Proceedings of the International Workshop on Intelligent Information Systems
IIS2011, September 13-14, 2011, Chisinau, Republic of Moldova

Formal Strategies in P Systems

Sergiu Ivanov

Abstract

The process of design and study of P system often consists in creating a basic model to which a series of further modifications is applied. These modifications are often generic and do not depend on the details of a certain instance of a P system. The modifications can therefore be abstracted as separate mathematical objects. This paper focuses on such objects which describe modifications to P systems.

Keywords: P systems, strategies, modularity, composition.

1 Introduction

Membrane computing is a theoretical framework of parallel distributed multiset processing. It has been introduced by Gheorghe Păun in 1998, and has been an active research area.; see [1] for the comprehensive bibliography and [2],[3] for a systematic survey. Membrane systems are also called P systems.

During the development of P systems some common modifications are often applied. Such modifications do not usually depend too much on the structure of the P system and can thus be generalised. This paper focuses on formalising such modifications in the form of P system strategies.

2 Definitions

A *transitional* membrane system is defined by a tuple

$$\begin{array}{lcl} \Pi & = & (O, \mu, w_1, w_2, \ldots, w_m, R_1, R_2, \ldots, R_m, i_0), \text{ where} \\ O & & \text{is a finite set of objects,} \\ \mu & & \text{is a hierarchical structure of } m \text{ membranes, bijectively} \\ & & \text{labelled with } 1, \ldots, m, \\ w_i & & \text{is the initial multiset in region } i, 1 \le i \le m, \\ R_i & & \text{is the set of rules of region } i, 1 \le i \le m, \\ i_0 & & \text{is the output region.} \end{array}$$

The rules have the form $u \to v$, where $u \in O^+$, $v \in (O \times Tar)^*$. The target indications from $Tar = \{here, out\} \cup \{in_j \mid 1 \le j \le m\}$, where j are the labels of the corresponding inner membranes. The target *here* is typically omitted. In case of non-cooperative rules, $u \in O$.

3 Strategies

We will call a *set transformation of* A, $A \subseteq U$, any function $T : A \cup \{\emptyset\} \to 2^U$, where U is the universal set of entities. A *multiset transformation of* A is any function $M : A \times \mathbb{N} \to U^*$.

We define an application of the set transformation T to a set $B \subseteq A$ in the following way:

$$T(B) = \begin{cases} \bigcup_{b \in B} T(b), & \text{for } B \neq \emptyset \\ T(\emptyset), & \text{for } B = \emptyset \end{cases} \tag{1}$$

An application of a set transformation T to the set B may be perceived as iterating over the elements $b \in B$ and substituting b for $T(b)$.

The application of M to a multiset $w \in A^*$ is defined in an analogous way, using the multiset union operation:

$$M(w) = \begin{cases} \biguplus_{b \in supp(w)} M(b, |w|_b), & \text{for } w \neq \lambda \\ M(\lambda, 0), & \text{for } w = \lambda \end{cases} \tag{2}$$

where $supp(w)$ is the set of entities contained in w.

A *non-conflicting set transformation* of A is such a transformation of A that

$$T(a) \cap A \subseteq \{a\} \tag{3}$$

A non-conflicting multiset transformation is defined in an analogous way, without taking multiplicities into consideration.

We define a *parametrised set transformation* T^P of a set A to be any function $T : (A \cup \{\emptyset\}) \times P \to 2^U$, where P is a set of valid parameter values. A *parametrised multiset transformation* M^P is any function $M : (A \cup \{\emptyset\}) \times \mathbb{N} \times P \to U^*$.

A *strategy* for a transitional P system Π is defined by the following tuple:

$$\begin{aligned} S_\Pi &= (T_O, M_{w_1}, M_{w_2}, \dots, M_{w_m}, T_{R_1}, T_{R_2}, \dots, T_{R_m}),\ \text{where} \\ T_A &\ \text{are set transformations of } A, A \in \{O\} \cup \{R_i \mid 1 \leq i \leq m\}, \\ M_{w_i} &\ \text{are non-conflicting multiset transformations of } O \\ &\ \text{associated with } w_i, 1 \leq i \leq m. \end{aligned} \tag{4}$$

A *parametrised strategy* S_Π^P is a strategy in which all transformations are parametrised over P.

The application of a strategy S_Π to a transitional P system Π is defined as the application of all the transformations in S_Π to the corresponding components of Π.

Example 1 (Increase the non-determinism)**.** *We would like to write a strategy which would add "waiting" rules of the form $a \to a$ to a transitional P system. The task is not straightforward, because the set transformations of the sets of rules of Π do not take the alphabet of the system as a parameter. The solution is as follows:*

$$\begin{gathered} Nondet = (T_O, M_{w_1}, M_{w_2}, \dots, M_{w_m}, T_{R_1}, T_{R_2}, \dots, T_{R_m}),\ \textit{where} \\ T_{R_i}(u \to v) = \{u \to v\} \cup \{a \to a \mid a \in uv\},\ T_{R_i}(\emptyset) = \emptyset, \\ T_O(a) = a,\ T_O(\emptyset) = \emptyset,\ M_{w_i}(a, n) = a^n,\ M_{w_i}(\lambda, 0) = \lambda, \\ 1 \leq i \leq m \end{gathered} \tag{5}$$

Note that no waiting rules are added for symbols which are not employed in any rule.

4 Conclusion

A strategy can be perceived as a way of modifying an aspect of the functionality of a P system. Strategies offer the advantage of extensibility because they allow building the necessary behaviour out of specific elements. Strategies are based on set and multiset transformations which are simple functions; this makes the concept of a strategy quite flexible and formalised.

Parametrised strategies, on the other hand, push the flexibility further because they may act differently depending on their parameter. This may either influence the general effect of the strategy on a given P system or may make it focus on a certain component of the system and fine-tune the model in order to achieve the necessary functionality.

References

[1] P systems webpage. `http://ppage.psystems.eu/`

[2] Gh. Păun: *Membrane Computing. An Introduction*, Springer, 2002.

[3] Gh. Păun, G. Rozenberg, A. Salomaa, Eds.: *Handbook of Membrane Computing*. Oxford University Press, 2010.

[4] G. Rozenberg, A. Salomaa, Eds.: *Handbook of Formal Languages*, vol. 1-3, Springer, 1997.

Sergiu Ivanov[1,2]

[1]Institute of Mathematics and Computer Science
Email: sivanov@math.md

[2]Technical University of Moldova