Complexity of Semiautomatic Structures

Sanjay Jain[1]*, Bakhadyr Khoussainov[2]**, Frank Stephan[3]***,
Dan Teng[3] and Siyuan Zou[3]

[1] Department of Computer Science, National University of Singapore
13 Computing Drive, COM1, Singapore 117417, Republic of Singapore
sanjay@comp.nus.edu.sg

[2] Department of Computer Science, University of Auckland, New Zealand
Private Bag 92019, Auckland, New Zealand
bmk@cs.auckland.ac.nz

[3] Department of Mathematics, The National University of Singapore
10 Lower Kent Ridge Road, S17, Singapore 119076, Republic of Singapore
fstephan@comp.nus.edu.sg, tengdanqq930@hotmail.com, zousiyuan@hotmail.com

Semiautomatic structures generalise automatic structures in the sense that for some of the relations and functions in the structure one only requires the derived relations and structures to be automatic when all but one input are filled with constants. One can also permit that this applies to equality in the structure so that only the sets of representatives equal to a given element of the structure are regular while equality itself is not an automatic relation on the domain of representatives. Initial results were in a publication at CSR 2014:

> Sanjay Jain, Bakhadyr Khoussainov, Frank Stephan, Dan Teng, Siyuan Zou. Semiautomatic Structures. Computer Science - Theory and Applications - 9th International Computer Science Symposium in Russia, CSR 2014, Moscow, Russia, June 7-11, 2014. Proceedings. Lecture Notes in Computer Science 8476, Springer 2014, pages 204–217.

In this conference publication, it was shown that one can find semiautomatic representations for the field of rationals and also for finite algebraic field extensions of it. Furthermore, one can show that infinite algebraic extensions of finite fields have semiautomatic representations in which the addition and equality are both automatic. Further prominent examples of semiautomatic structures are term algebras, any relational structure over a countable domain with a countable signature and any permutation algebra with a countable domain. Furthermore, examples of structures which fail to be semiautomatic were provided.

An example of a semiautomatic structure is the ring $(\mathbb{Z}, +, <, =; \cdot)$ which can be represented by a regular set in a way that the operations before the semicolon are automatic, that is, $+$ is an automatic function as a function with two inputs and $<$ and $=$ are also automatic relations; however, the multiplication is only semiautomatic, that is, every multiplication with a fixed constant is automatic while the multiplication as a two-input function is not automatic.

The current talk will sumarise these results and then concentrates on connections to complexity theory and recursion theory which go beyond the material of CSR 2014. These connections include the following:

* S. Jain was supported in part by NUS grants C252-000-087-001, R146-000-181-112 and R252-000-420-112.
** B. Khoussainov is partially supported by Marsden Fund grant of the Royal Society of New Zealand. The paper was written while B. Khoussainov was on sabbatical leave to the National University of Singapore.
*** F. Stephan was supported in part by NUS grants R146-000-181-112 and R252-000-420-112.

- If $(A; \circ, =)$ is a semiautomatic group then the word problem of each finitely generated subgroup can be solved in quadratic time — the proofs in the fields of Cayley automatic groups and Thurston automatic groups carry over. Here the word problem consists of two expressions $a_1 \circ a_2 \circ \ldots \circ a_n$, $b_1 \circ b_2 \circ \ldots \circ b_m$ and asks whether they represent the same word, where $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m$ are all taken from the finite set of generators and its inverses. However, for each set B one can make a finitely generated semiautomatic monoid $(A; \circ, =)$ in which the word problem W satisfies $B \equiv_T^p W$; in the case that B is r.e., the corresponding word problem W is also an r.e. set.
- Let F be the set of all functions from $\mathbb{Q}^k$ to $\mathbb{Q}^k$ which can be computed by a program with finitely many steps of the following type: The steps are carried out in the numerical order of their labels (= line numbers) and some can be omitted due to branching instructions. The first k variables are the components of the tuple and further variables might be introduced by the program, in each step at most one variable. An assignment assigns to a variable the value of a linear combination of the tuple members and their downrounded integer values and a constant. So if the variables are x_1, x_2, x_3, a new variable x_4 could take the value $2 + 3 \cdot x_1 + 1/2 \cdot x_3 + Floor(x_2)$. The conditional or unconditional jump always has as target a position with a larger line number, so the program cannot jump backwards and cannot have loops. Permitted tests for the conditional jump are those which check whether a variable value is positive, whether a variable value is 0 and whether a variable value is an integer. The final output is the tuple consisting of the first k variables after having performed the last step of the program. The structure $(\mathbb{Q}^k, F; +, \cdot, =)$ is semiautomatic; that is, the domain is coded as a regular set and all functions in F are automatic; furthermore, the pointwise addition and multiplication of vectors and comparison of vectors are semiautomatic. Note that for these semiautomatic operations, one vector consists of constants and the other one is any member of $\mathbb{Q}^k$.
- If one augments F to G by allowing assignments of the type $x_i = 1/x_j$ (where $1/0$ gives the value 0), then one can prove using Matiyasevich's undecidable result and the coding of integer-valued polynomials that $(\mathbb{Q}^k, G; +, =)$ is not semiautomatic.