Studia Humana
Volume 5:1 (2016), pp. 16—30
DOI: 10.1515/sh-2016-0002

Logics for Physarum Chips

Andrew Schumann

University of Information Technology and Management in Rzeszow
Rzeszow, Poland

e-mail: andrew.schumann@gmail.com

Krzysztof Pancerz

University of Rzeszow,
Rzeszow, Poland

e-mail: kkpancerz@gmail.com

Abstract:
The paper considers main features of two groups of logics for biological devices, called *Physarum* Chips, based on the plasmodium. Let us recall that the plasmodium is a single cell with many diploid nuclei. It propagates networks by growing pseudopodia to connect scattered nutrients (pieces of food). As a result, we deal with a kind of computing. The first group of logics for *Physarum* Chips formalizes the plasmodium behaviour under conditions of nutrient-poor substrate. This group can be defined as standard storage modification machines. The second group of logics for *Physarum* Chips covers the plasmodium computing under conditions of nutrient-rich substrate. In this case the plasmodium behaves in a massively parallel manner and propagates in all possible directions. The logics of the second group are unconventional and deal with non-well-founded data such as infinite streams.
Keywords: Physarum polycephalum, unconventional computing, modern Voronoi diagram, p-adic valued logic, Petri nets.

1. Introduction

Physarum Chip is a biological sensing and computing device implemented in vegetative stage of acellular slime mould *Physarum polycephalum* [1]. Notice that *Physarum polycephalum* belongs to the species of order *Physarales*, subclass *Myxogastromycetidae*, class *Myxomycetes*, division *Myxostelida*. Plasmodium is its ‘vegetative’ phase represented as a single cell with a myriad of diploid nuclei. The *Physarum* Chips, designed in our project *Physarum Chip Project: Growing Computers From Slime Mould* supported by FP7, are 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.

The *P. polycephalum* plasmodium 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. In the project we have proposed several logical methods for designing the *Physarum* Chips: conventional as well as unconventional. In this paper, we have generalized our results obtained for the whole period of the three-year project [1] – [42].

In Section II we consider conventional logics which are applicable for the case of nutrient-poor substrate of plasmodium feeding. In Section III we examine unconventional logics which can program the plasmodium behaviour in the case of nutrient-rich substrate. In Section IV we enumerate mathematical tools used in our formalization of *Physarum* computing. In Section V we propose a game-theoretic interface for *Physarum* computing.

2. Computing on Nutrient-Poor Substrate

2.1. Storage Modification Machines and Object-oriented Programming Language

Let us assume that the plasmodium feeds under the condition of *nutrient-poor substrate*. In this case the plasmodium can distinguish all attractants and repellents, and, as a result, those attractants and repellents involved in the stimulation of plasmodium gives a topology which can be defined as a *Voronoi diagram* [10], [19]. Within one Voronoi cell a reagent has a full power to attract or repel the plasmodium. In other words, within this cell the reagent determines the plasmodium behaviour completely. The distance for the cell 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. 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. Within the same Voronoi cell two active zones will fuse.

Now, suppose that the plasmodium feeds under the condition of *nutrient-rich substrate*. This means that for the plasmodium there is an information noise and the plasmodium cannot define where precisely attractants and repellents are located indeed.

Hence, in the case of nutrient-poor substrate with well distinguished localizations of attractants and repellents we can fully manage the plasmodium behaviour and propose a biological version of *storage modification machines*. These machines are defined in a new *object-oriented programming language* designed by us for *Physarum polycephalum* computing (OPL-Ph), [28], [29], [39], [42]. 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 [31] and game theory [8], [30]. The proposed OPL-Ph can be used for developing programs for *P. polycephalum* by the spatial configuration of stationary nodes. Geometrical distribution of stimuli can be identified with a low-level programming language for *Physarum* machines.

2.2. Programmable Logic Controllers

At the beginning, we have proposed to construct logic gates through the proper geometrical distribution of stimuli for *P. polycephalum*. This approach has been adopted from the ladder diagram language widely used to program *Programmable Logic Controllers*. Flowing power has been replaced with propagation of plasmodium of *P. polycephalum*. Plasmodium propagation is stimulated by attractants and repellents. 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.

In the proposed approach, we have assumed that each attractant (repellent) is characterized by its region of influence 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. The 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. Controlling the plasmodium propagation is realised by activating/deactivating stimuli.

Logic values for inputs have the following meaning in terms of states of stimuli: 0 – attractant/repellent deactivated, 1 – attractant/repellent activated. Logic values for outputs have the following meaning in terms of states of stimuli: 0 – absence of *P. polycephalum* at the attractant, 1 – presence of *P. polycephalum* at the attractant.

