Molecular Computation Models in ACL2: a Simulation of Lipton's Experiment Solving SAT.*

F.J. Martín-Mateos, J.A. Alonso, M.J. Pérez-Jiménez and F. Sancho-Caparrini
http://www.cs.us.es/{~fmartin,~jalonso,~marper,~fsancho}
Dpto. de Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

April 5, 2002

Abstract

In this paper we present an ACL2 formalization of a molecular computing model: Adleman's restricted model [2]. This is a first step to formalize unconventional models of computation in ACL2. As an application of this model, an implementation of Lipton's experiment solving SAT [4] is described, based on the formalization given in [8]. We use ACL2 to make a formal proof of the completeness and soundness properties of the function implementing the experiment.

1 Introduction

At the beginning of the fifties the analogy between some mathematical procedures and biological processes starts to be established. L.M. Adleman [1] proved this relation in 1994, showing that it was possible to use biological processes to solve difficult mathematical problems: he designed a biological experiment based on DNA manipulation to solve instances of the Hamiltonian path problem, a well known NP-complete problem. This experiment can be considered as a first step to build a prototype of a molecular computer.

In 1995, R.J. Lipton [4] solved an instance of the satisfiability problem of propositional logic, using the method of Adleman. Lipton's experiment is interesting because the initial test tube does not depend on the propositional formula, but on the number of its variables. In this way, the experiment is a molecular solution to every instance of the satisfiability problem with a fixed set of variables and hence, it provides a molecular algorithm.

Adleman and Lipton's experiments are the starting point of molecular computation and reveals its huge advantage in parallelism with respect to the conventional electronic computers. In 1995 [2] the first formal models of molecular computing appeared, based on a set of basic biochemical operations. In 1996, D. Beaver [3] proved that these models are computationally complete, in the sense that every computation of a Turing machine can also be achieve by a molecular machine.

In this paper we present an ACL2 formalization of a molecular computing model: Adleman's restricted model. This is a first step to formalize unconventional models of computation in ACL2. The functions implementing the main operations in Adleman's model are disabled after proving its properties. So, the subsequent development is generic in that the specific operations are not important, but the necessary properties of these operations are important.

In [8] a formalization of Lipton's experiment is given as an iterative algorithm based on the elemental operations of Adleman's restricted model. We define recursive functions implementing

*This work has been supported by MCyT: Project TIC2000-1368-CO3-02

this formalization and we prove the completeness and soundness theorems of these functions. A transcription of these proofs is provided in this paper.

Finally we focus on two remarks of our implementation. First, a function providing the initial test tube for Lipton's experiment is provided and second, a final test is incorporated to the Lipton's experiment to decide the satisfiability of a propositional formula. We also show the completeness and soundness theorems in both cases and explain the ideas we have followed to prove them.

References

[1] Leonard M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 11, 1994.

[2] Leonard M. Adleman. On constructing a molecular computer. In R.J. Lipton and E.B. Baum, editors, *DNA Based Computers*, number 27 in DIMACS Series, pages 1–21. American Mathematical Society, 1996.

[3] Donald Beaver. A universal molecular computer. In R.J. Lipton and E.B. Baum, editors, *DNA Based Computers*, number 27 in DIMACS Series, pages 29–36. American Mathematical Society, 1996.

[4] Richard J. Lipton. DNA solution of hard computational problems. *Science*, 268:542–545, 28, 1995.

[5] Gheorghe Păun. Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000.

[6] M. J. Pérez Jiménez and F. Sancho. A formalization of transition P–systems. *Fundamenta Informaticae*, 49 (1-3):261–272, 2002.

[7] M. J. Pérez Jiménez and F. Sancho. Verifying a P–system generating squares. *Romanian Journal of Information, Science and Technology*, 5 (1-2): 181–191, 2002.

[8] M. J. Pérez Jiménez, F. Sancho, C. Graciani, and A. Romero. Soluciones moleculares del problema SAT (*in spanish*). In A. Nepomuceno and all, editors, *Lógica, Lenguaje e Información, JOLL'2000*, pages 243–252. Ed. Kronos, 2000.