

## Android Automata Simulator

Daniela Chuda, Jakub Trizna, Peter Kratky

**Abstract:** The paper presents automata simulator for Android platform. Software simulation tools supporting a teaching process are highly accepted by both teachers and students. The main purpose of this paper is to propose key features and requirements of well-designed Android automata simulator and to present our tool CMSimulator – integrated simulator of finite automaton, pushdown automaton and Turing machine for Android platform. The work with the simulator will enable our bachelor students to comprehend the principle of automata anytime, anywhere.

**Key words:** Simulation, visualization of theoretical computer science, finite automata, push down automata, Turing machine, educational software.

### INTRODUCTION

The theoretical computer science courses are traditionally taught without any software assistance [5]. However, students of informatics and software engineering like using educational software simulation tools [1, 3, 4]. We are aware of that fact, so we have included automata simulation tools for PC platform in our course. Nowadays, most students constantly carry mobile devices to use various applications, therefore an automata simulator for mobile platform could be useful, especially Android platform which is very popular. The work with the simulator will enable our bachelor students to comprehend the principle of automata anytime, anywhere.

The aim of this paper is to report our experience with development of automata simulator on android platform and our experience gained in teaching of the course “Fundamentals of Theoretical Computer Science” in Bachelor educational program in software engineering held at our faculty.

### RELATED WORK

The number of existing solutions on android platform is not very large and support for all type of automaton is not suffice, as well as the formal specification of automaton is missing. In Figure 1 there is a comparison of 3 screens of simulators – first is our solution.

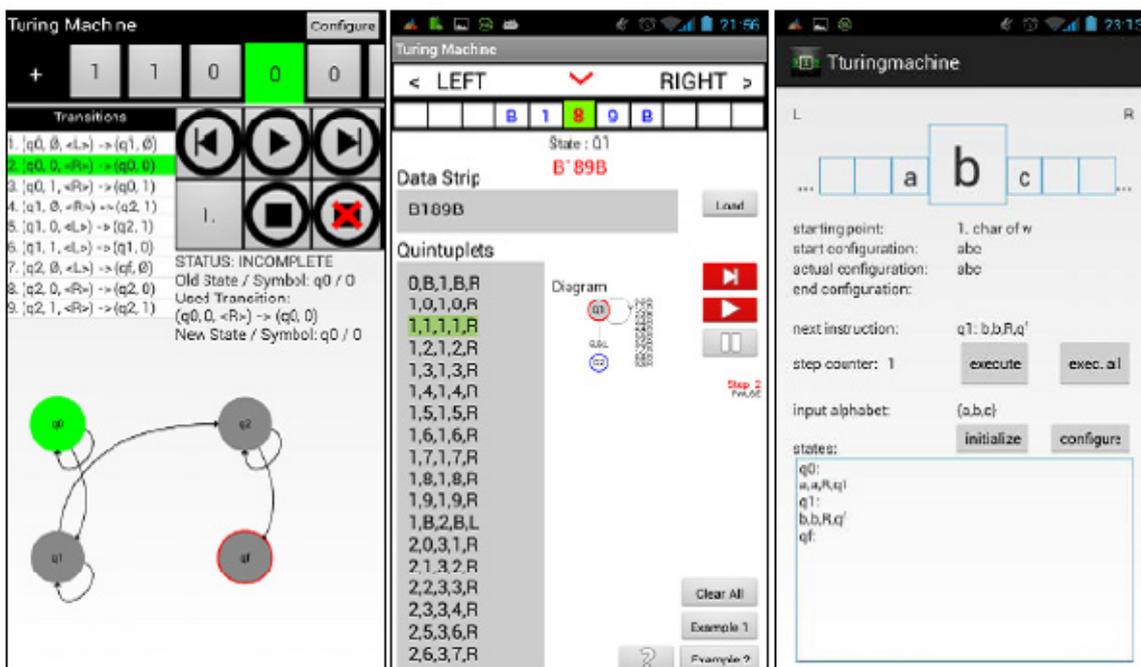


Figure 1. Comparison of simulation screen – CMSimulator (our solution), TuringMachine, Tturing machine

In the analysis phase we found some imperfections in the area of simulators [2]. Most of the problems occurred in entering the simulation programs and also the intuitiveness of application control was insufficient. On other hand, many features of existing simulators were suggestive. The phase of analysis brought positive ideas and thoughts, which helped to fulfil the appointed aim – to design and implement a simulator which would include a lot of features at preserved easy control. This goal cannot be fulfilled without proper testing with many people. In Table 1, there is a comparison of finite state automata simulators. We chose 21 characteristics to compare 3 finite state automata simulators and then we compounded the final rating.