2.3.Petri Nets

At the second stage, we have adopted more abstract models than distribution of stimuli to program *P. polycephalum* machines which can be identified with programming in the high-level language. The choice fell on *Petri nets*. Petri nets were first developed by C.A. Petri. Petri nets are a powerful graphical language for describing processes in digital hardware. We have shown how to build Petri net models, and next implement as *P. polycephalum* machines, of basic logic gates AND, OR, NOT, and simple combination circuits [25], [29]. In our approach, we use Petri nets with inhibitor arcs. Inhibitor arcs are used to disable transitions. Inhibitor arcs test the absence of tokens in a place. A transition can only be if all its places connected through inhibitor arcs are empty. This ability of Petri nets with inhibitor arcs is used to model behaviour of repellents. Plasmodium of *Physarum* avoids light and some thermo- and salt-based conditions and this fact can be modelled by inhibitor arcs. The Petri net model (code in the high-level language) can be translated into the code in the low-level language, i.e., geometrical distribution of attractants and repellents of the *Physarum* machine.

In the proposed Petri net models, we can distinguish three kinds of places:

(1) Places representing *P. polycephalum*. For such a place, the presence of the token means that plasmodium of *Physarum* is present in the origin point. Otherwise, the absence of the token means that there is no plasmodium.

(2) Places representing input attractants or repellents. For such a place, the presence of the token means that the attractant/repellent is activated. Otherwise, the absence of the token means that attractant/repellent is deactivated.

(3) Places representing output attractants. For such a place, the presence of the token denotes the present of *Physarum* at the attractant (*Physarum* occupies the attractant). Otherwise, the absence of the token denotes the absence of *Physarum polycephalum* at the attractant.

In the *AND gate*, the transitions *T* represents the flow (propagation) of plasmodium from the origin place to the output attractant. *T* is enabled to fire if both attractants are activated, Fig.1.

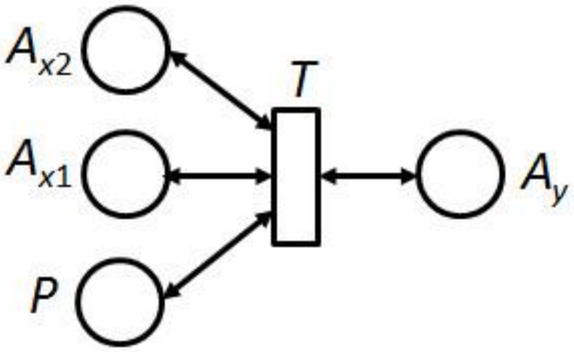

Fig. 1. AND gate

In the *OR gate*, the transitions T_1 and T_2 represent the alternative flows of plasmodium from the origin place to the output attractant. T_1 is enabled to fire if the first attractant is activated. T_2 is enabled to fire if the second attractant is activated, Fig.2.

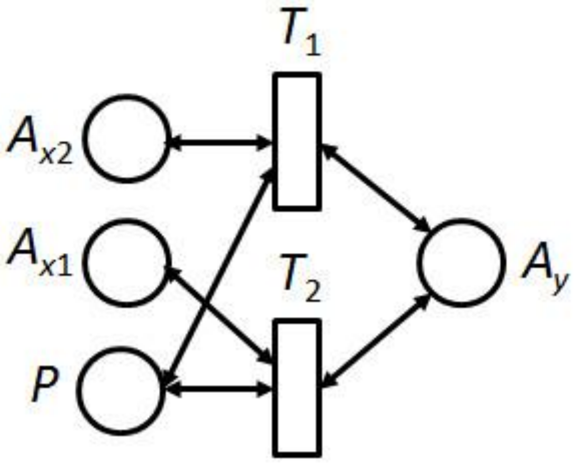

Fig. 2. OR gate

In the *NOT gate*, the transition T represents the flow (propagation) of plasmodium from the origin place to the output attractant. T is enabled to fire if the repellent is deactivated, Fig.3.

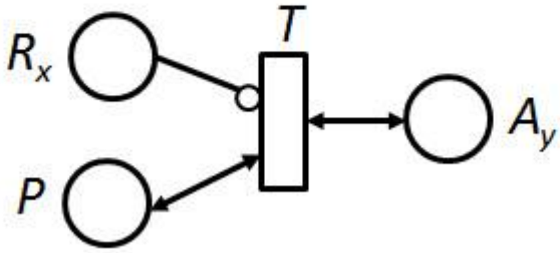

Fig. 3. NOT gate

Then, experimentally, we have built a *P. 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. So, using the ladder diagram approach, we can determine geometrical distribution of attractants and repellents for the 1-to-2 demultiplexer [32].

