), respectively. Distributions of stimuli can be treated as programs for these machines.

Figure 3. The structure of the *Physarum* machine for: (a) a production rule of type 1, (b) a production rule of type 3.



In the further research, we will consider more complex rule-based systems. However, we are aware of the topological constraints if the *Physarum* machine is implemented in the two-dimensional space (e.g., on the Petri dish). In this case, propagation of protoplasmic veins forming a planar graph is admissible only.

Another challenging problem is to use *Physarum* machines in the process of optimization of rule-based systems. *Physarum polycephalum* is originally famous as a computing biological substrate due to its alleged ability to approximate shortest path from its inoculation site to a source of nutrients (cf. [1]).

Acknowledgments

This research is being fulfilled by the support of FP7-ICT-2011-8.

References

[1] Adamatzky, A., *Physarum Machines: Computers from Slime Mould,* World Scientific, 2010.

[2] Adamatzky, A., Erokhin, V., Grube, M., Schubert, T., Schumann, A., *Physarum chip project: Growing computers from slime mould,* International Journal of Unconventional Computing 8(4), pp. 319–323, 2012.

[3] Agerwala, T., Flynn, M., *Comments on capabilities, limitations and 'correctness' of Petri nets,* [in:] *Proceedings of the 1st Annual Symposium on Computer Architecture (ISCA'1973),* Atlanta, USA, pp. 81–86, 1973.

[4] Brachman, R., Levesque, H., *Knowledge Representation and Reasoning,* Morgan Kaufmann Publishers, San Francisco, 2004.

[5] Chen, S.M., Ke, J.S., Chang, J.F., *Knowledge representation using fuzzy Petri nets,* IEEE Transactions on Knowledge and Data Engineering, 2(3), pp. 311–319, 1990.

[6] Craig, I., *Object-Oriented Programming Languages: Interpretation,* Springer-Verlag, London, 2007.

[7] Konar, A., *Cognitive Engineering: A Distributed Approach to Machine Intelligence,* Springer-Verlag, London, 2005.

[8] Looney, C., Alfize, A., *Logical controls via Boolean rule matrix transformations*, IEEE Transactions on Systems, Man and Cybernetics 17(6), pp. 1077–1082, 1987.

[9] Nakagaki, T., Yamada, H., Toth, A., *Maze-solving by an amoeboid organism,* Nature 407, pp. 470–470, 2000.

[10] Pancerz, K., Schumann, A., *Principles of an object-oriented programming language for Physarum polycephalum computing,* [in:] *Proceedings of the 10th International Conference on Digital Technologies (DT'2014)*, Zilina, Slovak Republic, pp. 273–280, 2014.

[11] Pancerz, K., Schumann, A., *Some issues on an object-oriented programming language for Physarum machines,* [in:] Bris, R., Majernik, J., Pancerz, K., Zaitseva, E. (eds.), *Applications of Computational Intelligence in Biomedical Technology,* Studies in Computational Intelligence, vol. 606, Springer International Publishing, Switzerland, pp. 185–199, 2016.

[12] Petri, C.A., *Kommunikation mit automaten,* Schriften des IIM, no. 2, Institut für Instrumentelle Mathematik, Bonn, 1962.

[13] Schumann, A., Pancerz, K., *Towards an object-oriented programming language for Physarum polycephalum computing,* [in:] Szczuka, M., Czaja, L., Kacprzak, M. (eds.),

Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P'2013), Warsaw, Poland, pp. 389–397, 2013.

[14] Schumann, A., Pancerz, K., *Timed transition system models for programming Physarum machines: Extended abstract*, [in:] Popova-Zeugmann, L. (ed.), *Proceedings of the Workshop on Concurrency*, Specification and Programming (CS&P'2014), Chemnitz, Germany, pp. 180–183, 2014.

[15] Schumann, A., Pancerz, K., *Towards an object-oriented programming language for Physarum polycephalum computing: A Petri net model approach. Fundamenta Informaticae 133(2-3)*, pp. 271–285, 2014.

[16] Schumann, A., Pancerz, K., *PhysarumSoft - a software tool for programming Physarum machines and simulating Physarum games*, [in:] *Proceedings of the FedCSIS 2015*, Lodz, Poland, pp. 619–626, 2015.

[17] Schumann, A., Pancerz, K., Jones, J., *Towards logic circuits based on Physarum polycephalum machines: The ladder diagram approach*, [in:] Cliquet Jr., Plantier, G., Schultz, T., Fred, A., Gamboa, H. (eds.), *Proceedings of the International Conference on Biomedical Electronics and Devices (BIODEVICES'2014)*, Angers, France, pp. 165–170, 2014.

[18] Suraj, Z., *A survey of selected classes of logical Petri nets*, [in:] Wróbel, Z., Marszał-Paszek, B., Paszek, P. (eds.), *Monografia jubileuszowa dla uczczenia 45 lat pracy naukowej Prof. A. Wakulicz-Deji*, Wyd. Uniwersytetu Śląskiego, Katowice, pp. 157–181, 2013.

[19] Szpyrka, M., Szmuc, T., *D-nets - Petri net form of rule-based systems. Foundations of Computing and Decision Sciences 31(2)*, pp. 175–190, 2006.

[20] Verbeek, H., Wynn, M., van der Aalst, W., ter Hofstede, A., *Reduction rules for reset/inhibitor nets*, Journal of Computer and System Sciences 76(2), pp. 125–143, 2010.

Originally published: Schumann, A., *Go Games on Plasmodia of Physarum Polycephalum*, *Annals of Computer Science and Information Systems*, [in:] Proceedings of the 2015 Federated Conference on Computer Science and Information Systems, IEEE Xplore, 2015.

XIV
Go Games on Plasmodia of *Physarum Polycephalum*

Andrew Schumann

Abstract

We simulate the motions of *Physarum polycephalum* plasmodium by the game of Go, the board game originated in ancient China more than 2,500 years ago. Then we concentrate just on Go games, where locations of black and white stones simulate syllogistic reasoning, in particular reasoning of Aristotelian syllogistic and reasoning of performative syllogistic. For the first kind of reasoning we need a special form of coalition games. For the second kind of reasoning we appeal to usual antagonistic games.

Introduction

In the *Physarum Chip Project: Growing Computers From Slime Mould* [1] we are working on designing a biological computer, where logic circuits are represented by programmable behaviors of *Physarum polycephalum* plasmodium, the one-cell organism that behaves according to different chemical stimuli called attractants and repellents and propagates networks connecting all reachable food attractants [2], [12]. The behavior of plasmodia is very sensitive and intelligent [4], [5], [6], [7], [8], [15], [17], [18], [19]. This behavior can be represented as a bio-inspired game theory on plasmodia [14], i.e. an experimental game theory, where, on the one hand, all basic definitions are verified in the experiments with *Physarum polycephalum* and *Badhamia utricularis* and, on the other hand, all basic algorithms are implemented in the object-oriented language for simulations of plasmodia [13].

We show that the slime mold can be a model for concurrent games and context-based games defined in [11]. In context-based games, players can move concurrently as well as in concurrent games, but the set of actions is ever infinite. In our experiments, we follow the following interpretations of basic entities: (1) attractants as payoffs; (2) attractants occupied by the plasmodium as states of the game; (3) active zones of plasmodium as players; (4) logic gates for behaviors as moves (available actions) for the players; (5) propagation of the plasmodium as the transition table which associates, with a given set of states and a given move of the players, the set of states resulting from that move.

In this game theory we can demonstrate creativity of primitive biological substrates of plasmodia. The point is that plasmodia do not strictly follow spatial algorithms like Kolmogorov-Uspensky machines, but perform many additional actions. So, the plasmodium behavior can be formalized within strong extensions of spatial algorithms, e.g. within concurrent games or context-based games [11].

In this paper we show how we can represent the plasmodium behavior as a *Go game*. It is board game with two players (called Black and White) who alternately place black and white stones, accordingly, on the vacant intersections (called points) of a board with a 19x19 grid of lines. Black moves first. Stones are placed until they reach a point where stones of another color are located. There are the following two basic rules of the game: (1) each stone must have at least one open point (called liberty) directly next to it (up, down, left, or right), or must be part of a connected group that has at least one such open point; stones which lose their last liberty are removed from the board; (2) the stones must never repeat a previous position of stones. The aim of the game is in surrounding more empty points by player's stones. At the end of game, the number of empty points player's stones surround are counted, together with the number of stones the player captured. This number determines who the winner is. The Go game originated in ancient China more than 2,500 years ago and it is very popular till now.

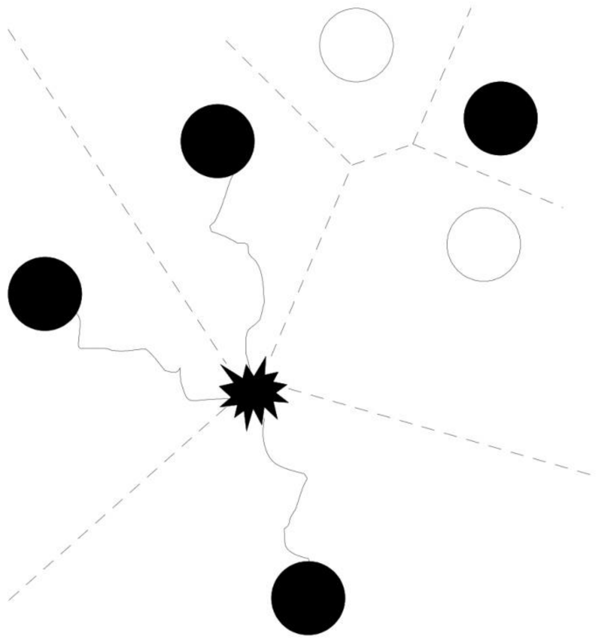
We can consider the game of Go as a model of plasmodium motions. In this view the black stones are considered attractants occupied by the plasmodium and the white stones are regarded as repellents. By this interpretation, we have two players, also: Black (this player places attractants) and White (this player places repellents). The winner is determined by the number of empty points player's stones surround.

Notice that the number of possible Go games is too large, 10^{761}. Therefore it is better to focus just on games, where locations of black and white stones simulate spatial reasoning. In this paper we propose two logics in the universe of possible Go games: (1) Aristotelian syllogistic [3]; (2) performative syllogistic [9], [10].

14.1. Classical game of Go on plasmodia of *Physarum polycephalum*

The plasmodium of *Physarum polycephalum* moves to attractants to connect them and in the meanwhile it avoids places, where repellents are located. The radius, where chemical signals from attractants (repellents) can be detected by the plasmodium to attract (repel) the latter, determines the structure of natural *Voronoi cells*, where each Voronoi cell is a place, where a chemical signal holds (see Fig. 1).

Figure 1. The six Voronoi cells in accordance with the four attractants denoted by the black stones and with the two repellents denoted by the white stones. The plasmodium located in the center of the picture connects the three attractants by the three protoplasmic tubes. It cannot see the fourth attractant because of the two repellents



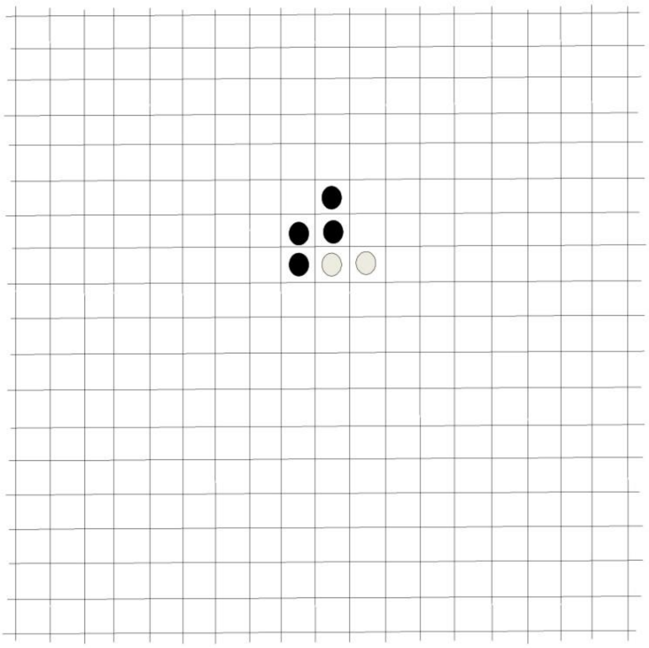
For the plasodium of fig. 1 we have just the four neighbor cells. Notably that in Go games at each point we have only four neighbors everywhere. So, we can design the space for plasmodia in the way to have just four neighbors at each point. So, the cells of a Go game board are considered Voronoi cells with the same radius of intensity and power of attractants and repellents located in these cells. We associate black stones with attractants occupied by plasmodia and white stones with repellents. For the sake of convenience and more analogy with Voronoi cells, let us consider cells (not intersections of lines) as points for stone locations (see Fig. 2). Then we can use all rules of Go games to simulate plasmodium motions.

Thus, our Go game is represented in the universe of 18x18 Voronoi cells (see Fig. 2).

14.2. Aristotelian Go game on plasmodia

Let us assume that the game of Go is a coalitional game with two players who choose only strategies to place black and white stones so that their locations can be interpreted as a spatial Aristotelian syllogistic reasoning.

Figure 2. The Go game board with two white stones designating repellents and four black stones designating attractants.



Let us recall that the axiomatization of the Aristotelian syllogistic was laid first by Łukasiewicz [3]. In his axiomatization, the alphabet consists of the syllogistic letters *S*, *P*, *M*, …, the syllogistic connectives *a*, *e*, *i*, *o*, and the propositional connectives $\neg$, $\vee$, $\wedge$, $\Rightarrow$. Atomic propositions are defined as follows: *SxP*, where $x \in \{a, e, i, o\}$. All other propositions are defined in the following way: (i) each atomic proposition is a proposition, (ii) if *X*, *Y* are propositions, then $\neg X$, $\neg Y$, *X* å *Y*, where å $\in \{\vee, \wedge, \Rightarrow\}$, are propositions, also. The axioms proposed by Łukasiewicz are as follows:

$$SaP := (\exists A(A \text{ is } S) \wedge \forall A(A \text{ is } S \Rightarrow A \text{ is } P)); \tag{1}$$

$$SiP := \exists A(A \text{ is } S \wedge A \text{ is } P); \tag{2}$$

$$SeP := \neg(SiP); \tag{3}$$

$$SoP := \neg(SaP); \tag{4}$$

$$SaS; \tag{5}$$

$$SiS; \tag{6}$$

$$(MaP \wedge SaM) \Rightarrow SaP; \quad (7)$$

$$(MaP \wedge MiS) \Rightarrow SiP. \quad (8)$$

In the Go implementation of Aristotelian syllogistic, the syllogistic letters *S*, *P*, *M*, … are interpreted as single cells of the board with the 18x18 Voronoi cells. The letter *S* is understood as empty if and only if the white stone is located on an appropriate cell denoted by *S*. Let us recall that in our plasmodium interpretation of Go games white stones mean ever repellents so that their location in a Voronoi cell means that this cell cannot be occupied by plasmodia. This letter *S* is treated as non-empty if and only if the black stone is located on an appropriate cell denoted by *S*. Recall that black stones mean ever attractants so that their location in a Voronoi cell means that this cell is occupied by plasmodia. If a cell does not contain any stone, this means that this cell is out of the game.

Hence, in the *Physarum* interpretation of this Go game, the non-empty syllogistic letters *S*, *P*, *M*, …, i.e. the cells denoted by *S*, *P*, *M*, … containing black stones are considered attractants and the empty syllogistic letters *S*, *P*, *M*, …, i.e. the cells denoted by *S*, *P*, *M*, … containing white stones are considered repellents. So, a data point *S* is regarded as non-empty if and only if an appropriate attractant located in *S* is occupied by plasmodium. This data point *S* is regarded as empty if and only if an appropriate repellent located in *S* repell plasmodium.

Thus, in the Aristotelian version of the Go game we have syllogistic strings of the form *SP* with the following interpretation: '*S* is *P*', and with the following meaning: *SP* is true if and only if *S* and *P* are neighbors and both *S* and *P* are not empty, otherwise *SP* is false. We can extend this meaning as follows: *SP* is true if and only if *S* and *P* are not empty and there is a line of non-empty cells between points *S* and *P*, otherwise *SP* is false. By the definition of true syllogistic strings, we can define atomic syllogistic propositions as follows:

SaP

In the formal syllogistic: there exists *A* such that *A* is *S* and for any *A*, if *A* is *S*, then *A* is *P*. *In the Go game model*: there is a cell *A* containing the black stone and for any *A*, if *AS* is true, then *AP* is true. *In the Physarum model*: there is a plasmodium in the cell *A* and for any *A*, if *AS* is true, then *AP* is true.

SiP

In the formal syllogistic: there exists *A* such that both *AS* is true and *AP* is true. *In the Go game model*: there exists a cell *A* containing the black stone such that *AS* is true and *AP* is true. *In the Physarum model*: there exists a plasmodium in the cell *A* such that *AS* is true and *AP* is true.

SeP

In the formal syllogistic: for all *A*, *AS* is false or *AP* is false. *In the Go game model*: for all cells *A* containing the black stones, *AS* is false or *AP* is false. *In the Physarum model*: for all plasmodia *A*, *AS* is false or *AP* is false.

SoP

In the formal syllogistic: for any *A*, *AS* is false or there exists *A* such that *AS* is true and *AP* is false. *In the Go game model*: for all cells *A* containing the black stones, *AS* is false or there exists A such that AS is true and AP is false. *In the Physarum model*: for any plasmodia A, AS is false or there exists A such that AS is true and AP is false.

Formally, this semantics is defined as follows. Let M be a set of attractants. Take a subset $|X| \subseteq M$ of cells containing the black stones (i.e. of cells containing attractants and occupied by the plasmodium) as a meaning for each syllogistic variable X. Next, define an ordering relation $\subseteq$ on subsets $|S|, |P| \subseteq M$ as: $|S| \subseteq |P|$ iff all attractants from $|P|$ are reachable for the plasmodium located at the attractants from $|S|$, i.e. iff for all cells of $|S|$ with black stones there are lines of black stones connecting them to cells of $|P|$ also containing black stones. Hence, $|S| \cap |P| \neq \varnothing$ means that some attractants from $|P|$ are reachable for the plasmodium located at the attractants from $|S|$ and $|S| \cap |P| = \varnothing$ means that no attractants from $|P|$ are reachable for the plasmodium located at the attractants from $|S|$. In the Go game model $|S| \cap |P| \neq \varnothing$ means that some cells from $|P|$ occupied by the black stones are connected by the lines of black stones with the cells from $|S|$ occupied by the black stones and $|S| \cap |P| = \varnothing$ means that there are no lines of black stones from the cells of $|P|$ to the cells of $|S|$.

