

# An Android Application on Theory of Computation WE-TOC

Mr. Ishaan Als<sup>1</sup> Mr. Amit Katole<sup>2</sup> Ms. Parul Sukhdeve<sup>3</sup> Ms. Snehal Vaidya<sup>4</sup>  
<sup>1,2,3,4</sup>Student

<sup>1,2,3,4</sup>Department of Computer Technology

<sup>1,2,3,4</sup>Rajiv Gandhi College of Engineering & Research, R.T.M.N.U, India

*Abstract*— In today’s scenario android applications prove to be very useful since they are open source and provide numerous features to the users. The android application developed will prove to be useful and feasible especially for the students who desire to extract or obtain solutions of complex automata efficiently and immediately along with accuracy. Difficulties are often experienced by students during the introductory phase of learning of numerous concepts of formal languages and automata. To decimate this gap between the students and Theory of Computations, an interactive, feasible, and visual android application is being developed.

**Key words:** Android, Automata, Visual Representation

## I. INTRODUCTION

Android, a mobile OS is based on Linux kernel and is fundamentally designed for tablets and smart phones. Over many years android applications have been used vastly. These applications have abruptly changed the lives of many people. Android user interface is designed in such a way that user’s direct handling, manipulates the operating system with ease. Most of the android devices comprises of the free and open source applications and software, namely YouTube, Gmail, Drive, Calendar. A wide community has accepted Android because of such an open nature.

Android is an open source and Linux-based Operating System for mobile devices such as smart phones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers only need to develop for Android, and their applications should be able to run on different devices powered by Android.

## II. LITERATURE SURVEY

Java Formal Languages and Automata Packages is a tool for interacting with automata theory concepts. With this, one can create a finite automata, pushdown automata and a one or two tape Turing machine, drawn as a transition diagram. The user can enter input strings and watch the execution, even for the nondeterministic machines. In addition, one can explore the conversion of several different forms including NFA to DFA, DFA to minimum state DFA, NFA to regular expression, non PDA to context free grammar, context free grammar to non PDA [1].

This tool has been designed with general definitions and can be used with many automata theory through textbooks. Without JFLAP, students create these automata on paper and turn them usually with errors. It is too tedious to trace through to check for correctness. With JFLAP the automata come to life with visual representation [2].

Though the tool is only available only for desktops and PC’s, it is unusual for users to perform all the operations

on it. Android, being fluently available everywhere is agreeable by everyone [3].

This advantage allow students to experiment with concepts that would be difficult and tedious to do on paper. A single application that covers a large numbers of topics takes enormous development time, but can be easier on instructors and students in moving from one concept to another. User will have to restart every time they reach a dead end. [4]

## III. PROPOSED SYSTEM

WE-TOC is a user friendly and easy to use application. Any user can use the application in an effective way and will be very beneficial for him/her. The application has three transition as for now, first is for PDA, second for Turing Machine, and third for Moore to Mealy conversion.

In PDA: The basic rule i.e.

$L = \{a^n b^n \mid n > 0\}$  and  $L = \{a^n b^{2n} \mid n > 0\}$  are verified so as to get a practical visual presentation of the knowledge. String is checked, if the inputs are valid or invalid. The algorithm is as follows: For every input symbol “a”, add 1 to stack and for every input symbol “b” delete 1 from stack. Finally when the stack is empty i.e. Z0 it will be replaced with empty symbol.

In Turing machine: A machine that recognizes the presence of substring “101” and replaces it with “110”. At any given time when a user want to recognize the presence of 101 he/she will have to use this task. Language is variable so string may start with 0 or 1 and may end with 0 or 1. The algorithm being:

- 1)  $q_0$ , be initial state of machine and first find the combination of 101.
- 2) If  $q_0$  transit with #, change the state to  $q_1$ .
- 3) State  $q_0$  will not transit #, replace that # with 1 and move in right direction with state  $q_2$ . If  $q_2$  transit with 1, 1 it moves to the state  $q_2$  itself.
- 4) State  $q_2$  will not transit with 1, replace that 1 by 0 and move in right direction with state  $q_3$ .

In Moore to Mealy conversion: The table will take entries up to 3 present states. The next state with two inputs as 0 or 1 will determine where the state will move to, and output can be achieved from the question itself. The user will have to note down all the reading from the question and input them into the table. What happens next, is the machine converts the moore machine to mealy machine. The explanation of this conversion is also added. This shows how and why the conversion took place.

Basic steps involved in starting the application:

- 1) Install

The user needs to install this application on their android device or use a virtual android device to start the application. The virtual android device is available on android studio.

- 2) Open Application

Once the installation is done either on the android device or android virtual device we then start this application. After starting the application we need to select which topic to study.

3) Start with any of three topics

Once the topic is selected, the user should complete the situation given. The user can go back at any given time.

4) Output

Once we are done with the work. We need to exit the application. The flow of application is given in a diagrammatic manner below.

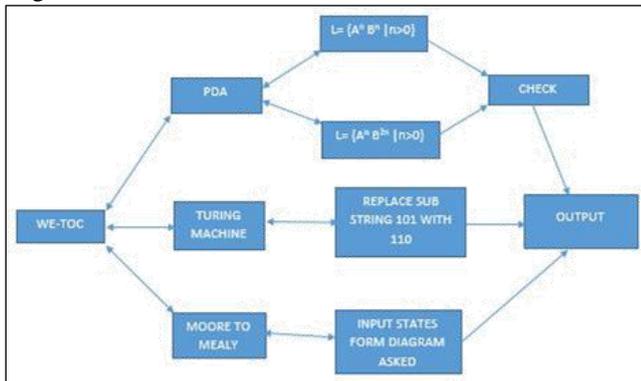


Fig. 3.1: Block Diagram of WE-TOC

#### IV. CONCLUSION

Thus conclude that this application is user friendly. Grasping knowledge is easy, as concepts within the application are pure basic rules. The idea of putting out a desktop application into an android application can be beneficiary. This android application is useful in learning and solving automata for everyone.

#### REFERENCES

- [1] O.G. Kakde, "Theory of computation", University science press, 2007.
- [2] Wei-Meng Lee, "Beginning Android™ Application Development", Wiley Publishing, Inc.
- [3] Android Tutorials, Guide for learning Android app development with source codes.