Table 1. Comparison of finite state automata simulator

	<b>DFA Tester Simulator</b>	<b>FSM Simulator</b>	<b>SimStudio - Finite Automaton</b>
Platform	Android	JavaScript	.NET
Examples	-	X	X
Help – central	-	X	-
Local help buttons	-	-	-
Import and export file	-	-	X
Menu	-	-	X
Separate screen for configuration and simulation	X	-	-
Input tape	-	-	X
Editor input tape	X	X	X
Definition of input alphabet	X	X	X
Editor of transition function	X	X	X
Support of nondeterminism	-	X	X
Input forms	X	-	X
Creation of state diagram	-	X	X
Definition of start and end state	X	X	X
Bi-directional simulation step	-	X	-
Parameters available for visual simulation	-	X	X
Description of actual step in simulation	-	X	X
Summary of process simulation	X	-	X
Possibility of generate acceptable / nonacceptable input	-	X	-
Representation of non – determinism simulation	-	-	-
Final results	3.	2.	1.

The first simulator, DFA Tester<sup>14</sup>, is an open source Android application from student Shudmanul Chowdhury. It is very simple and user-friendly application for only deterministic finite state automata simulation with testing on input tape.

FSM simulator<sup>15</sup> is a javascript simulator – a single webpage with deterministic and non-deterministic simulation created by Ivan Zuzak and Vedrana Jankovic.

<sup>14</sup> <https://github.com/seedform/DFA-Tester>

<sup>15</sup> [http://ivanzuzak.info/noam/webapps/fsm\\_simulator/](http://ivanzuzak.info/noam/webapps/fsm_simulator/)

The last one, SimStudio<sup>16</sup>, is our successful simulator solution for PC platform [5].

We also compared two turing machine simulators for Android: Turing Machine Simulator<sup>17</sup> from developer Paola Romani and Tturing Machine Simulator<sup>18</sup> from developer Tobi Holy. Design of both applications is depicted in Figure 1. Both applications simulate only Turing Machine.

In course of analysis we did an inquiry about requirements analysis. We evaluated 10 answers from 47 students. Students answered questions about using simulators, standards, errors detection, user controls, effectiveness, design, help and documentation. Students prefer errors detection, minimalistic design and local help.

## OUR SOLUTION

We designed an automata simulator for Android platform. The most important part was to find a way to place all segments of simulation and automata onto the small screen of mobile devices. The application includes multiple screens with logically divided functionality:

- main menu screen,
- simulation screen,
- configuration of automata.

The interactive simulation screen provides user with an overview of the input tape, transition functions, as well as the state diagram and it includes control buttons. Important feature is that the simulation could be stepped not only forwards but also backwards. Moreover, our mobile application is the only one that supports non-deterministics solution, it allows user to view every thread separately. Visualization of the simulation screen is shown in Figure 2.