This gives rise to models $\mathsf{M} = \langle M, |\cdot| \rangle$ such that

$\mathsf{M} \models SaP$ iff $|S| \subseteq |P|$;

$\mathsf{M} \models SiP$ iff $|S| \cap |P| \neq \varnothing$;

$\mathsf{M} \models SeP$ iff $|S| \cap |P| = \varnothing$;

$\mathsf{M} \models p \wedge q$ iff $\mathsf{M} \models p$ and $\mathsf{M} \models q$;

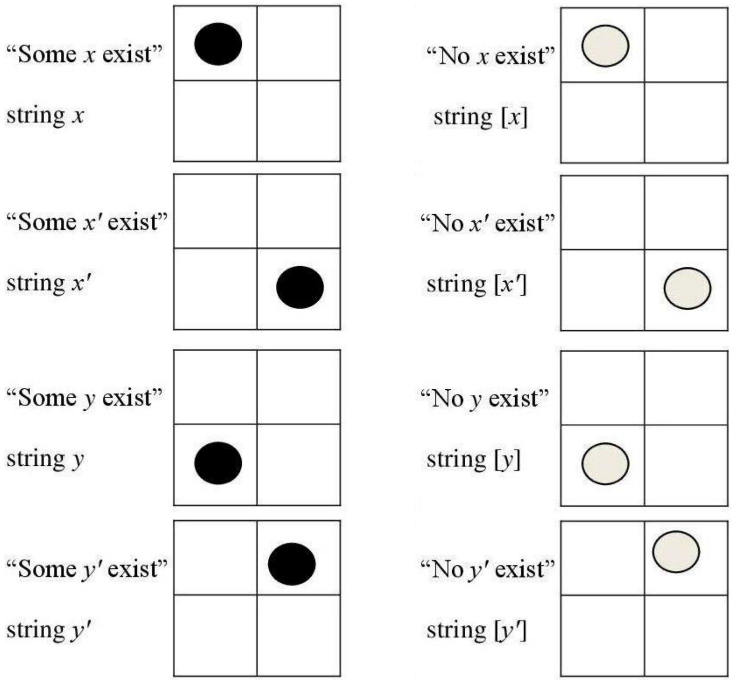
$\mathsf{M} \models p \vee q$ iff $\mathsf{M} \models p$ or $\mathsf{M} \models q$;

$\mathsf{M} \models \neg p$ iff it is false that $\mathsf{M} \models p$.

Proposition 1. *The Aristotelian syllogistic is sound and complete relatively to* M *if we understand* $\subseteq$ *as an inclusion relation (it is a well-known result [16]).*

However, relatively to all possible Go games (plasmodium behaviors) the Aristotelian syllogistic is not complete. Indeed, the relation $\subseteq$ can have the following verification on the *Aristotelian Go game model on plasmodia* according to our definitions: $|S| \subseteq |P|$ and $|S| \subseteq |P'|$, where $|P| \cap |P'| = \varnothing$, i.e. all attractants from $|P|$ are reachable for the plasmodium located at the attractants from $|S|$ and all attractants from $|P'|$ are reachable for the plasmodium located at the attractants from $|S|$, but between $|P|$ and $|P'|$ there are no paths. In this case $\subseteq$ is not an inclusion relation and proposition 1 does not hold. Hence, we need repellents to make $\subseteq$ the inclusion relations in all cases. Therefore, to obtain the Aristotelian Go game model on plasmodia we shall deal with the coalitional game, where two players will cooperate to build spatial reasoning satisfying the rules of Aristotelian syllogistic. The first player places black stones to designate places of growing plasmodia. The second player places white stones to designate places of repelling plasmodia. So, both players follow coalitional strategies to simulate Aristotelian syllogistic reasoning.

Figure 3. The Aristotelian Go game diagrams for the basic existence strings.



14.2.1. Coalition Go game for verifying Aristotelian reasoning

In the Aristotelian Go game model for verifying all the basic syllogistic propositions, we will use the following four cells: x, y, x', y' of the game board with the 19x19 grid of lines, where x' means all cells which differ from x, but they are neighbors for y, and y' means all cells which differ from y and are neighbors for x. These cells express appropriate meanings of syllogistic letters. The corresponding universe of discourse will be denoted by means of the following diagram:

x	y'
y	x'

Assume that a black stone denotes an attractant and if it is placed within a cell x, this means that “this Voronoi cell contains an attractant N_x activated and occupied by the plasmodium”. It is a verification of the syllogistic letter S_x at cell x of the board. A white stone denotes a repellent and if it is placed within a cell x, this means that “this Voronoi cell contains a repellent R_x activated and there is no plasmodium in it”. It is a verification of a new syllogistic letter $[S_x]$. For the sake of convenience, we will denote S_x by x and $[S_x]$ by $[x]$. Using these stones, we can verify all the basic existence syllogistic propositions (see Fig. 3).

Aristotelian Go game strings of the form xy, yx are interpreted as particular affirmative propositions “Some x are y” and “Some y are x” respectively, strings of the form $[xy]$, $[yx]$, $x[y]$, $y[x]$ are interpreted as universal negative propositions “No x are y” and “No y are x”. A universal affirmative proposition “All x are y” are presented by a complex string $xy \,\&\, x[y']$. The sign & means that we have strings xy and $x[y']$ simultaneously and they are considered the one complex string. All these strings are verified on the basis of the diagrams of Fig. 4. So, we use only black stones for building particular propositions, only white stones for building universal negative propositions, and we combine black and white stones for building universal affirmative propositions. Consequently, we need a cooperation of two players to implement a spatial version of Aristotelian syllogistic within coalition Go games.

For verifying syllogisms we will use the following diagrams symbolizing some neighbor cells:

	m	m'	
m'	x	y'	m
m	y	x'	m'
	m'	m	

The motion of plasmodium starts from one of the central cells (x, y, x', y') and goes towards one of the four directions (northwest, southwest, northeast, southeast). The syllogism shows a connection between two not-neighbor cells on the basis of its joint neighbor and says if there was either multiplication or fusion of plasmodia (i.e. either splitting or fusion of the lines of black stones). As a syllogistic conclusion, we obtain another diagram:

x	m'
m	x'

Different syllogistic conclusions derived show directions of plasmodium's propagation. Some examples are provided in fig. 5 – 7.

Figure 4. The Aristotelian Go game diagrams for syllogistic propositions.

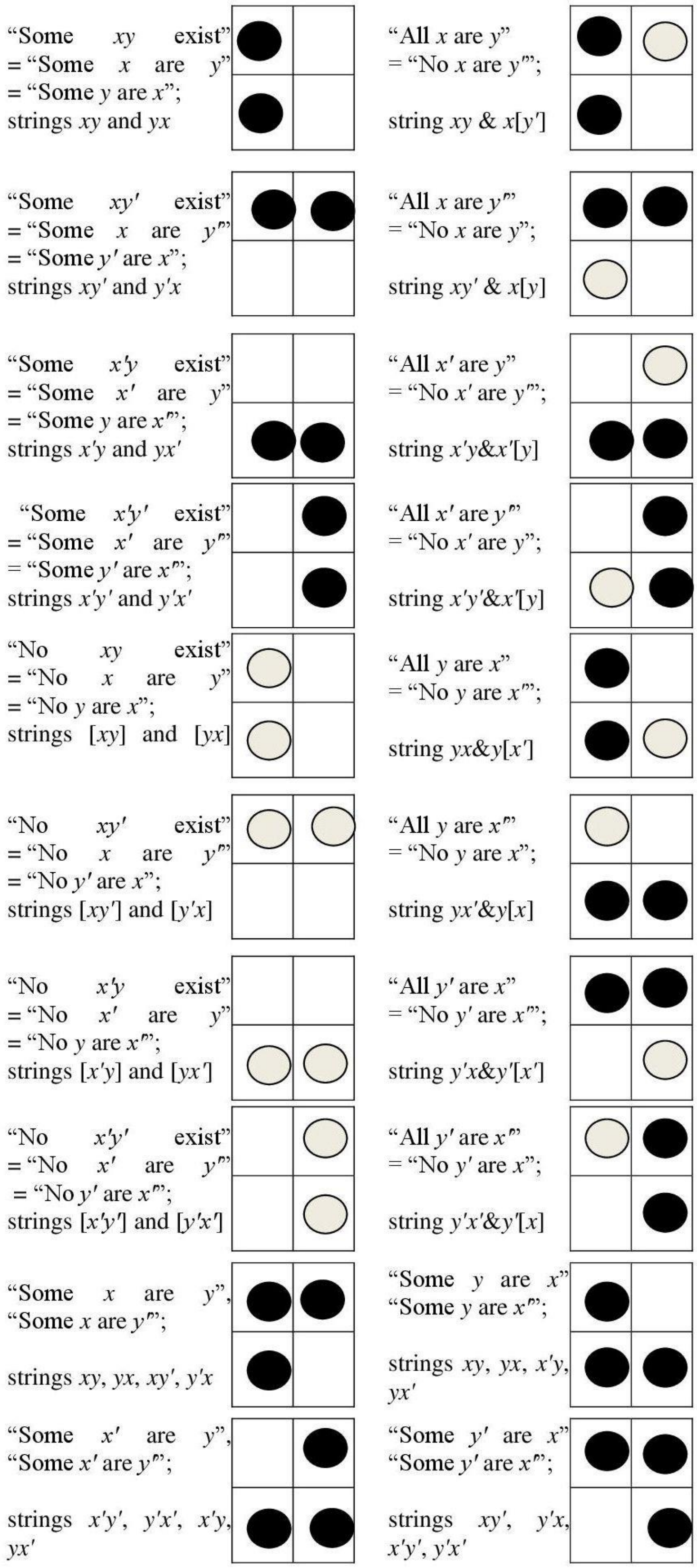
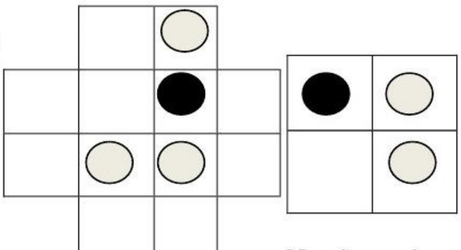


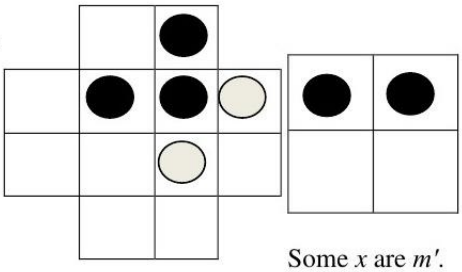
Figure 5. The Aristotelian Go game diagrams for syllogisms (part 1).

1. No y are x';
All y' are m.



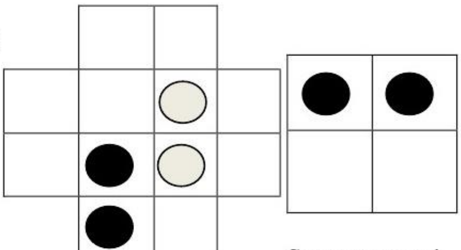
No x' are m'.

2. All y' are x;
All y' are m'.



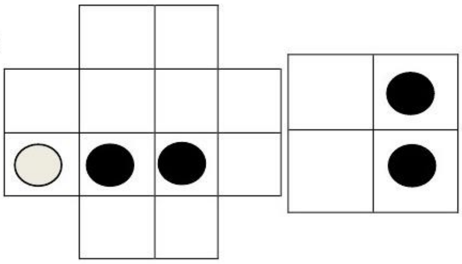
Some x are m'.

3. No x' are y';
Some y are m'.



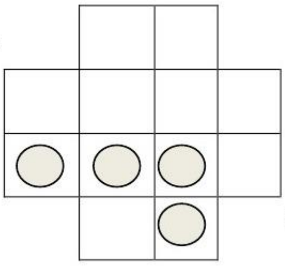
Some x are m'.

5. Some y are x';
No m are y.



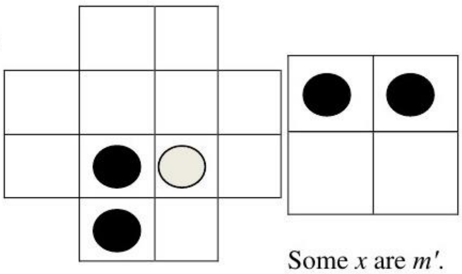
Some x' are m'.

6. No x' are m;
No m are y.



There is no conclus

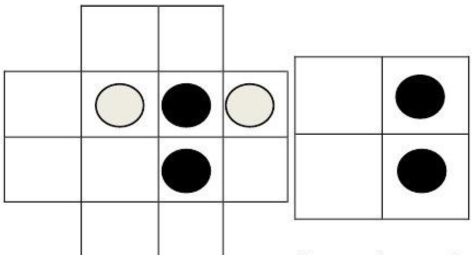
7. No y are x';
Some m' are y.



Some x are m'.

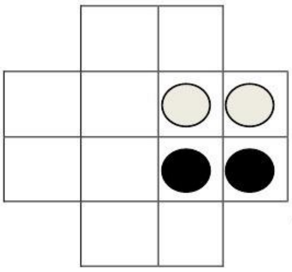
Figure 6. The Aristotelian Go game diagrams for syllogisms (part 2).

8. All y' are x';
No y' are m.



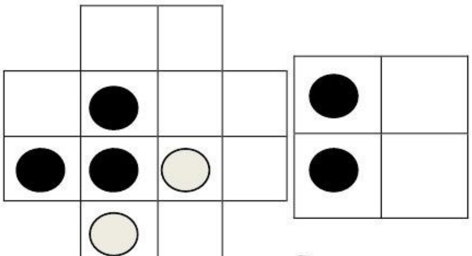
Some x' are m'.

9. Some x' are m';
No m are y'.



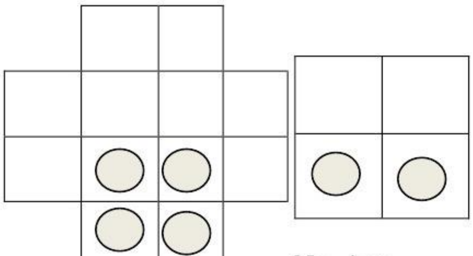
There is no conclusion.

10. All y are x;
All y are m.



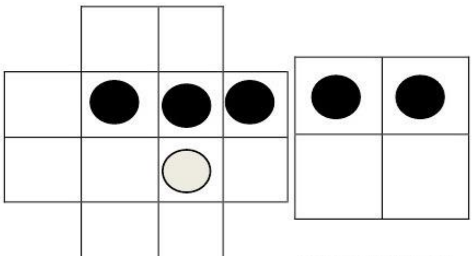
Some m are x.

11. No x' are m;
No m' are y.



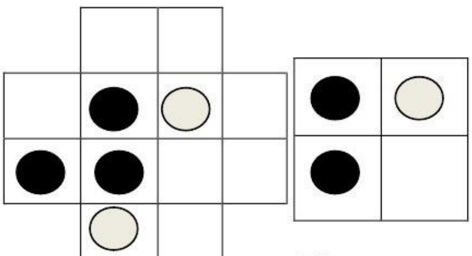
No x' are m.

12. All y' are x;
Some m are y'.



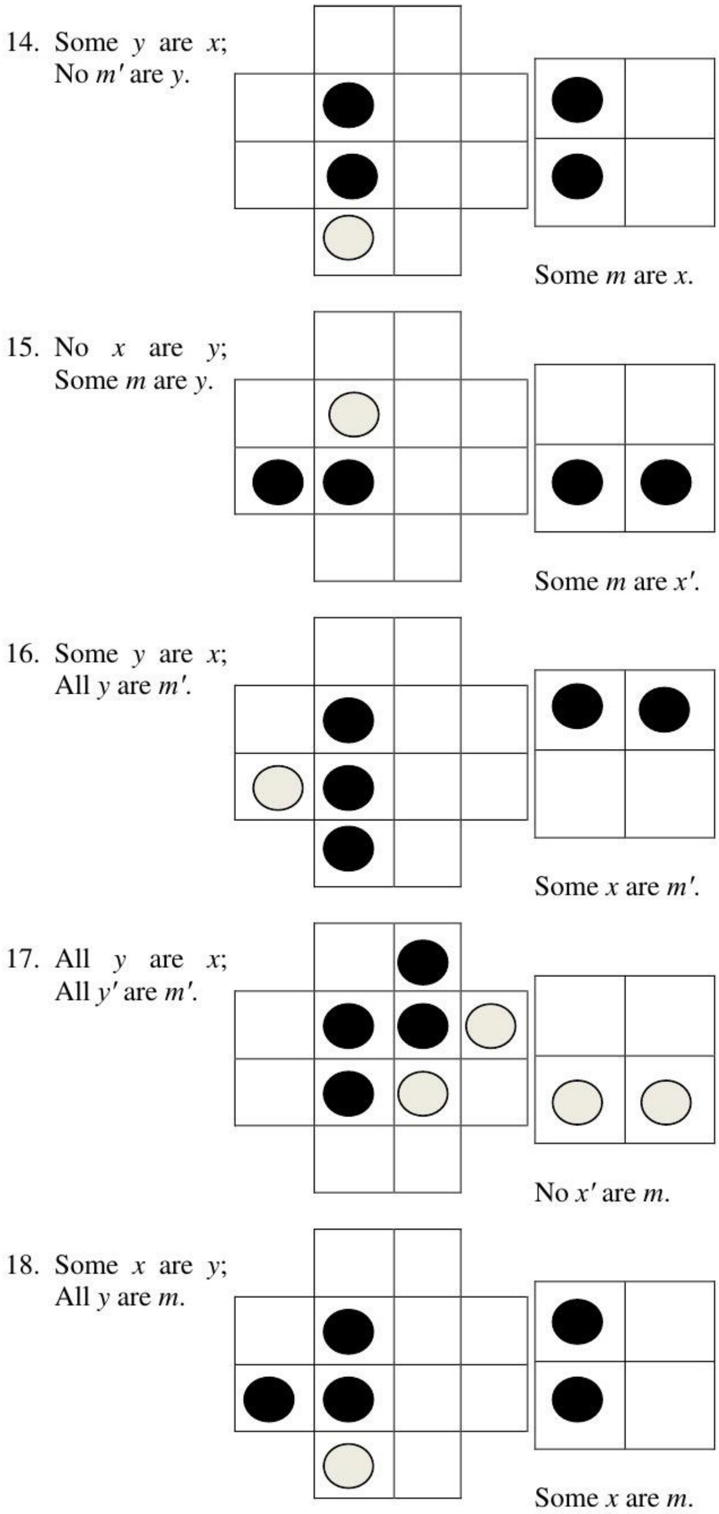
Some x are m'.

