A Note on the Probabilistic Evolution for P Systems

Sergey Verlan

Laboratoire d'Algorithmique, Complexité et Logique,
Université Paris Est – Créteil Val de Marne,
61, av. gén. de Gaulle, 94010, Créteil, France
verlan@univ-paris12.fr

Summary. In this note we propose a method that permits to describe in a uniform manner variants of probabilistic/stochastic P systems. We give examples of such a description for existing models of P systems using probabilities.

1 Introduction

The idea of enriching P systems with probabilities and using a probabilistic or stochastic evolution appears very early in the development of the area [7, 6]. Such kind of models (we shall call them probabilistic P systems) were shown to be very useful for simulations of biological phenomena, we cite here only [3, 1, 10]. While some of these models are using the Gillespie stochastic simulation algorithm (SSA) [4, 5] for the evolution step, the others are introducing different approximations of it or choose a completely different strategy. The definitions used to define the models often use specific notions and terminology, so their comparison and understanding quickly becomes a difficult task.

In this note we propose a systematical approach to the definition of such systems based on the association of a probability to a group of rules, which is a natural generalization of a probability for a single rule. The used method permits to establish a framework that can be used to compare existing definitions and gives a possibility to extend them. As an example of the application of the method we translate the definition of the evolution step of two variants of probabilistic P systems into our framework and we show the equivalent strategies of computation of individual rule probabilities leading to a corresponding group probability.

2 Preliminaries

We do not present here standard definitions. We refer to [12] for all details. We will denote by $|M|$ the cardinality of a set M or the size of a multiset M. By $|M|_x$ we will denote the number of elements x in the multiset M.

We also assume that the reader is familiar with standard notions of P systems, which can be consulted in the books [8] and [9] or at the web page [11]. We shall only focus on the semantics of the evolution step. We will follow the approach given in [2], however we will not enter into deep details concerning the notation and the definition of derivation modes given there. Consider a P system Π of any type evolving in any derivation mode. The key point of the semantics of P systems is that according to the type of the system and the derivation mode δ for any configuration of the system C a set of multisets (over $\mathcal{R}$) of applicable rules, denoted by $Appl(\Pi, C, \delta)$, is computed. After that, one of the elements from this set is chosen, non-deterministically, for the further evolution of the system.

We remark that from the point of view of the computer simulation of P systems the non-deterministic choice can be considered equivalent to a probabilistic choice where each multiset of rules has an equal probability to be chosen. Permitting these multisets to have a *different probability* is the main idea of this paper. More precisely, for each multiset of rules $R \in Appl(\Pi, C, \delta)$ we compute the probability $p(R, C)$ based on the propensity function $f : \mathcal{R}^\circ \times (\mathbb{N} \times O^\circ)^* \to \mathbb{R}$ that associates a real value for a multiset of rules with respect to a configuration. Hence the value $f(R, C)$ depends not only on the multiset of rules R, but also on the configuration C.

The probability to choose a multiset $R \in Appl(\Pi, C, \delta)$ is defined as the normalization of corresponding probabilities:

$$p(R, C) = \frac{f(R, C)}{\sum_{R' \in Appl(\Pi, C, \delta)} f(R', C)} \tag{1}$$

3 Discussion

In the previous section we didn't discuss the propensity function f, which is the main ingredient of the model. Below we will give examples of simple propensity functions each leading to different execution strategies.

Constant strategy: each rule r from $\mathcal{R}$ has a constant contribution to f and equal to c_r:

$$f(R, C) = \prod_{r \in R} c_r \tag{2}$$

Multiplicity-dependent strategy: each rule r from $\mathcal{R}$ has a contribution to f proportional to the number of times this rule can be applied and to a stochastic constant c_r that only depends on r:

$$N_r(C) = \min_{x \in lhs(r)} \left[\frac{|C|_x}{|lhs(r)|_x} \right] \tag{3}$$

$$f(R, C) = \prod_{r \in R} c_r N_r(C) \tag{4}$$

Concentration-dependent strategy: each rule r from $\mathcal{R}$ has a contribution to f proportional to $h_r(C)$, the number of distinct combinations of objects from C that activate r, and to a stochastic constant c_r that only depends on r (below we denote by $\binom{a}{b}$ the binomial function):

$$h_r(C) = \prod_{x \in lhs(r)} \binom{|C|_x}{|lhs(r)|_x} \tag{5}$$

$$h_R(C) = \prod_{r \in R} c_r h_r(C) \tag{6}$$

$$f(R, C) = h_R(C) \tag{7}$$

Gillespie strategy: each rule r from $\mathcal{R}$ has a contribution to f that depends on the order in which it was chosen and it is equal to $c_r \cdot h_r(C')$, where C' is the configuration obtained by applying all rules that were chosen before r.

We remark that the concentration-dependent strategy is not equal to Gillespie strategy. More precisely, in a Gillespie run the probability to choose a new rule depends on the objects consumed and produced by previously chosen rules. We can consider a Gillespie run as a sequence of sequential (single-rule) applications using concentration-dependent strategy.

We also remark that the Gillespie algorithm uses the notion of time that we do not consider in this paper. However, the definitions can be easily adapted for to handle this case.

4 Examples

4.1 Dynamical Probabilistic P Systems

Dynamical probabilistic P (DPP) systems were introduced in [10]. We present below the definition of the evolution step. For the sake of the simplicity we will consider only one compartment, however the discussion below can be easily generalized to several compartments.

Let C be the current configuration and $\mathcal{R}$ be the set of all rules. Then the system evolves from C to C' as follows.

1. For each rule $r \in \mathcal{R}$ the propensity of $a_r(C) = c_r * h_r(C)$ (h_r being defined as in Equation (5)) is computed.

2. The propensities are normalized giving a probability for a rule r to be chosen: $p_r(C) = \frac{a_r(c)}{\sum_{r' \in \mathcal{R}} a_{r'}(C)}$.
3. The rules to be applied are chosen according to their probabilities. If a non-applicable rule is chosen, the choice is repeated.
4. The process stops when a maximal (parallel) multiset of rules R is obtained.
5. The multiset of rules obtained at the previous step is applied and yields a new configuration C'.

It can be easily seen that since the probabilities to apply a rule (p_r) are computed only at the beginning of each step, then the maximal multiset of rules R is composed from independent rules (the order in which the rules were chosen has no influence). Hence the probability to choose a multiset of rules R is equal to the product of the probabilities of each rule: $p_R(C) = \prod_{r \in R} p_r$. Now if we normalize this value with respect to all possible maximally parallel multisets of rules we obtain:

$$\frac{\prod_{r \in R} p_r(C)}{\sum_{R' \in Appl(\Pi, C, max)} \prod_{z \in R'} p_z(C)} = \frac{\prod_{r \in R} \frac{a_r(C)}{\sum_{r' \in \mathcal{R}} a_{r'}(C)}}{\sum_{R' \in Appl(\Pi, C, max)} \prod_{z \in R'} \frac{p_z(C)}{\sum_{r' \in \mathcal{R}} a_{r'}(C)}} = \frac{\prod_{r \in R} a_r(C)}{\sum_{R' \in Appl(\Pi, C, max)} \prod_{z \in R'} a_z(C)} \tag{8}$$

Since for the concentration-dependent strategy we have $f(R, C) = \prod_{r \in R} a_r(C)$, it follows that (8) equals to (1). Hence we just showed that DPP systems can be translated to probabilistic P systems with a concentration-dependent strategy.

4.2 Probabilistic Functional Extended P Systems

Probabilistic functional extended P (PFEP) systems where introduced in [1] as a part of a framework used to model eco-systems. In order to simplify the presentation we consider a flattening of the structure of the P system, hence using only multiset rewriting rules. We also consider that the rules having the same left-hand side form a partition of the set of rules $\mathcal{R}$ into n subsets $\mathcal{R} = \mathcal{R}_1 \dots \mathcal{R}_n$, where $r_1, r_2 \in \mathcal{R}_i \Rightarrow lhs(r_1) = lhs(r_2)$, $1 \leq i \leq n$.

