Length-based communication in tissue-like P systems with string objects

Erzsébet Csuhaj-Varjú[1]
and
Pramod Kumar Sethy[2]

[1]Department of Algorithms and Applications
[2]PhD School in Informatics
Faculty of Informatics,
Eötvös Loránd University, Budapest, Hungary
{csuhaj,pksethy}@inf.elte.hu

Contents

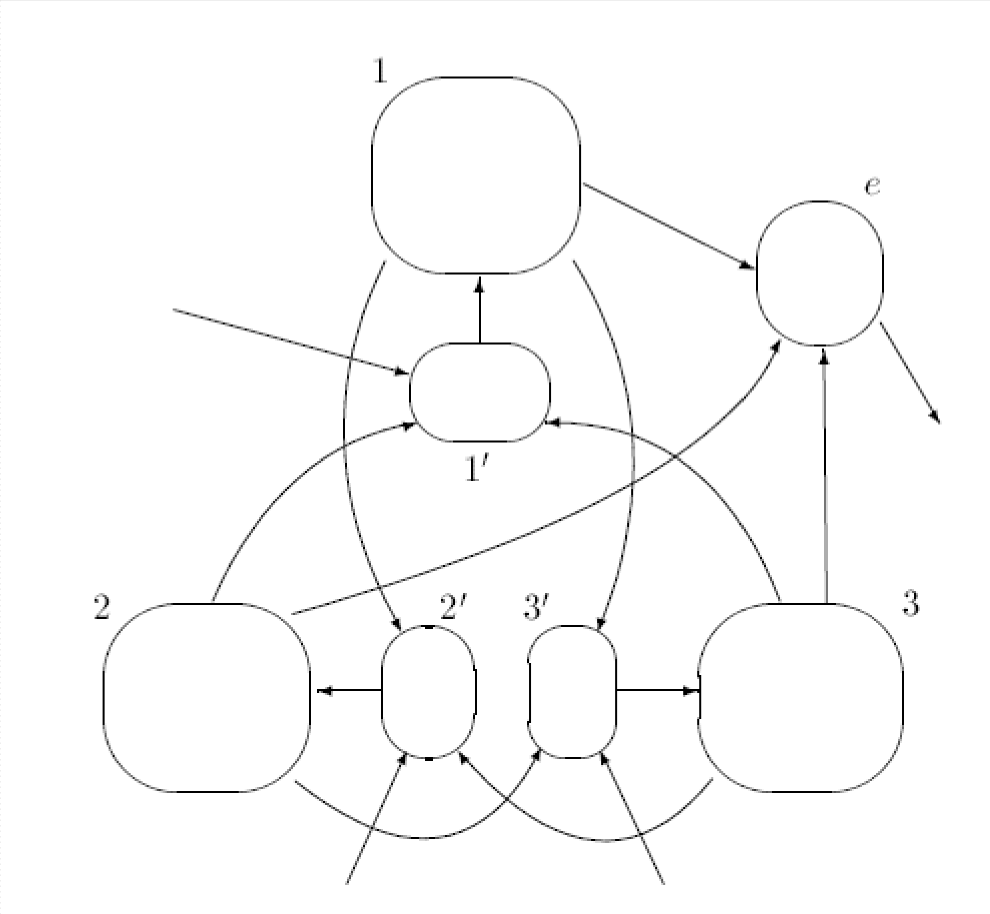
- Tissue-like P systems, communication
- Communication based on lengths of strings
- Problems, expected results
- Conclusions

Tissue-like P Systems with String Objects (stP systems)

Components of a basic model:

- A (directed) **graph**, where the nodes, called the **cells** contain **multisets of strings** over an alphabet of **objects**.
- Each cell is associated with a finite set of **multiset rewriting rules** (evolution rules, communication rules).
- The system may interact with a multiset of objects, called the **environment**.

Tissue-like P systems



See: The Oxford Handbook of Membrane Computing. Chapter 9. F.Bernardini, M. Gheorghe,

Tissue-like P Systems with String Objects

The P system has an **initial configuration**: the collection of initial multisets in the cells.

It **functions with changing its configurations**:
a configuration change consists of a rewriting step and/or communication.