13. All y are m;
All x are y.



All x are m.

Figure 7. The Aristotelian Go game diagrams for syllogisms (part 3).



Continuing in the same way, we can construct a syllogistic system, where conclusions are derived from three premises. The Aristotelian Go game (i.e. the suitable motion of plasmodium) starts from one of the central cells (x, y, x', y') and goes towards one of the four directions (northwest, southwest, northeast, southeast), then towards one of the eight directions (north-northwest, west-northwest, south-southwest, west-southwest, north-northeast, east-northeast, south-southeast, east-southeast), etc.

Hence, a Go game or spatial expansion of plasmodium is interpreted as a set of syllogistic propositions. The universal affirmative proposition xy & $x[y']$ means that the plasmodium at the place x goes only to y and all other directions are excluded. The universal negative proposition $x[y]$ or $[xy]$ means that the plasmodium at the place x cannot go to y and we know nothing about other directions. The particular affirmative proposition xy means that the plasmodium at the place x goes to y and we know nothing about other directions. Syllogistic conclusions allow us to mentally reduce the number of syllogistic propositions showing plasmodium's propagation.

For the implementation of Aristotelian syllogistic we appeal to repellents to delete some possibilities in the plasmodium propagation. So, model M defined above should be understood as follows:

$\mathsf{M} \models$ *All x are y* iff xy & $x[y']$, i.e. the plasmodium is located at x and can move only to y and cannot move towards all other directions (the black stone is placed at x and we can build the line of black stones only to y);

$\mathsf{M} \models$ *Some x are y* iff xy, i.e. the plasmodium is located at x and can move to y (the black stone is placed at x and we can build the line of black stones to y);

$\mathsf{M} \models$ *No x are y* iff $x[y]$ or $[xy]$, i.e. the plasmodium cannot move to y in any case (there is no line of black stones to y).

It is evident in this formulation that the Aristotelian syllogistic is so unnatural for plasmodia. Without repellents (the coalition game of two players), this syllogistic system cannot be verified in the medium of plasmodium propagations (Go game). In other words, we can prove the next proposition:

Proposition 2. *The Aristotelian syllogistic is not sound and complete on the plasmodium without repellents. In other words, the Aristotelian syllogistic is not sound and complete in the Go game without a coalition of two players.*

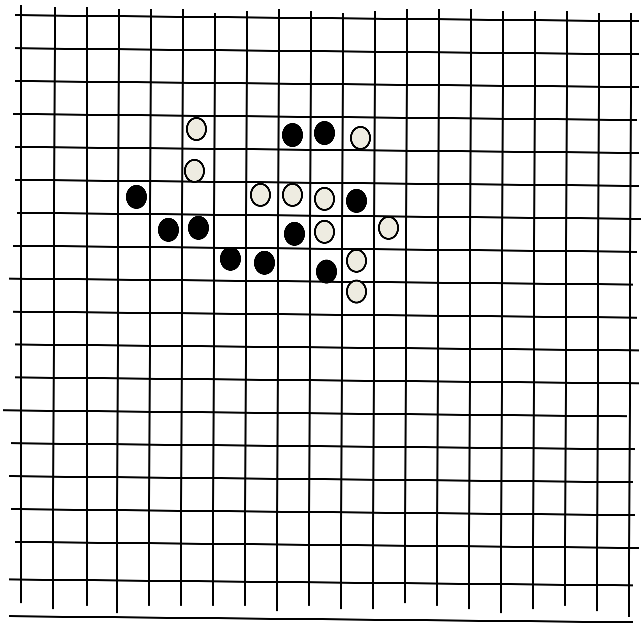
In other words, the Aristotelian syllogistic reasoning can be implemented as a Go game if and only if two players agree to play cooperatively to place black and white stones in accordance with spatial implementation of syllogisms.

14.2.2. Examples of Aristotelian Go game

Let us consider a game of Go at time step 10, i.e. when White and Black players have placed the 10 white stones and the 10 black stones respectively. Let this game be pictured in figure 8. Each Voronoi cell is denoted from $S_{1,1}$ to $S_{18,18}$. So, in figure 8 syllogistic letters $S_{6,4}$, $S_{7,5}$, $S_{7,6}$, $S_{8,7}$, $S_{8,8}$, $S_{7,9}$, $S_{8,10}$, $S_{6,11}$, $S_{4,9}$, $S_{4,10}$ are understood as non-empty and syllogistic letters $S_{4,6}$, $S_{5,6}$, $S_{6,8}$, $S_{6,9}$, $S_{6,10}$, $S_{4,11}$, $S_{7,10}$, $S_{7,12}$, $S_{11,8}$, $S_{12,8}$ as empty. As a result, we can build some true syllogistic propositions in this universe like that: 'Some $S_{7,5}$ are $S_{7,6}$', 'Some $S_{8,7}$ are $S_{8,8}$', 'Some $S_{4,9}$ are $S_{4,10}$', 'No $S_{4,6}$ are $S_{5,6}$', 'No $S_{6,8}$ are $S_{6,9}$', 'No $S_{6,9}$ are $S_{6,10}$', 'No $S_{6,10}$ are $S_{7,10}$', 'No $S_{11,8}$ are $S_{12,8}$', etc.

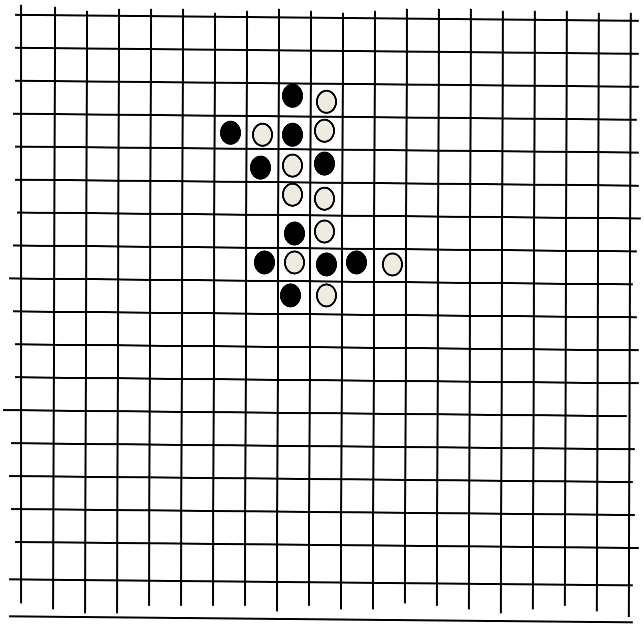
Let us notice that in the universe of fig. 8 we do not have universal affirmative propositions. But we can draw some syllogistic conclusions such as 'If $S_{4,10}[S_{4,11}]$ and $S_{4,9}$ $S_{4,10}$, then $S_{4,9}[S_{4,11}]$' (i.e. 'If no $S_{4,10}$ are $S_{4,11}$ and some $S_{4,9}$ are $S_{4,10}$, then no $S_{4,9}$ are $S_{4,11}$').

Figure 8. The Aristotelian Go game 1 at time step 10.



On the contrary, in the universe pictured in the Go game of fig. 9 we have universal affirmative propositions such as 'All $S_{8,10}$ are $S_{8,11}$' and 'All $S_{4,9}$ are $S_{3,9}$'. Some possible conclusions: 'If no $S_{8,11}$ are $S_{8,12}$ and all $S_{8,10}$ are $S_{8,11}$, then no $S_{8,10}$ are $S_{8,12}$' and 'If no $S_{3,9}$ are $S_{3,10}$ and all $S_{4,9}$ are $S_{3,9}$, then no $S_{4,9}$ are $S_{3,10}$'.

Figure 9. The Aristotelian Go game 2 at time step 10.



14.3. Non-Aristotelian syllogistic Go game on plasmodia

While in Aristotelian syllogisms we are concentrating on one direction of many *Physarum* motions, and dealing with acyclic directed graphs with fusions of many protoplasmic tubes toward one data point, in most cases of *Physarum* behavior, not limited by repellents, we observe a spatial expansion of *Physarum* protoplasm in all directions with many cycles. Under these circumstances it is more natural to define all the basic syllogistic propositions *SaP*, *SiP*, *SeP*, *SoP* in a way they satisfies the inverse relationship when all converses are valid: *SaP* $\Rightarrow$ *PaS*, *SiP* $\Rightarrow$ *PiS*, *SeP* $\Rightarrow$ *PeS*, *SoP* $\Rightarrow$ *PoS*. In other words, we can draw more natural conclusions for protoplasmic tubes which are decentralized and have some cycles. The formal syllogistic system over propositions with such properties is constructed in [9], [10]. This system is called the *performative syllogistic*. The alphabet of this system contains as descriptive signs the syllogistic letters *S*, *P*, *M*, …, as logical-semantic signs the syllogistic connectives *a*, *e*, *i*, *o*, and the propositional connectives $\neg$, $\vee$, $\wedge$, $\Rightarrow$. Atomic propositions are defined as follows: *SxP*, where $x \in \{a, e, i, o\}$. All other propositions are defined thus: (i) each atomic proposition is a proposition, (ii) if *X*, *Y* are propositions, then $\neg X$, $\neg Y$, *X* å *Y*, where å $\in \{\vee, \wedge, \Rightarrow\}$, are propositions, too.

Let us consider Go games with the two different kinds of plasmodia: (i) plasmodia of *Physarum polycephalum* and (ii) plasmodia of *Badhamia utricularis* [14]. They try to occupy free attractants antagonistically. So, if an attractant is occupied by the plasmodium of *Physarum polycephalum*, it cannot be occupied by the plasmodium of *Badhamia utricularis* and if it is occupied by the plasmodium of *Badhamia utricularis,* it cannot be occupied by the plasmodium of *Physarum polycephalum*. In this way we observe a competition between two plasmodia.

In order to implement the performative syllogistic in the Go games with *Physarum polycephalum* and *Badhamia utricularis* plasmodia, we will interpret data points denoted by appropriate syllogistic letters as black stones (attractants) if we assume that appropriate cells are occupied by the plasmodium of *Physarum polycephalum* and we will interpret data points denoted by appropriate syllogistic letters as white stones (attractants) if we assume that appropriate cells are occupied by the plasmodium of *Badhamia utricularis*. A data point *S* is considered empty for the Black player if and only if an appropriate attractant denoted by *S* is occupied by the white stone (plasmodium of *Badhamia utricularis*). A data point *S* is considered empty for the White player if and only if an appropriate attractant denoted by *S* is occupied by the black stone (plasmodium of *Physarum polycephalum*). Let us define syllogistic strings of the form *SP* with the following interpretation: (i) '*S* is *P*': *SP* is true for the Black player if and only if *S* and *P* are reachable for each other by the plasmodium of *Physarum polycephalum* and both *S* and *P* are not empty for the Black player, otherwise *SP* is false; (ii) '*S* is *P*': *SP* is true for the White player if and only if *S* and *P* are reachable for each other by the plasmodium of *Badhamia utricularis* and both *S* and *P* are not empty for the White player, otherwise *SP* is false. In other words, *SP* is true for the Black player (respectively, for the White player) if and only if *S* and *P* are not empty for the Black player (respectively, for the White player) and there is a line of non-empty cells for the Black player (respectively, for the White player) between points *S* and *P*, otherwise *SP* is false. Using this definition of syllogistic strings, we can define atomic syllogistic propositions as follows:

SaP

In the formal performative syllogistic: there exists *A* such that *A* is *S* and for any *A*, *AS* is true and *AP* is true. *In the Go game model:* there is a black (white) stone in *A* connected by black (white) stones to *S* and connected by black (white) stones to *P*. *In the Physarum model*: a plasmodium of *Physarum polycephalum* (a plasmodium of *Badhamia utricularis*) in *A* occupies *S* and for any plasmodia *A* of *Physarum polycephalum* (for any plasmodia *A* of *Badhamia utricularis*) which is a neighbor for *S* and *P*, there are strings *AS* and *AP*. This means that we have a massive-parallel occupation of the region by plasmodia of *Physarum polycephalum* (plasmodia of *Badhamia utricularis*) where the cells *S* and *P* are located.

SiP

In the formal performative syllogistic: for any *A*, both *AS* is false and *AP* is false. *In the Go game model*: for any cell *A* there are no lines of black (white) stones connecting *A* to *S* and *A* to *P*. *In the Physarum model*: for any plasmodium of *Physarum polycephalum* (of *Badhamia utricularis*) *A* which is a neighbor for *S* and *P*, there are no strings *AS* and *AP*. This means that the plasmodium of *Physarum polycephalum* (of *Badhamia utricularis*) cannot reach *S* from *P* or *P* from *S* immediately.

SeP

In the formal performative syllogistic: there exists *A* such that if *AS* is false, then *AP* is true. *In the Go game model*: there exists a cell *A* with the black (white) stone which is a neighbor for cells *S* and *P* such that there is a string *AS* or there is a string *AP*. *In the Physarum model*: there exists the plasmodium of *Physarum polycephalum* (of *Badhamia utricularis*) *A* which is a neighbor for *S* and *P* such that there is a string *AS* or there is a string *AP*. This means that the plasmodium of *Physarum polycephalum* (of *Badhamia utricularis*) occupies *S* or *P*, but not the whole region where the cells *S* and *P* are located.

SoP

In the formal performative syllogistic: for any *A*, *AS* is false or there exists *A* such that *AS* is false or *AP* is false. *In the Go game model*: for any cell *A* with the black (white) stone which is a neighbor for *S* and *P* there is no string *AS* or there exists a black (white) stone in *A* which is a neighbor for *S* and *P* such that there is no string *AS* or there is no string *AP*. *In the Physarum model*: for any plasmodium of *Physarum polycephalum* (of *Badhamia utricularis*) *A* which is a neighbor for *S* and *P* there is no string *AS* or there exists *A* which is a neighbor for *S* and *P* such that there is no string *AS* or there is no string *AP*. This means that the plasmodium of *Physarum polycephalum* (of *Badhamia utricularis*) does not occupy *S* or there is a neighboring cell which is not connected to *S* or *P* by a protoplasmic tube.

Notice that there are the following semantic correlations between propositions in the sense of the Black player and propositions in the sense of the White player:

- *SaP* is true for the Black player iff *SiP* is true for the White player with the same cells *S* and *P*;
- *SoP* is true for the Black player iff *SeP* is true for the White player with the same cells *S* and *P*;
- *SaP* is false for the Black player iff *SiP* is false for the White player with the same cells *S* and *P*;
- *SoP* is false for the Black player iff *SeP* is false for the White player with the same cells *S* and *P*.

Composite propositions are defined in the standard way.

In the performative syllogistic we have the following axioms:

$$SaP := (\exists A(A \text{is} S) \wedge (\forall A(A \text{is} S \wedge A \text{is} P))); \tag{9}$$

$$SiP := \forall A(\neg(A \text{is} S) \wedge \neg(A \text{is} P)); \tag{10}$$

$$\begin{aligned} &SoP := \neg(\exists A(A \text{is} S) \vee (\forall A(A \text{is} P \wedge A \text{is} S))), i.e. \\ &(\forall A \neg(A \text{is} S) \wedge \exists A(\neg(A \text{is} P) \vee \neg(A \text{is} S))); \end{aligned} \tag{11}$$

$$\begin{aligned} &SeP := \neg \forall A(\neg(A \text{is} S) \wedge \neg(A \text{is} P)), i.e. \\ &\exists A(A \text{is} S \vee A \text{is} P). \end{aligned} \tag{12}$$

$$SaP \Rightarrow SeP; \tag{13}$$

$$SaP \Rightarrow PaS; \tag{14}$$

$$SiP \Rightarrow PiS; \tag{15}$$

$$SaM \Rightarrow SeP; \tag{16}$$

$$MaP \Rightarrow SeP; \tag{17}$$

$$(MaP \wedge SaM) \Rightarrow SaP; \tag{18}$$

$$(MiP \wedge SiM) \Rightarrow SiP. \tag{19}$$

The formal properties of this axiomatic system are considered in [9], [10]. In the performative syllogistic we can analyze the collective dimension of behavior. Within this system we can study how the plasmodium of *Physarum polycephalum* and the plasmodium of *Badhamia utricularis* occupy all possible attractants in any direction if they can see them. So, this system shows logical properties of a massive-parallel behavior (i.e. the collective dimension of behavior). One of the most significant notions involved in this implementation of the performative syllogistic in plasmodia topology is a *neighborhood*. We can define a distance for the neighborhood differently, i.e. we can make it broader or narrower. So, from different neighborhoods it will follow that we deal with different 'universes of discourse'.

14.3.1. Antagonistic Go game for verifying performative syllogistic reasoning

In the Go game diagrams for the performative syllogistic, the 'universe of discourse' cover cells x, y, non-x (which be denoted by x'), non-y (which be denoted by y'):

x	y'
y	x'

where x, y, x', y' are neighbor cells containing black stones (interpreted as attractants for *Physarum polycephalum*) and white stones (interpreted as attractants for *Badhamia utricularis*), x' are all neighbors for y which differ from x, and y' are all neighbors for x which differ from y. Let us consider the Go game from the point of view just of the Black player. Suppose that we have black, white, and grey stones and (i) if a black stone is placed within a cell, this means that "this cell is occupied by the plasmodium of *Physarum polycephalum*" (i.e. "there is at least one thing in it for the Black player"), (ii) if a white stone is placed within a cell, this means that "this cell is not occupied the plasmodium of *Physarum polycephalum*" (i.e. "there is not thing in it for the Black player"), (ii) if a grey stone is placed within a cell, this means that "it is not known if this cell is occupied by the plasmodium of *Physarum polycephalum*" or "it is not known what color of stone placed within a cell is from the point of view of the Black player". All possible combinations of Go game diagrams for atomic propositions within our universe of discourse are pictured in fig. 10.