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.

2.4.Concurrent Games

In our OPL-Ph we have analysed biological mechanisms for *Physarum* decision making and reconstructed the so-called *concurrent games* of plasmodia [30]. So, attractants may be regarded as *payoffs* for *Physarum*. Plasmodia occupy attractants step by step. By different localizations of attractants we can affect on *Physarum* motions differently. We can interpret the stimuli for *Physarum* motions as Boolean functions on payoffs. Boolean functions are designed by logical gates mentioned above (Fig.1–3).

In this way we have designed the zero-sum game between plasmodia of *Physarum polycephalum* and *Badhamia utricularis,* the so-called *PHY game* [15], [18], [23], [30], [33], [38], [40]. To simulate *Physarum* games, we have created a specialized software tool. 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 OPL-Ph. 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 OPL-Ph 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*.

Strategies in games between *Physarum polycephalum* and *Badhamia utricularis* are described by rough sets defined on transition systems [26], [38], [40].

We claim that bio-inspired games might wake new interests in designing new games and new game platforms [33].

Thus, *Physarum* computing on nutrient-poor substrate can be considered storage modification machines and these machines are well formalizable in OPL-Ph. In this language we can define Petri nets and concurrent games. This language can be applied as programming language for *Physarum* logic circuits on nutrient-poor substrate. Hence, the programming language for *Physarum* behaviour we have constructed, on the one hand, simulates the *Physarum* behaviour and, on the other hand, shows which mathematical tools can be implemented in its behaviour.

3. Computing on Nutrient-Rich Substrate

3.1.Non-Aristotelian Extensions of Storage Modification Machines

Notice that under the conditions of nutrient-rich substrate, storage modification machines on plasmodia cannot be conventional. The point is that under these conditions we cannot approximate elementary (atomic) acts, i.e. we deal with a massive-parallel behaviour of plasmodia (they are to be expanded in all possible directions) [6], [17], [20]. Hence, we must extend our OPL-Ph by some unconventional tools to make OPL-Ph applicable for the case of nutrient-rich substrate, also.

For fixing *Physarum* computing on nutrient-rich substrate, we have constructed a *non-Aristotelian syllogistic* (*performative syllogistic*) [9], [10], [15], [19], [36] that models *Physarum* simultaneous propagations in all directions (i.e. it is massive-parallel). 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) [36]. Also, this syllogistic system of propagation can be used as basic logical theory for quantum logic (without logical atoms) [21]. 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 can 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 [3], [14].

Notice that the Aristotelian syllogistic is implementable for *Physarum* computing on nutrient-poor substrate, while performative syllogistic is applicable for *Physarum* computing on nutrient-rich substrate [9], [10], [15], [19], [36]. Performative syllogistic is an extension of Aristotelian one and includes the latter as its own part.

3.2.Reflexive Games

We have constructed an unconventional logic, which deals with non-well-founded data, namely with infinite streams and wave sets (sets of mutually defined infinite streams) [8], [17], [30]. 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 the *Physarum* propagations with much localization of plasmodia.

Within the unconventional logic, we have introduced the notion of *payoff cellular automata* [8] 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 have 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.

We have disproven the Aumann's agreement theorem (the so-called reflexion disagreement theorem) [11], [13]. 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 infinite reflexive level). And these results are used by us for formalizing the game theory of plasmodia. Richer substrate due nutrients, more reflexive plasmodium game.

3.3.Theories on Hybrid Actions

We have defined an extension of process algebra, where simple actions of labelled transition systems cannot be atomic; consequently, their compositions cannot be inductive [8]. 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.

The group theory proposed by us can be used as the new design method to construct reversible logic gates on plasmodia [5], [14]. In this way, we should appeal to the so-called *non-linear permutation groups* [5], [8]. 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.

Richer substrate due nutrients, more hybrid plasmodium action.

3.4.Neural Properties of Slime Mould and Modal Logics

There are two main properties of neural networks: lateral activation and lateral inhibition. The same properties are observed in *Physarum polycephalum* networks. We have generalized our studies of lateral inhibition effects in *P. polycephalum* behaviour in the way of constructing new syllogistics and modal logics. So, we have shown that there are two main possibilities of pairwise comparisons analysis in computer science: first, pairwise comparisons within a lattice, in this case these comparisons can be measurable by numbers (this one corresponds to lateral inhibition); second, comparisons beyond any lattice, in this case these comparisons cannot be measurable in principle (this one corresponds to lateral activation of neural networks). We have shown that the first approach to pairwise comparisons analysis is based on the conventional square of opposition and its generalization, but the second approach is based on unconventional squares of opposition [22].

