Power and Size of Generalized Communicating P Systems with Minimal Interaction Rules ⋆

Erzsébet Csuhaj-Varjú, Sergey Verlan

[1] Computer and Automation Research Institute
Hungarian Academy of Sciences
Kende u. 13-17, 1111 Budapest, Hungary, and
Eötvös Loránd University, Faculty of Informatics
Department of Algorithms and Their Applications
Pázmány Péter sétány 1/c, 1117 Budapest, Hungary
csuhaj@sztaki.hu

[2] Laboratoire d'Algorithmique, Complexité et Logique
Département Informatique, Université Paris Est
61, av. Général de Gaulle, 94010 Créteil, France
verlan@univ-paris12.fr

Summary. In this paper, we present results on the power and the size of generalized communicating P systems in the case of eight restricted variants of communication rules. These constructs are purely communicating tissue-like membrane systems with communication rules which allow the movement of only pairs of objects. We show that seven of these restricted variants are computationally complete, even with limited size, while systems belonging to the remaining one variant are able to compute only finite singletons of non-negative integers. The obtained results complete the investigations of the computational power of generalized communicating P systems.

1 Introduction

The theory of P systems provides several examples for computational models with large computational power and at the same time with simple architecture and small size complexity.

One of the main research directions in P systems theory is the study of the computational power of purely communicating membrane systems. Adequate examples of these constructs are the symport/antiport P systems [1]. Motivated by the problem how to define a common generalization of various purely communicating models in the framework of P systems, the concept of a generalized communicating P system was introduced in [3].

⋆ The research of the first author was supported in part by the Hungarian Scientific Research Fund "OTKA", Grant No. K75952. The second author acknowledges the Science and Technology Center in Ukraine, project 4032.

A generalized communicating P system, or a GCPS for short, corresponds to a graph where each node, called a cell, contains a multiset of objects which - by communication - may move between the cells. The communication rules are rather restricted, any rule identifies four cells, two input cells and two output cells, such that a pair of objects from the two input cells move synchronously to the two output cells. The form of a communication rule is $(a,i)(b,j) \to (a,k)(b,l)$ where a and b are objects and i,j,k,l are numbers that identify the input and the output cells. Such a rule means that an object a from cell i and an object b from cell j move synchronously to cell k and cell m, respectively. It can easily be seen that these very simple communication rules can also be interpreted as interaction rules. Although a GCPS realizes a graph structure, the cells are defined implicitly, since the system is given as a set of communication rules over an alphabet.

Depending on the relation of i,j,k,l, nine restricted variants of communication rules (modulo symmetry) can be distinguished. (For example, $i \neq j \neq k \neq l$ is one of these restrictions, called a parallel-shift rule.) When the GCPS has only one type of these restricted rules, we speak of generalized communicating P systems with minimal interaction, a GCPSMI for short.

In this article, we consider generalized communicating P systems which use only one type of the above interaction operations. In [3] it was shown that any register machine can be simulated by a GCPS having 19 cells and using only parallel-shift rules. Continuing the examination of the power of GCPSMIs, we study the remaining eight restricted variants of communication rules. We prove that in most of the cases (7 of 8) computational completeness is obtained, i.e., the corresponding GMPCSs are able to determine any recursively enumerable set of non-negative integers; the only exception determines only finite singletons of natural numbers. The constructions in the proofs also demonstrate that this large expressive power can be obtained by P systems with relatively small numbers of cells and simple graph architectures.

2 Definitions

In this section we recall some basic notions and notations commonly used in membrane computing and some basic concepts of formal language theory that we need throughout the paper.

We consider register machines with two types of instructions: $(p, A+, q, s)$ and $(p, A-, q, s)$, where p, q, s are states and A is a register. Register machines generate NRE.

Next we present the basic definitions concerning generalized communicating P system; for further details and motivations of these constructs, see [3].

