

Chapter 6.10

Forward and Backward Chaining with P Systems

Sergiu Ivanov

Academy of Sciences of Moldova and Technical University of Moldova, Moldova

Artiom Alhazov

Academy of Sciences of Moldova, Moldova, and Università degli Studi di Milano-Bicocca, Italy

Vladimir Rogojin

Helsinki University, Finland and Academy of Sciences of Moldova, Moldova

Miguel A. Gutiérrez-Naranjo

Academy of Sciences of Moldova, Moldova, and University of Sevilla, Spain

ABSTRACT

One of the concepts that lie at the basis of membrane computing is the multiset rewriting rule. On the other hand, the paradigm of rules is profusely used in computer science for representing and dealing with knowledge. Therefore, establishing a “bridge” between these domains is important, for instance, by designing P systems reproducing the modus ponens-based forward and backward chaining that can be used as tools for reasoning in propositional logic. In this paper, the authors show how powerful and intuitive the formalism of membrane computing is and how it can be used to represent concepts and notions from unrelated areas.

1. INTRODUCTION

The use of rules is one of the most common paradigms in computer science for dealing with knowledge. Given two pieces of knowledge V and W , expressed in some language, the rule $V \rightarrow W$ is usually considered as a causal relation between V and W . This representation is universal in sci-

ence. For example, in chemistry, V and W can be metabolites and $V \rightarrow W$ a chemical reaction. In this case, V represents the reactants which are consumed in the reaction and W is the obtained product. In propositional logic, $V \rightarrow W$, with $V = v_1 \vee v_2 \vee \dots \vee v_n$ and $W = w_1 \vee w_2 \vee \dots \vee w_m$, is a representation of the clause $\neg v_1 \vee \neg v_2 \vee \dots \vee \neg v_n \vee w_1 \vee w_2 \vee \dots \vee w_m$.

An important problem is deriving new knowledge: given a knowledge base $KB = (A, R)$, where

DOI: 10.4018/978-1-61350-456-7.ch6.10

A is a set of known atoms and R is a set of rules of type $V \rightarrow W$, the problem is to know if a new atom g can be obtained from the known atoms and rules. We will call this problem a *reasoning problem* and it will be denoted by $\langle A, R, g \rangle$.

In computer science, there are two basic methods for seeking a solution of a reasoning problem, both of them based on the inference rule known as Generalized Modus Ponens: the former is data-driven and it is known as *forward chaining*, the latter is query-driven and it is called *backward chaining* (Apt, 1990).

As one should observe, even though logic inference rules and multiset rewriting rules originate from totally different areas of mathematics and computer science and represent unrelated notions, their concepts have some similarities. In particular, no information about the ordering of elements in both left- and right-hand sides of the rules of both types is used. On the other hand, the inference rules could be thought of as set rewriting rules, while multiset rewriting rules operate at multisets. However, multiset rewriting rules could be interpreted as set rewriting rules if one ignores the multiplicity of elements of the multiset. Therefore we could represent sets of facts in P systems as multisets of objects and inference rules as multiset rewriting rules. When one considers the set of facts represented in a region of a P system, one only considers the underlying set of the region's multiset.

2. DEFINITIONS

2.1. Formal Logic Preliminaries

An *atomic formula* (also called an *atom*) is a formula with no deeper structure. An atomic formula is used to express some fact in the context of a given problem (Jago, 2007). The *universal set* of atoms is denoted with U . U is finite. For a set A , $|A|$ is the number of elements in this set (cardinality).

A *knowledge base* is a construct $KB = (A, R)$ where $A = \{a_1, a_2, \dots, a_n\} \subseteq U$ is the set of known atoms and R is the set of rules of the form $V \rightarrow W$, with $V, W \subseteq U$.

In propositional logic, the *derivation* of a proposition is done via the inference rule known as Generalized Modus Ponens:

$$\frac{P_1, P_2, \dots, P_n, \quad P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q}{Q}$$

The meaning of this is as follows: if $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q$ is a known rule and $\{P_1, P_2, \dots, P_n\} \subseteq A$ then, Q can be derived from this knowledge. Given a knowledge base $KB = (A, R)$ and an atomic formula $g \in U$, we say that g can be derived from KB , denoted by $KB \sqsubseteq g$, if there exists a finite sequence of atomic formulas F_1, F_2, \dots, F_k such that $F_k = g$ and for each $i \in \{1, 2, \dots, k\}$ one of the following claims holds:

- $F_i \in A$;
- F_i can be derived via Generalized Modus Ponens from R and the set of atoms $\{F_1, F_2, \dots, F_{i-1}\}$.

It is important to remark that for rules $V \rightarrow W$ we can require $|W| = 1$ without losing generality (Lloyd, 1987).

This definition of derivation provides two algorithms to answer the question of knowing if an atom g can be derived from a knowledge base KB . The first one is known as *forward chaining* and it is an example of data-driven reasoning, i.e., the starting point is the known data. The dual situation is the *backward chaining*, where the reasoning is query-driven (Bratko, 2001).

A deep study of both algorithms is out of the scope of this paper. We briefly recall their basic forms.

In this paper we present several different transformations of a tuple $\langle A, R, g \rangle$ into P systems and prove that forward chaining and backward

8 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/forward-backward-chaining-systems/62527

Related Content

Modeling Software Development Process Complexity

Vyron Damasiotis, Panos Fitsilis and James F. O'Kane (2021). *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming* (pp. 526-553).

www.irma-international.org/chapter/modeling-software-development-process-complexity/261041

Meta-Heuristic Paradigms and Swarm-Based Models for Large-Scale Optimization

D. Renuka Devi and T. A. Swetha Margaret (2025). *Harnessing AI for Control Engineering* (pp. 229-258).

www.irma-international.org/chapter/meta-heuristic-paradigms-and-swarm-based-models-for-large-scale-optimization/377543

A Framework of Statistical and Visualization Techniques for Missing Data Analysis in Software Cost Estimation

Lefteris Angelis, Nikolaos Mittas and Panagiota Chatzipetrou (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 433-460).

www.irma-international.org/chapter/a-framework-of-statistical-and-visualization-techniques-for-missing-data-analysis-in-software-cost-estimation/192887

Big Data Analytics in Smart Cities Traffic Light

Rushikumar Ghetiya, Mihir Sutariya, Yash Ghanshyambhai Dudharejiya, Damodharan Palaniappan, T. Premavathi, Rituraj Jain and Kumar J. Parmar (2025). *Leveraging Urban Computing for Sustainable Urban Development* (pp. 233-260).

www.irma-international.org/chapter/big-data-analytics-in-smart-cities-traffic-light/375376

Exploring Diseases Relationships: An Ontology-Based Spreading Activation Approach

Said Fathalla, Heba Mohamed and Yaman Kannot (2019). *Computational Methods and Algorithms for Medicine and Optimized Clinical Practice* (pp. 133-159).

www.irma-international.org/chapter/exploring-diseases-relationships/223787