Furthermore, the first approach corresponds to lateral inhibition in transmission signals and the second approach corresponds to lateral activation in transmission signals. For combining lateral inhibition and lateral activation in the same behaviour we introduced 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.

Under the conditions of nutrient-rich substrate the plasmodium has lateral activation effects and under the conditions of nutrient-poor substrate the plasmodium has lateral inhibition effects. We have shown that modal logic **K** corresponds to the lateral inhibition property and we can construct new modal logics, alternative to **K**, for embodying the lateral activation property [22].

Thus, we have extended our OPL-Ph by adding new tools: performative syllogistic, reflexive games, theories on hybrid actions, and modal logics for lateral activation, which allow us to simulate *Physarum* computing on nutrient-rich substrate.

4. Formalisation of Storage Modification Machines

In designing logics in OPL-Ph for *Physarum* Chips we have used the following mathematical tools:

4.1.Process Algebra

In OPL-Ph we have been based on *process-algebraic* formalizations of *Physarum* storage modification machines [34]. So, we have considered some instructions in *Physarum* machines in terms of process algebra like: add node, remove node, add edge, remove edge. 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 behaviour, we have proposed a high-level model, based on timed transition systems [26], [27].

In this model we have defined the following four basic forms of *Physarum* transitions (motions): direct (direction: 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 plasmodium from one active point into two active points, where two neighbouring attractants with a similar power of intensity are located), and repel (repelling of plasmodium or inaction).

In *Physarum* motions, we can perceive some ambiguity influencing on exact anticipation of states of *Physarum* machines in time. In case of splitting plasmodium, there is some uncertainty in determining next active points (attractants occupied by plasmodium), if a given active point is known. This uncertainty does not occur in case of direction, where the next active point is uniquely determined. To model ambiguity in anticipation of states of *Physarum* machines, we propose to use rough set theory [26], [38], [40], [41]. Analogously to the lower and upper approximations, we define the lower and upper predecessor anticipations of states in the *Physarum* machine. Behaviour of *Physarum* machines can also be modelled using Bayesian networks with probabilities defined on rough sets [41].

Thus, we have proposed some timed and probabilistic extensions of standard process algebra to implement timed and rough set models of behaviour of *Physarum* machines in OPL-Ph.

4.2.Computation on Trees

Computations on tree are usually represented by spatial algorithms like Kolmogorov-Uspensky machines. Theoretically, Turing machines, Kolmogorov-Uspensky machines, Schönhage’s storage modification machines, and random-access machines have the same expressibility power. In other words, the class of functions computable by these machines is the same. Unfortunately, the computational power of their implementations on the *Physarum polycephalum* medium is low.

The point is that not every computable function can be simulated by plasmodium behaviours: first, the motion of plasmodia is too slow (several days are needed to compute simple functions such as 5-bit conjunction, 3-bit adder, etc., but the plasmodium stage of *Physarum polycephalum* is time-limited, therefore there is not enough time for realizations, e.g., of thousands-bit functions); second, the more attractants or repellents are involved in designing computable functions, the less accuracy of their implementation is, because the plasmodium tries to be propagated in all possible directions and we will deal with indirected graphs and other problems; third, the plasmodium is an adaptive organism that is very sensitive to environments, therefore it is very difficult to organize the same laboratory conditions for calculating the same *k*-bit functions, where *k* is large.

To make computations on tree more expressive we have proposed the performative syllogistic – syllogistic system of propagation [9], [10], [15], [19]. This system can logically simulate a massive-parallel behaviour in the propagation of collectives of parasites [36]. So, protoplasmic trees can be interpreted as syllogistic trees. In this way while Aristotelian syllogistic may describe concrete directions of *Physarum* spatial expansions, performative syllogistic proposed by us 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 performative syllogistic we do not need them. Performative syllogistic has a p-adic valued semantics and satisfies p-adic valued probabilities [6], [15]. The syllogistic can be extended to a more general theory of context-based games. This theory is proposed in [8]. Within this theory we can define algorithms for computing on protoplasmic tree.

Computations on protoplasmic tree are understood as a kind of extension of concurrent processes defined in concurrent games proposed by Samson Abramsky [30]. This extension is called context-based processes and they are defined in the theory of *context-based games* proposed by us.

