Final Write Up

Jacob Itz

Jacob.Itz1@marist.edu

May 17, 2019

1 Abstract

For my final project, I constructed two Deterministic Finite Automatons(DFAs) that validate a Tic Tac Toe game, one checks if X wins, the other checks if O wins. The game state is passed to the DFA in a string of digits(0, 1, or 2). If accepted the DFA returns True, otherwise it returns False. Both the DFAs and the Graphical game were constructed using Java.

2 Introduction

As mentioned above, the Tic Tac Toe game relies on the return value of the DFA driver. The state of the game is stored in a 9 digit string where each digit corresponds to a box on the Tic Tac Toe board. When the player clicks a box where they want their next symbol to be, a digit corresponding to their symbol(X == 1, O == 2, Empty == 0) is inserted into the game state string which is then processed by the DFA for the player who took their turn.

3 Detailed System Description

All parts of this project were coded in Java. The window that the user interacts with was built using Eclipse WindowBuilder.

3.1 User Interaction

The user can choose to play against another person on the same machine, or they can play against the computer. The user solely sees the GUI Tic Tac Toe board, and clicks the box they want to take when it is their turn to play. If player is playing against the computer, the computer will make a move immediately following the player's move. When the game is over(win, lose, or tie) the user is given a choice to either play again, quit, or do nothing. Every other aspect of the program is behind closed doors.

3.2 The DFA

The DFA for this project is very messy, so to prevent further confusion, I decided to break the problem up into two halves: one DFA to validate a win for X, and one to validate a win for O. In the game string there are 8 patterns that can lead to a win for either player. In the rows: 111000000, 000111000, 000000111. In the columns: 100100100, 010010010, 001001001. Or in the diagonals: 100010001 or 001010100. Matching each of these patterns alone is not too hard, but figuring out the correct transitions when the other digits are not zero was tough. It was even more difficult to account for ones in positions that do not lead to a win and that led to an even messier diagram.

4 Requirements

The requirements to run the project are simple. You only need a computer with Java 8 installed.

5 Literature Survey

There are a few similar DFA driven Tic Tac Toe games out there, but none quite like mine. Someone in class made a similar one, but mine has a GUI so you tell me who did it better. As for literature, I didn't find any. I did find a DFA diagram on reddit, but it was virtually unreadable with over 100 states, so I didn't even bother considering it. Knowing the 8 patterns to win was enough information to throw together a DFA. Apart from the DFA everything else is pretty basic java.

6 User Manual

The system is used primarily to pass time. Alternatively it can be used to work out the nuances of beating a child at a game on the back of a kids menu. The use is fairly self-explanatory, but I will give a brief summary.

1. Install Java
2. Add java and javac to your system Path
3. Download the project
4. Open a Command Prompt or bash shell
5. Navigate to the directory containing the project
6. Switch to the src directory
7. Type *javacGameBoard.java* and press Enter
8. Type *javaGameBoard* then press enter

Once the game has launched, you can use the drop-down menu on the bottom left to switch between 2 player mode and 1 player mode. That's it.

7 Conclusion

This project outlines one possible use for a DFA. It is not necessarily the best use of a DFA, but it is definitely one use. It would have been much easier to use regular expressions for this task, but that would have been way too easy. The significance of this project is hard to define exactly. It had some significance to me because I made it. I'd like to think it also has some significance to you since you got to enjoy it in all of its glory, but that seems unrealistic. Maybe if I had used one DFA to validate wins for both players the DFA would have been a little more impressive, but I realized after a week or two that that was just not in the cards. In conclusion, you *can* use a DFA to validate a Tic Tac Toe game, but you probably don't want to.