The Winner Decision Model of Tic Tac Toe Game by using Multi-Tape Turing Machine

Sneha Garg
M.tech Scholar
Department of Computer Science and Engineering
Govt. Women Engineering College
Ajmer, India
snehagarg229@gmail.com

Dalpat Songara
Assistant Professor
Department of Computer Science and Engineering
Govt. Women Engineering College
Ajmer, India
dalpatsongara@gmail.com

Abstract— **The Tic tac toe is very popular game having a 3 X 3 grid board and 2 players. A Special Symbol (X or O) is assigned to each player to indicate the slot is covered by the respective player. The winner of the game is the player who first cover a horizontal, vertical and diagonal row of the board having only player's own symbols. This paper presents the design model of Tic tac toe Game using Multi-Tape Turing Machine in which both player choose input randomly and result of the game is declared. The computational Model of Tic tac toe is used to describe it in a formal manner.**

Keywords-Tic tac toe; Automata Theory; Turing Machine;

I. INTRODUCTION

The Tic tac toe game is the most common and popular game that is played on a board of 3 by 3 grid having 9 slots as shown in fig.1. A Special Symbol (X or O) is assigned to each player to indicate that this particular slot is covered by the respective player. Both players choose a place to mark on the board alternatively. The players choose empty slots to mark, already filled slots can't be chosen. It is a perfect information game which means that each player has knowledge of all the actions that occurred previously at the moment of making decision. This game is also known as "Noughts and Crosses" or "Three in a Row" game.

X	O	O
O	X	
X		X

Figure 1. Tic tac toe Game Board

The major purpose of the players of Tic tac toe game is to form a vertical, horizontal or diagonal row on the board by their own special symbol. The winner of the game is the player who cover a winning row of the board first, having only player's own symbols. The game result in a draw if all 9 slots are covered and winning row is not covered by any player. Players of Tic tac toe have a unique strategy of playing the game. However, the probability of winning this game is high for the player who has played this game previously as compared to the new player. If both the players play in an intelligent manner, then game results in a draw. A Player follows an approach or a trick for playing the game so that either he wins or a draw occurs but should never lose the game [1].

A. Tricks to solve Tic tac toe

The Tic tac toe is an Extensive form of game. Every participant of the game should know the rules for playing this game and must have information of the preceding events. The board of Tic tac toe game with its corresponding positions are represented in fig 2.

1	2	3
4	5	6
7	8	9

Figure 2. Tic tac toe Board with Positions

1) A Player P who starts the game, makes choice as below:
An intelligent player places his assigned symbol on one of the corners of the board and create a tough situation for the opponent player. Therefore, the opponent player O can either win or lose the game depending upon his choice of the remaining positions on the board. A player P always wins the game by forming a fork. A fork is a condition when there are winning cases on two rows in the next turn.

a) The following conditions occur when the Player P starts by choosing a corner –
   - If Player O also selects any of the remaining corners, then Player P can surely win the game by choosing any other corner position on the board.
   - If Player O selects any of the mid positions on the board, then player P can surely win the game by choosing center position on the board.
   - If Player O selects the center position on the board, then the game results in a draw if both the players play smartly.

b) Conditions when Player P starts by choosing center position–

- If Player O selects any of the mid positions on the board, then the Player P can surely win the game by choosing the remaining corner positions of opposite side of Player O's move and form fork.
- If Player O selects any corner then game result in a draw if both the players play smartly.

c) If the Player P starts the game by selecting any of the mid positions, then player P will lose the game if Player O plays the game optimally. Therefore, a good approach is to never select any mid-position at the beginning of the game.

2) *Player P makes choices specified as below when Player O start the game:*

a) If Player O selects any of the corner or mid positions, then the Player P should choose center of the board so that the game results in a draw.

b) If Player O selects the center, then the Player P should choose any of the corner positions on the board so that the game results in a draw.

B. Unique and Identical Game States

There are 3^9 possible ways to fill all the 9 squares of Tic tac toe game, but these all game states are not practically possible.

Practically Possible game states are:

- If a player wins the game then there is no need for the opponent to mark his symbol and should stop the game immediately.
- The player who start the game have one more or equal turn to the turn of second player.