Thus, we have defined some unconventional algorithms on non-well-founded trees to make calculations on plasmodia more expressive. These algorithms can be used for constructing an alternative quantum logic (without logical atoms) [21] and a simulation model for propagating parasites [35].

4.3.Cellular Automata

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 neighbourhood of the set's elements in accordance with the spreading of attractants or repellents. As a result, we obtain Voronoi cells. In this structure we can implement cellular automata [19]. Taking into account the fact that the plasmodium is very sensitive to the environment and can change its strategies we can extent the standard notion of cellular automata and assume that transition rules can change in an appropriate neighbourhood in accordance with some outer conditions at time step $t = 0, 1, 2...$ The theory of these automata with changing transition rules is proposed in [8], [17].

The plasmodium implements *cellular automata with changing transition rules*. Within these automata we can define context-based concurrent formal theories [5], [14], [17].

4.4.p-Adic Logic and Arithmetics

The slime mould is considered a natural fuzzy processor with fuzzy values on the set of p-adic integers [6], [7], [20], [24]. 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, $t = 0, 1, 2, \dots$, 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 syllogistics we constructed for describing the propagation of the slime mould [6]. This syllogistics can be extended to a *p-adic valued logic* and *p-adic valued arithmetics* [7], [20], [24]. Within this logic we can develop a context-based game theory [8], [30]. All these logical tools can be implemented on plasmodia by conventional and unconventional reversible logic gates [3], [14].

We have proposed to use p-adic valued fuzziness and probabilities for measuring behaviours which cannot be measured additively [11], [13]. Then we have constructed a natural deductive system for describing all possible experiments with *P. polycephalum* [7]. This system is p-adic many valued. We have considered possibilities for applying p-adic valued logic **BL** to the task of designing the *Physarum* Chips. If it is assumed that at any time step t of propagation the slime mould can discover and reach not more than $p - 1$ attractants, then this behaviour can be coded in terms of p-adic numbers. As a result, this behaviour implements some p-adic valued arithmetic circuits and can verify p-adic valued logical propositions.

We have offered two unconventional arithmetic circuits: adder and subtractor defined on finite p-adic integers [24]. Adder and subtractor are designed by means of spatial configurations of several attractants and repellents which are stimuli for the plasmodium behaviour. As a result, the plasmodium could form a network of protoplasmic veins connecting attractants and original points of the plasmodium. Occupying new attractants is considered in the way of adders and leaving some attractants because of repelling is considered in the way of subtracters. On the basis of p-adic adders and subtractors we can design complex p-adic valued arithmetic circuits within a p-adic valued logic proposed by us.

So, p-adic valued logic and p-adic valued arithmetic are implementable on plasmodia. In the meanwhile, *Physarum* computing on nutrient-poor substrate is expressible by finite p-adic integers and *Physarum* computing on nutrient-rich substrate is expressible by infinite p-adic integers.

4.5.Non-well-founded Formal Theories

We have proposed two *formal theories on hybrid actions* (non-well-founded data): group theory and Boolean algebra [17]. Both theories possess many unusual properties such as the following one: the same member of this group theory behaves as multiplicative zero in respect to some members and as multiplicative unit in respect to other members. This group theory is used to design reversible logic gates on plasmodia,

Hence, to formalize *Physarum* computing we have proposed new formal theories such as theory of non-linear permutation groups to design unconventional reversible logic gates [5], [14].

4.6.Reversible Logic Gates

We have considered different ways of designing reversible logic gates on *P. polycephalum* motions using controlling stimuli such as attractants and repellents [3], [5], [14]. Repellents are needed because of uncertainty in the direction of plasmodium propagation to eliminate some directions as unimportant. In this way, we can construct conventional reversible logic gates: the CNOT gate, the FREDKIN gate, the TOFFOLI gate, etc. Combinations of reversible logic gates are regarded as matrix multiplications. Nevertheless, the plasmodium in its networking can permanently change its decisions and without repellents we have an extension of matrix multiplication group theory. Within this extension we can design unconventional reversible logic gates, where the number of inputs and outputs is uncertain. For designing logic gates we have proposed 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).

4.7.Actin Filament Networks

Plasmodia consist mainly of actin filaments. We have proposed *artificial actin filament networks* where inputs are different stresses and outputs are formations and destructions of filaments, on the one hand, and as assemblies and disassemblies of actin filament networks, on the other hand [16]. Hence, under different external conditions we observe dynamic changes in the length of actin filaments and in the outlook of filament networks. As we see, the main difference of actin filament networks from others including neural networks is that the topology of actin filament networks changes in responses to dynamics of external stimuli. Some new filaments/processors can appear in one conditions and they can disappear in other conditions.

