Abstract machine simulator

Robin BÁBÍČEK, Matúš CORANIČ, Matúš ČELKO, Celestín ČERNÁK, Daniela MILOŇOVÁ, Katarína POLÁKOVÁ *

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
besttpteam@fiit.stuba.sk

Abstract. Students of information technology all over the world take classes on formal languages and automatons, usually without visual simulators involved. They often deal with issues like how to effectively explain function and operation of various abstract state machines, how to represent their non-deterministic behaviour or how to interconnect state diagram, transition table and input data. All these issues can be easily addressed and demonstrated by a visual simulator. The aim of this project is to design and implement a visual simulator of abstract state machines with emphasis on easy-to-understand simulation, representation of nondeterminism and possibility to design your own automaton.

1 Introduction

Students taking courses of theoretical computer science face the task to understand and be able to explain functionality of various abstract machines, such as finite state automaton, push-down automaton, Turing machine, RAM and abacus machine. This task can be mastered much easier with help of visual simulators. The aim of this project is to create a visual simulator of abstract state machines that can be used in courses of formal languages and automata at our faculty.

2 Related work

We reviewed 12 different simulators, either student projects or other freeware simulators. We evaluated all features of these simulators and listed the results in a comparative table. They offer easy simulation of predefined abstract machines, often

[*] Supervisor: Dr. Daniela Chuda, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava.

supporting specific features, which include support for different input and output methods, graphical representation, nondeterministic behaviour, etc.

However, they lack several aspects that would certainly prove to be very useful and thus are main objectives of our project:

- easy and user friendly usage
- possibility to design your own automaton
- possibility to apply operations (union, concatenation, etc.) on computational model
- improved visualization of nondeterministic behaviour
- possibility to test students by generating exemplary issues along with correct answers
- possibility to import and export the automata

3 Design and implementation

Our solution is designed to incorporate all of the aforementioned features. The goal is to create a flexible framework that would address different automata types in similar fashion and allow definition of custom automata on end-user level. It is conceived as a Java application to ensure maximum portability and maintainability.

The application logic relies on XML technology, utilizing the JAXB API. Automata and simulation settings are stored in XML format and are dynamically converted to Java classes at runtime.

The user interface was designed with ease of use and minimization of learning curve in mind. Visualization of state diagrams and computation trees is handled by the JUNG graph framework.

One of the key problems in abstract automata simulation is nondeterministic computation and its visualization. We have addressed the extensive state space with optimized breadth-first search. Unlike other simulators, we attempt to visualize the computation even when the computation tree contains a large number of nodes. If the number of displayed nodes exceeds the capabilities of graphical visualization, it is possible to switch to textual representation.

4 Conclusions

The simulator of abstract machines is designed for students of formal languages and automata at our faculty. Main contribution of our simulator is ease of use, possibility to design custom automata from scratch (either using graphical or text-based form) and extensive options in visualizing of nondeterministic behaviour.

Future work includes support for mathematical operations and generating exemplary tests for students.