Theoretical Computer Science 673 (2017) 42–57

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A toolbox for simpler active membrane algorithms

Alberto Leporati, Luca Manzoni, Giancarlo Mauri, Antonio E. Porreca *, Claudio Zandron

Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Viale Sarca 336/14, 20126 Milano, Italy

ARTICLE INFO

Article history:
Received 15 September 2016
Received in revised form 13 March 2017
Accepted 17 March 2017
Available online 22 March 2017
Communicated by M.J. Pérez-Jiménez

Keywords:
Membrane computing
Computational complexity
P system with active membranes

ABSTRACT

We show that recogniser P systems with active membranes can be augmented with a priority over their set of rules and any number of membrane charges without loss of generality, as they can be simulated by standard P systems with active membranes, in particular using only *two* charges. Furthermore, we show that more general accepting conditions, such as sending out several, possibly contradictory results and keeping only the first one, or rejecting by halting without output, are also equivalent to the standard accepting conditions. The simulations we propose are always without significant loss of efficiency, and thus the results of this paper can hopefully simplify the design of algorithms for P systems with active membranes.

1. Introduction

P systems with active membranes [14] have been extensively investigated as computing devices, both from the computability and the computational complexity standpoints.

By analysing the algorithms for P systems with active membranes described in the literature, it is possible to identify a number of useful and recurring techniques or "design patterns". A standard one is using elementary membrane division to produce all assignments of a set of variables $x_1, \ldots, x_n$ [14]; the results of evaluating a Boolean formula under those assignments can then be combined in several ways:

- by disjunction, allowing the solution of the SAT problem, and thus all **NP**-complete problems [21];
- by counting the number of satisfying assignments against a threshold, allowing the solution of **PP**-complete counting problems [17];
- by alternating disjunctions and conjunctions by means of a tree-shaped membrane structure of depth n, allowing the solution of **PSPACE**-complete problems [20].

Other techniques involve simulating register machines [6] or Turing machines [3], also in their nondeterministic version, by simulating nondeterminism with parallelism as above for solving **NP**-complete problems [10]. Membranes at different nesting levels can also be employed as "subroutines", simulating multiple Turing machines and becoming functionally equivalent to oracles for subproblems [10].

* Corresponding author.
E-mail addresses: leporati@disco.unimib.it (A. Leporati), luca.manzoni@disco.unimib.it (L. Manzoni), mauri@disco.unimib.it (G. Mauri), porreca@disco.unimib.it (A.E. Porreca), zandron@disco.unimib.it (C. Zandron).

http://dx.doi.org/10.1016/j.tcs.2017.03.015
0304-3975/

While the main ideas behind those constructions are generally straightforward and show clear affinity with techniques from the theory of traditional computing devices, their implementation unfortunately often involves a number of technical details which obfuscate the big picture. One of the main culprits are the ubiquitous timer objects, which keep the different parts of the P system synchronised and allow the halting of the computation immediately after producing the output, a condition that is usually imposed by the definition of *recogniser P systems* [16] and that often requires extra work to be met.

One of the crucial aspects of the definition of P systems with active membranes is the number of possible membrane charges, which is three in the original definition. Although charges are not needed to solve **PSPACE**-complete problems in polynomial time[1] [4] and two charges suffice to achieve universality [2], having access to a number of charges growing with the size of the input allows a simpler implementation of many algorithms. For instance, when this is allowed, the simulation of bounded-tape Turing machines becomes trivial [10]. In that paper, an arbitrary number of charges was reduced to three without loss of efficiency, but only in a very restrictive set of circumstances (essentially, no communication with adjacent membranes is allowed, and the membranes must behave deterministically).

The purpose of this paper is twofold. On the one hand, we want to understand which features of recogniser P systems with active membranes are actually essential to characterise their behaviour. On the other hand, we want to provide an array of useful extensions which can be added to P systems with active membranes but can be simulated by the original model without loss of efficiency. This will hopefully reduce the amount of "boilerplate code" (repetitive rules unrelated to the main algorithm) in proofs and allow focusing on a higher-level description of P systems, such as dividing membranes working in parallel and their communication patterns.

The formally redundant but convenient features we describe in this paper are the ability to use any number of charges, any partial priority ordering of rules (as in the original definition of transition P systems [13]), and the ability to output the result of the computation in less restrictive ways, such as not requiring the P system to halt after having sent out the result, or rejecting by halting without output. Furthermore, we show that all these enhancements can be simulated efficiently by standard recogniser P systems with active membranes using only *two* charges (even when working in super-polynomial time).

2. Basic notions

For an introduction to membrane computing and the related notions of formal language theory and multiset processing, we refer the reader to *The Oxford Handbook of Membrane Computing* [15] or to any introductory membrane computing survey [18]. Here we recall the formal definition of P systems with active membranes using weak non-elementary division rules [14,22].

Definition 1. A *P system with active membranes with weak non-elementary division rules* of initial degree $d \geq 1$ is a tuple

$$\Pi = (\Gamma, \Lambda, \mu, w_{h_1}, \ldots, w_{h_d}, R)$$

where:

- Γ is an alphabet, i.e., a finite non-empty set of symbols, usually called *objects*;
- Λ is a finite set of labels for the membranes;
- μ is a membrane structure (i.e., a rooted *unordered* tree, usually represented by nested brackets) consisting of d membranes labelled by elements of Λ in a one-to-one way;
- $w_{h_1}, \ldots, w_{h_d}$, with $h_1, \ldots, h_d \in \Lambda$, are multisets (finite sets with multiplicity) of objects in Γ, describing the initial contents of the d regions of μ;
- R is a finite set of rules.

Each membrane possesses, besides its label and position in μ, another attribute called *electrical charge*, which can be either neutral (0), positive (+) or negative (−) and is always neutral before the beginning of the computation.

The rules in R are of the following types:

(a) *Object evolution rules*, of the form $[a \to w]_h^\alpha$
They can be applied inside a membrane labelled by h, having charge α and containing an occurrence of the object a; the object a is rewritten into the multiset w (i.e., a is removed from the multiset in h and replaced by the objects in w).

(b) *Send-in communication rules*, of the form $a[\,]_h^\alpha \to [b]_h^\beta$
They can be applied to a membrane labelled by h, having charge α and such that the external region contains an occurrence of the object a; the object a is sent into h becoming b and, simultaneously, the charge of h is changed to β.

[1] However notice that, in the absence of membrane dissolution rules, the lack of charges seems to reduce the efficiency of P systems [9].