Repetitive Substructures for Efficient Representation of Automata

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Michal Šedý

supervised by doc. Mgr. Lukáš Holík, Ph.D.

Excel@FIT 2024

Motivation

In many automata, especially in those representing regular expressions, there exist repetitive substructures, that can no be eliminated by state of the art reduction tool RABIT/Reduce tool [1] as shown in Figure 1.

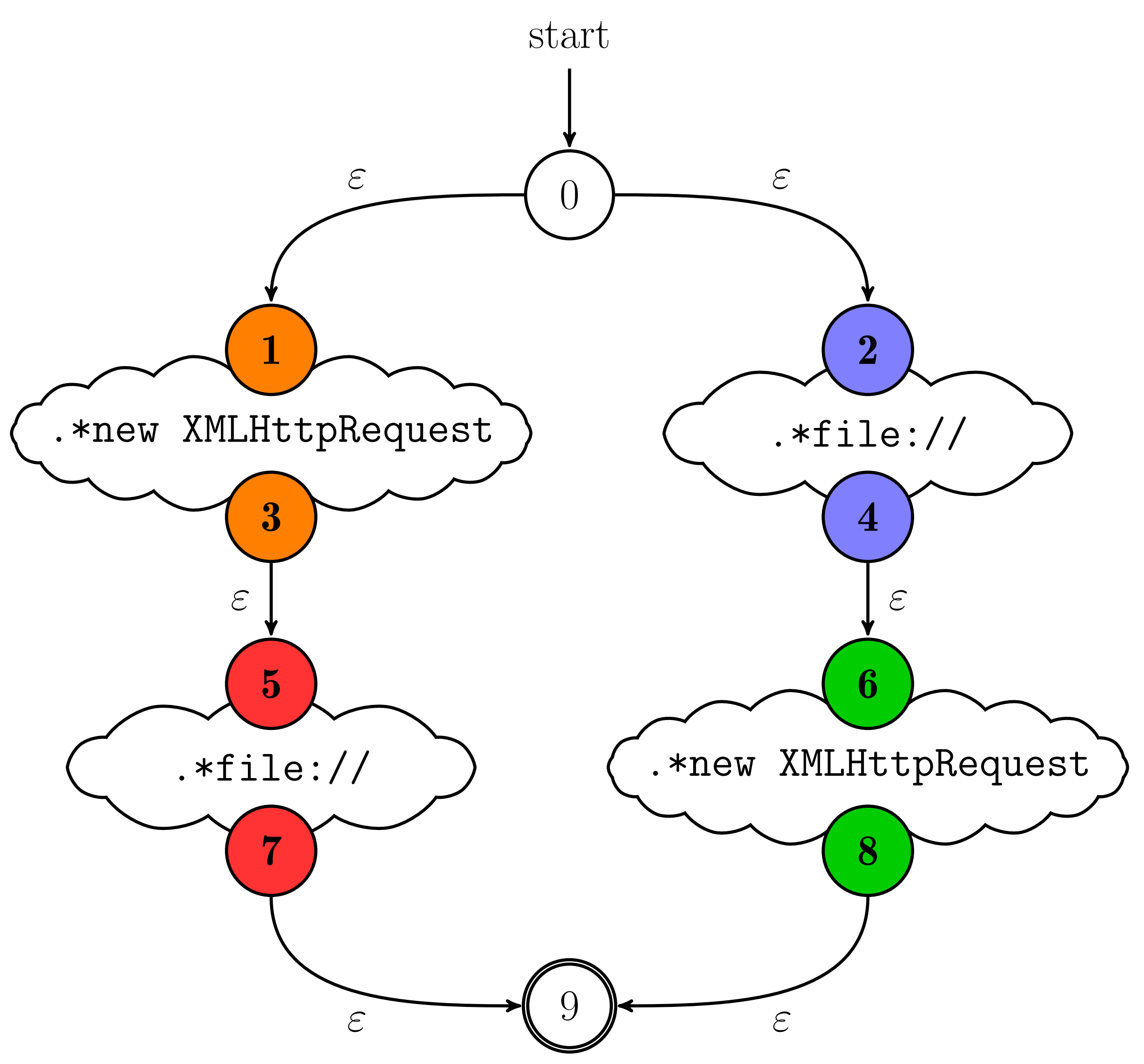

Fig. 1: Network filtering automaton.

We propose a new method based on pushdown automata and so called procedures, that represents the repetitive substructures only once.

Usage of a push-down automaton and procedures is an analogy to call stack and functions from programming languages.

Procedures

To represent automata efficiently without duplicate substructures, we introduce a new concept called procedures. Each redundant substructure is represented only once as a procedure. The automaton uses a stack to determine from which state the procedure is entered and to which state it should return. The symbol on the stack can be also used to guard transitions that are specific to only some substructures, that the procedure represents.

