Chapter 5

Asynchronous P Systems

Tudor Bălănescu
University of Piteşti, Romania

Radu Nicolescu
University of Auckland, New Zealand

Huiling Wu
University of Auckland, New Zealand

ABSTRACT

In this paper, the authors propose a new approach to fully asynchronous P systems, and a matching complexity measure, both inspired from the field of distributed algorithms. The authors validate the proposed approach by implementing several well-known distributed depth-first search (DFS) and breadth-first search (BFS) algorithms. Empirical results show that the proposed P algorithms have shorter descriptions and achieve a performance comparable to the corresponding distributed algorithms.

1. INTRODUCTION

A P system is a computational model inspired by the structure and interactions of cell membranes, introduced by Păun (1998) and Păun, Rozenberg, and Salomaa (2010). The essential specification of a P system includes a membrane structure, objects and rules. Cells evolve by applying rules in a non-deterministic and (potentially maximally) parallel manner. These characteristics make P systems a promising candidate as a model for distributed and parallel computing.

The traditional P system model is *synchronous*, i.e., all cells evolution is controlled by a single global clock. P systems with various *asynchronous* features have been investigated by recent research (Frisco, 2004; Casiraghi, Ferretti, Gallini, & Mauri, 2005; Cavaliere & Sburlan, 2004; Cavaliere et al., 2008; Cavaliere & Mura, 2008; Cavaliere et al., 2009; Freund, 2005; Gutiérrez-Naranjo & Pérez-Jiménez, 2010; Kleijn & Koutny, 2006; Pan, Zeng, & Zhang, 2011; Yuan & Zhang, 2007). Here we are looking for similar but simpler definitions, closer to the definitions used in

DOI: 10.4018/978-1-4666-4253-9.ch005

distributed algorithms (Lynch, 1996; Tel, 2000), which should in the future enable us to consider essential distributed features, such as fairness, safety, liveness and infinite evolutions. In our approach, algorithms are non-deterministic, not necessarily constrained to return exactly the same result.

Fully asynchronous P systems are characterized by the absence of any system clock, let alone a global one; however, an outside observer may very well use a clock to time the evolutions. Our approach, based on classical notions in distributed algorithms, as presented by Tel (2000), does *not* require any change in the *static* description of P systems, only their *evolutions* differ (i.e., the underlying runtime engine works differently): (1) for each cell, each step starts after a *random delay t* (after the preceding step); (2) for each cell, each step, once started, takes *zero* time (i.e., it occurs instantaneously); (3) for each message, its transmission *delay t* is *random* (from its origin until it arrives at its target).

A full version of this paper appears as a CDMTCS report (Nicolescu & Wu, 2011b).

2. PRELIMINARIES

We assume that the reader is familiar with the basic terminology and notations, such as alphabets, strings, multisets, relations, graphs, nodes (vertices), edges, directed graphs (digraphs), directed acyclic graphs (dags), arcs, depth-first search (DFS), breadth-first search (BFS), spanning tree, DFS and BFS spanning trees.

In this paper, we use a *simple P module*—an umbrella concept, which is general enough to cover several basic *synchronous* P system families, with *states*, *priorities*, *promoters* and *duplex* channels. In this basic model, the cells evolve *synchronously*. For the full definition of P modules and modular compositions, we refer readers to Dinneen, Kim, and Nicolescu (2010c). Essentially, a simple P module is a system, $\Pi = (O, \sigma 1, \sigma_2, \ldots, \sigma_n, \delta)$, where:

