**Project Report**

**Deterministic Finite Automata  
(with TIC TAC TOE)**

**Submitted**

**K. Prasanth & A. Kiran Kumar**

**(110cs0141) (110cs0125)**

***Implemented in Java***

**IDE : *Net beans***

**CONTENTS:**

* **Overview**
* **About TIC TAC TOE**
* **DFA Description**
* **Transition Table**
* **Transition Diagram**
* **Methodology**
* **Conclusion**

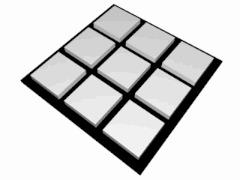
**PROJECT DESCRIPTION**

1. **Overview :**

This project is to implement **“TIC TAC TOE”** game using Deterministic finite automata (Created in GUI using Java).

1. **About TIC TAC TOE :**

* Tic-tac-toe, originally called zerosand crosses (here for convenience, it is implemented using 0’s and 1’s)
* Two player game who take turns marking the blocks in a 3×3 grid.
* The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins.

****

1. **DFA Description :**

**M = (*Q*, Σ, δ, *q0*, *q6*)**

Where**, Q** is the finite set of internal states

**( Q = { *q0, q1, q2, q3, q4, q5, q6*}**

**∑** is the finite set of input symbols (∑ = {0, 1} )

***q0***is the initial state

***q6*** is the final state

**δ : Q × ∑ -> Q** is the transition function

1. **Transition Table :**

The transition table is represented as a **2-Dimensional array**.

|  |  |
| --- | --- |
| **0** | **1** |

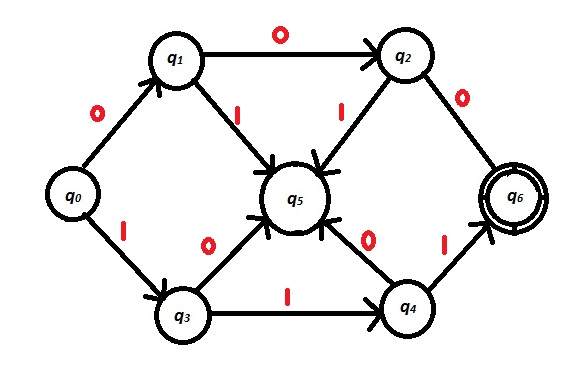
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***q0***   |  |  | | --- | --- | | ***q1*** | ***q3*** | | ***q2*** | ***q5*** | | ***q6*** | ***q5*** | | ***q5*** | ***q4*** | | ***q5*** | ***q6*** | | ***q5*** | ***q5*** | | ***q5*** | ***q5*** | |
| ***q1*** |
| ***q2*** |
| ***q3*** |
| ***q4*** |
| ***q5*** |
| ***q6*** |

Here, ***q0*** represents initial state

***q5*** represents dead state

***q6*** represents final state .

1. **Transition Diagram :**

****

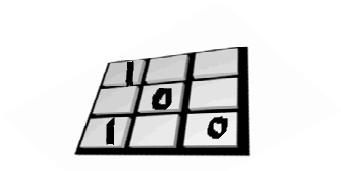
1. **Methodology :**

**Convention:** In the GUI, **q5** is taken as **qd** (dead state), **q6** is taken as **qf** (final state). **Player 1’s choice =1, Player 2’s choice =0.**

If any three consecutive blocks either in horizontal, vertical, or diagonal sequences are filled, then the 3 entries in a given sequence are passed to the automata as input string. If the automata accept the string **“111”** then Player 1 is the winner, if automata accept **“000”** then Player 2 is the winner. If all the blocks of tic-tac-toe are filled before any of them wins, then the game is tied. The respective transition when a particular block is clicked is shown in the transition diagram on the GUI.

(If the filling of any entry makes a triplet either in horizontal, vertical or any of the diagonals, then that particular triplet’s transition is shown in the transition diagram. Also only the latest transition will be shown after filling of each entry. If filling of an entry causes 2 or 3 triplets, then the transition of the highest priority is only shown.)

***NOTE:***



In the above case, 1 triplet is filled, when that triplet is passed to automata, the string can be taken as “100” or ”001” (i.e. it can be taken as reverse). So, in transition function, one of the transitions is shown in full red color, the other is shown as orange color (although in the orange transition, the start state and the dead state remains in red color only) 

In the above case, if the latest entry is “1” in the 3x1 block, two triplets are filled with it (111 and 100). For like these cases, the transition “111” is shown directly irrespective of other triplets because it is an acceptable string for the automata.

1. **Conclusion :**

The GUI Interface for TIC TAC TOE is successfully implemented using Deterministic Finite Automata.