The universe of discourse for simulating performative syllogisms by means of *Physarum* behaviors covers cells x, y, m, x', y', m' in the following manner:

y'	m	m'	x'
m'	x	y'	m
m	y	x'	m'
x	m'	m	y

The motion of plasmodium starts from one of the central cells (x, y, x', y') and goes towards one of the four directions (northwest, southwest, northeast, southeast). The Go game diagram for syllogistic conclusions is as follows:

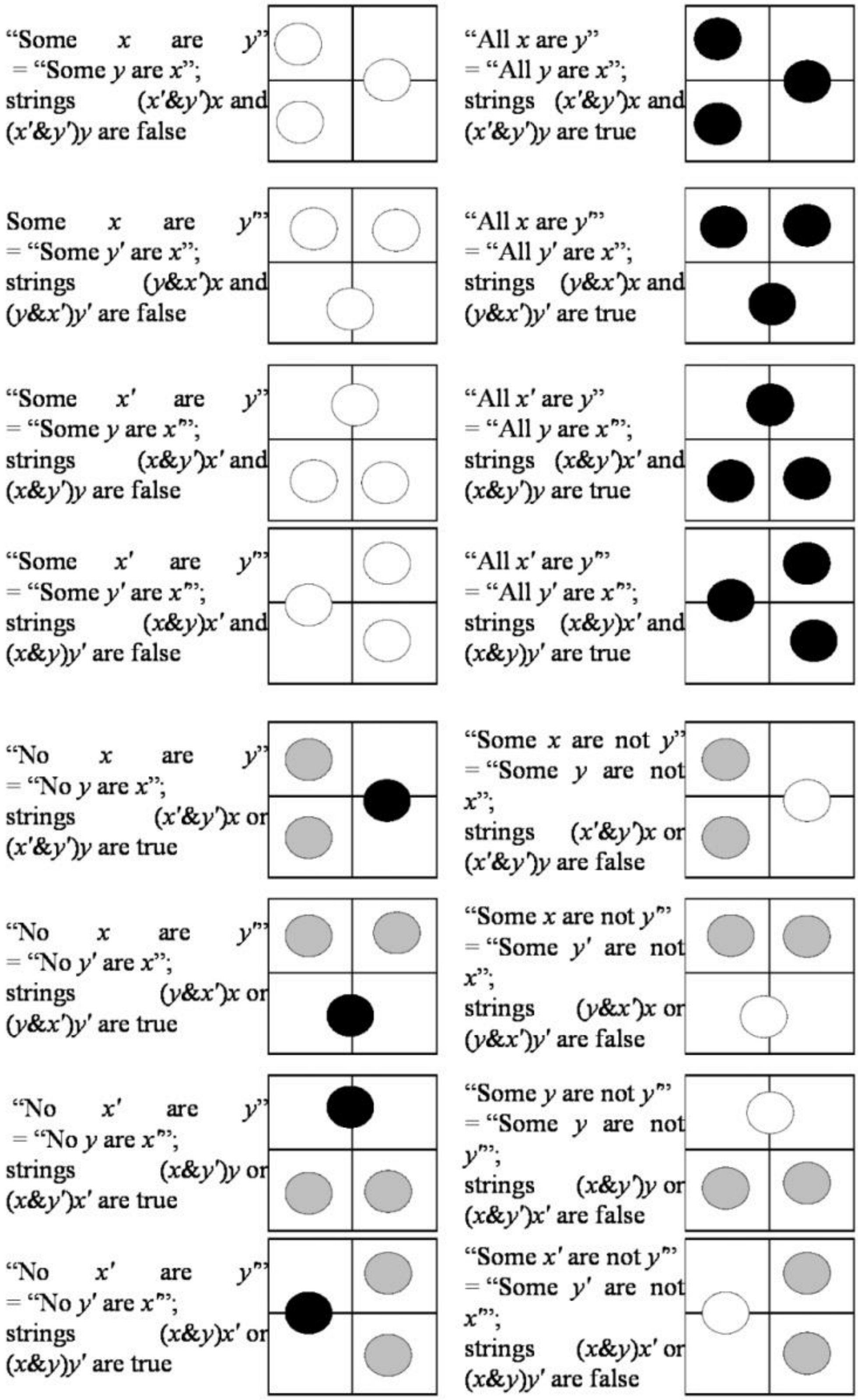
x	m'
m	x'

Some examples of performative syllogistic conclusions are regarded in fig. 11.

Thus, the performative syllogistic allows us to study different zones containing attractants for *Physarum polycephalum* and *Badhamia utricularis* if they are connected by protoplasmic tubes homogenously.

A model $\mathsf{M}' = \langle M', |\cdot|_x \rangle$ for the performative syllogistic, where M' is the set of attractants and $|X|_x \subseteq M'$ is a meaning of syllogistic letter X which is understood as all attractants reachable for the plasmodium of *Physarum polycephalum* from the point x, is defined as follows:

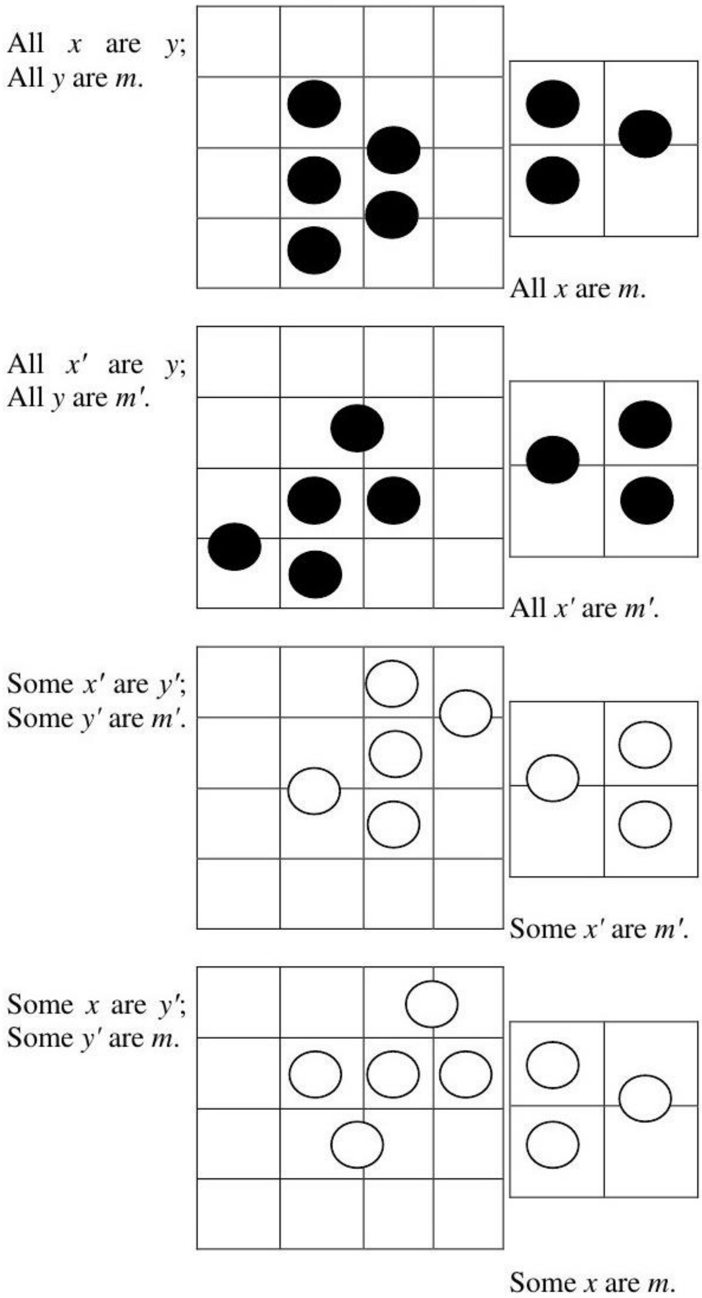
Figure 10. The Go game diagrams for premises of performative syllogisms. Strings of the form $(x' \& y')x$ mean that in cells x' and y' there are neighbors A for x such that Ax, i.e. $(x' \& y')$ is a metavariable in $(x' \& y')x$ that is used to denote all attractants of x' and y' which are neighbors for the attractant of x.



$\mathsf{M}' \models$ *All x are y* iff $|X|_x \neq \varnothing$, $|X|_y \neq \varnothing$, and $|X|_x \cap |X|_y \neq \varnothing$, more precisely both $(x' \,\&\, y')x$ and $(x' \,\&\, y')y$ hold in M', i.e. the plasmodium of *Physarum polycephalum* can move from neighbors of y to x and it can move from neighbors of x to y (we can place black stones in the line from neighbors of y to x and from neighbors of x to y);

$\mathsf{M}' \models$ *Some x are y* iff $y \notin |X|_x$ and $x \notin |X|_y$, more precisely neither $(x' \,\&\, y')x$ nor $(x' \,\&\, y')y$ hold in M', i.e. the plasmodium of *Physarum polycephalum* cannot move from neighbors of y to x and it cannot move from neighbors of x to y (we can place white stones in the line from neighbors of y to x and from neighbors of x to y);

Figure 11. The Go game diagrams for performative syllogisms with true conclusions from the point of view of the Black player.



$\mathsf{M}' \models$ *No x are y* iff $y \in |X|_x$ or $x \in |X|_y$, more precisely $(x' \,\&\, y')x$ or $(x' \,\&\, y')y$ hold in M', i.e. the plasmodium of *Physarum polycephalum* can move from neighbors of y to x or it can move from neighbors of x to y (we can place black stones in the line from neighbors of y to x or from neighbors of x to y);

$\mathsf{M}' \models$ *Some x are not y* iff $y \notin |X|_x$ or $x \notin |X|_y$, more precisely $(x' \,\&\, y')x$ or $(x' \,\&\, y')$ do not hold in M', i.e. the plasmodium of *Physarum polycephalum* cannot move from neighbors of y to x or it cannot move from neighbors of x to y (we can place white stones in the line from neighbors of y to x or from neighbors of x to y);

$\mathsf{M}' \models p \wedge q$ iff $\mathsf{M}' \models p$ and $\mathsf{M}' \models q$;

$\mathsf{M}' \models p \vee q$ iff $\mathsf{M}' \models p$ or $\mathsf{M}' \models q$;

$\mathsf{M}' \models \neg p$ iff it is false that $\mathsf{M}' \models p$.

Proposition 3. *The performative syllogistic is sound and complete in* M'.

For more details on formal properties of performative syllogistic, please see [9], [10]. This syllogistic describes the logic of plasmodium propagation in all possible directions. For the implementation of this syllogistic we do not need repellents. It is a natural system. The performative syllogistic as a Go game is an antagonistic game, where two players draw own conclusions without any coalition (see Fig. 10).

14.3.2. Examples of performative-syllogistic Go game

Let us also examine a game of Go at time step 10 to provide an example for performative syllogistic. Let this game be shown in figure 12. As usual, each Voronoi cell is denoted from $S_{1,1}$ to $S_{18,18}$. In the universe of fig. 12 there are no universal affirmative propositions and particular affirmative propositions. We face only universal negative propositions and particular negative propositions such as ‘No $S'_{4,9}$ are $S'_{4,10}$’, where $S'_{4,9}$ are neighbors for $S_{4,10}$ differing from $S_{4,9}$ and $S'_{4,10}$ are neighbors for $S_{4,9}$ differing from $S_{4,10}$, and ‘Some $S'_{4,11}$ are not $S'_{5,11}$’, where $S'_{4,11}$ are neighbors for $S_{5,11}$ differing from $S_{4,11}$ and $S'_{5,11}$ are neighbors for $S_{4,11}$ differing from $S_{5,11}$.

In the universe of figure 13 there is a universal affirmative proposition: ‘All $S_{6,5}$ are $S_{6,6}$’, and a particular affirmative proposition: ‘Some $S_{5,8}$ are $S_{5,9}$’.

Conclusion

We have just shown that we can simulate the plasmodium motion as a Go game, where black stones are interpreted as attractants and white stones as repellents. We can consider configurations of stones as spatial reasoning. If we implement the Aristotelian syllogistic, we need a coalition of two players. If we implement the performative syllogistic [9], [10], we deal with an antagonistic game. In the latter case black stones are interpreted as attractants for *Physarum polycephalum* and white stones as attractants for *Badhamia utricularis.*

Figure 12. The performative syllogistic Go game 1 at time step 10. The black stones are attractants occupied by the plasmodium of *Physarum polycephalum* and the white stones are attractants occupied by the plasmodium of *Badhamia utricularis*. So, we construct syllogisms from the point of view of the Black player.

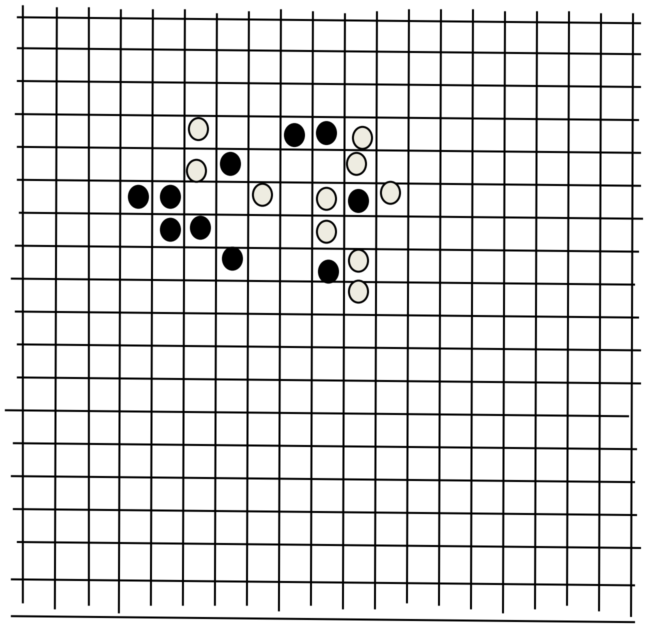
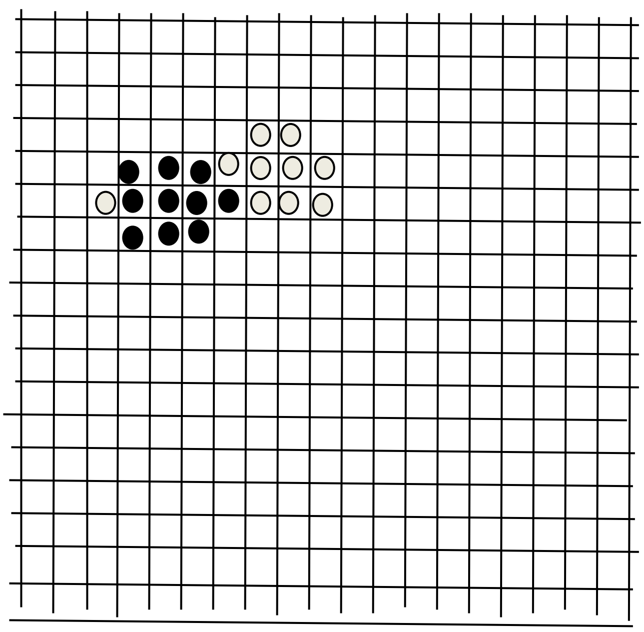


Figure 13. The performative syllogistic Go game 2 at time step 10. The black stones are attractants occupied by the plasmodium of *Physarum polycephalum* and the white stones are attractants occupied by the plasmodium of *Badhamia utricularis*. We build syllogisms from the point of view of the Black player.



References

[1] Adamatzky, A., Erokhin, V., Grube, M., Schubert, Th., Schumann, A., *Physarum Chip Project: Growing Computers From Slime Mould*, International Journal of Unconventional Computing, vol. 8, no. 4, pp. 319–323, 2012.

[2] Adamatzky, A., *Physarum Machines: Computers from Slime Mould* (World Scientific Series on Nonlinear Science, Series A), World Scientific Publishing Company, 2010.

[3] Łukasiewicz, J., *Aristotle's Syllogistic From the Standpoint of Modern Formal Logic*, Oxford Clarendon Press, 2nd edition, 1957.

[4] Nakagaki, T., Yamada, H., Toth, A., *Maze-solving by an amoeboid organism*, Nature, vol. 407, pp. 470–470, 2000.

[5] Nakagaki, T., Yamada, H., Tothm, A., *Path finding by tube morphogenesis in an amoeboid organism*, Biophysical Chemistry, vol. 92, pp. 47–52, 2001.

[6] Nakagaki, T., Iima, M., Ueda, T., Nishiura, Y., Saigusa, T., Tero, A., Kobayashi, R., Showalter, K., *Minimum-risk path finding by an adaptive amoeba network*, Physical Review Letters, vol. 99, pp. 68–104, 2007.

[7] Saigusa, T., Tero, A., Nakagaki, T., Kuramoto, Y., *Amoebae Anticipate Periodic Events*, Phys. Rev. Lett, vol. 100, no. 1, 2008.

[8] Schumann, A., Akimova, L., *Simulating of Schistosomatidae (Trematoda: Digenea) Behavior by Physarum Spatial Logic*, *Annals of Computer Science and Information Systems, vol. 1., Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, IEEE Xplore, pp. 225–230, 2013.

[9] Schumann, A., *On Two Squares of Opposition: the Lesniewski's Style Formalization of Synthetic Propositions*, Acta Analytica, vol. 28, pp. 71–93, 2013.

[10] Schumann, A., *Two Squares of Opposition: for Analytic and Synthetic Propositions*, Bulletin of the Section of Logic, vol. 40, no. 3/4, pp. 165–178, 2011.

[11] Schumann, A., *Payoff Cellular Automata and Reflexive Games*, Journal of Cellular Automata, vol. 9, no. 4, pp. 287–313, 2014.

[12] Schumann, A., Adamatzky, A., *Physarum Spatial Logic*, New Mathematics and Natural Computation, vol. 7, no. 3, pp. 483–498, 2011.

[13] Schumann, A., Pancerz, K., *Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing*, [in:] *Proceedings of the Workshop on Concurrency, Specification and Programming* (CS&P'2013), Warsaw, Poland, September 25-27, pp. 389–397, 2013.

[14] Schumann, A., Pancerz, K., Adamatzky, A., Grube, M., *Bio-Inspired Game Theory: The Case of Physarum Polycephalum*, [in:] *8th International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, ICST, 2015, doi: http://dx.doi.org/10.4108/icst.bict.2014.257869.

[15] Shirakawa, T., Gunji, Y.P., Miyake, Y., *An associative learning experiment using the plasmodium of Physarum polycephalum*, Nano Communication Networks, vol. 2, pp. 99–105, 2011.

[16] Smith, R., *Completeness of an ecthetic syllogistic*, Notre Dame J. Formal Logic, vol. 24, no. 2, pp. 224–232, 1983.

[17] Tero, A., Nakagaki, T., Toyabe, K., Yumiki, K., Kobayashi, R., *A Method Inspired by Physarum for Solving the Steiner Problem*, *IJUC*, vol. 6, no. 2, pp. 109–123, 2010.

[18] Tsuda, S., Aono, M., Gunji, Y.P., *Robust and emergent Physarum-computing*, BioSystems, vol. 73, pp. 45–55, 2004.