Impractical game states shown in fig.3 in which X plays first.

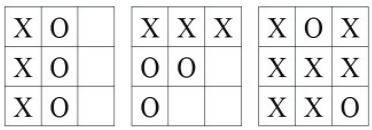

Figure 3. Impractical States of Tic tac toe

If the symmetry of this board is not considered, then total possible game states are 255,168. In fig.4 similar game states are shown that are achieved by revolving game board 90, 180 or 270 degree or mirroring across horizontal or vertical axes at any state. Total possible games are 26,830 if symmetry is considered. The useful and isolated states are 765 that are achieved by discarding similar and impractical game states [1].

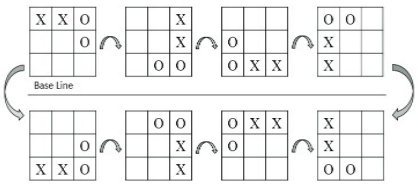

Figure 4. Identical States

C. Automata Theory and Formal Verification

The theory of computation shows learning of logic and mainly logic inside mathematics of a problem and how optimally it can be solved by a computation model using an algorithm. These studies work by analyzing problems and get the solutions of these problems. Automata is used to solve computational problems by using Automata Theory [2].

Automata Theory contain a mathematical model of computing that divides the computational machinery for the problem [3]. Practical programming problems are solved using real programming languages. They are easy to use due to their flexible nature but it is not easy for them to describe problem in a formal way. Therefore, to define a problem in formal way mathematical models are used. A mathematical model is also called a "FSM (finite-state machine)" or as a real computer, that means if a problem can be solved using these models, then it can also be solved on a real computer and vice versa.

Formal verification is a method for checking that a model satisfies some requirements [4]. So for the design verification, the model is converted into a simple verifiable format and it is known as a set of interacting system, which have finite states and transitions among states. Therefore, the design verification contains a complete model description. Configurations of FSM are written as a function of present state and Tape Symbols, called transition functions.

A Turing Machine (TM) is an abstract machine or automaton which has a tape for storage and this tape is stripped into cells that can store a symbol as shown in fig. 5. The TM also has a read-write (R/W) head that can move either in a left direction or a right direction on the tape by reading or writing a single symbol on each move to the tape. TM can be represented as Instantaneous Descriptions using move-relations, Transition tables and Transition diagrams [5].

A Turing Machine TM is defined as

$$TM = (Q, \Sigma, \Gamma, \delta, q_0, b, f)$$

Where Q is a non-empty finite set of States, Γ is a non-empty finite set of Tape Alphabets, δ is Transition Function, $q_0 \in Q$ Initial State, $b \in \Gamma$ Special Symbol called blank, $\Sigma \subseteq \Gamma$ Finite set of Input alphabets and $f \subseteq Q$ is non-empty finite set of Final States.

The transition function δ is defined as

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$$

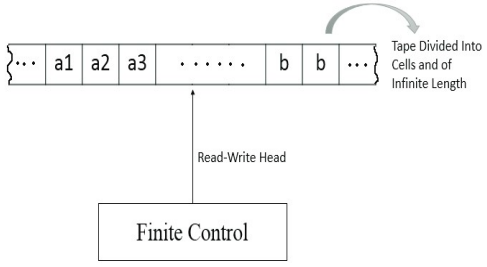

Figure 5. Turing Machine Modal

A Multi-Tape Turing Machine is a TM with several tapes. Each tape has its own independently controlled read-write head. A move depends on the current state and *n* tape symbols under *n* tape heads [3].

The Transition Function δ for TM with n-Tape is defined as

$$\delta : Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \{L, R, S\}^n$$

II. PREVIOUS WORK

There are various methods to play Tic tac toe. The previous techniques for playing this game and various games designed by an automata are following:-

Amarjeet Singh proposed a method to play the Tic tac toe using objective functions. These functions need to be maximized and best solution is chosen according to the rules described in the paper [1].

