Environments for generalized communicating P systems

Ákos Balaskó

Institute for Computer Science and Control, HAS
Budapest, Hungary
balasko.akos@mta.sztaki.hu

Erzsébet Csuhaj-Varjú

Eötvös Loránd University, Budapest, Hungary
csuhaj@inf.elte.hu

Generalized Communicating P Systems

Models inspired by the problem how to define a **common generalization of various purely communicating models** in the framework of **P systems**.

[Verlan, Bernardini, Gheorghe, Margenstern, 2008]

Generalized Communicating P Systems (GCPS)

$$\Pi = (O, E, w_1, \ldots, w_n, R, h)$$

1. O is a finite alphabet, called the *set of objects* of Π;
2. $E \subseteq O$; called the *the set of environmental objects* of Π;
3. $w_i \in O^*$, for all $1 \leq i \leq n$, are strings which represent the multiset of objects *initially associated to cell* i;
4. R is a finite set of *interaction rules* of the form $(a, i)(b, j) \rightarrow (a, k)(b, l)$, 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., $a \notin E$ and/or $b \notin E$;
5. $h \in \{1, \ldots, n\}$, $1 \leq h \leq n$, is the *output cell*.

[Verlan, Bernardini, Gheorghe, Margenstern, 2008]

Interaction rules

$$(a, i)(b, j) \rightarrow (a, k)(b, l),$$

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.

Configuration, transition, computation

$$\Pi = (O, E, w_1, \dots, w_n, R, h)$$

Configuration:

$$(z_0, z_1, \dots, z_n)$$

with $z_0 \in (O \setminus E)^*$ and $z_i \in O^*$, for all $1 \leq i \leq n$;

Transition:

$$u = (z_0, z_1, \dots, z_n) \Longrightarrow u' = (z'_0, z'_1, \dots, z'_n)$$

A **transition sequence** is said to be a **successful generation** by Π if it starts with its initial configuration and ends with one of its **halting configurations**.

We say that Π **generates a non-negative integer** n if there is a successful generation by Π with n being **the size of the multiset of objects present inside the output cell in the halting configuration**.

Restricted variants of interaction rules

1. $i = j = k \neq l$: the **conditional-uniport-out rule** sends b to cell l provided that a and b are in cell i.

2. $i = k = l \neq j$: the **conditional-uniport-in rule** brings b to cell i provided that a is in that cell.

3. $i = j,\ k = l,\ i \neq k$: the **symport2 rule** corresponds to the minimal symport rule, i.e., a and b move together from cell i to k.

4. $i = l,\ j = k,\ i \neq j$: the **antiport1 rule** corresponds to the minimal antiport rule, i.e., a and b are exchanged in cells i and k.

5. $i = k$ and $i \neq j$, $i \neq l$, $j \neq l$: the **presence-move rule** moves the object b from cell j to l, provided that there is an object a in cell i and i, j, l are pairwise different cells.

Restricted variants of interaction rules

1. $i = j$, $i \neq k$, $i \neq l$, $k \neq l$: the **split rule** sends a and b from cell i to cells k and l, respectively.

2. $k = l$, $i \neq j$, $k \neq i$, $k \neq j$: the **join rule** brings a and b together to cell i.

3. $i = l$, $i \neq j$, $i \neq k$ and $j \neq k$: the **chain rule** moves a from cell i to cell k while b is moved from cell j to cell i, i.e., to the cell where a was previously.

4. i, j, k, l are pairwise different numbers: the **parallel-shift rule** moves a and b from two different cells to another two to different cells.

Computational power

Generalized communicating P systems are **computationally complete computing devices**, even with a **small number of cells**, or even with **only one symbol**.

[Csuhaj-Varjú, Verlan, 2011]

[Csuhaj-Varjú, Vaszil, Verlan, 2010]

[Krishna, Gheorghe,Dragomir, 2013]

Related areas, topics

- Petri nets,
- Workflows,
- Multi-agent systems with a dynamic environment

Role of the Environment

- **Supports** or **restricts** the functioning of the system

- **Environment + GCPS** = dynamical system

Open problem: classification of the environments

Generalized Communicating P Systems with Dynamic Environment (deGCPS)

Definition 1. *A generalized communicating P system with dynamic environment (a deGCPS, for short) of degree n, where $n \geq 1$, is an $(n+5)$-tuple*

$$\Pi = (O, M, w_0, w_1, \ldots, w_n, R, h)$$

where

1. *O is a finite alphabet, called the set of objects of Π;*
2. *$M = (O, P)$ is a multiset rewriting scheme;*
3. *w_0 is a finite multiset representing the initial environment and $w_i \in O^*$, for every i, $1 \leq i \leq n$, is a finite multiset of objects initially associated to cell i;*
4. *R is a finite set of interaction rules of the form $(a, i)(b, j) \rightarrow (a, k)(b, l)$, where $a, b \in O$, $0 \leq i, j, k, l \leq n$;*
5. *$h \in \{1, \ldots, n\}$ is the* output cell.