The rewriting step is performed in some mode (**maximally parallel, sequential, minimal**, etc mode).

Communication is performed according to some **communication protocol**.

Communication

In string-object case, **communication** is usually based on qualitative conditions (for example, **context conditions**).

An **interesting problem** is, how large computational power can be obtained if we consider **the lengths of the strings as base of communication.**

(**No contextual information** is used)

Length-Based Communication

Type 1:

$\Pi = (O, G, (R_1, A_1, H_1), \dots, (R_n, A_n, H_n), i_0)$, - tP system with string objects

(R_i, A_i, H_i) - cell i, rewriting rules, axiom strings, length set

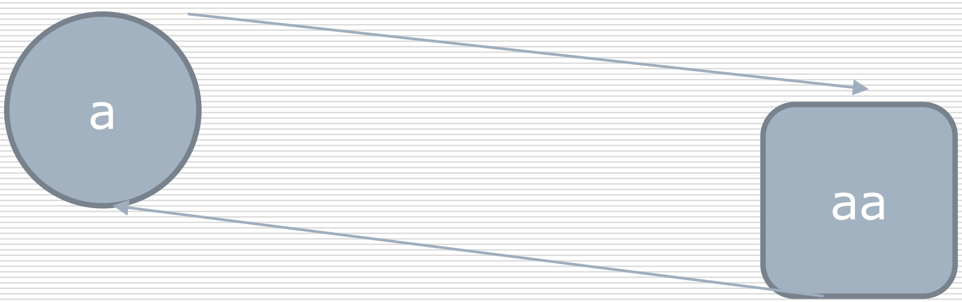
H_i - length set (a set of non-negative integers)

After evolution (rewriting) a copy of those strings which are with length in H_i is sent to each neighbouring cell.

Length-Based Communication - Example

Type 1:

rule: λ→a



$V_1 = \{ 2^n \mid n \geq 0 \}$

$V_2 = \{ 2^n \mid n \geq 0 \}$

Length-Based Communication

Type 2:

$\Pi = (O, G, (R_1, A_1, V_1), \ldots, (R_n, A_n, V_n), i_0)$, - tP system with string objects

(R_i, A_i, V_i) - cell i, rewriting rules, axiom strings, set of Parikh vectors

V_i - set of Parikh vectors over O

After evolution (rewriting), a copy of those strings which are with Parikh vector in V_i is sent to each neighbouring cell.

Length-Based Communication - Question

Type 2:

$$\Pi = (O, G, (R_1, A_1, V_1), \dots, (R_n, A_n, V_n), i_0)$$

How large computational power can be obtained if V_i is a semilinear set for each i, and the rule set of each cell consists of point mutation rules?

Point mutation rules: insertion/deletion/replacement of one object

Conjecture: some non-context-free context-sensitive languages can be obtained

Length-Based Communication

Type 3:

$\Pi = (O, G, (R_1, A_1, \rho_1), \ldots, (R_n, A_n, \rho_n), i_0)$, - tP system with string objects

(R_i, A_i, ρ_i) - cell i, rewriting rules, axiom strings, relation

$\rho_i \in \{+, -, =\}$

Length-Based Communication - continued

Type 3:

Every string is the object of the application of one rule.

If ρ_i is +, then (a copy of) those strings where the new string is longer than the original word (leaves) leave to every neighbouring cell.

If ρ_i is -, then (a copy of) those strings where the new string is shorter than the original word (leaves) leave to every neighbouring cell.

If ρ_i is =, then (a copy of) those strings where the new string is of the same length as the original word (leaves) leave to every neighbouring cell.

Length-Based Communication - continued

Type 3:

Question:

What about the computational power of stP systems of type 3?

Conjecture: if the stP system is with point mutation rules, than these constructs are as powerful as the register machines.

Register Machine

$$M = (Q, R, q_0, q_f, P)$$

where

1. Q is a finite non-empty set, called the set of *states*;

$R = \{A_1, \ldots, A_k\}$, $k \geq 1$, is a set of *registers*;
$q_0 \in Q$ is the *initial state*;
$q_f \in Q$ is the *final state*;
P is a set of *instructions* of the following forms:
(a) $(p, A+, q, s)$, where $p, q, s \in Q$, $p \neq q_f$, $A \in R$, called an *increment instruction*,
(b) $(p, A-, q, s)$, where $p, q, s \in Q$, $p \neq q_f$, $A \in R$, called a *decrement instruction*.

