Verification of Non Deterministic Transition P Systems Solving SAT Problem

Mario J. Pérez–Jiménez, Fernando Sancho–Caparrini

Dpt. Computer Science and Artificial Intelligence. University of Seville. Spain

Abstract. Formal verification of programs is a hard task, even more if we use a procedural model as P systems are. In this paper we establish the formal verification of two non deterministic solutions to SAT problem of Propositional Logic, in the following sense: to every propositional formula, in conjuctive normal form, a transition P system, is associated in such a way that the formula is satisfiable if and only if there exists a successful computation of the associated P system. The analysis of the execution time of every computation, the number of possible computations and the cost to construct the P systems are presented.

1 Introduction

Models of natural computing use the inherent parallelism in the nature to obtain more efficient algorithms to solve problems computationally intractable for conventional models. The transition P systems [?] offers us a new natural computing model where parallelism and non-determinism plays an important role. It is a procedural model. The confirmation that the P system designed to solve a certain problem actually solves this problem is, even, less intuitive and harder. For this reason, to find tools of formal verification in order to guarantee the soundness and completeness of the designed P system, is more necessary.

As a first step to generate these tools, in [?] a complete mathematical formalization of transition P systems was presented (it can be easily adapted to other variants of P systems) allowing to treat the transitions, or steps, of the execution of the P system as branches of a computation tree, where evolution of all the necessary elements (membranes, rules, objects,...) are perfectly determined.

In this formalization the following concepts are essential: *configuration* (each one of the "states" the membrane structure can get, showing the present objects in them, during the execution of the P system) and *applicability matrix* (group of rules that can be applied to a certain configuration to obtain next configuration). As a result of one computation of a P system, we say that the P system *halts* if it reaches a configuration where no possible rules can be applied, and among them, we are interested in *successful* configurations, where the output membrane is elementary.

Verification of programs usually is a hard problem to abord. Even more if the model where we work is a procedural model, as P systems are. In this paper formal verification of non deterministic solutions to satisfiability problem of

Propositional Logic using transition P systems is given. In this kind of solutions an input data is accepted if and only if the system has, at least, an accepting computation. The solution we present here is addressed with completely different techniques in [?] and [?], where P systems working with multisets of strings is used: the strings are processed by replication, splitting, mutation and recombination in *P systems with worm–objects* ([?]), and by replicated rewritting rules in *Replicated Rewriting Simple P systems* ([?]).

In section 3 we present a transition P system solving the **SAT** problem of Propositional Logic in the following strong sense: to every propositional formula, φ, in conjuctive normal form, a transition P system, Π_φ (cooperative, with prioriy, with dissolution and without catalysts) is associated in such way that for each assignment verifying the formula, there exists a successful computation of Π_φ in whose output a partial assignment verifying the formula φ is encoded. This one works with multisets of objects and rewriting rules but doesn't make use of the massive parallelism. Nevertheless, the result is a P system working in linear time in all its computations (all of them halt), due to the inherent non-determinism.

In section 4, the formal verification of this P system, by means of invariant formulas, in certain sense, is given. We emphasize that the characterization of successful computations depends on the dissolution of all membranes of the P system. In section 5 the number of steps of any computation, the number of possible computations and the cost to construct the P system, are analyzed.

In section 6 we show a new transition P system that uses massive parallelism and non-determinism solving SAT problem in constant time. Formal verification of this solution is directly obtained from the execution of the P system. An analysis of the costs is presented.

References

1. Castellanos, J.; Păun, Gh; Rodríguez Patón, A. P systems with worm–objects, *Proceedings of IEEE Conference SPIRE'2000*, La Coruña, 64–74, 2000.
2. Krishna, S. N. *Languages of P Systems: Computability and Complexity*, Ph.D. Thesis, 2001.
3. Manca, V. String rewriting and metabolism: A logical perspective, in *computing with Bio-Molecules. Theory and Experiments*, Springer-Verlag, Singapore, 1998, 36–60.
4. Paŭn, Gh. Computing with membranes, *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for Computer Science – TUCS Report N 208*, 1998 (www.tucs.fi)
5. Păun Gh., Rozenberg, G. A guide to membrane computing, *Theoretical Computer Science*, 287, 2002, 73–100.
6. Pérez-Jiménez, M.J.; Sancho, F. A formalization of transition P systems, *Fundamenta Informaticae*, 49(1–3), 2002, 273–288.
7. Pérez-Jiménez, M.J.; Sancho, F. Verifying a P system generating square, *Romanian Journal of Information Science and Technology*, 5(2–3), 2002, to appear.