The $n+3$-tuple $(O, w_1, \ldots, w_n, R, h)$ is said to be the core (core GCPS) of Π.

[Balaskó, Csuhaj-Varjú, Vaszil, 2015]

Configuration, Computation, Result

$$\Pi = (O, M, w_0, w_1, \ldots, w_n, R, h)$$

$$(z_0, z_1, \ldots, z_n)$$

A transition in Π consists of two steps:

first, the current multiset of the environment is changed by applying the rules of M in maximally parallel, and then rules of R are applied in a maximally parallel manner.

A successful generation in Π is a sequence of transitions starting from the initial configuration and ending in a final configuration, i.e., in a configuration of the form $c_f = (u_0, u_1, \ldots, u_n)$, where no rule of R can be applied to c_f and no rule of P can be applied to u_0.

The result of a successful generation in a deGCPS Π is the number of objects present in the output cell, cell h.

Computational power

$$NOdetP_4(split) = NRE.$$

$$NOdetP_4(join) = NRE.$$

$$NOdetP_5(parallel\_shift) = NRE.$$

[Balaskó, Csuhaj-Varjú, Vaszil, 2015]

Proof technique

$$NOdetP_4(split) = NRE$$

Simulation of the register machine

increment instruction: (p, A_i+, r, s)

Rules of R:

1. $(p,0)(c_i,0) \to (p,2)(c_i,3)$,
2. $(X,1)(Y,1) \to (X,2)(Y,3)$,
3. $(Y,3)(c_i,3) \to (Y,1)(c_i,2)$,
4. $(X,2)(p,2) \to (X,1)(p,0)$,

Rules of M:

$$(a) : Q_p \to pc_iQ'_p,\ (b) : Q'_p \to Q''_p,\ (c) : pQ''_p \to Q_r,\ (d) : pQ''_p \to Q_s$$

On the Environment …

The interesting fact that **both ''infinite'' and finite environments** turn out to be universal is due to that in the ''infinite'' case in any computation step the necessary number of objects that should be/can be imported from the environment is a finite number.

Thus, by **a suitable multiset rewriting system**, the **necessary environmental objects for the core GCPS can be provided.**

On the Environment …

Considering that real-life situations e.g. **workflow enactment** on a **dynamic and heterogeneous distributed infrastructure** has no control on the change of the environment.

In addition, **it must prevent its execution and adapt itself to the unpredictable environmental changes.**

We should consider the case when **the environment is not created just for supporting correctness of the computation,** but it may **cause harmful exceptions** in the course of the execution of the core GCPS.

Classifying the rules

Critical rules perform
interaction between the cell and the environment.

Definition 2. *Let* $\Pi = (O, S, R, w_1, w_n, s, d)$ *be a* deGCPS, $a, b \in O$, *moreover* $1 \leq i, j, k \leq n$. $r \in R$ *is a* critical rule out *if it is in form* $(a, i)(b, k) \to (a, j)(b, 0)$ *or* $(a, i)(b, j) \to (a, 0)(b, k)$. *In addition* $r \in R$ *is a* critical rule in *if it is in form* $(a, i)(b, 0) \to (a, j)(b, k)$ *or* $(a, 0)(b, j) \to (a, i)(b, k)$. R^c *denotes the set of critical rules in* Π, $R^c \subseteq R$.

Applied rule: a rule which is performed during some transition

Classifying the Environments

Semi-responsible, responsible environments:

provide the necessary number of objects for functioning

Definition 4. *An environment M is* semi-responsible *for a given* GCPS *in reference of derivation, if there is at least one derivation in where M provides all of the critical symbols in the required number at least at the time when the belonging communication rules are applied. If M is semi-responsible for* all *of the possible derivations, then M is* responsible

Classifying the Environments

Semi-intruder, intruder, destructive environments:

significantly modify the functioning (GCPS versus deGCPS)

Definition 5. *An environment M is* semi-intruder *for a given* GCPS *in reference of derivation, if there is at least one derivation in where some of the critical rules cannot be applied, but this malfunction does not halts the computation.If M is semi-intruder for* all *of the possible derivations, then M is* intruder *for the given GCPS.*

Results

For any recursively enumerable set S of numbers there exists a deGCPS Π of type either split, or join, or parallel-shift with a **responsible environment** M such that N(Π)=S and M is an **MIL system**.

For a deGCPS Π in a certain subclass of deGCPSs and for any **MD0L** system we **can decide** whether M is a **responsible**, a **semi-intruder**, or an **intruder** environment for Π.

(hints of the proof: periodical applications of critical rules provide some base of the construction)

Future research

- Studying environments other types,
- providing further descriptions of the interaction of the environment and the GCPS.