The evolution of a PFEP system is done as follows:

1. A maximally parallel multiset of rules R is chosen.
2. R is partitioned into submultisets based on the left-hand side of rules: $R_i = \{r \in R \mid r \in \mathcal{R}_i\}$.
3. For each non-empty partition R_i, $|R_i|$ rules from $\mathcal{R}_i$ are chosen according to the given probability $f_r(a)$, where $r \in R_i$ and a is a moment of time.
4. The resulting multiset of rules is applied yielding a new configuration.

From the description of the strategy it is clear that it corresponds to the multiplicity-dependent strategy for a maximally parallel derivation mode (and where the constant c_r is replaced by $f_r(a)$).

5 Final Remarks

In this note we presented a new method to describe P systems working with probabilities. The main aim of this method is to provide a common framework permitting to describe variants of probabilistic P systems. Such a framework could be useful for the comparison of different variants and for the extension of existing ones.

The used method is based on the assignment of a probability to a multiset of applicable rules (according to some derivation mode). The evolution step then chooses a multiset of rules according to its probability and then applies it. We were particularly interested by the cases when the probability of a multiset of rules R can be represented as a product of individual probabilities of rules $r \in R$. We gave example of three strategies for the computation of the individual probabilities of a rule. The first strategy supposes that the rule probability is constant, the second strategy supposes that the rule probability is proportional to the number of its applications, while the third strategy corresponds to the mass action law. We showed that the DPP systems from [10] are using the third strategy, while the PFEP systems from [1] the second. An interesting direction for the further research is to consider the above strategies in combination with different derivation modes.

We remark that the obtained strategies are not equivalent to the Gillespie stochastic simulation algorithm (SSA), except if the sequential derivation mode is used, because they do not take into account the intermediate modifications of the configuration. In some sense they correspond to the tau-leaping method, which is an approximation of the Gillespie SSA. An interesting topic for a further research could be the expression of different Gillespie-based strategies in the proposed framework. This can give rise to new variants of P systems suitable for stochastic simulations.

References

1. M. Cardona, M. A. Colomer, A. Margalida, A. Palau, I. Pérez-Hurtado, M. J. Pérez-Jiménez, D. Sanuy, A computational modeling for real ecosystems based on P systems. *Natural Computing*, **10**(1): 39–53, 2011.
2. R. Freund, S. Verlan: A formal framework for static (tissue) P systems. In *Proc. of WMC 2008* (G. Eleftherakis et al., eds.), Thessaloniki, Greece, Springer, 2007, LNCS 4860, 271–284.
3. M. Gheorghe, V. Manca, F. J. Romero-Campero, Deterministic and stochastic P systems for modelling cellular processes. *Natural Computing* **9**(2): 457–473, 2010.
4. D.T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comput. Phys. 22 (4)(1976) 403-434.
5. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.
6. M. Madhu, Probabilistic rewriting P systems. *International Journal of Foundations of Computer Science*, **14**(1): 157–166, 2003.

7. A. Obtulowicz, Gh. Păun. (In search of) Probabilistic P systems, *BioSystems*, **70**(2): 107–121, 2003.
8. Gh. Păun, *Membrane Computing. An Introduction*. Springer–Verlag, 2002.
9. G. Păun, G. Rozenberg, and A. Salomaa. *The Oxford Handbook Of Membrane Computing*. Oxford University Press, 2009.
10. D. Pescini, D. Besozzi, G. Mauri, and C. Zandron. Dynamical probabilistic P systems. *International Journal of Foundations of Computer Science*, **17**:183–204, 2006.
11. *The Membrane Computing Web Page*: `http://ppage.psystems.eu`
12. G. Rozenberg, A. Salomaa, eds., *Handbook of Formal Languages*. Springer–Verlag, Berlin, 1997.