1. O is a finite non-empty alphabet of *elementary objects*;
2. $\sigma 1, \sigma_2, \ldots, \sigma_n$ are cells, of the form $\sigma_i = (Q_i, S_{i,0}, w_{i,0}, R_i)$, $1 \leq i \leq n$, where:
    a. Q_i is a finite set of *states*;
    b. $S_{i,0} \in Q_i$ is the *initial state*;
    c. $w_{i,0} \in O^*$ is the *initial multiset* of objects;
    d. R_i is a finite *ordered* set of rewriting/communication *rules* of the form: $S\ x \rightarrow_{\alpha} S'\ x'\ (y)_{\beta}\ |_{z}$, where: $S, S' \in Q_i$, $x, x', y, z \in O^*$, $\alpha \in \{\min, \max\}$, $\beta \in \{\uparrow, \downarrow, \updownarrow\}$.
3. δ is a set of *digraph* (*head,tail*) arcs on $\{1, 2, \ldots, n\}$, without reflexive arcs, representing *duplex* channels between cells.

The membrane structure is a digraph with duplex channels, so heads can send messages to tails *and* tails to heads. Rules are prioritized and are applied in *weak priority* order (Păun, 2006). The general form of a rule, which transforms state S to state S', is $S\ x \rightarrow_{\alpha} S'\ x'\ (y)_{\beta}\ |_{z}$. This rule consumes multiset x, and then (after all applicable rules have consumed their left-hand objects) produces multiset x', in the same cell (*here* mode). Also, it produces multiset y and sends it, by *replication* (*repl* mode), to all heads ("up"), to all tails ("down") or to all heads and tails ("up and down"), according to the target indicator $\beta \in \{\uparrow, \downarrow, \updownarrow\}$.

Additionally, we also use a targeted sending, $\beta = \uparrow_j, \downarrow_j, \updownarrow_j$, where j is either an arc label. Each arc has two labels—at its tail and at its head. Arc labels can be used for directing messages to a specific target cell. An arc, $\gamma = (\sigma_i, \sigma_j)$, which is not explicitly labelled, is implicitly labelled with the indices of the two cells, i.e., in this case, γ's labels are j, for its tail, and i, for its head.

Operator $\alpha \in \{\min, \max\}$ describes the rewriting mode. In the *minimal* mode, an applicable rule is applied once. In the *maximal* mode, an applicable rule is used as many times as possible and all rules with the same states S and S' can be applied in the maximally parallel manner. Finally,

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/asynchronous-p-systems/80058

Related Content

Algorithms and Computations in BL-Algebras
Celestin Lele (2010). *International Journal of Artificial Life Research (pp. 29-47).*
www.irma-international.org/article/algorithms-computations-algebras/49682

Multi-Objective Optimization Evolutionary Algorithms in Insurance-Linked Derivatives
M. J. Perez (2007). *Handbook of Research on Nature-Inspired Computing for Economics and Management (pp. 885-908).*
www.irma-international.org/chapter/multi-objective-optimization-evolutionary-algorithms/21172

Assessment of Multi-Engine Machine Translation for English to Hindi Language (MEMTEHiL): Using F&A and iBLEU Metrics
Pankaj K. Goswami, Sanjay K. Dwivediand C. K. Jha (2016). *International Journal of Artificial Life Research (pp. 30-45).*
www.irma-international.org/article/assessment-of-multi-engine-machine-translation-for-english-to-hindi-language-memtehil/177183

Using a Hybrid System Composed of Neural Networks and Genetic Algorithms for Financial Forecasting
Mihaela Dumitrescu (2017). *Nature-Inspired Computing: Concepts, Methodologies, Tools, and Applications (pp. 999-1007).*
www.irma-international.org/chapter/using-a-hybrid-system-composed-of-neural-networks-and-genetic-algorithms-for-financial-forecasting/161059

Bio-Inspired Algorithms Used in Medical Image Processing
K. Ezhilarasan, K. Somasundaram, T. Kalaiselvi, Praveenkumar Somasundaram, S. Karthigai Selviand A. Jeevarekha (2024). *Bio-Inspired Optimization Techniques in Blockchain Systems (pp. 25-60).*
www.irma-international.org/chapter/bio-inspired-algorithms-used-in-medical-image-processing/338083