[19] Watanabe, Sh., Tero, A., Takamatsu, A., Nakagaki, T., *Traffic optimization in railroad networks using an algorithm mimicking an amoeba-like organism*, Physarum plasmodium, Biosystems, vol. 105, no. 3, pp. 225–232, 2011.

Originally published: Schumann, A., Pancerz, K., *A Rough Set Version of the Go Game on Physarum Machines*, [in:] *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT 2015),* New York City, New York, USA, December 3-5, 2015.

XV
A Rough Set Version of the Go Game on *Physarum* Machines

Andrew Schumann, Krzysztof Pancerz

Abstract

We make use of a *Physarum* machine that is a biological computing device implemented in the plasmodium of *Physarum polycephalum* and/or *Badhamia utricularis* which are one-cell organisms able to build complex networks for solving different computational tasks. The plasmodium behavior can model an ancient Chinese game called Go. In the paper, we describe implementation of the Go game on the *Physarum* machines. A special version of the game is presented, where payoffs are assessed by means of the measure defined on the basis of rough set theory. Theoretical foundations given in the paper are supplemented with description of a specialized software tool developed, among others, for simulation of the described game.

Introduction

Go is a game, originated in ancient China, in which two people play with a Go board and Go stones (cf. [5]). In general, the two players alternately place black and white stones, on the vacant intersections of a board with a 19×19 grid of lines, to surrounding territory. Whoever has more territory at the end of the game is the winner. Vertically and horizontally adjacent stones of the same color form a group. One of the basic principles of Go is the fact that stones must have at least one *liberty* to remain on the board. A liberty of a given stone is a vacant intersection adjacent to it. If a stone has at least one liberty, then the next stone of a given player can be placed on it to extend his/her group.

As it was shown in [12], the Go game can be simulated by means of a *Physarum* machine. A *Physarum* machine is a programmable amorphous biological computing device experimentally implemented in the plasmodium of *Physarum polycephalum*, also called true slime mould [1]. *Physarum polycephalum* is a single cell organism belonging to the species of order *Physarales*. General assumptions for games implemented on *Physarum* machines were presented in [16] and [17]. In our case, we assume that *Physarum* machines are experimentally implemented in the plasmodium of *Physarum polycephalum*, as well as in the plasmodium of *Badhamia utricularis* (cf. [7], [17]). *Badhamia utricularis* is another species of order *Physarales*. The plasmodium of *Physarum polycephalum* or *Badhamia utricularis*, spread by networks, can be programmable by adding and removing attractants and repellents. Two syllogistic systems implemented as Go games were considered in [12], namely, the Aristotelian syllogistic as well as the performative syllogistic. In the first case, the locations of black and white stones are understood as locations of attractants and repellents, respectively. In the second case, the locations of black stones are understood as locations of attractants occupied by plasmodia of *Physarum polycephalum* and the locations of white stones are understood as locations of attractants occupied by plasmodia of *Badhamia utricularis*. The Aristotelian syllogistic version of the Go game is a coalition game. The performative syllogistic version of the Go game is an antagonistic game.

In the paper, we will consider the second case (i.e., an antagonistic game implemented in plasmodia of *Physarum polycephalum* and *Badhamia utricularis*). For this case, we propose a new version of the Go game. In the presented approach, payoffs are assessed by means of the measure defined on the basis of rough set theory. Rough sets are a tool to deal with rough (ambiguous, imprecise) concepts in the universe of discourse (cf. [10]). In our previous research, we used rough sets to describe behavior of *Physarum* machines (see [8] and [14]). In this case *Physarum* machines were modeled by means of transition systems or timed transition systems and rough set models were more abstract models built over transition systems. As it was shown, in rough set descriptions of *Physarum* machines, both a standard definition of rough sets proposed by Z. Pawlak [10] and the Variable Precision Rough Set Model (VPRSM) can be used. The VPRSM approach is a more relaxed and generalized rough set approach, proposed by W. Ziarko in [18]. Basic definitions, notions and notation concerning rough set theory as well as the VPRSM approach are recalled in Section 15.1. A rough set based approach was also used by us to describe strategy games implemented on *Physarum* machines [7]. The strategies of such games were approximated on the basis of a rough set model, describing behavior of the *Physarum* machine, created according to the VPRSM approach.

In the paper, another rough set based approach to assessing payoffs of games implemented on *Physarum* machines is proposed in Section 15.2. This time, for the Go game, we do not use transition system models describing behavior of *Physarum* machines, but our attention is focused on the adjacent surroundings of stones. Rough sets are used to determine how (i.e., exactly or roughly) these surroundings approximate sets of intersections occupied by either black or white stones (i.e., occupied by either plasmodia of *Physarum polycephalum* or plasmodia of *Badhamia utricularis*).

To support research on programming *Physarum* machines and simulating *Physarum* games, we are developing a specialized software tool, called the *Physarum* software system, shortly *PhysarumSoft* (see [13]). The approach presented in this paper is also implemented in *PhysarumSoft*. A short description of the *Physarum* game module is given in section 15.3.

15.1. The rudiments of rough sets

In this section, we recall necessary definitions, notions and notation concerning rough sets.

The idea of rough sets (see [10]) consists of the approximation of a given set by a pair of sets, called the lower and the upper approximation of this set. Some sets cannot be exactly defined. If a given set X is not exactly defined, then we employ two exact sets (the lower and the upper approximation of X) that define X roughly (approximately).

Let $U \neq \emptyset$ be a finite set of objects we are interested in. U is called the universe. Any subset $X \subseteq U$ of the universe is called a concept in U. Let R be any equivalence relation over U. We denote an equivalence class of any $u \in U$ by $[u]_R$. With each subset $X \subseteq U$ and any equivalence relation R over U, we associate two subsets:

$$R_*(X) = \{u \in U : [u]_R \subseteq X\},$$

$$R^*(X) = \{u \in U : [u]_R \cap X \neq \emptyset\},$$

called the R-lower and R-upper approximation of X, respectively. A set $BN_R(X) = R^*(X) - R_*(X)$ is called the R-boundary region of X. If $BN_R(X) = \emptyset$, then X is sharp (exact) with respect to R. Otherwise, X is rough (inexact).

The definitions given earlier are based on the standard definition of set inclusion. Let U be the universe and $A, B \subseteq U$. The standard set inclusion is defined as

$$A \subseteq B \; if \; and \; only \; if \; \bigvee_{u \in A} u \in B.$$

In some situations, the application of this definition seems to be too restrictive and rigorous. W. Ziarko proposed in [18] some relaxation of the original rough set approach. His proposition was called the Variable Precision Rough Set Model (VPRSM). The VPRSM approach is based on the notion of the majority set inclusion. Let U be the universe, $A, B \subseteq U$, and $0 \leq \beta < 0.5$. The majority set inclusion is defined as

$$A \overset{\beta}{\subseteq} B \; if \; and \; only \; if \; 1 - \frac{card(A \cap B)}{card(A)} \leq \beta,$$

where $card$ denotes the cardinality of the set. $A \overset{\beta}{\subseteq} B$ means that the specified majority of elements belonging to A belongs also to B. One can see that if $\beta = 0$, then the majority set inclusion becomes the standard set inclusion.

By replacing the standard set inclusion with the majority set inclusion in definitions of approximations, we obtain the following two subsets:

$$R_*^{\beta}(X) = \{u \in U : [u]_R \overset{\beta}{\subseteq} X\},$$

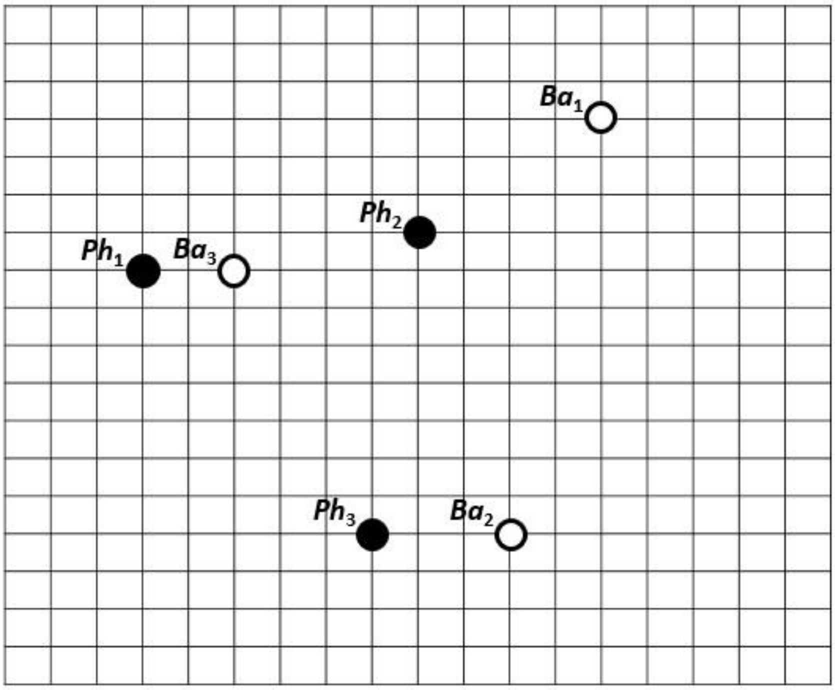
$$R^{*\beta}(X) = \{u \in U : \frac{card([u]_R \cap X)}{card([u]_R)} > \beta\},$$

called the R_β-lower and R_β-upper approximation of X, respectively.

15.2. Rough set based assessment of payoffs

In this section, we present basic principles of the rough set version of the Go game implemented on the *Physarum* machine. Let a board for the Go game, with a 19×19 grid of lines, be in use. The set of all intersections of the grid is denoted by I. At the beginning, the fixed numbers of original points of both the plasmodia of *Physarum polycephalum* and the plasmodia of *Badhamia utricularis* are randomly deployed on intersections. An example of the initial configuration of the Go game is shown in figure 1. In this case, three original points of the plasmodia of *Physarum polycephalum*, Ph_1, Ph_2, and Ph_3, understood as black stones, as well as three original points of the plasmodia of *Badhamia utricularis*, Ba_1, Ba_2, and Ba_3, understood as white stones, are deployed on intersections.

Figure 1. An example of the initial configuration of the Go game implemented on the *Physarum* machine.



During the game, the two players alternately place attractants on the vacant intersections of the board. The first player plays for the *Physarum polycephalum* plasmodia, the second one for the *Badhamia utricularis* plasmodia. The plasmodia look for attractants, propagate protoplasmic veins towards them, feed on them and go on. In a real-life implementation of *Physarum* machines, attractants are sources of nutrients or pheromones, on which the plasmodium feeds (see [1]). The attractants occupied by plasmodia of *Physarum polycephalum* are treated as black stones whereas the attractants occupied by plasmodia of *Badhamia utricularis*, as white stones.

Remark 1 We assume that the state of the Go game (i.e., the state of the *Physarum* machine) is observed only at discrete time instants, i.e., after each move in the game. Therefore, whenever time instant t is used, it means that $t = 0,1,2,\ldots$.

For our version of Go, we assume the following formal structure of the *Physarum* machine. The *Physarum* machine is a triple:

$$\mathsf{PM} = (P, B, A),$$

where:

- $P = \{Ph_1, Ph_2, \ldots, Ph_k\}$ is a set of original points of the plasmodia of *Physarum polycephalum*.
- $B = \{Ba_1, Ba_2, \ldots, Ba_l\}$ is a set of original points of the plasmodia of *Badhamia utricularis*.
- $A = \{A^t\}_{t=0,1,2,\ldots}$ is the family of the sets of attractants, where $A^t = \{A_1^t, A_2^t, \ldots, A_{r_t}^t\}$ is the set of all attractants present at time instant t in PM.

One can see that the structure of the *Physarum* machine PM is changing in time (new attractants are added by the players). Positions of original points of the plasmodia as well as attractants are considered in the two-dimensional space of intersections. Hence, each intersection $i \in I$ is identified by two coordinates x and y. This fact will be denoted by $i(x, y)$. For each intersection $i(x, y)$, we can distinguish its adjacent surroundings:

$$\begin{aligned} Surr(i) \quad & = \{i'(x', y') \in I : \\ & (x' = x-1 \vee x' = x+1) \wedge (x' \geq 1) \wedge (x' \leq 19) \\ & \wedge \\ & (y' = y-1 \vee y' = y+1) \wedge (y' \geq 1) \wedge (y' \leq 19)\}. \end{aligned}$$

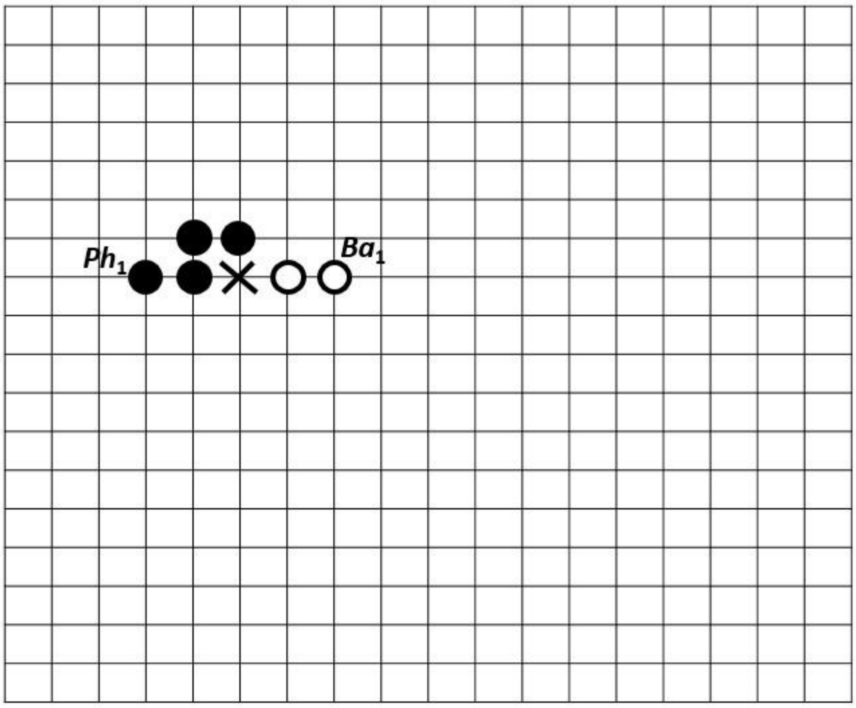
It is worth noting that liberties of stones placed at intersections are defined only in adjacent surroundings of stones.

After each move (i.e., placement of a new attractant on the board by one of the players), the following rules of behavior of the plasmodia are applied:

- As soon as the attractant is placed on an intersection and the adjacent surroundings contain the plasmodium of either *Physarum polycephalum* or *Badhamia utricularis*, this new attractant is occupied by the plasmodium.
- If there exist more than one intersection occupied by both the plasmodia of *Physarum polycephalum* and the plasmodia of *Badhamia utricularis* in the adjacent surroundings of the intersection where the attractant was placed, then only the plasmodia of one type (randomly selected) are attracted. The plasmodium of *Physarum polycephalum* and the plasmodium of *Badhamia utricularis* cannot occupy the same attractants.

In case of Rule 2, the fusion of the plasmodia of one type can hold. Moreover, Rule 2 causes that, in some situations, given moves are taken under the players' own risk. Let us consider an illustrative configuration of the Go game shown in figure 2. There are two attractants occupied by the plasmodia of *Physarum polycephalum* (black stones) and one attractant occupied by the plasmodia of *Badhamia utricularis* (white stone) in the adjacent surroundings of the intersection marked with $\times$. If the first player places a new attractant $A_\times$ at intersection $\times$, two situations, mutually exclusive, will be possible. In the first situation, two plasmodia of *Physarum polycephalum* will be attracted by $A_\times$ and fused on it. In this case, a new black stone will appear. In the second situation, the plasmodium of *Badhamia utricularis* will be attracted by $A_\times$. In this case, a new white stone will appear. It means that the first player's move may work to the advantage of the opponent.

Figure 2. An illustrative configuration of the Go game, where a move is taken under the player's own risk.



Formally, during the game, at given time instant t, we can distinguish three kinds of intersections in the set I_t of all intersections:

- $I_t^{\varnothing}$ - a set of all vacant intersections at t.
- $I_t^{\bullet}$ - a set of all intersections occupied by plasmodia of *Physarum polycephalum* at t (black stones).
- I_t° - a set of all intersections occupied by plasmodia of *Badhamia utricularis*) at t (white stones).

One can see that $I_t = I_t^{\varnothing} \cup I_t^{\bullet} \cup I_t^{\circ}$, where $I_t^{\varnothing}$, $I_t^{\bullet}$, and I_t° are pairwise disjoint.

A set I_t^{π} of all intersections occupied by given plasmodia π (either the plasmodia of *Physarum polycephalum* or the plasmodia of *Badhamia utricularis*) at given time instant t can be approximated (according to rough set definition) by surroundings of intersections occupied by these plasmodia. Such approximation will be called surroundings approximation.

The lower surroundings approximation $Surr_*(I_t^{\pi})$ of I_t^{π} is given by configuration of the Go game.

$$Surr_*(I_t^{\pi}) = \{i \in I_t^{\pi} : Surr(i) \neq \varnothing \wedge Surr(i) \subseteq I_t^{\pi}\},$$

where π is either $\bullet$ or $\circ$. The lower surroundings approximation consists of all intersections, occupied by plasmodia π, whose all of not vacant adjacent intersections are also occupied by π. Each intersection $i \in I$ such that $i \in Surr_*(I_t^{\pi})$ is called a full generator of the payoff of the player playing for the plasmodia π.

The upper surroundings approximation $Surr^*(I_t^{\pi})$ of I_t^{π} is given by

$$Surr^*(I_t^{\pi}) = \{i \in I_t^{\pi} : Surr(i) \cap I_t^{\pi} \neq \varnothing\},$$

where π is either $\bullet$ or $\circ$. The upper surroundings approximation consists of all intersections, occupied by plasmodia π, whose adjacent surroundings cover at least one intersection also occupied by π.

The set $BN_{Surr}(I_t^{\pi}) = Surr^*(I_t^{\pi}) - Surr_*(I_t^{\pi})$ is referred to as the boundary region of surroundings approximation of I_t^{π} at time instant t. Each intersection $i \in I$ such that $i \in BN_{Surr}(I_t^{\pi})$ is called a partial generator of the payoff of the player playing for the plasmodia π.

