CISC-499 Projects 2017–18

Kai Salomaa
School of Computing, Queen's University
`ksalomaa [AT] cs [DOT] queensu [DOT] ca`

1. **Transforming regular expressions to finite-state machines**

   Given a regular expression of length n, what is the worst-case size of the minimimal deterministic finite automaton (DFA) for the language? An exponential upper bound is known but average regular expressions can be implemented more efficiently.

   The main goal of this project is to generate libraries of "random" regular expressions and determine their state complexity. The regular expression–to–DFA transformations, as well as, the minimization of the DFAs have been automated in various libraries such as

   - Fado `http://fado.dcc.fc.up.pt/`, or,
   - Vaucanson `http://vaucanson-project.org/?eng`.

   The software libraries provide a collection of operations to convert regular expressions finite-state machines or vice versa and for minimizing finite state machines.

   The second goal in the project can be to find different types of "bad" examples: regular expressions where the equivalent minimized DFA is large.

   The project requires an understanding of the basics of finite automata and regular expressions. The amount of programming required is not large, but you should expect to run a significant number of simulations and other experiments.

2. **Finite automata on trees: implementing basic operations**

   Tree automata are an extension of finite automata that operate on trees instead of strings. Many of the well known constructions and algorithms for finite automata extend in a natural way to tree automata that process inputs from the leaves to the root. General information on tree automata can be found at `http://tata.gforge.inria.fr/` (For the purposes of this project we need only basic definitions from section 1.1 of the TATA book project.)

   The goal of this project is to implement some basic operations for tree automata: (for example) the cross-product construction for union and intersection and the determinization of nondeterministic automata. One should design a suitable symbolic "Grail-like"representation for the tree automata. (*Grail* is a symbolic computation environment for ordinary finite state automata.)

3. **Algorithmic complexity of decision problems for regular languages**

   This a theoretical topic requiring an understanding of basic notions in computational complexity and familiarity with time/space bounded reducibility. The topic would be suitable for students who are taking CISC-462.

   It is known that all natural problems, like membership, emptiness, equivalence etc. are decidable for finite state automata. However, what is the complexity of these problems?

   The goal of this project is *survey* what are the known complexity results for the basic decision problems for deterministic and nondeterministic finite automata and for regular expressions. Often the questions are complete for PSPACE for NFAs, or log-space complete for DFAs. In particular, it will be useful to identify

   - examples of natural problems for finite automata where the precise complexity is unknown,
   - examples finite automaton problems that are not known to be solvable, or that are known to be unsolvable.

   The goal of the project is to present the findings in the report in a systematic way (the terminology in different articles appearing in literature may not always be consistent). The project involves a fairly *large amount of literature search* since the complexity results are not included in typical textbooks. I can provide some survey articles to be used as a starting point.

Other projects. I have available some additional topics – please come to see me in my office or send me email. If you have your own idea for a project related to my research, please come to talk with me about it.