BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024
http://excel.fit.vutbr.cz

Repetitive Substructures for Efficient Representation of Automata

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

Abstract

Nondeterministic finite automata are widely used across almost every field of computer science, such as for a representation of regular expressions, in network intruction detection systems for monitoring high-speed networks, in abstract regular model checking, program verification, or even in bioinformatics for searching sequences of nucleotides in DNA. To obtain smaller automata and thus reduce computational resources, the state-of-the-art minimization techniques, such as state merging and transition prunning, are used. However, these methods can still leave duplicate substructures in the resulting automata. This work presents a novelty approach to automata minimization based on the transformation of a NFA into a nondeterministic pushdown automaton. The transformation identifies and represents similar substructures only once by so called procedures. By this, we can further reduce automata by up to another 50%.

*xsedym02@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Nondeterministic finite automat (NFA), as its name suggests, can nondeterministically traverse from one state into multiple states based on the same input. This allows NFAs to represent the language more compactly than its deterministic counterpart that can be only at one state at the time. Despite the complicated minimization caused by nondeterminism, NFAs are applied in many fields of computer science, such as representing regular expressions, network intrusion detection systems for monitoring high-speed networks [**?**, **?**], in abstract regular model checking [**?**], in verifying programs that manipulate strings [**?**] or in decision procedures in the WS1S and WS2S logic [**?**, **?**].

Minimization techniques

To reduce computational resources when working with NFAs, it is crucial to reduce their size. For this purpose, the state merging [**?**, **?**, **?**] and transition pruning [**?**, **?**] techniques are being used. The state merging technique can merge two states if one of them fully covers the logic of the other. On the other hand the transition pruning removes a transition if there already exists a duplicit transition with the same logic. This minimization approaches are implemented in state-of-the-art tool RABIT/Reduce [**?**].

There are Repetitive Substructures

Despite the good minimization potential that standard minimization techniques carry, the resulting automata can still contain redundant substructures. This automata ofter represents a regular expressions. For example an automaton for network traffic scanning containing the union of regular expression from network intrusion detection systems (NIDS). There are also types of automata that cannot be minimized by this methods at all.

Our Novelty Approach

In our work we present a novelty reduction approach, that incorporates a transformation of NFA to a nondeterministic pushdown automaton (NPDA) that uses a stack. The method identifies automaton's substructures, with the similar logic, and represents them only once by so-called procedure. The stack is then used to denote from which state the procedure have been entered and where to return af the end. This transformation can be understood as the conversion of a purely sequential program to program that uses functions and call stack. By applying our approach on results of the standard minimization techniques we were able to achieve an additional reduction of automaton by up to 50% states and transitions.

2. Motivation

The automaton representing a regular expression `(.*new XMLHttpRequest.*file://)|(.*file://.*new XMLHttpRequest)` from network intrusion detection system Snort [**?**] is shown in Figure 1. Besides the epsilon transitions and `.*`, this is the most minimal form that can be achieved by standard minimization techniques. This is caused by the lack of language inclusions as `Request` and `file` substructures are compactly different. As a result the automaton contains two substructures where each has one redundant copy, making the NFA representation unefficient.

3. One Procedure No Duplicate

4. Experiments

4.1 Parametric Regular Expressions

4.2 Network Intrusion Detection Systems

5. Conclusion

References