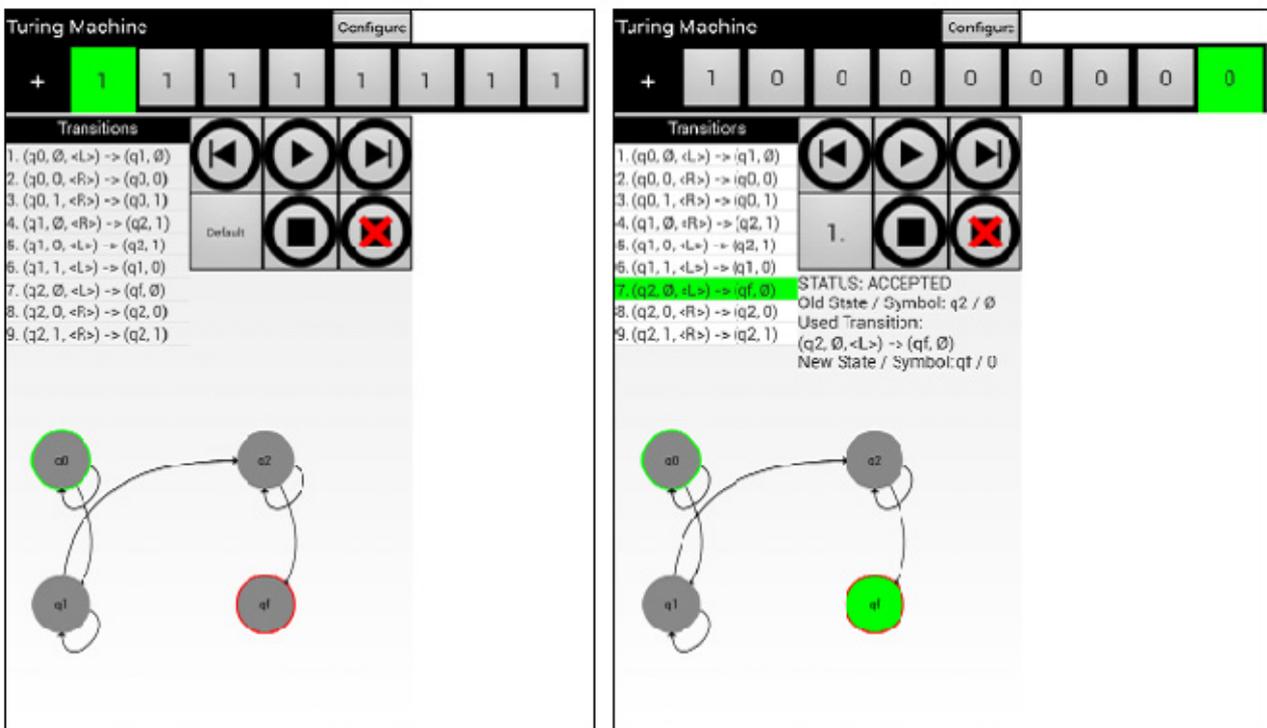


Figure 2. CMSimulator - visualization of the simulation (before and after simulation steps)

16 <http://www2.fkit.stuba.sk/~chuda/TZI/simulatory/SimStudio.rar>

17 <https://play.google.com/store/apps/details?id=dez.ch.turingmachine>

18 <https://play.google.com/store/apps/details?id=com.tturingmachine.tturingmachine>

In the screen for editing configuration of the automata user can manage transition function. It is necessary to define alphabet and states here, so that the user is protected against providing incorrect inputs. This is our solution to a very common issue of bugs in configuration.

In Figure 3 we can see the main menu of our application CMSimulator<sup>19</sup>. The application CMSimulator consists of four formal models simulator:

- finite automata simulator,
- push down automata simulator,
- linear bounded automata simulator,
- turing machine.

Our application allows to export the solution to an external file and to reopen the file. We support the format .jff – format of JFLAP simulator [9, 10] for PC platform. Design of CMSimulator application is simple and transparent, and at the same time it consists all important parts (see also Figure 1). The application provides slovak as well as english language pack. At this time Google Play does not contain such a learning application for simulation all formal models like our application.

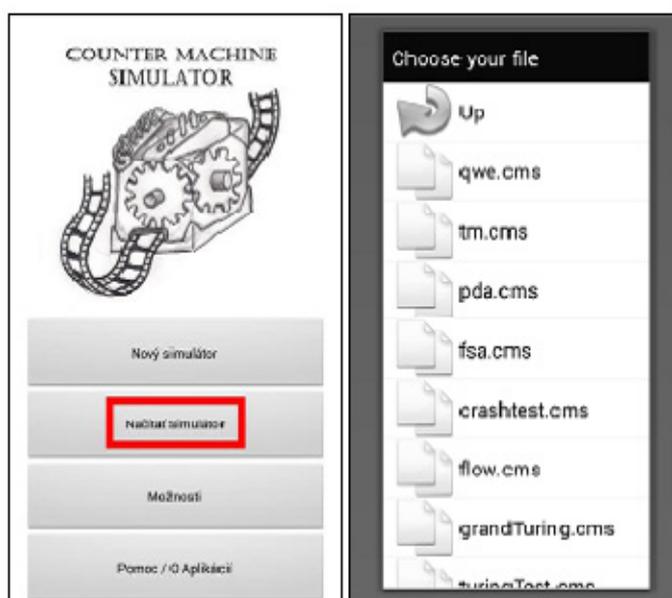


Figure 3. CMSimulator – Main Menu

## CONCLUSIONS AND FUTURE WORK

In the work we described development of Automata Simulator for Android platform which was also tested by teachers and students and it integrates improvements according to their comments and suggestions. Our simulator CMSimulator complements a set of tools for learning of theoretical informatics. It offers fast simulation and intuitive environment to students, they can use this simulator anytime, anywhere. New android application of automata simulator – CMSimulator brings more opportunities for experimenting and increases interest for learning.

