Invited Talk: The Complexity of Word Equations

Claudio Gutiérrez

Computer Science Department
Universidad de Chile

A *word* over an alphabet A is a finite sequence of elements of A. The problem this talk surveys has an almost trivial statement: Given alphabets $C = \{c_1, \ldots, c_n\}$ (of constants) and $V = \{v_1, \ldots, v_m\}$ (of variables) and two words w_1, w_2 over the alphabet $C \cup V$, are there words $u_1, \ldots, u_m$ over C such that after replacing each v_i by u_i in w_1, w_2 we get two identical words? This kind of equations have received several names, like *word equations* (A. A. Markov), *string diophantine equations* (H. Wang), *equations in free semigroups* (in the mathematical community), *associative unification* (in the computer science community).

The problem gained attention in the early sixties in relation to the problem of solving Diophantine equations in the work of A. A. Markov. In the early seventies the problem attracted the interest of the people in Computer Science in relation to formal languages, automata, variable length codes, unification.

Several partial results were obtained before G. S. Makanin, in 1977, solved the problem in its complete generality given an algorithm which when inputing an arbitrary word equation answer yes or no depending if it has a solution or not. Makanin's procedure gained fame as one of the most complex decision procedures in the literature. After this breakthrough, the problem spread in several directions, the computational complexity of it playing a central role. Using Makanin's algorithm it was shown that the problem of satisfiability of word equations is in EXPSPACE. Recently, in the late 90's, Rytter and Plandowski proved that minimal solutions of word equations are highly compressible. Form here, by purely word combinatorial methods, in 1999 Plandowski devised an algorithm which gives a PSPACE bound for the general problem, which is the best known so far. Proving lower bounds has shown to be much simpler, and it is known that the problem is NP-hard. Today it is conjectured that the problem is NP-complete.

In this survey we are going to restrict ourselves to complexity issues. We will take a look at Lentin's Pig-pug method (published independently by Plotkin), Makanin's algorithm, Rytter and Plandowski's result about compression and Plandowski's algorithm. We will discuss the role played by the variables. There are two paradigmatic cases, both known: if we restrict to equations which have at most two distinct variables (any number of occurrences) the satisfiability problem can be solved in polynomial time. On the contrary, for equations in which every variable occurs at most twice, the problem was shown to be NP-hard. We will finish with some open problems.