By replacing the standard set inclusion with the majority set inclusion in the definition of the lower surroundings approximation (according to the VPRSM approach recalled in section 2), we obtain the β-lower surroundings approximation:

$$Surr_*^{\beta}(I_t^{\pi}) = \{i \in I_t^{\pi} : Surr(i) \neq \emptyset \wedge Surr(i) \overset{\beta}{\subseteq} I_t^{\pi}\},$$

Each intersection $i \in I$ such that $i \in Surr_*^{\beta}(I_t^{\pi})$ is called a full quasi-generator of the payoff of the player playing for the plasmodia π.

On the basis of lower surroundings approximations, we define a measure assessing payoffs of the players. For the first player playing for the *Physarum polycephalum* plasmodia, the payoff measure has the form:

$$\Theta^{\bullet} = card\,(Surr_*(I_t^{\bullet})).$$

For the second player playing for the *Badhamia utricularis* plasmodia, the payoff measure has the form:

$$\Theta^{\circ} = card(Surr_*(I_t^{\circ})).$$

In a more relaxed case, we have respectively:

$$\Theta^{\bullet} = card\,(Surr_*^{\beta}(I_t^{\bullet})).$$

and

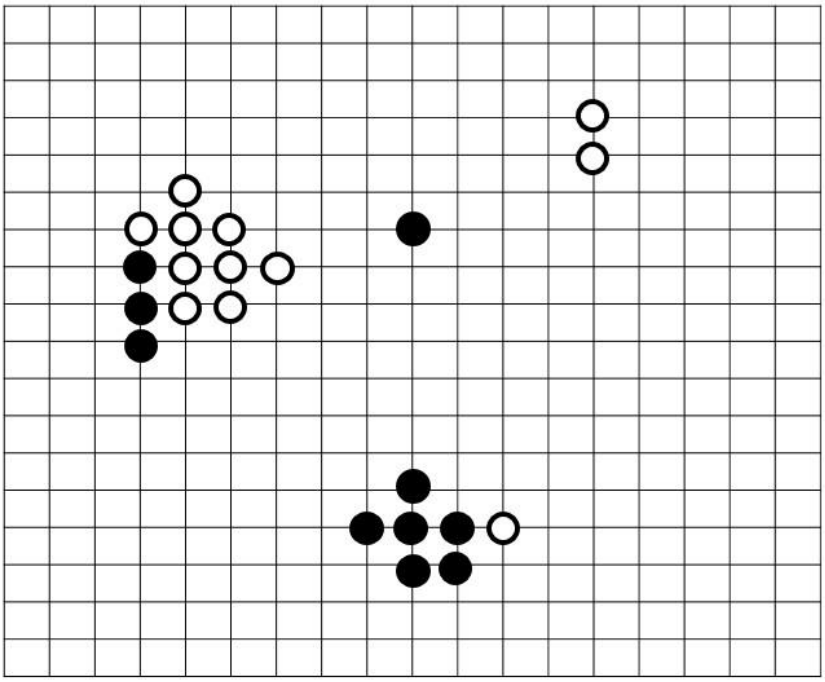
$$\Theta^{\circ} = card\,(Surr_*^{\beta}(I_t^{\circ})).$$

One can see that the VPRSM approach enables us to set different levels of difficulty of the Go game.

The goal of each player is to maximize its payoff.

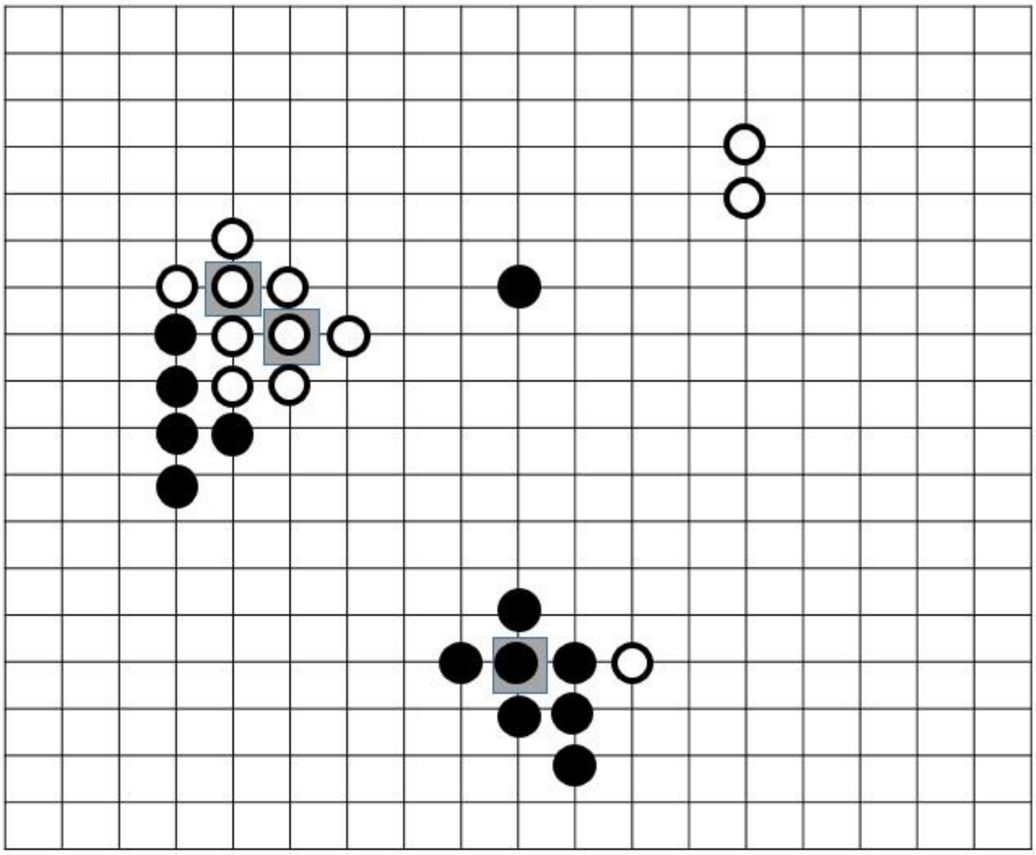
Let us consider an illustrative configuration of the Go game after several moves shown in figure 3.

Figure 3. An illustrative configuration of the Go game after several moves.



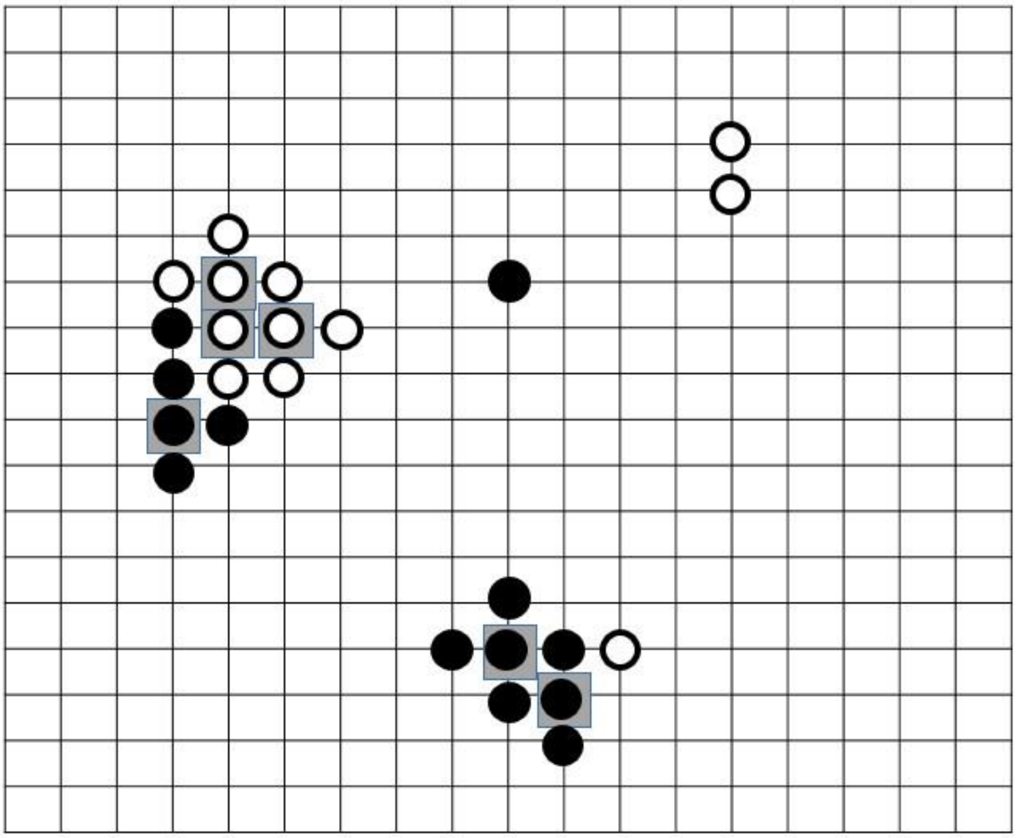
In case of a standard definition of rough sets (i.e., the most rigorous case), intersections belonging to lower surroundings approximations $Surr_*(I_t^\bullet)$ and $Surr_*(I_t^\circ)$ of $(I_t^\bullet)$ and (I_t°), respectively, are marked with grey rectangles in figure 4. It is worth noting that all of the intersections from $Surr_*(I_t^\bullet)$ and $Surr_*(I_t^\circ)$ are full generators of the payoffs of the players playing for the *Physarum polycephalum* plasmodia and *Badhamia utricularis* plasmodia, respectively. Hence, we obtain $\Theta^\circ = 1$ and $\Theta^\bullet = 2$. The second player, playing for the *Badhamia utricularis* plasmodia, wins.

Figure 4. A configuration of the Go game after several moves (payoffs defined on the basis of a standard definition of rough sets).



In case of the VPRSM approach (i.e., a more relaxed case), for $\beta = 0.25$, intersections belonging to lower surroundings approximations $Surr_*^{0.25}(I_t^\bullet)$ and $Surr_*^{0.25}(I_t^\circ)$ of $(I_t^\bullet)$ and (I_t°), respectively, are marked with grey rectangles in figure 5. It is worth noting that some of intersections from $Surr_*(I_t^\bullet)$ and $Surr_*(I_t^\circ)$ are full quasi-generators of the payoffs of the players playing for the *Physarum polycephalum* plasmodia and *Badhamia utricularis* plasmodia, respectively. Hence, we obtain $\Theta^\circ = 3$ and $\Theta^\bullet = 3$. No player wins.

Figure 5. A configuration of the Go game after several moves (payoffs defined on the basis of the VPRSM approach for $\beta = 0.25$).



Let us consider now how we can define strategies in our Go game if we deal with the standard surroundings approximation. A mapping from the intersections belonging to upper surroundings approximations $Surr^{*}(I_t^{\pi})$ at t to the intersections belonging to lower surroundings approximations $Surr_{*}(I_{t+k}^{\pi})$ at $t+k$ is said to be a set of rational strategies of π of radius k to win. A mapping from the intersections belonging to β-lower surroundings approximations $Surr_{*}^{\beta}(I_t^{\bullet})$ at t to the intersections belonging to the union $Surr_{*}^{\gamma}(I_{t+k}^{\circ}) \cup Surr_{*}^{\beta}(I_{t+k}^{\bullet})$ at $t+k$, where $0 < \beta \leq \gamma$ and

$$card(Surr_{*}^{\beta}(I_t^{\bullet})) \leq card(Surr_{*}^{\gamma}(I_{t+k}^{\circ}) \cup Surr_{*}^{\beta}(I_{t+k}^{\bullet})) < card(Surr_{*}(I_{t+k}^{\bullet})),$$

is said to be a set of rational strategies of $\circ$ (the player playing for *Badhamia utricularis*) of radius k not to lose. A mapping from the intersections belonging to β-lower surroundings approximations $Surr_{*}^{\beta}(I_t^{\circ})$ at t to the intersections belonging to the union $Surr_{*}^{\gamma}(I_{t+k}^{\bullet}) \cup Surr_{*}^{\beta}(I_{t+k}^{\circ})$ at $t+k$, where $0 < \beta \leq \gamma$ and

$$card(Surr_{*}^{\beta}(I_t^{\circ})) \leq card(Surr_{*}^{\gamma}(I_{t+k}^{\bullet}) \cup Surr_{*}^{\beta}(I_{t+k}^{\circ})) < card(Surr_{*}(I_{t+k}^{\circ})),$$

is said to be a set of rational strategies of $\bullet$ (the player playing for *Physarum polycephalum*) of radius k not to lose.

The agent is rational if (s)he follows one of the rational strategies to win or not to lose in moves. Also, we can define strategies in the Go game if we deal with the VPRSM surroundings approximation. A mapping from the intersections belonging to γ-lower surroundings approximations $Surr_*^\gamma(I_t^\pi)$ at t to the intersections belonging to β -lower surroundings approximations $Surr_*^\beta(I_{t+k}^\pi)$ at $t+k$ is said to be a set of rational β-strategies of π of radius k to win if $\gamma < beta$. A mapping from the intersections belonging to γ-lower surroundings approximations $Surr_*^\gamma(I_t^\bullet)$ at t to the intersections belonging to the union $Surr_*^\delta(I_{t+k}^\circ) \cup Surr_*^\gamma(I_{t+k}^\bullet)$ at $t + k$, where $0 < \beta \le \gamma$ and $0 < \beta \le \delta$ and $card(Surr_*^\gamma(I_t^\bullet)) \le card(Surr_*^\delta(I_{t+k}^\circ) \cup Surr_*^\gamma(I_{t+k}^\bullet)) < card(Surr_*^\beta(I_{t+k}^\bullet))$, is said to be a set of rational β-strategies of $\circ$ (the player playing for *Badhamia utricularis*) of radius k not to lose. A mapping from the intersections belonging to γ -lower surroundings approximations $Surr_*^\gamma(I_t^\circ)$ at t to the intersections belonging to the union $Surr_*^\delta(I_{t+k}^\bullet) \cup Surr_*^\gamma(I_{t+k}^\circ)$ at $t+k$, where $0 < \beta \le \gamma$ and $0 < \beta \le \delta$ and $card(Surr_*^\gamma(I_t^\circ)) \le card(Surr_*^\delta(I_{t+k}^\bullet) \cup Surr_*^\gamma(I_{t+k}^\circ)) < card(Surr_*^\beta(I_{t+k}^\circ))$, is said to be a set of rational β-strategies of $\bullet$ (the player playing for *Physarum polycephalum*) of radius k not to lose.

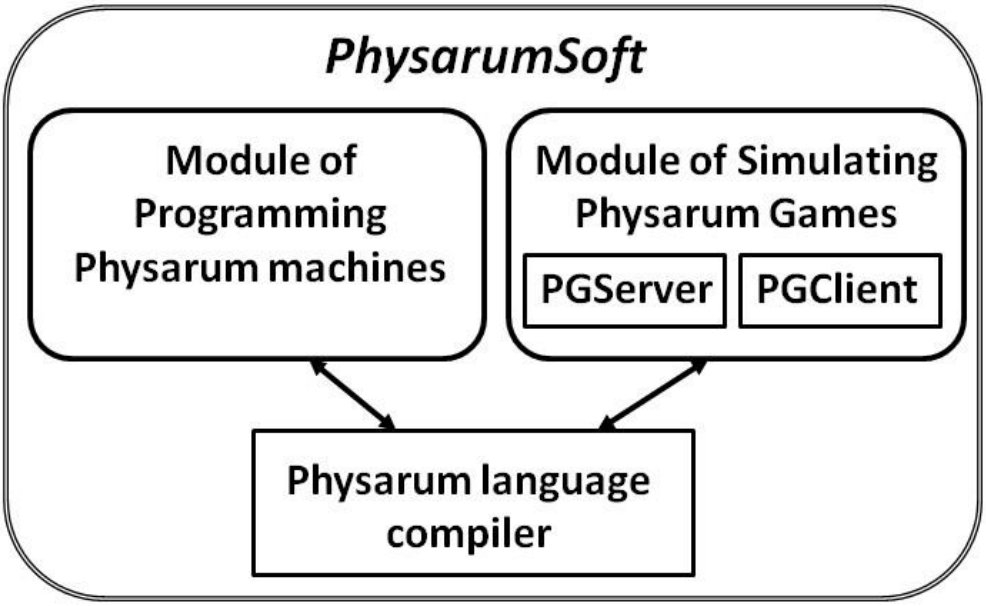
The agent is β-rational if (s)he follows one of the rational β-strategies to win or not to lose in moves.

15.3. A software tool

In [13], we described selected functionality of the current version of a new software tool, called *PhysarumSoft*, developed by us for programming *Physarum* machines and simulating *Physarum* games. In this section, we recall important features of this tool, as well as present additional information about the part of the tool responsible for simulating a rough set version of the Go game.

PhysarumSoft was designed for the Java platform. A general structure of this tool is shown in figure 6.

Figure 6. A general structure of *PhysarumSoft*.



We can distinguish three main parts of *PhysarumSoft*:

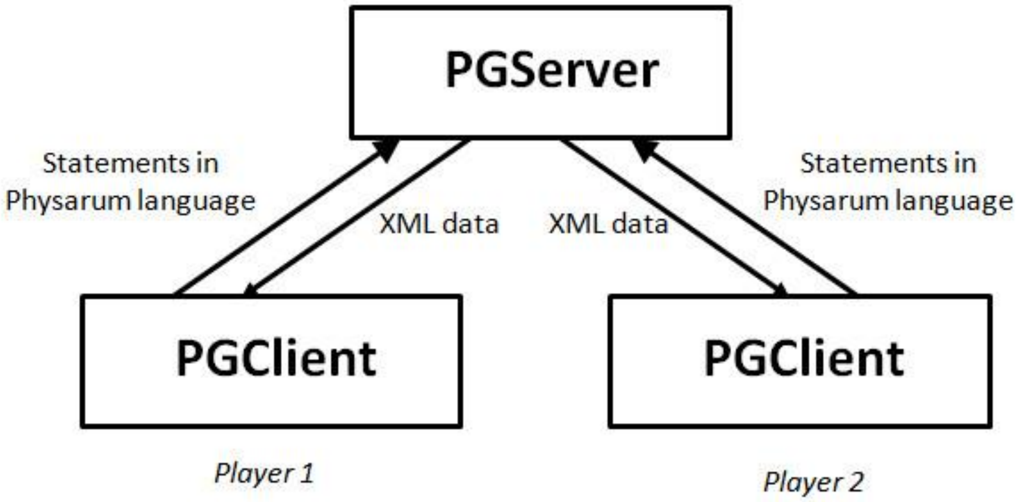
- *Physarum* language compiler.
- Module of programming *Physarum* machines.
- Module of simulating *Physarum* games.

