Optik 127 (2016) 8231–8239

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Original research article

Design a membrane system for matrix multiplication

Lian Ye [a,b,*], Ping Guo [a,b]

[a] College of Computer Science, Chongqing University, Chongqing 400030, China
[b] Chongqing Key Laboratory of Software Theory and Technology, Chongqing 400044, China

ARTICLE INFO

Article history:
Received 19 April 2016
Accepted 8 June 2016

Keywords:
Cell-like *P* systems
Matrix multiplication
Parallel computing
Membrane computing

ABSTRACT

Membrane systems are a framework of distributed parallel computing models, inspired by some basic features of biological membranes. In this paper, a membrane computing system named Π_{MM} is proposed to solve multiplication operation with matrices by its parallelism. It mainly includes three stages: firstly, construct new embedded membranes and input the values of two matrices; secondly, divide the membranes to generate new membranes according to the rows of matrix; finally, the solutions are obtained in each row membranes under the relatively calculation rules side by side. An instance is given to show the whole procedure how to solve the matrix multiplication by the Π_{MM}.

1. Introduction

Matrix operations have been widely used in the field of data mining, such as image processing, text mining, recommendation system, bioinformatics and so on. Matrix multiplication is one of important matrix arithmetic operations.

Membrane computing is an emerging branch within natural computing [1], this new model of computation starts from the assumption that the processes taking place in the compartmental structure of a living cell can be interpreted a computations. Membrane system is a computing system based on the principle of membrane computing, which consists of a hierarchical arrangement of membranes embedded in a skin membrane, and delimiting regions or compartments where multi-sets of objects and sets of evolution rules are placed. *P* systems are not used as a computing paradigm, but rather as formalism for describing the behavior of the system to be modeled.

On the one hand, it is commonly accepted the belief that living systems perform some information processing to keep them far from dynamical equilibrium as well as to accomplish several particular tasks [2] allowing them to react and to adapt to the environment they live in.

On the other hand, people look at the way nature computes with the purpose to abstract its information processing mechanisms in such a way to obtain new computational paradigms that can be put beside traditional ones to tackle specific problems, like NP problems in a feasible time. A SAT problem was solved by the splitting rule of membrane system [3], a HPP problem was also solved by the generating rules [4], and a hybrid membrane evolutionary algorithm was proposed for solving constrained optimization problems [5].

In this paper, we propose a *P* system named Π_{MM} to solve matrix arithmetic, which is a new field in membrane computing. Of course, the new computing system is based on our previous works. We designed the single-membrane arithmetic *P* system without priority rules to simplify the structure of the system greatly and improve the efficiency of computations at first [6]. Then we have designed an arithmetic expression evaluations system to provide an effective method for applications

* Corresponding author at: College of Computer Science, Chongqing University, Chongqing 400030, China.
E-mail addresses: ylredleaf@cqu.edu.cn (L. Ye), guoping@cqu.edu.cn (P. Guo).

http://dx.doi.org/10.1016/j.ijleo.2016.06.040

of numerical computation with membrane computing [7], and proposed different membranes for implementing primary Boolean and relational operations respectively for evaluating a logical expression [8]. In Ref. [9], a fraction arithmetic *P* system was presented for performing addition, subtraction, multiplication and division on fractions through designing the rules with priority, and then a method of fraction reduction was proposed [10]. The fraction arithmetic *P* system is considered as a calculator component in Π_{MM}, once the former operands are sending into the component, the operands of arithmetic result will be given out.

The organization of the paper is as follows: the foundation of *P* systems and the standard notations used for matrices and definition of multiplication operation with matrices are introduced in Section 2; the details of Π_{MM}, including the membrane structure and evolving rules are proposed in Section 3. In Section 4, an example is presented to show how the Π_{MM} works. The conclusions are proposed in the final section.

2. Preliminaries

In membrane computing there are basically three ways to consider computational devices: cell-like systems [11,12], tissue-like systems [13] and Neural-1ike Systems [14,15]. The first one, using the biological membranes arranged hierarchically, inspired from the structure of the cell; the second one using the biological membranes placed in the nodes of a graph, inspired from the cell inter-communications in tissues; the third one, Spiking neural *P* systems are a class of distributed parallel computing devices inspired from the way neurons communicate by means of spikes. Our work is focused on the cell-like systems. So all the basic knowledge about cell-like systems and the description of matrix operation are given in this section.

2.1. Cell-like P systems

The main syntactic ingredients of a cell-like *P* system are the membrane structure, the multi-sets, and the evolution rules. A membrane structure consists of several membranes arranged in a hierarchical structure inside a main membrane (the skin), and delimiting regions (the space in-between a membrane and the immediately inner membranes). Each membrane identifies a region inside the system. Regions defined by a membrane structure contain objects corresponding to chemical substances present in the compartments of a cell. The objects can be described by symbols or by strings of symbols, in such a way that multi-sets of objects are placed in regions of the membrane structure. The objects can evolve according to given evolution rules, associated with the regions. The semantics of the cell-like membrane systems is defined through a nondeterministic and synchronous model, by introducing the concepts of configuration, transition step, and computation.

A basic transition *P* system of degree $m \geq 1$ is a tuple [1],

$$\Pi = (O, \mu, \omega_1, \ldots, \omega_m, (R_1, \rho_1), (R_2, \rho_2), \ldots, (R_m, \rho_m), i_o) \tag{1}$$

where,

(i) O is the alphabet of the system;
(ii) μ is a membrane structure consisting of m membranes, which are labeled by numbers in the set $\{1, \ldots, n\}$;
(iii) $\omega_1 \cdots \omega_m$ are multi-sets, representing the objects initially presented in the regions $(1, \ldots, m)$ of the system;
(iv) $R_1, \ldots, R_m$ are finite sets of evolution rules associated with the regions (1, . . ., m) of μ; $(\rho_1, \ldots, \rho_m)$ are strict partial order relations defined over $(R_1, \ldots, R_m)$ respectively, specifying a priority relation among the evolution rules; The rule can be described as the form $(u \to v, \rho_i)$, where, $u \to v$ is a rewrite rule, and $\rho_i (1 \leq i \leq m)$ indicates the priority.
(v) i_o is the output region.

2.2. Matrix multiplication

More generally, it is possible to multiply a matrix *A* times a matrix *B* if the number of columns of *A* equals the number of rows of *B*. The first column of the product is determined by the first column of *B*; that is, the first column of *AB* is Ab_1, the second column of *AB* is Ab_2, and so on. Thus the product *AB* is the matrix whose columns are $Ab_1, Ab_2, \ldots, Ab_n$:

$$AB = (Ab_1, Ab_2, \ldots, Ab_n) \tag{2}$$

The (i,j) entry of *AB* is the i_{th} entry of the column vector Ab_j. It is determined by multiplying the i_{th} row vector of *A* times the j_{th} column vector of *B*.

Definition 1. if $A = (a_{ij})$ is a $(m \times k)$ matrix and $B = (b_{ij})$ is a $(k \times n)$ matrix, then the product $AB = C = (c_{ij})$ is the $(m \times n)$ matrix whose entries are defined by

$$C_{ij} = a_i b_j = \sum_{l=1}^{k} a_{il} b_{lj} \tag{3}$$