A Perfect Shuffle Algorithm for Reduction Processes and its Simulation with P Systems[1]

Rodica CETERCHI[2], Mario J. PÉREZ JIMÉNEZ[3]

[2]Faculty of Mathematics and Computer Science, University of Bucharest
Academiei 14, RO-010014, Bucharest, Romania

[3] Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: rc@funinf.cs.unibuc.ro, mario.perez@cs.us.es

Abstract. We present in this paper an algorithm for reduction based on the parallel architecture known as *shuffle–exchange network*, or *perfect shuffle*. Next, we use a new version of P systems with communication, designed to simulate this parallel architecture, in order to show how and when they can be used to solve reduction.

Keywords: Membrane Computing, Perfect Shuffle Architecture, Communication P Systems.

1 Introduction

A parallel machine consists of a large number of processors (each one having an arithmetic logic unit with registers and a private memory) able to solve problems in a cooperative way; that is, the machine is capable of executing several instructions in the same time unit.

P systems are powerful computational devices, with a high degree of parallelism, whose functioning is inspired by biological processes at the level of the cells, and of their membranes ([4],[5]). Among these processes, communication plays an important role (see [3]).

In [2] we have studied the perfect shuffle **SIMD** computer, where the processor interconnections are based on the *shuffle* and the *exchange* connections, and we have analyzed the possibility to simulate this parallel architecture by cellular computing with membranes. A new version of P systems, called with *periodic dynamic communication* graphs, has been proposed. They are P systems with communication, for which the communication graphs are not fixed, but have a dynamic evolution, and for which different communication rules can be associated to different communication graphs. A first simulation of some parallel algorithms, using P systems with communication, can be considered [1], where emphasis was on the problem of sorting.

In this paper we present an algorithm for reduction based on the perfect shuffle computer. Next, we show how such an algorithm can be simulated using the new P systems introduced in [2].

The paper is organized as follows. In Section **??** we give some preliminary notions on the perfect shuffle (PS) architecture. In Section **??**, following [2], we introduce a variant of P systems with communication modeling perfect shuffle machines. Section **??** is devoted to the presentation of a reduction algorithm based on the PS architecture, and the proof of its correctness. In section **??** we present the P systems with periodic dynamic communication used to simulate such algorithms, and we show under what conditions such a simulation is possible. Finally, some conclusions of our study are presented.

[1]This work has been partially supported by the Ministerio de Ciencia y Tecnología of Spain and the EU program FEDER, project TIC2002-04220-C03-01

References

[1] Ceterchi, R., Martín–Vide, C.: P systems with Communication for Static Sorting. In M. Cavaliere, C. Martín–Vide and Gh. Păun (eds.), *Proceedings of the Brainstorming Week on Membrane Computing*, Report GRLMC 26/03, 2003, 101–117.

[2] Ceterchi, R., Pérez Jiménez, M. J.: Simulating Shuffle–Exchange Networks with P Systems. In Gh. Păun, A. Riscos, F. Sancho and A. Romero (eds.), *Proceedings of the Second Brainstorming Week on Membrane Computing*, Report RGNC 01/04, 2004, 117-129.

[3] Păun, A., Păun, G.: The power of Communication: P Systems with Symport/Antiport. *New Generation Computers*, 20, 3(2002), 295–306.

[4] Păun, G.: Computing with Membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for CS-TUCS Report* No. 208, 1998

[5] Păun, G.: *Membrane Computing. An Introduction*, Springer-Verlag, Berlin, Heidelberg, 2002

[6] Quinn, M.J. *Parallel Computing. Theory and Practice*, McGraw–Hill Series in Computer Science, 1994.