Petri Net Models of Simple Rule-Based Systems for Programming Physarum Machines

Extended Abstract

Andrew Schumann[1] and Krzysztof Pancerz[1,2]

[1] University of Information Technology and Management
Sucharskiego Str. 2, 35-225 Rzeszow, Poland
aschumann@wsiz.rzeszow.pl

[2] University of Management and Administration
Akademicka Str. 4, 22-400 Zamosc, Poland
kpancerz@wszia.edu.pl

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).

Key words: Physarum polycephalum, unconventional computing, Petri nets, rule-based systems

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 $\ldots$ 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 [10], [11], [13], 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 [16]). Our language is based on the prototype-based approach (cf. [6]). 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. 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

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	$false$

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. In these models, the places R_{di1}, R_{di2}, ..., R_{di3} correspond to propositions in

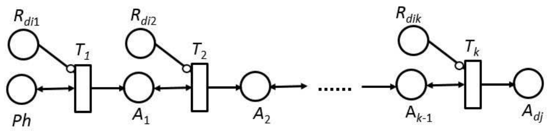

Fig. 1. A Petri net model of a rule of type 1: the repellent-based control approach

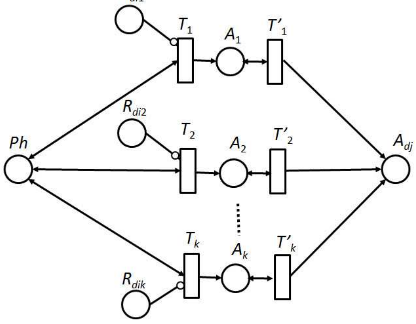

Fig. 2. A Petri net model of a rule of type 3: the repellent-based control approach

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. In

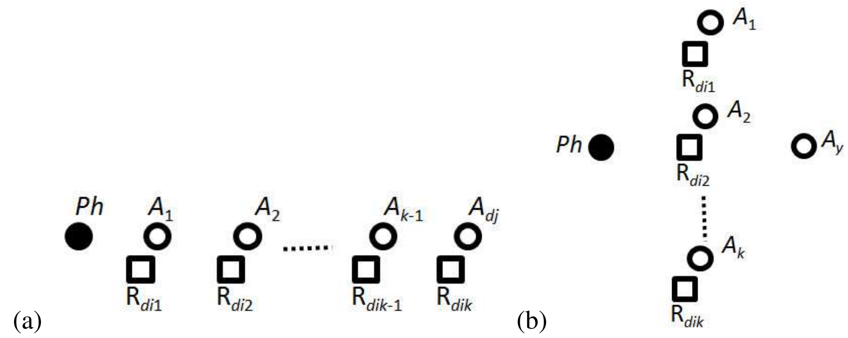

Fig. 3. The structure of the *Physarum* machine for: (a) a production rule of type 1, (b) a production rule of type 3

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.

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. 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). pp. 81–86. Atlanta, USA (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), 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), 1077–1082 (1987)
9. Nakagaki, T., Yamada, H., Toth, A.: Maze-solving by an amoeboid organism. Nature 407, 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). pp. 273–280. Zilina, Slovak Republic (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, pp. 185–199. Springer International Publishing, Switzerland (2016)
12. Petri, C.A.: Kommunikation mit automaten. Schriften des IIM nr. 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). pp. 389–397. Warsaw, Poland (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). pp. 180–183. Chemnitz, Germany (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), 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. pp. 619–626. Lodz, Poland (2015)
17. Schumann, A., Pancerz, K., Jones, J.: Towards logic circuits based on physarum polycephalum machines: The ladder diagram approach. In: Cliquet Jr., A., Plantier, G., Schultz, T., Fred, A., Gamboa, H. (eds.) Proceedings of the International Conference on Biomedical Electronics and Devices (BIODEVICES'2014). pp. 165–170. Angers, France (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, pp. 157–181. Wyd. Uniwersytetu Śląskiego, Katowice (2013)
19. Szpyrka, M., Szmuc, T.: D-nets - Petri net form of rule-based systems. Foundations of Computing and Decision Sciences 31(2), 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), 125–143 (2010)