4.8.Swarm Computing

By modelling the plasmodium behaviour in the *Physarum* Chips we can simulate some patterns of collective intelligent behaviours of animal or insect groups: flocks of birds, colonies of ants, schools of fish, swarms of bees, etc. for which there are ever emergent patterns which cannot be reduced to a linear composition of elementary subsystems properly [2], [4], [31]. In swarm intelligence the Travelling Salesman Problem can be solved: more shorter distance between cities (pieces of food for the plasmodium), more attracting they are, as well as the Generalized Assignment Problem can be solved: the tubes of the plasmodium are regarded as agents, the nutrient sources as tasks, the amount of nutrient as profit, and the distance as cost. We show that by using p-adic integers we can code different emergent patterns so that the implementation of some unconventional algorithms of p-adic arithmetics and logic can be more applicable than conventional automata.

5. Game-Theoretic Interface for Storage Modification Machine

5.1.Chemical Interface

In moving, the plasmodium switches its direction or even multiplies in accordance with different bio signals 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.

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 by chemical signals [23].

5.2.Bio-inspired Game Theory

We have proposed a bio-inspired game theory on plasmodia, 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 OPL-Ph [15], [18], [23], [30], [33], [38], [40]. Our results allow us to claim that the slime mould can be a model for concurrent games and context based games. 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:

- Attractants as payoffs;
- Attractants occupied by the plasmodium as states of the game;
- Active zones of plasmodium as players;
- Logic gates for behaviours as moves (available actions) for the players;
- 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 the *Physarum* 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 behaviour can be formalized within strong extensions of spatial algorithms, e.g. within concurrent games or context-based games.

5.3.Go Games

In the universe of 5-adic integers, we have simulated the motions of *P. polycephalum* plasmodium by the game of Go [15], [18], [38]. We have considered two syllogistic systems implemented as Go games: the Aristotelian syllogistic and performative syllogistic. In the Aristotelian syllogistic, the locations of black and white stones are understood as locations of attractants and repellents, respectively. In the performative syllogistic, we consider the locations of black stones as locations of attractants occupied by plasmodia of *P. polycephalum* and the locations of white stones as locations of attractants occupied by plasmodia of *Badhamia utricularis*. The Aristotelian syllogistic version of Go game is a coalition game. The performative syllogistic version of Go game is an antagonistic game. We described 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. We proposed a rough set approach for description of a strategy game created on the *Physarum* machine. The strategies of such a game are approximated on the basis of a rough set model, describing behavior of the *Physarum* machine, created according to the VPRSM (Variable Precision Rough Set Model) approach [38].

Thus, we have proposed game-theoretic interface for *Physarum* Chips.

Conclusion

We have built up two basic groups of logics for the *Physarum* Chips: the first ones deal with the plasmodium behaviour under the conditions of nutrient-poor substrate and the second one deal with the plasmodium behaviour under the conditions of nutrient-rich substrate. The first logics can be conventional (classical, multi-valued, modal, or fuzzy), while the second are strong extensions of the first logics and cannot be conventional and they engage the so-called non-well-founded data such as infinite streams.

Acknowledgment
This research is supported by FP7-ICT-2011-8. We would like to express our especially warm gratitude to Andy Adamatzky, the leader of the project – the collaboration with whom was very interesting and promising for us.

