Reformulation, Extension, and Application of the Formal Framework for P Systems (Extended Abstract)

Ren Tristan A. de la Cruz

June 22, 2020

Abstract

Membrane computing is a field of theoretical computer science that studies different models of computation known as *P systems*. The term '*P systems*' refers to a family of models of computation which are inspired by biological processes. P system models use abstractions of biological processes as computational operations. For example, different types of rules (operations) used by most P system variants are abstractions of processes like *chemical reaction* and *ion transport* that occur inside biological cells. Most P system variants use *object symbols* as the objects of computation. One can think of these object symbols as abstraction of physical *molecules* or *ions*. P systems store *multisets* of these object symbols inside regions enclosed by *membranes*. A P system has a collection of these membranes with multisets of objects symbols inside. The membranes can be 'connected' to each other to form a *membrane structure*.

Aside from studying P systems themselves, membrane computing also deals with applications of P systems. P systems have been used to solve 'hard' (NP-complete) problems. They have been used to model natural phenomena like biological oscillations, population dynamics, and sodium-potassium pump mechanism. Other P system applications include boolean circuit simulation, fault diagnosis models for electric locomotive, linguistic applications, etc.

There are tens, if not hundreds, of P system variants. Their syntax are well-defined but the semantics are often described in an informal manner. *Formal framework* is an attempt to formalize not only syntax but also the procedural semantics of a wide variety of P systems. There are currently three versions of the framework, one for P systems with static membrane structures, one for P systems with dynamic membrane structures, and another for static P systems with input-output. The formal frameworks can be used to analyze, compared, and extended P systems.

The *first formal framework (FF1)* [4] is an attempt to formally define procedural semantics for a large number of well-known variants of tissue P systems with static membrane structures. The *second formal framework (FF2)* [3] is a similar attempt for P systems with dynamic membrane structures. The *third formal framework (FF3)* [8] is an extension of FF1 with the purpose of expressing spiking neural P systems [5] in this extended framework.

The formal frameworks use and formalize notions/concepts common to most of their target P systems. The main notions available in all three versions of the framework are the following:

1. *configuration* of the system
2. *rules* of the system
3. *applicability* of a combination of rules
4. *allowed* combinations of rules
5. how to *apply* a combination of rules
6. *halting* condition for the system

In [7], both FF1 and FF2 were used to implement different features of P system variants. These features include *membrane thickness/polarity/labels*, *rule priorities*, and *membrane dissolution*. P system variants like *catalytic P systems*, *P systems with symport/antiport* [6], *P colonies*, and *probabilistic P systems* were also implemented as specific FF models. [7] shows that the formal framework can be used to model a wide variety of P systems with diverse features and semantics.

The FF can be used to compare different P system variants. By translating P system variants to their corresponding FF models, one can directly compare these FF models. The FF can be used a common language to analyze, compare and extended different P system variants.

This research proposal has three aspects: (1) the reformulation of the formal framework, (2) the extension of the formal framework, and (3) the application of the formal framework.

Reformulating the formal framework means changing the framework by changing the notions/concepts used or using different formalizations for these notions but not affecting the usefulness of the framework. Extending the framework means adding new notions and formalizations to extended the scope or usefulness of the framework. An extended framework can mean it can model more P system variants or that there are more notions in the framework that can provide more insights to the workings of existing 'supported' P system variants. Application of the framework means using the framework to analyze, compare, and/or extended existing P system variants.

The following are the specific objectives of this proposal:

1. (Reformulation) Combine FF2 and FF3 into a single formal framework. This involves the addition of the *Input* and *Output* functions from FF3 to FF2. It also involves the use of regular multiset languages for *permitting* or *forbidding* conditions of the interaction rule. The purpose of this objective is to have a single formal framework (FF) that can be used of static or dynamic P systems.

2. (Reformulation) Reformulate the interaction rule in FF (from objective 1) in a *bottom-up* manner instead of the *top-down* approach of the FF. The rule in the FF (or specifically FF2) contains 11 components because it is trying to be the most general and unrestricted version of a rule such that the rule types from the P system variants are simply restricted versions of the more general FF interaction rule. We call this approach of finding the most general and unrestricted form of the rule as *top-down*. A rule can instead be defined as a 'combination' of simpler 'elementary' rules. We start from the *bottom* with this 'elementary' rules and use them to define a general rule which is a combination of these 'elementary' rules.

3. (Application) Perform a comprehensive survey of the different P system variants and use the FF to create the equivalent FF models of the different P system variants.

4. (Extension) While doing the comprehensive survey of P system variants, if there are variants that are difficult or impossible to create an FF model for, formalize the features of these variants and use them to extended the FF. Examples for such P system variants with features that are not (directly) represented are available in [1] and [2].

5. (Application) Create a simulator for FF models. Combining this simulator with the FF models of the P systems from the survey (objective 4) will result in a fairly general simulator than can simulate a wide variety of P systems.

References

[1] Artiom Alhazov, Sergiu Ivanov, and Yurii Rogozhin. Polymorphic p systems. In Marian Gheorghe, Thomas Hinze, Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa, editors, *Membrane Computing*, pages 81–94, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[2] Fernando Arroyo, Angel Baranda, Juan Castellanos, and Gheorghe Paun. Membrane computing: The power of (rule) creation. *Journal of Universal Computer Science*, 8:369–381, 01 2002.

[3] Rudolf Freund, Ignacio Pérez-Hurtado, Agustín Riscos-Núñez, and Sergey Verlan. A formalization of membrane systems with dynamically evolving structures. *International Journal of Computer Mathematics*, 90(4):801–815, 2013.

[4] Rudolf Freund and Sergey Verlan. A formal framework for static (tissue) p systems. In George Eleftherakis, Petros Kefalas, Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa, editors, *Membrane Computing*, pages 271–284, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[5] Mihai Ionescu, Gheorghe Păun, and Takashi Yokomori. Spiking neural p systems. *Fundam. Inf.*, 71(2,3):279–308, February 2006.

[6] Andrei Pâun and Gheorghe Pâun. The power of communication: P systems with symport/antiport. *New Generation Computing*, 20(3):295–305, Sep 2002.

[7] Sergey Verlan. Using the formal framework for p systems. In Artiom Alhazov, Svetlana Cojocaru, Marian Gheorghe, Yurii Rogozhin, Grzegorz Rozenberg, and Arto Salomaa, editors, *Membrane Computing*, pages 56–79, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[8] Sergey Verlan, Rudolf Freund, Artiom Alhazov, and Linqiang Pan. A formal framework for spiking neural p systems. In Gheorghe Păun, editor, *Proceedings of the 20th International Conference on Membrane Computing*. Bibliostar, 2019.