Theoretical Computer Science 608 (2015) 57–65

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Membrane fission versus cell division: When membrane proliferation is not enough

Luis F. Macías-Ramos, Mario J. Pérez-Jiménez, Agustín Riscos-Núñez *, Luis Valencia-Cabrera

Research Group of Natural Computing, Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, Sevilla, 41012, Spain

ARTICLE INFO

Article history:
Received 31 January 2015
Received in revised form 4 May 2015
Accepted 8 June 2015
Available online 11 June 2015

Keywords:
Cell division
Membrane fission
Membrane computing
Symport/antiport rules
Tractability frontier

ABSTRACT

Cell division is a process that produces two or more cells from one cell by *replicating* the original chromosomes so that each daughter cell gets a copy of them. Membrane fission is a process by which a biological membrane is split into two new ones in such a manner that the contents of the initial membrane get *distributed* or *separated* among the new membranes. Inspired by these biological phenomena, new kinds of models were considered in the discipline of *Membrane Computing*, in the context of P systems with active membranes, and tissue P systems that use symport/antiport rules, respectively.

This paper combines the two approaches: cell-like P systems with symport/antiport rules and membrane separation are studied, from a computational complexity perspective. Specifically, the role of the environment in the context of cell-like P systems with membrane separation is established, and additional borderlines between tractability and **NP**-hardness are summarized.

1. Introduction

Cell division is one of the basic processes in the cell life cycle and it allows producing two or more cells from one cell. Basically, there are three processes associated with cell division: *binary fission* (typical of prokaryotic cells), *mitosis* and *meiosis* (these two taking place in eukaryotic cells). In eukaryotic cells, mitosis generally involves forming two identical daughter cells by replicating and dividing the original chromosomes. Meiosis allows genetic variation through a process of DNA shuffling while the cells are dividing.

Several cell division inspired mechanisms were introduced in *Membrane Computing*, a distributed parallel computing paradigm inspired by the way the living cells process chemical substances, energy and information. In this discipline these mechanisms are called *cell division* rules and *membrane division* rules that are triggered by an object which is replaced in the two new cells by possibly new objects and the remaining objects are *duplicated* in both new cells/membranes. These two ways have given rise to cell-like P systems based models (*P systems with active membranes* [21]) and tissue-like P systems based models (*tissue P systems with cell division* [20]), respectively.

Lipid membranes separate the interior of a cell from its environment or surround membrane compartments (mitochondria, endosomes, Golgi complex, etc.) allowing the cells and compartments to have an identity. They serve as concentrations

* Corresponding author.
E-mail addresses: lfmaciasr@us.es (L.F. Macías-Ramos), marper@us.es (M.J. Pérez-Jiménez), ariscosn@us.es (A. Riscos-Núñez), lvalencia@us.es (L. Valencia-Cabrera).

http://dx.doi.org/10.1016/j.tcs.2015.06.025
0304-3975/

barriers allowing the incorporation of material from its environment, in the case of a cell, or exchange material between compartments (from a donor membrane to an acceptor membrane), implemented by membrane carriers. The formation of such carriers in cells follows a simple three-step process whose last step is *membrane fission* consisting in the splitting of the membranes in two new ones [15]. The biological phenomenon of membrane fission was incorporated in *Membrane Computing* through a new kind of rules, called *membrane separation rules*, in the framework of polarizationless P systems with active membranes [3]. Originally, each of these separation rules could have their own partition of the working alphabet. Nevertheless, in [16] a new definition of separation rules in the framework of P systems with active membranes was introduced. In the new definition, all separation rules are associated with the same partition of the working alphabet in two subsets, which is given in advance. By applying these kinds of rules, the object triggering them is consumed and the remaining objects are *distributed* both in the created membranes.

Networks of membranes which compute by communication only, using symport/antiport rules were considered in [18]. By means of this kind of rules, a change of the places of objects with respect to the membranes of a system takes place along computations but not a change/evolution of the objects themselves. Such rules are used both for communication with the environment and for direct communication between different membranes. It is worth noting that, in these systems, the environment plays an active role because we cannot only send objects outside the system, but we can also bring in objects from the environment.