References

1. A. Adamatzky, V. Erokhin, M. Grube, T. Schubert, A. Schumann, "Physarum Chip Project: Growing Computers From Slime Mould," *International Journal of Unconventional Computing*, vol. 8, no. 4, pp. 319-323, 2012.
2. A. Schumann, "Conventional and Unconventional Approaches to Swarm Logic," in *Advances in Unconventional Computing*, A. Adamatzky, Ed., Springer 2016, submitted for publication.
3. A. Schumann, "Conventional and unconventional reversible logic gates on Physarum polycephalum," *International Journal of Parallel, Emergent and Distributed Systems*, 2015, DOI: 10.1080/17445760.2015.1068775.
4. A. Schumann, "From Swarm Simulations to Swarm Intelligence," in *9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, ACM, 2015, submitted for publication.
5. A. Schumann, "Non-Linear Permutation Groups on Physarum Polycephalum," in *ICSAI 2014*, IEEE Xplore, 15-17 Nov. 2014, pp. 246–251.
6. A. Schumann, "p-Adic Valued Fuzzyness and Experiments with Physarum Polycephalum," in *FSKD 2014*, IEEE Xplore, 19-21 Aug. 2014, pp. 466–472.
7. A. Schumann, "p-Adic valued logical calculi in simulations of the slime mould behaviour," *Journal of Applied Non-Classical Logics*, 2015, DOI: 10.1080/11663081.2015.1049099.
8. A. Schumann, "Payoff Cellular Automata and Reflexive Games," *Journal of Cellular Automata*, vol. 9, no. 4, pp. 287–313, 2014.
9. A. Schumann, "Physarum Polycephalum Syllogistic L-Systems and Judaic Roots of Unconventional Computing," *Studies in Logic, Grammar and Rhetoric*, vl. 44, no. 57, pp. 181–201, 2016, DOI: 10.1515/slgr-2016-0011.
10. A. Schumann, "Physarum Syllogistic L-Systems," in *FUTURE COMPUTING 2014, The Sixth International Conference on Future Computational Technologies and Applications*, 2014.
11. A. Schumann, "Probabilities on Streams and Reflexive Games," *Operations Research and Decisions*, vol. 24, no. 1, pp. 71 – 96, 2014.
12. A. Schumann, "Proof-Theoretic Cellular Automata as Logic of Unconventional Computing," in *International Journal of Unconventional Computing*", vol. 8, no. 3, pp. 263–280, 2012.
13. A. Schumann, "Reflexive Games and Non-Archimedean Probabilities," *p-Adic Numbers, Ultrametric Analysis and Applications,* vol. 6, no. 1, pp. 66–79, January 2014.
14. A. Schumann, "Reversible Logic Gates on Physarum Polycephalum," in *International Conference of Numerical Analysis and Applied Mathematics*, September 22-28, Special Volume of the AIP Conference Proceedings 2014.
15. A. Schumann, "Syllogistic Versions of Go Games on Physarum Polycephalum," in *Advances in Physarum Machines,* A. Adamatzky, Ed., vol. 21 of the series Emergence, Complexity and Computation, Springer 2016, pp. 651–685.
16. A. Schumann, "Toward A Computational Model of Actin Filament Networks," in *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*, SCITEPRESS, 2016, submitted for publication.
17. A. Schumann, "Towards context-based concurrent formal theories," *Parallel Processing Letters,* vol. 25, 2015, 1540008.
18. A. Schumann, A. "Go Games on Plasmodia of Physarum Polycephalum," in *Annals of Computer Science and Information Systems, Proceedings of the Federated Conference on Computer Science and Information Systems*, IEEE Xplore, 2015, pp. 607–614.

