Efficient algorithms for weak forms of non-determinism

MICHAŁ SKRZYPCZAK

Finite automata form a mathematical model of machines that perform certain tasks. A typical application is recognition of a pattern (e.g. a regular expression) in a given input text — an automaton reads successive letters of the input text and when it reaches the end, it indicates whether the text matches the pattern or not.

There are two standard models of automata: deterministic, where the successive action is uniquely determined by the configuration of the machine; and non-deterministic, where more than one possible action is available. A non-deterministic automaton accepts a given input if at least one among many available executions on that input leads to acceptance. Informally, a non-deterministic automaton *guesses* the actions to take with the aim of accepting the given input.

One of the crucial results of automata theory states, that every non-deterministic automaton can be determinised: another automaton can be constructed that is deterministic and equivalent to the original one. It means that it is possible to simulate the *guessing* by a deterministic choice. The price of that lies in the size of the automaton: there are non-deterministic automata with n configurations such that the smallest equivalent deterministic automaton one has 2^n configurations. In other words, determinisation may require exponential blow-up in the size of the automata. Such an explosion leads to high time complexity of various algorithms operating on automata.

The aim of the present grant is to study the growth of the number of states when performing standard constructions on automata that use non-determinism in a restricted way. The most fundamental class of such automata are unambiguous ones: a non-deterministic automaton is unambiguous if for every input it has at most one accepting execution. Such an automaton still *guesses* the successive actions, but due to the restriction, at most one such choice leads to acceptance.

It is known that many operations and problems are more efficient for unambiguous automata than for general non-deterministic ones. An instance of such a problem is universality: verify if a given automaton accepts all the inputs. In the case of general non-deterministic automata the problem requires exponential time to solve (more precisely it is PSPACE-complete). In the case of unambiguous automata it is possible to solve this problem in polynomial time. The main focus of this grant is on the problem of complementation: given an automaton $\mathcal{A}$ construct an automaton $\mathcal{B}$ such that for every input, the automaton $\mathcal{B}$ accepts it if and only if $\mathcal{A}$ does not accept it. The following conjecture says that it is possible to effectively complement unambiguous automata.

Conjecture 1 (Colcombet)**.** *For every unambiguous automaton $\mathcal{A}$ there exists an unambiguous automaton $\mathcal{B}$ of polynomial size on the number of configurations of $\mathcal{A}$ such that $\mathcal{B}$ is a complement of $\mathcal{A}$.*

The conjecture concerns one of the fundamental open problems in automata theory. Essentially all the questions concerning the cost of standard operations on automata are solved for a long time now. The fact that the question about complementing unambiguous automata is still open indicates that the problem is hard. At the same time, any progress in this direction would be valuable from the theoretical point of view. Additionally, an efficient algorithm for complementing unambiguous automata would have a direct practical impact.

Apart from studying Conjecture 1 and it's simplified versions, the aim of the project is to study related questions for other weak forms of non-determinism. One of them asks again about complementing but this time of automata over words with data. An example of such a word can be a sequence of requests to a server: each request is of one of finitely many types and additionally it bears an identifier of the user who sent it. The number of such identifiers is not bounded. Similarly as in the case of regular expressions, one can use automata to recognise certain patterns in such sequences. An example of such a pattern asks if three distinct users logged in and out without performing any additional operation. It is known that some non-deterministic automata over words with data cannot be complemented at all. Within the project I plan to study the possibility of complementing unambiguous automata over words with data.