BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Comparing Languages and Reducing Automata Used in Network Traffic Filtering

Ing. Vojtěch Havlena, *supervisor*: Prof. Ing. Tomáš Vojnar, Ph.D.

Motivation

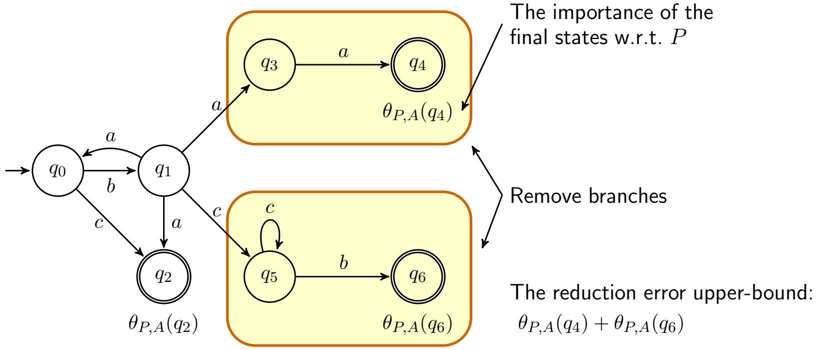
- Hardware filtering of malicious network traffic. Suspicious packets are described by regular expressions, which are converted to nondeterministic finite automata and then implemented into HW.
- **Problem:** The size of an NFA stored in HW.
- The classical reductions that preserve language need not be sufficient. Therefore we propose approach based on **approximate reduction** of NFAs.

Proposed Methods

- The **pruning reduction** (under-approximation) and the **self-loop reduction** (over-approximation).
- **Formal guarantees** with respect to **probabilistic distance**.
- The probabilistic distance utilizes probabilistic distribution of the input strings represented by a **probabilistic automaton** (PA) to express similarity of regular languages.
- The reduction can be parametrized by a **maximal error** with respect to the probabilistic distance between the language of the input NFA and the reduced NFA.

Pruning Reduction

- The pruning reduction selects **branches** of the input NFA that are later removed. The branches are chosen according to the input PA.



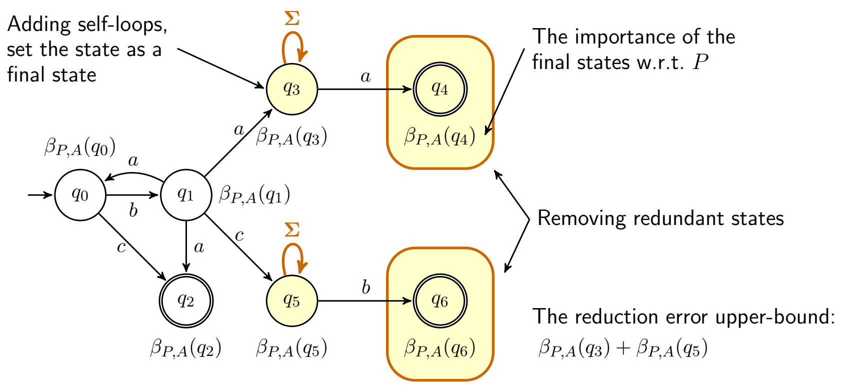
The importance of the final states w.r.t. P

Remove branches

The reduction error upper-bound: $\theta_{P,A}(q_4) + \theta_{P,A}(q_6)$

Self-loop Reduction

- **Adding self-loops** to certain states and making these states final, followed by removing all other transitions from these states and trimming the modified automaton.



Adding self-loops, set the state as a final state

The importance of the final states w.r.t. P

Removing redundant states

The reduction error upper-bound: $\beta_{P,A}(q_3) + \beta_{P,A}(q_5)$

Experiments

1. Learning of PA from a traffic sample.
2. Reductions of automata describing attacks/protocols with respect to the learned PA.
3. Evaluation of the real traffic error.

Automaton	**Number of states**	**Traffic error**	
	before/after reduction	(packets)	
`info.rules`	16/3 16/4	0.001 69 0.000 89	(10^6) (10^6)
`shellcode.rules`	95/29 95/48	0.000 016 0.000 014	(5×10^5) (5×10^5)
`chat.rules`	219/47 219/66	0.27 0.03	(10^5) (10^5)

- The considered automata can be reduced over **70 % of their size with the traffic error less than 3 %**.