Conformon P Systems and Topology of Information Flow

Pierluigi Frisco
School of Mathematical and Computer Sciences
Heriot-Watt University
EH14 4AS Edinburgh, UK
pier@macs.hw.ac.uk

Abstract. We survey some of the results about conformon P systems and link these results to the topology of information flow. This topology is studied with models of Petri nets. Several directions of research and open problems are given.

1 Introduction

The ten years young field of Membrane Computing saw, in between other things, the definition of a number of formal models of computation all sharing a well defined topological structure, locality of interaction and parallel processing \[30,10,28\]. These models allowed us to broaden our understanding of computation. Now, for instance, we know that the simple passage of symbols from one compartment to another in P systems with symport/antiport is sufficient to compute \[27\], that conformon P systems with either positive or negative values have similar computational power \[7\], that dissolution can play an important role in the computing power of P systems with active membranes \[17\], etc.

All these results told us a lot about how to perform computation. One important question that often went unanswered is why a certain model of P system could or could not compute a specific set of numbers. The answer to this why would:

link all the different models of P systems (and any formal system) based on multiset rewriting now regarded as different because of their definition;
generalise results to any other formal system based on multiset rewriting;
allow us to understand more fundamental features that have to be present in a formal system in order to compute;
classify formal systems of computation in different ways;
give new tools to prove the computational power and other properties of these formal systems.

Recently a way to answer this why, using the topology of information flow, has been suggested \[6,9,10,13\]. In this paper we survey the known results linking the topology of information flow to computational power and we show how these results can be applied to conformon P systems. No new result is introduced.
We refer readers to the original papers for the proofs of the used theorems. The description is kept rather informal even if formal definitions are provided. The given directions of research and open problems are meant to stimulate and inspire further developments in this line of research.

2 Preliminaries

We assume the reader to have familiarity with basic concepts of formal language theory [19], and in particular with the topic of membrane computing [10,29]. In this section we recall particular aspects relevant to our presentation. We denote with $\mathbb{N}_0$ the set of natural numbers $\{0, 1, 2, \ldots\}$ and $\mathbb{N} = \mathbb{N}_0 \setminus \{0\}$.

3 About Simulations

In order to study the topology of information flow of cP systems with P/T systems we have to relate configurations of the former with configuration of the latter. We do this through a definition of simulation (different than the ones in [24,25,33]). Here we generalise the definitions of simulation given in [10,6].

A multiset over a set $A$ is a total function $M : A \to \mathbb{N}$. For every $a \in A$, $M(a)$ denotes the multiplicity of $a$ in the multiset $M$. The support of a multiset $M$ is the set $\text{supp}(M) = \{a \in A \mid M(a) > 0\}$. The set of multisets over a set $A$ is denoted by $\mathbb{M}_A = \{M \mid M : A \to \mathbb{N}\}$. If $A$ and $B$ are sets, $\alpha \subseteq A \times B$ is a relation and if $(a, b) \in \alpha, a \in A, b \in B$, then we say that $b$ is returned by $\alpha$ (on $a$).

The formal systems we consider have configurations, that is, snapshots of relevant elements defining a formal system while it computes. Different formal systems have different relevant elements defining configurations. Even if we are not able to formally define configuration for a generic formal system, it is possible to formally define this concepts for specific formal systems. Formal systems can have an infinite set of configurations.

Given an initial configuration, a formal system can pass from one configuration to another. This process is called computation and the passage from one configuration to another is called transition. A transition occurs because some operations (rules, instructions, transitions, etc.) are applied to one configuration. If a computation is finite, then the last configuration is called final. We assume that final configurations have certain properties that make them such (that is, different from the other configurations). There is no transition from a final configuration. If a final configuration meets certain criteria, then it is said that the systems halts, otherwise it is said that the system stops.

Let $S$ be a formal systems with $O$ set of operations and $C = \{c_1, c_2, \ldots\}$ set of configurations.

We denote with $\Rightarrow$, $\sigma$ multiset over $O$, the transition from one configuration to another in a computation of $S$ according to the application of the operations $\sigma_1, \ldots, \sigma_n$. With $\Rightarrow^+$ we denote non-empty sequences of transitions from one configuration to another in a computation of $S$ according to the application of the