Future work includes the preparation pack of learning examples and final testing and experimenting in course “Fundamentals of Theoretical Computer Science”.

<sup>19</sup> [http://www2.fiit.stuba.sk/~chuda/TZI/simulatory/CMS\\_install.rar](http://www2.fiit.stuba.sk/~chuda/TZI/simulatory/CMS_install.rar)

## REFERENCES

- [1] Čerňanský, M., Nehéz, M., Chudá, D., Polický, I. On Using of Turing Machine Simulators in Teaching of Theoretical Computer Science, *Aplimat - Journal of Applied Mathematics*, 1, No. 2 (2008), Bratislava: SUT, pp. 301-312.
- [2] Chakraborty, P., Saxena, P. C., Katti, C. P. Fifty years of automata simulation: a review. *ACM Inroads* 2, 4 (December 2011), p. 59-70, 2011.
- [3] Chuda, D. Visualization in education of theoretical computer science. In *Proceedings of the 2007 international conference on Computer systems and technologies (CompSysTech '07)*, Boris Rachev, Angel Smrikarov, and Dimo Dimov (Eds.). ACM, New York, NY, USA, , Article 84 , 6 pages. 2007.
- [4] Chuda, D., Nehéz, M., Čerňanský, M. On using of random access machine simulators in teaching of theoretical computer science. In *Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing (CompSysTech '09)*, Boris Rachev and Angel Smrikarov (Eds.). ACM, New York, NY, USA, Article 84 , 6 pages. 2009.
- [5] Chuda, D., Rodina, D. Automata simulator. In *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies (CompSysTech '10)*, Boris Rachev and Angel Smrikarov (Eds.). ACM, New York, NY, USA, 394-399. 2010.
- [6] Gruska, J. *Foundations of Computing*. Boston: International Thomson Computer Press, 1997.
- [7] Kaplan, C.S.: Halting problem. <http://www.cgl.uwaterloo.ca/~csk/halt/>
- [8] Nielsen, J. 1995. 10 Usability Heuristics for User Interface Design {Online; 2014/12/06} <http://www.nngroup.com/articles/ten-usability-heuristics/>
- [9] Rodger, S.H., T. W. FINLEY: *JFLAP: An Interactive Formal Languages and Automata Package*, ISBN 0763738344, Jones & Bartlett Publishers, Inc., 2006.
- [10] Rodger, S., Qin, H., Su, J. Increasing the use of JFLAP in Courses, *Sixth Program Visualization Workshop (PVW 2011)*, Darmstadt, Germany, p.53-56, 2011.
- [11] Vieira, L.F.M., M.A.M. Vieira, N.J. Vieira. Language emulator, a helpful toolkit in the learning process of computer theory, In *Proc. 35th SIGCSE Tech. Symposium on Comp. Sci. Education*, ACM Press, 2004, pp. 135-139.
- [12] Xavier, S.P.E.: *Theory of automata, formal languages and computation*. New Delhi: New Age International (P) Ltd, Publishers, 2005. 345 s. ISBN 978-81-224-2334-1

## ACKNOWLEDGEMENT

This work was partially supported by the Scientific Grant Agency of the Slovak Republic, grant No. VG1/0752/14.

## ABOUT THE AUTHOR

Assoc. Prof. Mgr. Daniela Chuda, PhD., Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technology, Slovak University of Technology, Slovakia, Phone: +421 2 210 22 318, E-mail: [daniela.chuda@stuba.sk](mailto:daniela.chuda@stuba.sk)

Bc. Jakub Trizna, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technology, Slovak University of Technology, Slovakia, E-mail: [jakub.trizna@gmail.com](mailto:jakub.trizna@gmail.com)

Ing. Peter Krátky, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technology, Slovak University of Technology in Bratislava, Slovakia, Phone: +421 902 680 177, E-mail: [peter.kratky@stuba.sk](mailto:peter.kratky@stuba.sk)

**The paper has been reviewed.**