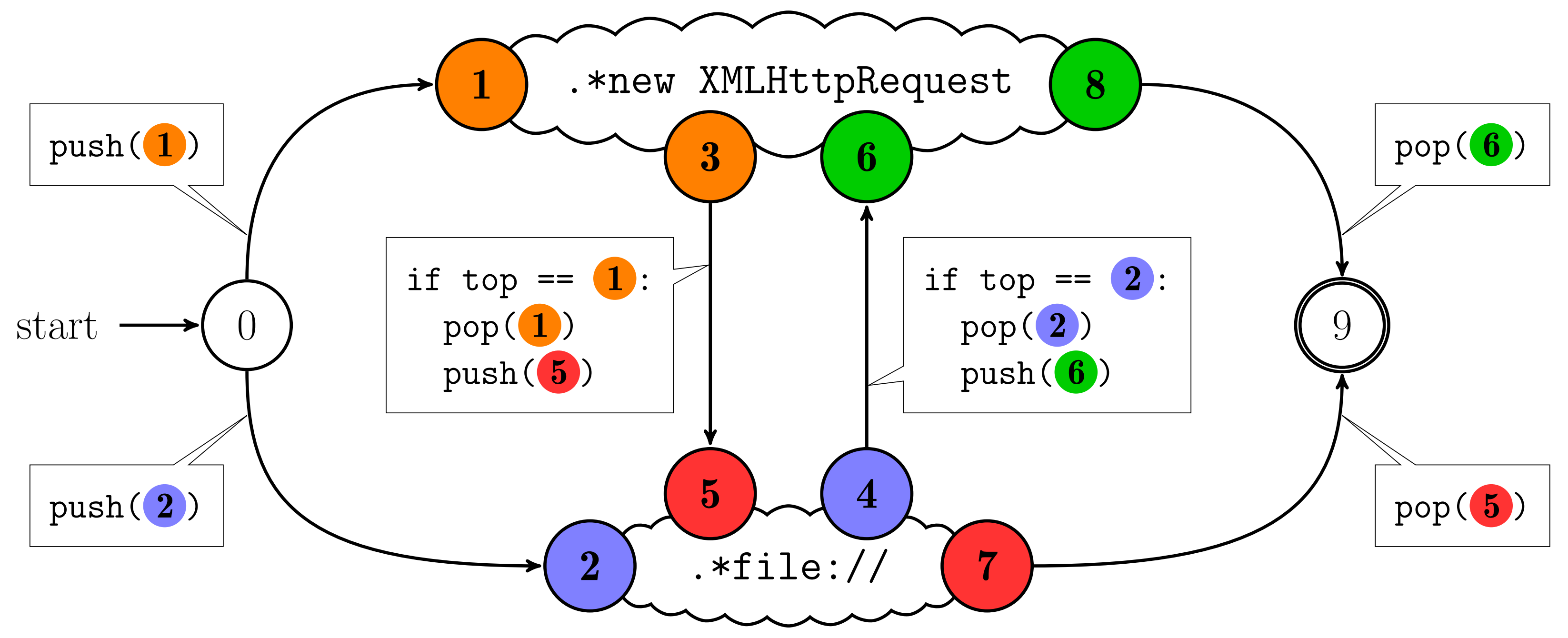

Fig. 2: Reduced network filtering automaton with two procedures.

Network Intrusion Detection System

To test the reduction capability of procedures in a real-world scenario, rules from Snort (well-known NIDS) were used. We generated seven automata, each representing a union of regular expressions from a single category of Snort rules.

Snort rules	Q_{in}	δ_{in}	Q_{RAB}	δ_{RAB}	$Q_{Proc} + \Gamma_{Proc}$	δ_{Proc}
p2p	33	1'090	32	1'084	25+6 (96.9%)	570 (52.6%)
worm	50	3'880	34	290	24+8 (94.1%)	284 (97.9%)
shellcode	162	3'328	56	579	48+2 (89.3%)	486 (83.9%)
mysql	235	30'052	91	14'430	45+18 (69.2%)	7'142 (49.5%)
chat	408	23'937	113	1'367	71+25 (76.7%)	1'058 (77.4%)
specific-threats	459	57'292	236	31'935	99+32 (55.5%)	12'680 (39.7%)
telnet	829	7'070	309	2'898	155+82 (50.0%)	2'164 (74.7%)

Tab. 1: Reduction results of RABIT/Reduce (RAB) and procedures (Proc) on seven sets of Snort rules. Q denotes the number of states δ the number of transitions, and Γ the number of stack symbols. The percentages refer to RABIT/Reduce results.

The two most significant reductions are highlighted in Table 1. The best reduction was attained on the automaton created from the `specific-threats` rule set. The RABIT/Reduce reduced the automaton by 48.6% of states and by 44.3% of transitions. By further application of procedures, another 43.5% reduction in states and 60.3% reduction in transitions was achieved. This experiment shows, that the procedures performs significant reduction even on real-world examples.

