Garching, 8. November 2021

Report on the dissertation
Efficient Automata Techniques and Their Applications
by Vojtěch Havlena

Topic

Automata Theory was born in the 1960s as the theoretical basis of compiler construction, where automata were used as abstract mathematical models of recognizers or parsers. Since the 1990s, Automata Theory has also become a fundamental tool for the development of Formal Methods and formal verification technology. In these areas, automata are used as a universal data structure to represent sets of objects encoded as strings or trees. The objects can be program states, program executions, or program inputs. Automata-based algorithms are used to implement the operations on the data structure required for each specific application.

The use of automata as data structures has led to the development of Automata Engineering, a new discipline for the design, analysis, implementation, optimization, profiling and experimental evaluation of algorithms manipulating (e.g. intersecting, determinizing, complementing . . .) automata.

The thesis of Vojtěch Havlena presents a (very) wide range of contributions to Automata Engineering. It studies three kinds of automata, NFAs, tree automata, and Büchi automata, and a number of applications.

Scientific Contributions

The thesis is divided into three parts, each of them divided into several chapters.

Part I studies approximate reduction of NFAs in the context of a very specific application: detection of suspicious patterns in network packets. Suspicious patterns are described by regular expressions, and pattern matching is done by translating the expression into an NFA, and simulating the NFA on the string. However, the application requires inspection speeds of at least 100 Gpbs, and standard algorithms do not scale. A very efficient hardware-based solution is to implement the NFA on a FPGA with one flip-flop per state, but the required speed can only be achieved for NFAs of size limited by the available hardware. This motivates the main problem studied by Havlena: given an NFA A and a bound b, find a NFA B with at most b states whose language is “closest” to the language of A. Havlena chooses the probabilistic distance between the automata as notion of closeness, defined as the probability of the symmetric difference of the languages w.r.t. the probability distribution defined by a given probabilistic automaton. He shows that the associated decision problem is PSPACE-complete, even in the restricted setting in which B can only be obtained from A by removing states or adding self-loops (and removing redundant states). So he studies heuristics that remove states or add loops in a greedy manner. He then conducts a thorough experimental evaluation, including an impressive real use case that implements a high speed intrusion detection system in a FPGA suitable for 100 Gpbs traffic and realistic regular expressions, and even partially for 400 Gpbs. In the final chapter, Havlena presents a more practical approach in which the probabilistic automaton modeling the distribution of traffic (which is difficult to obtain in reality) is replaced by a multiset of packets observed in typical traffic, and the goal of the reduction becomes minimizing the number of misclassified packets. The approach does not offer the guarantees of the probabilistic one, but the experimental evaluation, again very thorough, puts the interest of the approach beyond doubt.

Part II presents an automata-based algorithm for the satisfiability problem of WS2S, weak monadic second-order logic of two successors. WS2S has been applied to several aspects of program verification, and to the verification of parametric families of hardware circuits. It is well known that the logical operations of WS2S can be “mimicked” by automata operations on tree automata, which allows to construct for any formula a tree automaton recognizing the models of the formula. Havlena presents a lazy approach to this idea. A formula is transformed into an *automata term*, which can be seen as a recursive procedure whose instructions are automata operations, and which, when run, produces an automaton for the formula. Havlena designs an efficient execution procedure based on memoization of intermediate results, lazy evaluation, subsumption, product flattening, and nondeterministic union. All these techniques are known in automata engineering (memoization is essential for BDD libraries, lazy evaluation is well-known, and subsumption is the basis of antichain-based automata algorithms for language inclusion and universality). In the next chapter, he adds antiprenexing as a preprocessing step. Intuitively, antiprenexing pushes quantifiers down the syntax tree as much as possible. A competent experimental evaluation and detailed comparison with MONA shows that the approach is competitive.

The last chapter of this part has a different flavor. It presents an automata theoretic approach to the problem of solving word equations using Regular Model Checking (RMC). The idea is that RMC allows to give a symbolic implementation of implementation of Nielsen’s transformation, a technique for solving systems of equations which is known to be complete in the quadratic case. This is a very nice observation, leading to very good experimental results.

Part III studies efficient algorithm for complementing Büchi automata. It develops a competitive implementation based on Schewe’s procedure, introducing optimizations of two kinds. First, simulation procedures are applied to remove macrostates containing two states p, q of the original automaton such that p is simulated by q, but has bigger rank. Second, *super-tight* runs are introduced, and techniques are developed to remove macrostates that do not appear in any super-tight run. The experimental results are excellent, the new algorithm performs better than any other tool in about two thirds of the benchmarks.

Evaluation

Havlena's thesis presents a very impressive collection of contributions to automata engineering. The most distinctive features are the breadth of the contributions, the careful justification of the design choices, and the extensive experimental sections. The breadth is certainly remarkable. The thesis deals with automata on both finite and infinite words and tree automata, and with a very large variety of problems and application areas. It goes well beyond the standard scope of a thesis, which usually focuses on one class of automata and one application area. Havlena shows excellent knowledge of automata theory, complexity theory, logic, and programming, and combines his knowledge in all these areas very well; in particular, the design decisions are carefully justified with the help of hardness results, hardware, or software limitations. Finally, the methodology of the experimental results is excellent. Clear distinctions are made regarding the provenance, quantity, and quality of the benchmarks; reproducibility is paid the necessary attention; comparisons with other tools are constructive and fair. Also, the experimental validation takes place as close to real-life conditions as possible (depending of course on the nature of the application, which varies very much from one part to the other).

All the contributions of the thesis have been published at a very high level in seven conference papers and two journal papers. In particular, the thesis has resulted in publications in CADE, CONCUR, and TACAS, top conferences in their respective fields, and the Journal of Automated Reasoning, an excellent venue. This an excellent record achieved by only a small percentage of the thesis on formal verification.

The thesis is well written. In particular, all three parts start with introductory chapters giving excellent and very well researched expositions of related work.

For all of the above, I consider that the thesis meets all the requirements of the proceedings for obtaining a PhD.

Javier Esparza