Definition 1. *A* generalized communicating P system *of degree n (a* GCPS, *for short) is a construct: $\Pi = (O, E, w_1, \dots, w_n, R, i_o)$, where:*

1. O is a finite alphabet,

2. *$E \subseteq O$;*
3. *$w_i \in O^*$, for all $1 \leq i \leq n$, the multiset of objects* initially associated to cell *i;*
4. *R is a finite set of* interaction rules *of the form $(a,i)(b,j) \to (a,k)(b,l)$, where $a, b \in O$, $0 \leq i,j,k,l \leq n$, and if $i = 0$ and $j = 0$, then $\{a,b\} \cap (O \setminus E) \neq \emptyset$; i.e. $a \notin E$ or $b \notin E$*;
5. *$i_o \in \{1, \ldots, n\}$ is the* output cell.

The system consists of n cells, numbered from 1 to n, that contain multisets of objects over O; initially cell i contains the multiset w_i. There is also a special cell distinguished, numbered by 0, called the *environment.* The environment contains symbols of $E \subseteq O$ in an *infinite number of copies.*

The cells interact with each other by means of the rules in R of form $r = (a,i)(b,j) \to (a,k)(b,l)$, with $a, b \in O$ and $0 \leq i,j,k,l \leq n$. Such an interaction rule may be applied if there is an object a in cell i and an object b in cell j. As the result of the application of r, the object a moves from cell i to cell k and b moves from cell j to cell l. If two objects from the environment are moved to some other cell or cells, then at least one of them must not appear in the environment in an infinite number of copies.

Notice that the structure of a GCPS corresponds neither to a tree as in cell-like P systems nor to a graph as in tissue P systems, though some models of cell-like P systems and tissue P systems can be seen as special variants of GCPSs.

In general, for a given GCPS, every rule is defined over a block of cells which allows certain objects to pass from the input cells to the output cells; altogether these rules define a network of communicating cells.

Let $\Pi = (O, E, w_1, \ldots, w_n, R, i_o)$, $n \geq 1$, be a GCPS. A *configuration* of Π is a tuple $(z_0, z_1, \ldots, z_n)$ with $z_0 \in (O \setminus E)^*$ and $z_i \in O^*$, for all $1 \leq i \leq n$; z_0 is the multiset of objects possibly present in the environment in a finite number of copies, whereas, for all $1 \leq i \leq n$, z_i is the multiset of objects present inside cell i. The *initial configuration of Π* is the tuple $(\lambda, w_1, \ldots, w_n)$. Then, given a configuration of Π, a new configuration can be produced by applying the rules in a non-deterministic maximally parallel way: all the rules that can be applied to the objects currently present inside the cells and the environment must be applied in parallel at the same time; the only restriction is that an occurrence of an object has to be used by at most one rule. One such application of the rules represents a *transition step (in Π).* A *computation in Π* is any sequence of transition steps in Π which starts from its initial configuration. A *successful computation in Π* is any computation which produces a configuration where no more rules can be applied to the objects left inside the cells and in the environment. The result of a successful computation is the non-negative integer corresponding to the size of the multiset of objects inside the output cell i_o in the final configuration. The set of non-negative integers computed in this way by GCPS Π is denoted by $N(\Pi)$.

We may impose several restrictions on the interaction rules, some of these restrictions directly correspond to antiport or symport rules of size 2.

Below we define all possible restrictions (modulo symmetry): let O be an alphabet and consider an interaction rule $(a,i)(b,j) \to (a,k)(b,l)$ with $a,b \in O$, $i,j,k,l \geq 0$. Then we distinguish the following cases:

1. $i = j = k \neq l$: the *conditional-uniport-out rule* sends b to membrane l provided that a and b are in membrane i.
2. $i = k = l \neq j$: the *conditional-uniport-in rule* brings b to membrane i provided that a is in that membrane.
3. $i = j \neq k = l$: the *symport2 rule* corresponds to the minimal symport rule, i.e., a and b move together from membrane i to k.
4. $i = l \neq j = k$: the *antiport1 rule* corresponds to the minimal antiport rule, i.e., a and b are exchanged in membranes i and k.
5. $i = k \neq j \neq l$: the *presence-move rule* moves the symbol b from membrane j to l, provided that there is a symbol a in membrane i.
6. $i = j \neq k \neq l$: the *split rule* sends a and b from membrane i to membranes k and l, respectively.
7. $k = l \neq i \neq j$: the *joining rule* brings a and b together to membrane i.
8. $i = l \neq j \neq k$ or $i \neq j = k \neq l$: the *chain rule* moves a from membrane i to membrane k while b is moved from membrane j to membrane i, i.e., where a previously has been.
9. $i \neq j \neq k \neq l$: the *parallel-shift rule* moves a and b in independent membranes.