The main features of *PhysarumSoft* are the following:

- Portability. Thanks to the Java technology, the created tool can be run on various software and hardware platforms. In the future, the tool will be adapted for platforms available in mobile devices and as a service in the cloud.
- User-friendly interface.
- Modularity. The project of *PhysarumSoft* and its implementation covers modularity. It makes the tool extend easily in the future.

To simulate games on *Physarum* machines, we are developing a special module of *PhysarumSoft* called the *Physarum* game simulator. This module works under the client-server paradigm. A general structure of the *Physarum* game simulator is shown in figure 7.

Figure 7. A general structure of the *Physarum* game simulator.



In *PhysarumSoft*, communication between clients and the server is realized through text messages containing statements of the *Physarum* language. The *Physarum* language, is a new object-oriented programming language, utilizing the prototype-based approach (cf. [3]), designed by us to program *Physarum* machines, i.e., to set the spatial distribution (configuration) of stimuli (attractants, repellents) controlling propagation of protoplasmic veins of the plasmodium. For a detailed description of the *Physarum* language, we refer the reader to [6], [9], and [15].

The server sends to clients information about the current configuration of the *Physarum* machine (localization of the original points of *Physarum polycephalum* and *Badhamia utricularis*, localization of stimuli, as well as a list of edges, corresponding to veins of plasmodia, between active points) through the XML file. Each original point of the plasmodia of *Physarum polycephalum* or the plasmodia of *Badhamia utricularis* and each attractant occupied by these plasmodia is called an active point in the *Physarum* machines.

The server-side application of the *Physarum* game simulator is called *PGServer*. The main window of *PGServer* is shown in figure 8. In this window, the user can:

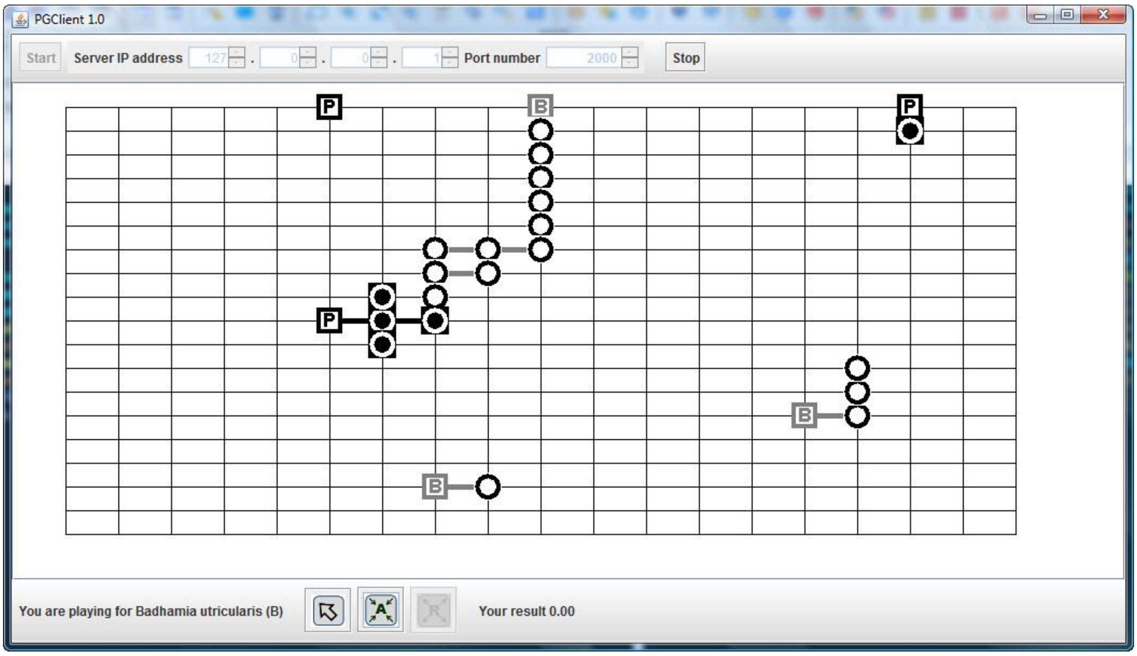
- select the port number on which the server listens for connections,
- start and stop the server,
- set the game:
  - a *Physarum* game with strategy based on stimulus placement (see [7]),
  - a *Physarum* game with strategy based on stimulus activation (see [7]),
  - a rough set version of the Go game (see Section 15.2),
- shadow information about actions undertaken.

Figure 8. The main window of *PGServer.*

The client-side application of the *Physarum* game simulator is called *PGClient*. The main window of *PGClient* is shown in figure 9. In case of the Go game, the user can:

- set the server IP address and its port number,
- start the participation in the game,
- put attractants at the vacant intersections of a board,
- monitor the current state of the game as well as the current assessment of payoffs.

Figure 9. The main window of *PGClient.*



Conclusions

A new version of the Go game, based on rough set theory and implemented on the *Physarum* machines, has been presented. The presented version is an antagonistic game. The locations of black stones are understood as intersections occupied by plasmodia of *Physarum polycephalum* and the locations of white stones are understood as intersections occupied by plasmodia of *Badhamia utricularis*. The main aim of our further research is to propose a coalition game based on a rough set assessment of payoffs. Moreover, we will extend the spectrum of measures by applying various rough set approaches.

Acknowledgments

This research is being fulfilled by the support of FP7-ICT-2011-8.

References

[1] Adamatzky, A., *Physarum Machines: Computers from Slime Mould*, World Scientific, 2010.

[2] Adamatzky, A., Erokhin, V., Grube, M., Schubert, T., Schumann, A., *Physarum Chip Project: Growing computers from slime mould. International Journal of Unconventional Computing*, 8(4):319–323, 2012.

[3] Craig, I., *Object-Oriented Programming Languages: Interpretation*, Springer-Verlag, London, 2007.

[4] Dubois, D., Prade, H., *Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems*, 17(2-3):191–209, 1990.

[5] Kim, J., Jeong, S., *Learn to Play Go*, Good Move Press, New York, 1994.

[6] Pancerz, K., Schumann, A., *Principles of an object-oriented programming language for Physarum polycephalum computing*, [in:] *Proceedings of the 10th International Conference on Digital Technologies (DT'2014)*, Zilina, Slovak Republic, pp. 273–280, 2014.

[7] Pancerz, K., Schumann, A., *Rough set description of strategy games on Physarum machines*, [in:] Adamatzky, A. (ed.), *Advances in Unconventional Computing*, Springer International Publishing, 2016, (to appear).

[8] Pancerz, K., Schumann, A., *Rough set models of Physarum machines. International Journal of General Systems*, 44(3):314–325, 2015.

[9] Pancerz, K., Schumann, A., *Some issues on an object-oriented programming language for Physarum machines*, [in:] Bris, R., Majernik, J., Pancerz, K., Zaitseva, E. (eds.), *Applications of Computational Intelligence in Biomedical Technology*, vol. 606 of

Studies in Computational Intelligence, Springer International Publishing, pp. 185–199, 2016.

[10] Pawlak, Z., *Rough Sets. Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[11] Pawlak, Z., Skowron, A., *Rudiments of rough sets*, Information Sciences, 177:3–27, 2007.

[12] Schumann, A., *Syllogistic versions of Go games on Physarum polycephalum*, [in:] Adamatzky, A. (ed.), *Advances in Physarum Machines*, Springer International Publishing, 2016 (to appear).

[13] Schumann, A., Pancerz, K., *PhysarumSoft - a software tool for programming Physarum machines and simulating Physarum games*, [in:] Ganzha, M., Maciaszek, L., Paprzycki, M. (eds.), *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems (FedCSIS'2015)*, Lodz, Poland, pp. 607–614, 2015.

[14] Schumann, A., Pancerz, K., *Roughness in timed transition systems modeling propagation of plasmodium*, [in:] Ciucci, D., Wang, G., Mitra, S., Wu, W.Z. (eds.), *Rough Sets and Knowledge Technology*, Lecture Notes in Artificial Intelligence, vol. 9436, Springer International Publishing, 2015.

[15] Schumann, A., Pancerz, K., *Towards an object-oriented programming language for Physarum polycephalum computing*, [in:] Szczuka, M., Czaja, L., Kacprzak, M. (eds.), *Proceedings of the Workshop on Concurrency, Specification and Programming (CS&P'2013)*, Warsaw, Poland, pp. 389–397, 2013.

[16] Schumann, A., Pancerz, K., *Interfaces in a game-theoretic setting for controlling the plasmodium motions*, [in:] *Proceedings of the 8th International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS'2015)*, Lisbon, Portugal, pp. 338–343, 2015.

[17] Schumann, A., Pancerz, K., Adamatzky, A., Grube, M., *Bio-inspired game theory: The case of Physarum polycephalum*, [in:] *Proceedings of the 8th International Conference on Bio-inspired Information and Communications Technologies (BICT'2014)*, Boston, Massachusetts, USA, 2014.

[18] Ziarko, W., *Variable precision rough set model*, Journal of Computer and System Sciences, 46(1):39–59, 1993.

Originally published: Schumann, A., Pancerz, K., *Interfaces in a Game-Theoretic Setting for Controlling the Plasmodium Motions*, [in:] BIOSTEC (Biosignals) 2014, Scitepress, 2014.

XVI
Interfaces in a Game-Theoretic Setting for Controlling the Plasmodium Motions

Andrew Schumann, Krzysztof Pancerz

Abstract

The plasmodium is the large one-cell organism containing a mass of multinucleate protoplasm. It is an active feeding stage of *Physarum polycephalum* or *Badhamia utricularis* and it moves by protoplasmic streaming which reverses every 30-60 s. In moving, the plasmodium switches its direction or even multiplies in accordance with different biosignals attracting or repelling its motions, e.g. in accordance with pheromones of bacterial food, which attract the plasmodium, and high salt concentrations, which repel it. So, the plasmodium motions can be controlled by different topologies of attractants and repellents so that the plasmodium can be considered a programmable biological device in the form of a timed transition system, where attractants and repellents determine the set of all plasmodium transitions. Furthermore, we can define p-adic probabilities on these transitions and, using them, we can define a knowledge state of plasmodium and its game strategy in occupying attractants as payoffs for the plasmodium. As a result, we can regard the task of controlling the plasmodium motions as a game and we can design different interfaces in a game-theoretic setting for the controllers of plasmodium transitions.

Introduction

Conventionally, the intelligent behavior of animals is explained by their nervous system that coordinates voluntary and involuntary actions of animal's body and transmits signals between different parts of its body, which allows animals to act intentionally and efficiently. There is an approach in artificial intelligence, consisting in building computational models inspired by these nervous systems, that is called *artificial neural network*.

Nevertheless, there are one-cell organisms like *Physarum polycephalum* or *Badhamia utricularis* (supergroup Amoebozoa, phylum Mycetozoa, class Myxogastria) without any nervous system and they are able at their plasmodial stage to build complex networks for solving different tasks: maze-solving (Nakagaki, Yamada and Toth 2000), minimum-risk path finding (Nakagaki, Yamada and Toth 2001), (Nakagaki et al. 2007), associative learning (Shirakawa, Gunji and Miyake 2011), etc. In other words, *Physarum polycephalum* and *Badhamia utricularis* demonstrate an intelligent behavior with intentionality and efficiency, although they do not have nervous systems at all. In particular, they demonstrate the ability to memorize and anticipate repeated events (Saigusa et al. 2008). Furthermore, by means of plasmodium behavior, it is possible to simulate the behavior of some collectives such as collectives of parasites (Schumann, Akimova 2013). Thus, the complex intelligent behavior of plasmodium is biologically unexplained still and shows the limits of our understanding what natural intelligence is.

Now, there are many attempts to involve the plasmodium into semi-electrical devices to obtain a semi-biological and semi-electrical chip in due course (Sun et al. 2009), (Tsuda, Aono and Gunji 2004), (Tsuda et al. 2011), (Adamatzky 2010). The point is that the plasmodium spread by networks can be programmable and thereby it may simulate different intelligent processes. We are working on this problem, too (Adamatzky et al. 2012). In this paper, we are going to present our results in modelling the plasmodium networks as timed transition systems (Section 16.1). Propagations in these systems can be calculated by means of p-adic valued probabilities and fuzziness (Khrennikov, Schumann 2006), (Schumann 2008), (Schumann 2010), (see Section 16.2). In terms of these probabilities we can define a knowledge state of plasmodium and its game strategy in occupying attractants as payoffs for the plasmodium (Section 16.3). Hence, we can control the plasmodium motions as a game (Section 16.4). As a consequence, user interfaces for the controllers of plasmodium propagations can have a natural form of game-theoretic setting.

16.1. Timed transition systems for programming the plasmodium motions

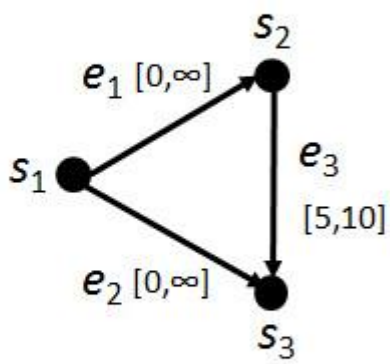
The plasmodium is an amorphous yellowish mass with networks of protoplasmic tubes, programmed by spatial configurations of attracting and repelling stimuli. Any motion of plasmodium proceeds from one stimuli to others. As a result, we deal with a kind of natural transition systems with states presented by attractants and events presented by plasmodium transitions between attractants. We can distinguish several operations (instructions) in the plasmodium networks like: add node, remove node, add edge, remove edge (Adamatzky 2010). Adding and removing nodes can be implemented through activation and deactivation of attractants, respectively. Adding and removing edges can be implemented by means of repellents put in proper places in the space. An activated repellent can avoid a plasmodium transition between attractants.

Adding and removing edges (in fact, adding and removing protoplasmic tubes) can change dynamically over time. To model such behavior, we have proposed to use timed transition systems as a high-level model of behavior of plasmodium. Let N be a set of nonnegative integers. Formally, a timed transition system $TTS = \langle S, E, T, s_0, l, u \rangle$ consists of the non-empty set of states S, the set of events E, the transition relations $T \subseteq S \times E \times S$, the initial state s_0 as well as a minimal delay function (a lower bound) $l : E \to N$ assigning a nonnegative integer to each event and a maximal delay function (an upper bound) $u : E \to N \cup \{\infty\}$ assigning a nonnegative integer or infinity to each event. Usually transition systems are based on actions which may be viewed as labelled events. If $\langle s, e, s' \rangle \in T$, then the idea is that TS can go from s to s' as a result of the event e occurring at s. In timed transition systems, timing constraints restrict the times at which events may occur. The timing constraints are classified into two categories: lower-bound and upper-bound requirements. A transition system can be presented as a graph structure with nodes corresponding to states and edges corresponding to transitions. In case of plasmodium, states represent attractants whereas edges represent protoplasmic tubes (plasmodium transitions between attractants).

To program computation tasks for the plasmodium propagations, we are developing a new object-oriented programming language (Schumann, Pancerz 2013) called the *Physarum language*, where the following three basic set descriptions are defined: (i) $TS.State$ – setting states of plasmodium, including initial states; (ii) $TS.Event$ – setting events transiting one states to others; (iii) $TS.Transition$ – setting transitions of plasmodium. The proposed language can be used for developing programs for plasmodium motions by the spatial configuration of stimuli.

Let us consider a simple timed transition system shown as a graph structure in figure 1 with the following timing constraints: $l(e_1) = 0$, $u(e_1) = \infty$, $l(e_2) = 0$, $u(e_2) = \infty$, $l(e_3) = 5$, $u(e_3) = 10$.

Figure 1. An example of timed transition system.



The code in the *Physarum* language has the following form:

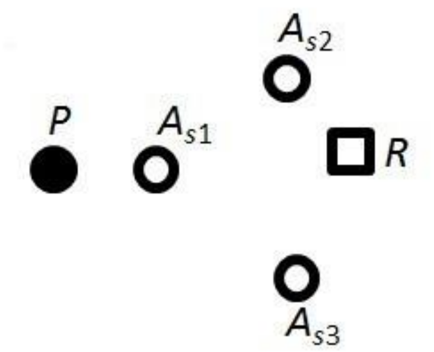
```
#TRANSITION_SYSTEM s1=new TS.State("s1");
s1.setAsInitial; s2=new TS.State("s2");
s3=new TS.State("s3");
e1=new TS.Event("e1");
t1=new TS.Transition(s1,e1,s2);
e2=new TS.Event("e2");
t2=new TS.Transition(s1,e2,s3);
e3=new TS.Event("e3");
e3.setTimingConstraints(5,10);
```

```
t3=new TS.Transition(s2,e3,s3);
```

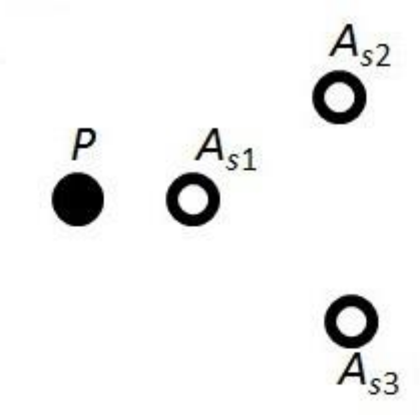
The default timing constraints are 0 as a lower bound and ∞ as an upper bound.

As a result of programming the plasmodium transitions, we obtain spatial configurations of stimuli presented in figure 2: (a) for the time instant $t = 4$, (b) for the time instant $t = 8$, where P is plasmodium, A_{s_1}, A_{s_2}, A_{s_3} are attractants, and R is a repellent. It is easy to see that the event e_3 is allowed only if actual time $t \in \{5,6,\ldots,10\}$. Therefore, in the model in figure 2(a), a repellent, avoiding the transition between states s_2 and s_3 as a result of the event e_3, is present, i.e. it is activated.

Figure 2. Spatial configurations of stimuli for the plasmodium motions.



(a)



(b)

16.2. p-Adic valued probabilities and fuzziness

We have supplemented our language with instructions enabling us to determine (in the simulation stage) possible properties of experiments in terms of the probability space:

- $setTimeStep$ – setting a time step from which the experiment starts, $t = 0,1,2,\ldots,n$,
- $setTimeEnd$ – setting a time end when the experiment stops, $t = 0,1,2,\ldots,\infty$,
- $getNeighCard$ – getting a cardinality number of neighboring attractants for a given attractant at the given $setTimeStep$ and $setTimeEnd$,
- $getAccessCard$ – getting a cardinality number of attractants accessible for a given attractant by protoplasmic tubes at the given $setTimeStep$ and $setTimeEnd$.