19. A. Schumann, A. Adamatzky, “Physarum Polycephalum Diagrams for Syllogistic Systems,” *IfCoLog Journal of Logics and their Applications*, vol. 2, no. 1, pp. 35–68, 2015.
20. A. Schumann, A. Adamatzky, “The Double-Slit Experiment with Physarum Polycephalum and p-Adic Valued Probabilities and Fuzziness,” *International Journal of General Systems*, 27 Jan 2015, DOI: 10.1080/03081079.2014.997530.
21. A. Schumann, A. Khrennikov, “Quantum Non-Objectivity from Performativity of Quantum Phenomena,” *Physica Scripta*, no. T163, pp. 15, 2014.
22. A. Schumann, J. Woleński, “Two Squares of Oppositions and Their Applications in Pairwise Comparisons Analysis,” *Fundamenta Informaticae*, 2016, submitted for publication.
23. A. Schumann, K. Pancerz, “Interfaces in a Game-Theoretic Setting for Controlling the Plasmodium Motions,” in *BIOSTEC (Biosignals),* Scitepress 2014.
24. A. Schumann, K. Pancerz, “p-Adic Computation with Physarum,” in *Advances in Physarum Machines*, A. Adamatzky, Ed., vol.21 of the series Emergence, Complexity and Computation, Springer, 2016, pp. 619–649.
25. A. Schumann, K. Pancerz, “Petri Net Models of Simple Rule-Based Systems for Programming Physarum Machines,” in *Proceedings of the 24th International Workshop on Concurrency, Specification and Programming*, Z. Suraj, L. Czaja, Eds., CEUR Workshop Proceedings 1492, CEUR-WS.org, September 28-30, Rzeszow - Poland 2015, pp. 155–160.
26. A. Schumann, K. Pancerz, “Roughness in Timed Transition Systems Modeling Propagation of Plasmodium,” in *Rough Sets and Knowledge Technology - 10th International Conference, RSKT, held as part of the International Joint Conference on Rough Sets, IJCRS, November 20-23, Proceedings,* D.Ciucci, G. Wang, S. Mitra, W.Wei-Zhi, Eds., Lecture Notes in Computer Science 9436, Springer, *Tianjin - China* 2015, pp. 482–491.
27. A. Schumann, K. Pancerz, “Timed Transition System Models for Programming Physarum Machines: Extended Abstract”, in *Proceedings of the Workshop on Concurrency,* L. Popova-Zeugmann, Ed., Specification and Programming (CS&P), Chemnitz, September 29 - October 1, Germany 2014, pp. 180–183.
28. A. Schumann, K. Pancerz, “Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing,” in *Proceedings of the Workshop on Concurrency,* Specification and Programming (CS&P), M. Szczuka, L.Czaja, M. Kacprzak, Eds. September 25-27, Warsaw - Poland 2013, , pp. 389–397.
29. A. Schumann, K. Pancerz, “Towards an Object-Oriented Programming Language for Physarum Polycephalum Computing: A Petri Net Model Approach,” *Fundamenta Informaticae*, vol. 133, no. 2-3, pp. 271–285, 2014.
30. A. Schumann, K. Pancerz, A. Adamatzky, M. Grube, “Bio-Inspired Game Theory: The Case of Physarum Polycephalum,” in *BICT*, ACM, 1-3 Dec. 2014.
31. A. Schumann, K. Pancerz, A. Szelc, “The Swarm Computing Approach to Business Intelligence,” *Studia Humana*, vol. 4, no. 3, pp. 41–50, 2015, DOI: 10.1515/sh-2015-0019.
32. A. Schumann, K. Pancerz, J. Jones, “Towards Logic Circuits Based on Physarum Polycephalum Machines: the Ladder Diagram Approach,” in *Proceedings of the 7th International Conference on Biomedical Electronics and Devices (BIODEVICES’2014)*, A. Cliquet, G. Plantier, T. Schultz, A. Fred, H. Gamboa, Eds. Angers, March 3-6, France 2014, pp. 165–170.
33. A. Schumann, K., Pancerz, “PhysarumSoft – a Software Tool for Programming Physarum Machines and Simulating Physarum Games,” in *Annals of Computer Science and Information Systems. Proceedings of the Federated Conference on Computer Science and Information Systems*, IEEE Xplore, 2015, pp. 615–626.
34. A. Schumann, L. Akimova, “Process Calculus and Illocutionary Logic for Analyzing the Behavior of Schistosomatidae (Trematoda: Digenea),” in *Computational Intelligence, Medicine and Biology - Selected Links.* K. Pancerz, E. Zaitseva, Eds. Studies in Computational Intelligence 600, Springer 2015.

35. A. Schumann, L. Akimova, “Simulating of Schistosomatidae (Trematoda: Digenea) Behaviour by Physarum Spatial Logic,” in *Annals of Computer Science and Information Systems*, vol.1. Proceedings of the Federated Conference on Computer Science and Information Systems. IEEE Xplore, 2013, pp. 225–2301.
36. A. Schumann, L. Akimova, “Syllogistic System for the Propagation of Parasites. The Case of Schistosomatidae (Trematoda: Digenea),” *Studies in Logic, Grammar and Rhetoric*, vol. 40, no. 1, pp. 303–319, 2015.
37. A. Schumann, Unconventional logic for massively parallel reasoning, in *Human System Interaction (HSI), The 6th International Conference on 6-8 June 2013*, IEEE Xplore, (2013), pp. 298–305, DOI 10.1109/HSI.2013.6577839
38. K. Pancerz, A. Schumann, “A Rough Set Version of the Go Game on Physarum Machines,” in: *9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, ACM, 2015, submitted for publication.
39. K. Pancerz, A. Schumann, “Principles of an Object-Oriented Programming Language for Physarum Polycephalum Computing,” in *Proceedings of the 10th International Conference on Digital Technologies (DT'2014)*, Zilina, July 9-11, Slovakia 2014, pp. 273–280.
40. K. Pancerz, A. Schumann, “Rough Set Description of Strategy Games on Physarum Machines,” in *Advances in Unconventional Computing*, A. Adamatzky, Ed., Springer 2016, submitted for publication.
41. K. Pancerz, A. Schumann, “Rough Set Models of Physarum Machines,” *International Journal of General Systems*, 27 Jan 2015, DOI: 10.1080/03081079.2014.997529.
42. K. Pancerz, A. Schumann, “Some Issues on an Object-Oriented Programming Language for Physarum Machines,” in *Applications of Computational Intelligence in Biomedical Technology*. R. Bris, J. Majernik, K. Pancerz, E. Zaitseva, Eds. Studies in Computational Intelligence 606, Springer 2016.