17th Brainstorming Week on Membrane Computing

Semaphore concept in tissue-like P systems with string-objects and point mutation rules

Pramod Kumar Sethy

joint work with Erzsébet Csuhaj-Varjú

Eötvös Loránd University
Faculty of Informatics

2019

- Introduction
- Semaphore
  - Types
  - Operations
- tP systems with string-objects and point mutation rules
- Sem-tP systems
- Open problems

Introduction:-

- To implement semaphore concept in a variant of tP systems
- Monitor the communication of strings across the tP system
- Update the semaphore value after each operation that takes place at each cell
- Minimize the semaphore operations
- Propose a new model, ***Sem-tP*** .

Semaphore:-

- A special type of variable used in operating systems
- Used to achieve process synchronization in multi-processing environment
- Also used to support concurrency and communication
- Here we use for the purpose of supporting communication only

Semaphore types:-

- The two most common kinds of semaphores:
  - Counting semaphore: uses non-negative integers
  - Binary semaphore: takes only 0 or 1
- We consider only counting semaphores, and we denote as **τ** .

Semaphore operations:-

- Initial value of **τ** on each node is 0.
- When any evolution rule is applied to a word in a node:

➢ For insertion, the value of **τ** increases by +1

➢ For deletion, the value of **τ** decreases by -1

➢ For substitution, there will be no change in the value of **τ** .

Tissue-like P system with string objects and semaphores (Sem-tP):-

- A Sem-tP is a construct as follows:

  $\Pi = (O, G, (M_1, A_1, \tau_1), \ldots, (M_n, A_n, \tau_n), i_0),\ n \geq 1$

where

- O is the alphabet of objects,
- G is a directed graph of n cells; for each i, $1 \leq i \leq n$, cell i is denoted by (M_i, A_i, τ_i), where
- M_i is a finite set of point mutation rules, A_i is a finite multiset of axiom strings.
- $\tau_i: Z \times Z \rightarrow Z$, $1 \leq i \leq n$, is called the semaphore function of cell i; its initial value is 0.
- $i_0 \in \{1, \ldots, n\}$, the label of the output node.

Sem-tP continued :-

Note:

M_i is a finite set of evolution rules of only one of the following types of rules:

- ✓ $a \rightarrow b, a, b \in V$ (substitution rules),
- ✓ $a \rightarrow \lambda$, $a \in V$ (deletion rules),
- ✓ $\lambda \rightarrow a$, $a \in V$ (insertion rules).

Sem-tP -functioning :-

Sem-tP system functions with changing its configurations.

A sequence of configurations following each other and starting with the initial configuration is called a computation.

The initial configuration is $((A_1, 0)\ldots, (A_n, 0))$.

Sem-tP –functioning continued :-

For two configurations C_1 and C_2, we say that

C_1 directly changes to C_2 if the following hold:

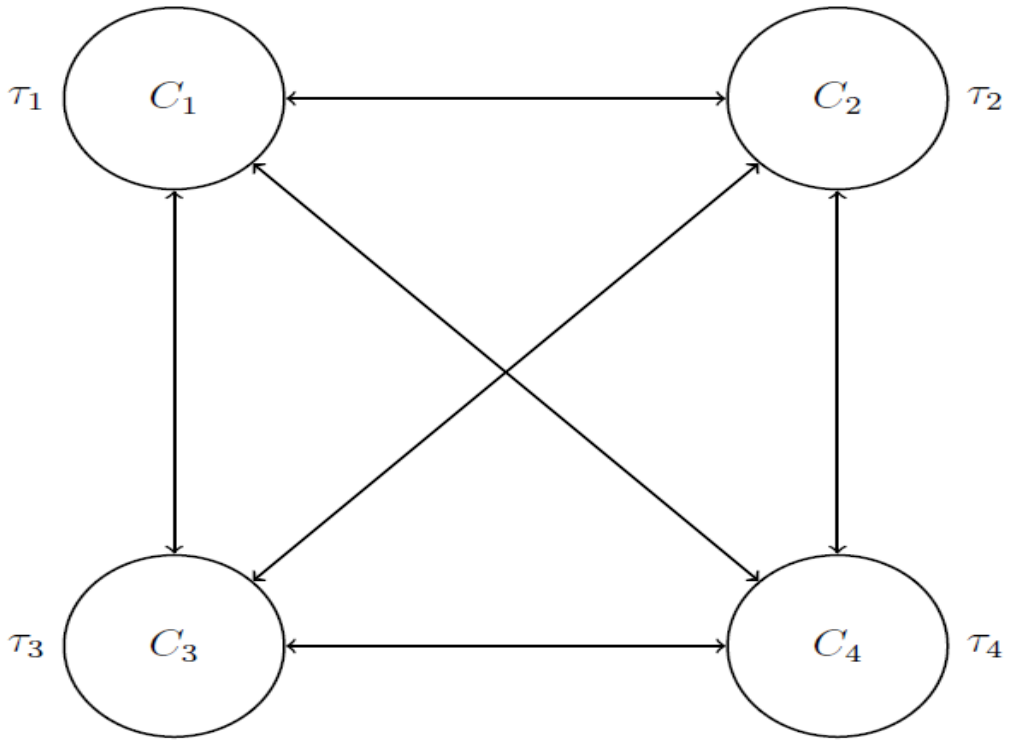
- The cells apply (an arbitrary number of) point mutation rules to the string they have.
- The new value of the semaphore function will be the total number of the values of the performed operations.
- If the total number of the performed operation values is positive, then a copy of those strings which were affected by some operation leaves to the neighbouring cells.
- The cell accepts strings if its semaphore function value is non-negative.

Sem-tP –functioning continued :-

The language of Sem-tP Π is the set of strings that appear in the output cell during the computation.

Example

Sem-tP with four nodes



Open problems

- What can we say about the computational power of Sem-tP systems?
- Can we solve NP complete problems with Sem-tP systems?

Thank you for your attention!