Furthermore, for every $p \in Q$, $(p \neq q_f)$, there is exactly one instruction of the form either $(p, A+, q, s)$ or $(p, A-, q, s)$.

Length-Based Communication

Type 3:

Ideas of simulating the register machine:

Registers' contents is represented by $a_i \ldots a_i$

Cells labelled by q - state

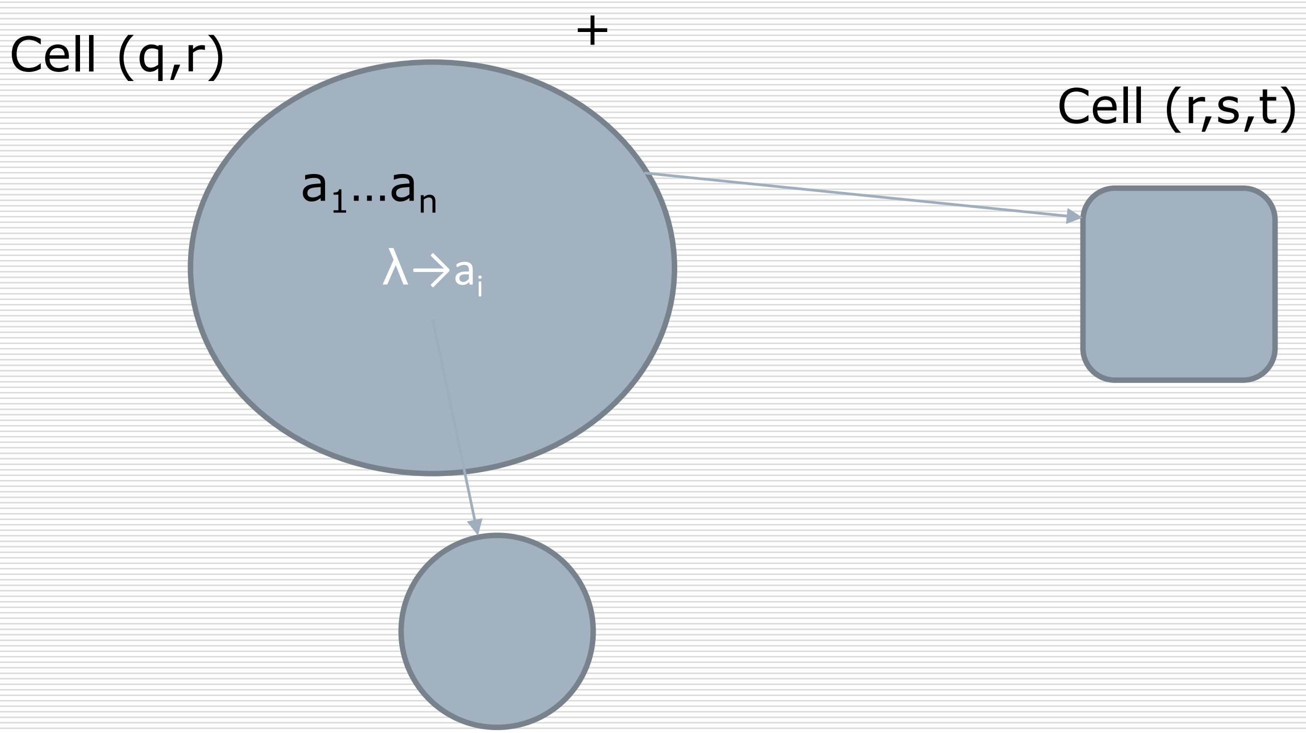
Neighbouring cells are defined by the instructions

(q, A_i+, r), $\left(q, A_j, r, s,\right)$

At any moment, the contents of all registers are stored in one cell in the form of one word.

Length-based Communication

Type 3:



Open Problems

How large computational power can be obtained with particular variants of these stP systems with length-based communication?

How can these systems used in problem solving?

This work was supported by the National Research, Development, and Innovation Office - NKFIH, Hungary, Grant no. K 120558.