Parametric Regular Expressions

We evaluated the reduction potential of procedures on 3'656 automata, with 207 states and 2'584 transitions on average, generated from parametric regular expressions from [2].

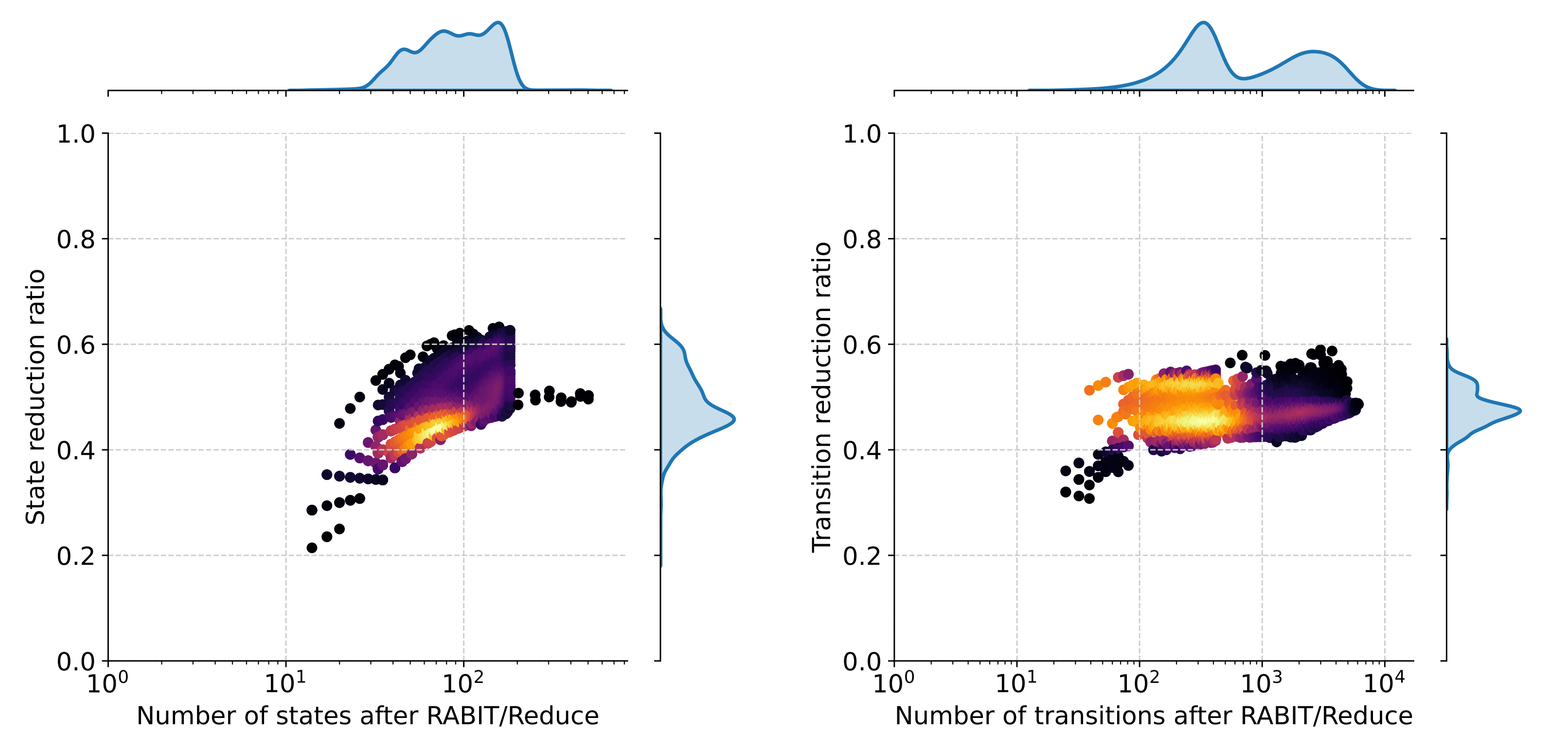

Fig. 3: Reduction ratios achived after utilizing procedures in results of the RABIT/Reduce tool. On average, the procedures improved reduction by 50.3% in states and by 47.9% in transitions.

The standalone usage of RABIT/Reduce resulted on average in 52.5% reduction of states and 48.4% reduction of transitions. The further reduction performed by our algorithm can be seen in Figure 3. The application of procedures reduced the automata to half of the size given by RABIT/Reduce.

References

[1] Abdulla, P. A. et al. *RABIT and Reduce.* `https://languageinclusion.org`.

[2] Gange, G. et al. Unbounded Model-Checking with Interpolation for Regular Language Constraints. Springer. 2013.