Course: Modern Theoretical Computer Science
Student: Denis Matoušek <xmatou08@stud.fit.vutbr.cz>
Date: October 11, 2015

Partial Determinization of Finite Automata

This work focuses on the topic of optimization of finite automata (FA) for use in hardware-based processing unit used for wire-speed pattern matching in the data of packets on high-speed networks. Typical use case is a network intrusion detection system (NIDS), which classifies packets according to a set of regular expressions (RE). Focusing on FPGA chips, there are specific requirements including use of parallelism to achieve desired throughput of data processing and maintaining reasonable usage of resources of an FPGA chip (memory blocks, logic, and flip-flop registers). Usage of resources is influenced by the method of implementation and number of FA states and transitions used.

There is a major difference in implementation of NFA and DFA within an FPGA chip. An NFA is typically implemented in logic elements and flip-flop registers, allowing more states to be active at the same time. This approach is logic-intensive. An DFA is typically implemented using memory blocks and a hashing scheme for determining next state. This approach is memory-intensive. Since the input, i. e. the set of REs, can be converted to an NFA, one can think about using the NFA approach. However, there is an issue with the amount of logic while processing more input symbols at the same time, which is a need to achieve required throughput. Another approach would be conversion of the NFA into a DFA. However, there are cases that cause state explosion (e. g. indefinitely-repeating wildcards), which leads up to exponential growth of the number of states of resulting DFA. This is the reason for considering another approaches.

There are several approaches based on division of original NFA into multiple FAs. [1] introduced an approach with a central DFA cooperating with several NFAs called *hybrid-FA*. However, it suffers from several shortcomings including possibility of decrease in throughput. [2] and [3] introduced an approach with a central NFA cooperating with several DFAs called *NFA Split*. An algorithm to find an equivalence relation on the set of states of original NFA (division into an NFA and multiple DFAs) was established. However, based on experiments, it was found out that the first generated DFAs has much more states than the other ones, which is undesirable feature considering memory requirements for implementation of each DFA in an FPGA chip. This work deals with analysis of merging the other DFAs with the first generated DFA with the most states in *NFA Split* approach to optimize the overall FA in terms of implementation in an FPGA chip.

References

[1] Michela Becchi and Patrick Crowley. A Hybrid Finite Automaton for Practical Deep Packet Inspection. In *Proceedings of the International Conference on emerging Networking Experiments and Technologies (CoNEXT)*, New York, NY, December 2007. ACM.

[2] Jan Kořenek. Rychlé vyhledávání regulárních výrazů s využitím technologie FPGA, disertační práce, Brno, FIT VUT v Brně, 2010

[3] Jan Kořenek. Fast Regular Expression Matching Using FPGA. *Information Sciences and Technologies Bulletin of the ACM Slovakia.* Bratislava: Vydavateľstvo STU, 2010, vol. 2, no. 2, pp. 103-111. ISSN 1338-1237.