A generalized communicating P system may have rules of several types as defined above. Moreover, we may allow uniport rules (i.e., rules of the form $(a,i) \to (a,k)$ specifying that, whenever an object a is present in cell i, this may be moved to cell k). In this case, GCPS with symport2 and uniport rules or with antiport1 and uniport rules become tissue P systems with minimal symport or minimal symport and antiport, respectively.

When only one type of rules is considered, we call the corresponding GCPS a *minimal interaction P system*, or a GCPSMI for short. We denote by $NOtP_k(x)$ the set of numbers generated by a minimal interaction P system of degree k having rules of type x, $x \in \{uout, uin, sym2, anti1, presence, split, join, chain, shift\}$.

3 Power and Size of Minimal Interaction P Systems

Minimal interaction P systems with any types of rules defined above, except antiport1, are computationally complete devices, i.e. they are able to compute any recursively enumerable set of non-negative integers. Moreover, the systems which are computational complete are able to reach this computational power with a number of cells limited by a small constant. In the case of split rules, 9 cells suffice.

Theorem 1. *$NOtP_9(split) = NRE$.*

Systems having symport 2 rules cannot generate NRE, however they can accept any recursively enumerable set of numbers. The proof of the result is based on a simulation of a split rule by symport2 rules. Then the result follows from Theorem 1.

Theorem 2. *For any* $S \in NRE$ *there is* $\Pi \in NOtP_8(sym2)$ *that accepts* S.

Theorem 3. $NOtP_*(anti1) \subset NFIN$.

The proof follows from the fact that the number of symbols in a cell cannot be changed by using only antiport1 rules. Hence, only finite singletons of non-negative integers can be generated.

The first equality below is proved by a direct simulation of the register machine, and the second one is based on a simulation of a join rule by by a sequence of chain rules.

Theorem 4. $NOtP_7(join) = NOtP_*(chain) = NRE$.

Theorem 5. $NOtP_{30}(uin) = NRE$.

This statement is proved by simulating the increment and the decrement instructions of a register machine. Instead of direct simulations of the instructions, we define sets of conditional-uniport-in rules, so-called (primitive) blocks, as it was done in [3] and [2], and then we show how a set of rules simulating an increment instruction or a decrement instruction can be composed from these blocks. For this purpose, we use three blocks: the so-called uniport block, the basic block or main block, and the zero block.

The uniport block corresponds to an uniport rule. The basic or main block is a variant of a minimal interaction rule that permits to move synchronously symbols a from cell i to cell k and b from cell j to cell l. If b is not present, then an infinite loop occurs. The zero block is a variant of a minimal interaction rule that moves symbol a from cell i to k providing that there are no symbols b in cell j. If there are symbols b in cell j then the computation enters into an infinite loop.

Next two equalities are proved in a similar way.

Theorem 6. $NOtP_{30}(uout) = NOtP_{36}(presence) = NRE$.

References

1. A. Păun, Gh. Păun: The power of communication: P systems with symport/antiport. *New Generation Computing* 20 (2002), 295–305.
2. S. Verlan, F. Bernardini, M. Gheorghe, M. Margenstern: On communication in tissue P systems: conditional uniport. In *Membrane Computing, 7th International Workshop, WMC 2006, Leiden, The Netherlands, Revised, Selected, and Invited Papers*. LNCS 4361, Springer, 521-535.
3. S. Verlan, F. Bernardini, M. Gheorghe, M. Margenstern: Generalized communicating P systems. *Theoretical Computer Science* 404 (1-2) (2008), 170–184.