With respect to tissue-like approaches, from the seminal definitions of tissue P systems [13,14], one of the most interesting variants of tissue P systems was presented in [20]. In that paper, the definition of tissue P systems with symport/antiport rules is combined with the one of P systems with active membranes, yielding *tissue P systems with cell division*. One of the latest studies on their computational power can be found in [22]. In tissue-like systems, the membrane fission phenomenon has been considered together with symport/antiport rules [17] by means of cell separation rules. These models are called *tissue P systems with cell separation* and its computational efficiency was investigated. Besides, a tractability border in terms of upper bound of the length of communication rules was obtained: passing from 1 to 8 amounts to passing from non-efficiency to efficiency, assuming that $\mathbf{P} \neq \mathbf{NP}$ [17]. Nevertheless, in [24] that frontier was refined in an optimal sense.

Cell-like P systems with symport/antiport rules were introduced in [19]. This kind of P systems was shown to be computationally complete.[1] In this paper we consider membrane separation in the framework of cell-like P systems with symport/antiport rules, and the computational efficiency of these systems is investigated.

The paper is organized as follows. Next section briefly describes some preliminaries in order to make the work self-contained. In Section 3, complexity classes of recognizer P systems with symport/antiport rules are introduced. In Section 4, the main result is presented: only tractability problems can be solved by families of P systems with symport/antiport rules, membrane separation and without environment. Section 5 summarizes different boundaries between tractability and **NP**-hardness in the framework of recognizer P systems with symport/antiport rules. Finally, conclusions and some open problems are drawn.

2. Preliminaries

An *alphabet* Γ is a non-empty set and their elements are called *symbols*. A *string* u over Γ is an ordered finite sequence of symbols, that is, a mapping from a natural number $n \in \mathbb{N}$ onto Γ. Number n is called the *length* of string u and it is denoted by $|u|$, that is, the length of a string is the number of occurrences of symbols it contains. The empty string (with length 0) is denoted by λ. The set of all strings over an alphabet Γ is denoted by Γ^*. A *language* over Γ is a subset of Γ^*.

A *multiset* over an alphabet Γ is an ordered pair (Γ, f) where f is a mapping from Γ onto the set of natural numbers $\mathbb{N}$. The *support* of a multiset $m = (\Gamma, f)$ is defined as $supp(m) = \{x \in \Gamma : f(x) > 0\}$. A multiset is finite (resp. empty) if its support is a finite (resp. empty) set. We denote by $\emptyset$ the empty multiset.

Let $m_1 = (\Gamma, f_1)$, $m_2 = (\Gamma, f_2)$ be multisets over Γ, then the union of m_1 and m_2, denoted by $m_1 + m_2$, is the multiset (Γ, g), where $g(x) = f_1(x) + f_2(x)$ for each $x \in \Gamma$. We say that m_1 is contained in m_2 and we denote it by $m_1 \subseteq m_2$, if $f_1(x) \leq f_2(x)$ for each $x \in \Gamma$. The relative complement of m_2 in m_1, denoted by $m_1 \setminus m_2$, is multiset (Γ, g), where $g(x) = f_1(x) - f_2(x)$ if $f_1(x) \geq f_2(x)$, and $g(x) = 0$ otherwise.

A *symport rule* (respectively, *antiport rule*) over an alphabet Γ is an expression (u, out) or (u, in), where u is a finite multiset over Γ such that $|u| > 0$ (resp. an expression $(u, out; v, in)$, where u, v are finite multisets over Γ such that $|u| > 0$ and $|v| > 0$). The length of rule (u, out) or (u, in) (resp. $(u, out; v, in)$) is defined as $|u|$ (resp. $|u| + |v|$). A *division rule* over Γ is an expression $[\,a\,]_i \rightarrow [\,b\,]_i\,[\,c\,]_i$, where $a, b, c \in \Gamma$. A *separation rule* over Γ is an expression $[a]_i \rightarrow [\Gamma_0]_i[\Gamma_1]_i$ where $a \in \Gamma$ and $\{\Gamma_0, \Gamma_1\}$ is a partition of Γ.

Let us recall that a *free tree* (*tree*, for short) is a connected, acyclic, undirected graph. A *rooted tree* is a tree in which one of the vertices (called *the root of the tree*) is distinguished from the others. In a rooted tree the concepts of ascendants and descendants are defined as follows. Given a node x (different from the root), if the last edge on the (unique) path from the root to the node x is $\{x, y\}$ (in this case, $x \neq y$), then y is **the** *parent* of node x and x is **a** *child* of node y. The root is the only node in the tree with no parent. A node with no children is called a *leaf* (see [4] for details).

[1] In [2], the authors show that systems with $s \geq 2$ symbols and $m \geq 1$ membranes (such that $m + s \geq 6$) are universal.