Instructions for the simulation stage are preceded with $\$$. Let us consider a simple timed transition system given earlier.

If we add the following instructions to the code:

```
$setTimeStep(0);
$setTimeEnd(10);
$getNeighCard(s2);
$getAccessCard(s2);
```

we obtain the following 3-adic streams:

- 22222222222 for getting a cardinality number of neighboring attractants for A_{s_2},
- 22222211111 for getting a cardinality number of attractants accessible for A_{s_2}.

Thus, if we have $p-1$ neighbor attractants for A_{s_2}, we deal with p-adic streams. If $setTimeStep(0)$ and $setTimeEnd(\infty)$ we deal with infinite *p*-adic streams. All these streams including both finite and infinite can be identified with *p*-adic integers. Let us recall that each *p*-adic integer has a unique expansion

$$n = \alpha_0 + \alpha_1 \cdot p + \ldots + \alpha_t \cdot p^t + \ldots = \sum_{t=0}^{\infty} \alpha_t \cdot p^t,$$

where $\alpha_t \in \{0,1,\ldots,p-1\}$, $\forall t \in N$. This number sometimes has the following notation: $n = \ldots\alpha_3\alpha_2\alpha_1\alpha_0$, where α_t can be interpreted as a value of α at time step $t = 0,1,2,\ldots,\infty$. We have used the latter notation in our example. The set of *p*-adic integers is denoted by $\mathbf{Z}_p$. For more details about *p*-adic numbers, please see (Koblitz, 1984). Now, *p*-adic analysis is used in many applications including quantum mechanics (Vladimirov, Volovich 1989), (Volovich 1987).

The set $\mathbf{Z}_p$ cannot be linearly ordered, but there are many possibilities to define a partial ordering relation. For example, we can assume that (i) for any finite *p*-adic integers $\sigma, \tau \in \mathbf{N}$, we have $\sigma \leq \tau$ in $\mathbf{N}$ iff $\sigma \leq \tau$ in $\mathbf{Z}_p$; (ii) each finite *p*-adic integer $n = \ldots \alpha_3 \alpha_2 \alpha_1 \alpha_0$ (i.e. such that $\alpha_i = 0$ for any $i > j$) is less than any infinite number τ, i.e. $\sigma < \tau$ for any $\sigma \in \mathbf{N}$ and $\tau \in \mathbf{Z}_p \setminus \mathbf{N}$; (iii) each infinite *p*-adic integer σ is less, than *p*-adic integer τ iff $\sigma_t \leq \tau_t$ for all t = 0, 1, 2, Let us denote this ordering relation by $\mathrm{O}_{\mathbf{Z}_p}$. We can see that there exist *p*-adic integers, which are incompatible by $\mathrm{O}_{\mathbf{Z}_p}$. For example, let p = 2 and let σ represents the *p*-adic integer $-1/3 =$ 10101 101 and τ the *p*-adic integer $-2/3 =$ 01010 010. Then the *p*-adic streams σ and τ are incompatible. Now we can define sup and $\inf$ digit by digit. Then if $\sigma \leq \tau$, so $inf(\sigma, \tau) = \sigma$ and $sup(\sigma, \tau) = \tau$. The greatest *p*-adic integer according to our definition is $-1 = \ldots xxxxxx$, where $x = p-1$, and the smallest is 0 = 00000.

Let us define the Boolean operations on attractants A_{s_i}, A_{s_j}, ... so that

$$getNeighCard(si \cap sj) :=$$

$$inf(getNeighCard(si), getNeighCard(sj));$$

$$getAccessCard(si \cap sj) :=$$

$$inf(getAccessCard(si), getAccessCard(sj));$$

$$getNeighCard(si \cup sj) :=$$

$$sup(getNeighCard(si), getNeighCard(sj));$$

$$getAccessCard(si \cup sj)$$

$$:= inf(getAccessCard(si), getAccessCard(sj));$$

$$getNeighCard(\neg si) := -1 - getNeighCard(si);$$

$$getAccessCard(\neg si) := -1 - getAccessCard(si).$$

Let Ω^* denote all attractants both activated and deactivated at each $t = 0, 1, 2, \ldots, \infty$. It is a union of all attractants A_{s_i}, A_{s_j}, ... at each time step. Its subsets will be denoted by $A^*, B^* \subseteq \Omega^*$.

Let us define *p*-adic fuzziness as follows: a *p-adic fuzzy measure* is a set function $F_{Z_p}(\cdot)$ defined for sets $A^*, B^* \subseteq \Omega^*$, it runs over the set $\mathbf{Z}_p$ and satisfies the following properties:

- $F_{Z_p}(\Omega^*) = -1$ and $F_{Z_p}(\varnothing^*) = 0$.
- If $A^* \subseteq \Omega^*$ and $B^* \subseteq \Omega^*$ are disjoint, i.e. $\inf(F_{Z_p}(A^*), F_{Z_p}(B^*)) = 0$, then $F_{Z_p}(A^* \cup B^*) = F_{Z_p}(A^*) + F_{Z_p}(B^*)$. Otherwise, $F_{Z_p}(A^* \cup B^*) = F_{Z_p}(A^*) + F_{Z_p}(B^*) - \inf(F_{Z_p}(A^*), F_{Z_p}(B^*)) = sup(F_{Z_p}(A^*), F_{Z_p}(B^*))$.
- If $A^*, B^* \subseteq \Omega^*$, then $F_{Z_p}(A^* \cap B^*) = inf(F_{Z_p}(A^*), F_{Z_p}(B^*))$.
- $F_{Z_p}(\neg A^*) = -1 - F_{Z_p}(A^*)$ for all $A^* \subseteq \Omega^*$, where $\neg A^* = \Omega^* \setminus A^*$.

A *p-adic probability measure* is a set function $P_{Z_p}(\cdot)$ defined for sets $A^*, B^* \subseteq \Omega^*$ thus:

- $P_{Z_p}(A^*) = -F_{Z_p}(A^*) \in \mathbf{Z}_p$
- $P_{Z_p}(A^* | B^*) \in \mathbf{Q}_p$ is characterized by the following constraint:

$$P_{Z_p}(A^* | B^*) = \frac{P_{Z_p}(A^* \cap B^*)}{P_{Z_p}(B^*)} = \frac{F_{Z_p}(A^* \cap B^*)}{F_{Z_p}(B^*)},$$

where $P_{Z_p}(B^*) \neq 0$, $P_{Z_p}(A^* \cap B^*) = \inf(P_{Z_p}(A^*), P_{Z_p}(B^*))$.

The measure $P_{Z_p}(\cdot)$ runs over the set $\mathbf{Q}_p$ of all *p*-adic numbers (not only integers). Notice that while $\mathbf{Z}_p$ is the ring of *p*-adic integers, $\mathbf{Q}_p$ is the field of *p*-adic numbers.

16.3. States of knowledge and strategies of plasmodium

Using *p*-adic valued fuzziness and probabilities, we can define games of plasmodia. So, in the given topology of attractants, active zones of plasmodia (initial states) can be considered players. Suppose, we have a set of N players, call them i = 1, ... , N. Agent i's *knowledge structure* is a function $\mathbf{P}_i$ which assigns to each attractant $\omega \in \Omega^*$ a non-empty subset of Ω^*, so that each thing ω belongs to one or more elements of each $\mathbf{P}_i$, i.e. Ω^* is contained in a union of $\mathbf{P}_i$, but $\mathbf{P}_i$ are not mutually disjoint. Then $\mathbf{P}_i(\omega)$ is called i's knowledge state at the attractant ω. This means that if the actual state is ω, the individual only knows that the actual state is in $\mathbf{P}_i(\omega)$.

We can interpret $\mathbf{P}_i(\omega)$ probabilistically as follows: $\mathbf{P}^i(\omega) = \{\omega': P^i_{Z_p}(\omega' \mid \omega) > 0\}$. Evidently that $P^i_{Z_p}(\omega \mid \omega) > 0$ for all $\omega \in \Omega^*$, therefore for all $\omega \in \Omega^*$, $\omega \in \mathbf{P}_i(\omega)$.

Now we consider the relation $A^* \subseteq \mathbf{P}_i(\omega)$, where $A^* \subseteq \Omega^*$, as the statement that at ω agent i *accepts the performance* A^*:

$$K_i A^* = \{\omega : A^* \subseteq \mathbf{P}_i(\omega)\}. \quad (1)$$

Let B_i^* mean ‘Attractants, which can be occupied by agent i’. After several steps, we expect fusions of all protoplasmic tubes so that all attractants are occupying by all agents. Does it mean that we observe a union of B_i^*? No, it does not. We face just the situation that since a time step $t = k$ the sets B_i^* are intersected. Let C_i^* mean ‘Attractants accessible for the attractant N_i by protoplasmic tubes’. Assume, $\omega \in B_i^*$ and $\omega' \in C_i^*$. Evidently, $P^i_{Z_p}(\omega' \mid \omega) > 0$. As a consequence, we assume according to our definitions that each agent i knows ω at ω' and knows ω' at ω, i.e. agent i accepts the performance B_i^* at ω' and i accepts the performance C_i^* at ω.

Let *getAcessSet*(*i*, *k*) be a set of all attractants such that i knows about them at the given *setTimeStep*(t_0) and *setTimeStep*(t_k). A strategy of a player i is a mapping $strat_{i,k} : getAccessSet(i,k) \to \Omega^*$ such that for any history knowledge *getAcessSet*(*i*, *k*) it is true that $strat_{i,k}$ belongs to the set of attractants accessible at k.

16.4. Game-theoretic interfaces for plasmodium

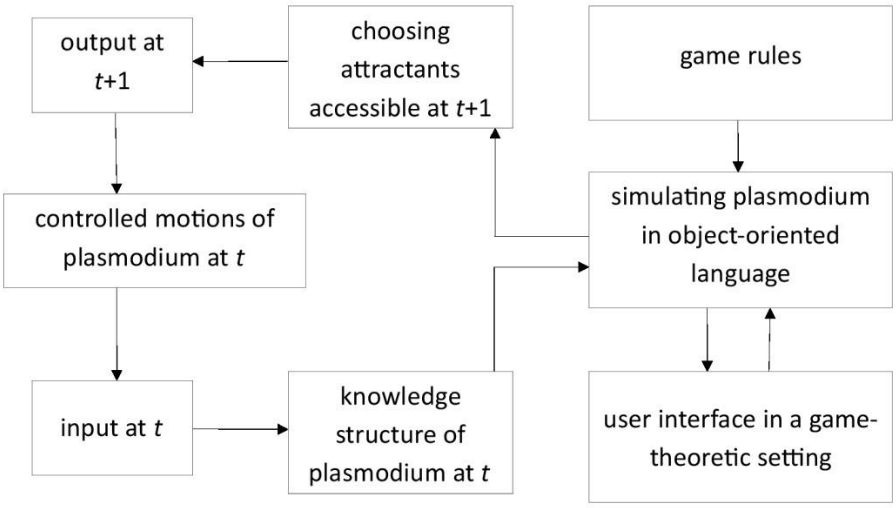
It is known due to the experiments performed by Andrew Adamatsky and Martin Grube that if there are only two agents of the plasmodium game, where the first agent is presented by a usual *Physarum polycephalum* plasmodium and the second agent by its modification called a *Badhamia utricularis* plasmodium, then both start to compete with each other. In particular, the *Physarum polycephalum* plasmodium grows faster and could grow into branches of *Badhamia utricularis*, while the *Badhamia utricularis* plasmodium could grow over *Physarum polycephalum* veins. So, we face an interesting form of zero-sum games.

The user interface for this game is designed on the basis of the following game steps:

- first, the system of *Physarum* language generates locations of attractants and repellents;
- second, we can chose *n* plasmodia/agents of *Physarum polycephalum* and *m* plasmodia/agents of *Badhamia utricularis*;
- third, we obtain the task, for example to reach as many as possible attractants or to construct the longest path consisting of occupied attractants, etc.;
- fourth, we can chose initial points for *Physarum polycephalum* transitions and initial points for *Badhamia utricularis* transitions;
- fifth, we start to move step by step;
- sixth, we define who wins, either *Physarum polycephalum* or *Badhamia utricularis*.

Thus, the plasmodium game has the form of cycle of figure 3.

Figure 3. The operative cycle of game-theoretic controller of plasmodium motions.



In this game, we have two players (the first plays for the *Physarum polycephalum* plasmodia, the second for the *Badhamia utricularis* plasmodia). The system places attractants and repellents automatically. Then the players choose which attractants are occupied before the game and which rules of the game hold (to reach as many as possible attractants or to construct the longest path consisting of occupied attractants, etc.). Then the system shows who wins and who loses.

Conclusion

The plasmodium motion is an intelligent way of constructing expanding networks for solving complex tasks. This motion has the form of transitions determined by locations of attractants and repellents. On these transitions, it is possible to define p-adic probabilities which are used for defining a knowledge state of plasmodium and its game strategy in occupying attractants as payoffs for the plasmodium. Consequently, the task of controlling the plasmodium motions is considered a game.

Acknowledgment

This research is supported by FP7-ICT-2011-8.

References

[1] Abramsky, S., Mellies, P.A., *Concurrent Games and Full Completeness, Proceedings of the 14th Symposium on Logic in Computer Science,* pp. 431–442, 1999.

[2] Adamatzky, A., *Physarum Machines: Computers from Slime Mould,* World Scientific Series on Nonlinear Science, Series A, World Scientific Publishing Company, 2010.

[3] Adamatzky, A., Erokhin, V., Grube, M., Schubert, Th., Schumann, A., *Physarum Chip Project: Growing Computers From Slime Mould*, Int. J. of Unconventional Computing 8(4): 319–323, 2012.

[4] Khrennikov, A., Schumann, A., *Logical Approach to p-adic Probabilities,* Bulletin of the Section of Logic 35(1): 49–57, 2006.

[5] Khrennikov, A., Schumann, A., *p-Adic Physics, Non-well-Founded Reality and Unconventional Computing*, P-Adic Numbers, Ultrametric Analysis and Applications 1(4): 297–306, 2009.

[6] Koblitz, N., *p-adic numbers, p-adic analysis and zeta functions,* Second edition, Springer-Verlag, 1984.

[7] Nakagaki, T., Yamada, H., Toth, A., *Maze-solving by an amoeboid organism,* Nature 407: 470–470, 2000.

[8] Nakagaki, T., Yamada, H., Toth, A., *Path finding by tube morphogenesis in an amoeboid organism,* Biophysical Chemistry 92: 47–52, 2001.

[9] Nakagaki, T., Iima, M., Ueda, T., Nishiura, Y., Saigusa, T., Tero, A., Kobayashi, R., Showalter, K., *Minimum-risk path finding by an adaptive amoeba network*, Physical Review Letters 99: 68–104, 2007.

[10] Neubert, H., Nowotny, W., Baumann, K., Marx, H., *Die Myxomyceten Deutschlands und des angrenzenden Alpenraumes unter besonderer Berücksichtigung Österreichs*, Bd. 2, Karlheinz Baumann Verlag, Gomaringen, 1995.

[11] Saigusa, T., Tero, A., Nakagaki, T., Kuramoto, Y., *Amoebae Anticipate Periodic Events*, Phys. Rev. Lett. 100(1): 018101, 2008.

[12] Shirakawa, T., Gunji, Y.P., Miyake, Y., *An associative learning experiment using the plasmodium of Physarum polycephalum*, Nano Communication Networks 2: 99–105, 2011.

[13] Schumann, A., *Non-Archimedean Fuzzy and Probability Logic*, J. of Applied Non-Classical Logics 18(1): 29–48, 2008.

[14] Schumann, A., *Non-Archimedean Valued Extension of Logic* $L\Pi$ *and p-Adic Valued Extension of Logic BL*, J. of Uncertain Systems 4(2): 99–115, 2010.

[15] Schumann, A., Akimova, L., *Simulating of Schistosomatidae (Trematoda: Digenea) Behavior by Physarum Spatial Logic*, Annals of Computer Science and Information Systems, vol. 1., *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, IEEE Xplore, pp. 225–230, 2013.

[16] Schumann, A., Pancerz, K., *Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing*, [in:] Szczuka, M., Czaja, L., Kacprzak, M., *Proceedings of the Workshop on Concurrency*, Specification and Programming (CS&P'2013), Warsaw, pp. 389–397, 2013.

[17] Sun, T., Tsuda, S., Zauner, K.P., Morgan, H., *Single cell imaging using electrical impedance tomography*, [in:] *NEMS '09: Proc. of the 2009 4th IEEE Int. Conf*, on Nano/Micro Engineered and Molecular Systems, pp. 858–863, 2009.

[18] Tsuda, S., Aono, M., Gunji, Y.P., *Robust and emergent Physarum-computing*, BioSystems 73: 45–55, 2004.

[19] Tsuda, S., Jones, J., Adamatzky, A., Mills, J., *Routing Physarum with electrical flow/current*, Int. J. Nanotechnology and Molecular Comput, 2011.

[20] Vladimirov, V.S., Volovich, I.V., *p-Adic Quantum Mechanics*, Commun. Math. Phys, 123: 659–676, 1989.

[21] Volovich, I.V., *Number theory as the ultimate theory*, CERN preprint, CERN-TH.4791/87, 1987.

Andrew Schumann (University of Information Technology and Management in Rzeszow, Poland) worked at the Belarusian State University, Minsk, Belarus. His research focuses on logic and philosophy of science with an emphasis on non-well-founded phenomena: self-references and circularity. He contributed mainly to research areas such as reasoning under uncertainty, probability reasoning, non-Archimedean mathematics, as well as their applications to cognitive science. He is engaged also in unconventional computing, decision theory, logical modeling of economics.

Krzysztof Pancerz received his M.Sc. in 1998 from the Rzeszow University of Technology in Electrical Engineering and Ph.D. in 2006 from the Institute of Computer Science, the Polish Academy of Sciences in Computer Science. His research interests concern computational intelligence, knowledge engineering, unconventional computing, and computer-aided diagnosis. He has published over 70 research papers in international journals, monographs and conference proceedings.

Physarum Machines. Selected Works

The slime mould of *Physarum polycephalum* is a giant single cell that contains many nuclei. This organism is capable for distributed sensing, concurrent information processing, and parallel computation. The project *Physarum Chip: Growing Computers from Slime Mould* we carried out is focused on theoretical and experimental laboratory studies on sensing and computing properties of slime mould, and development of mathematical and logical theories of plasmodial behaviour. This book contains some selected papers prepared within the *Physarum Chip* project.