Noman Sohaib Qureshi designed the arcade game and a roller Coaster game using automata theory. Non-Deterministic Finite Automata is created for designing these games and then converted to Deterministic Finite Automata for development and coding of the games [6-7].

Thomas Abtey proposed a simulation based generation approach to build winning heuristic moves of the Tic tac toe game. This paper shows learning by experience and use that experience to make better moves [8].

Sivaraman Sriram proposed a decision tree algorithm that never loses a game and show that minimax algorithm breakdown to play optimally when other player play sub-optimally and proposed algorithm run pure when both the players play optimally [9].

P. Varshney and H. Mohammadi proposed strategies to play Tic tac toe using Genetic Programming [10-11]. P. Varshney found more than 72 thousand strategies to play Tic tac toe by which game always won or draw and have better win to draw ratio.

Sunil Karamchandani proposed Artificial intelligence based Tic tac toe that helps computer to choose the best move to either win or draw the game in the least amount of time [12]. Pinaki Chakraborty proposed Artificial intelligence based Tic tac toe by using domain specific priority based heuristic [13].

Shaila Parekh described Electronic Tic tac toe game having Logic Gates and Boolean Algebra. The best solution is decided using idea of optimization and probability from its database [14]. S. Jain implemented Tic tac toe game using Robot with NXT Lego Mind storms kit and a microcontroller is used to execute an algorithm for solving Tic tac toe for Robot [15].

Sai Ho Ling implemented Tic tac toe game by using Genetic Neural Network [16]. At each game state, score of each possible move is calculated and then best move is selected with highest score.N. F. Rajani implemented Tic tac toe game by using Hamming Distance Classifier in Neural Network [17]. M. V. D. Steeg proposed Temporal Difference learning of 3D Tic tac toe using Multilayer Perceptron that trained through self-play and a fixed opponent with which they tested [18].

III. PROPOSED WORK

To build the working decision model of the tic tac toe game, a Multi-Tape Turing Machine is designed. In this model any player (p or o) can start the game and choose a place to mark from input tape of TM alternatively. It contains 4 tapes to store the symbols and a subroutine to check the sequences for the result. The machine declares the result by traversing the input given by the both players, so it is a winner decision model of Tic tac toe game.

This paper proposes a theoretical abstract model for declaring result of Tic tac toe game on which any game sequence can be analyzed and by this theoretical model, any real-world simulator or program can be easily designed.

A. Mathematical Model

The Proposed Multi-Tape Turing Machine for The Tic tac toe Game (fig. 6) is defined as

$$M\_TM = (Q, \Sigma, \Gamma, \delta, S, b, f)$$

Where $Q = (S, q_p, q_o, q_1, q_{cp}, q_{co}, Q_d, W_p, W_o)$

$\Sigma = (p, o, 1, 2, 3, 4, 5, 6, 7, 8, 9)$

$\Gamma = (C, p, o, 1, 2, 3, 4, 5, 6, 7, 8, 9, b, X, O, \$)$

C is Left end Marker and $\$$ is Right End Marker

X is Player p 's Symbol and O is Player o 's Symbol

$f = (Q_d, W_p, W_o)$

The Transition Function δ for M_TM is defined as

$$\delta : Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \{L, R, S\}^n \; where \quad n = 4$$

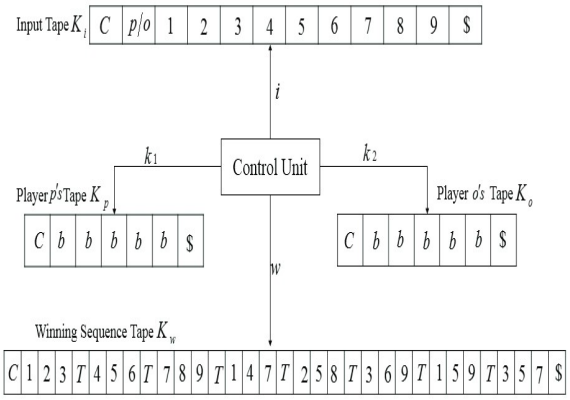

Figure 6. Block Diagram of Multi-tape Turing Machine for Tic tac toe Game

1) Key points and Procedure :

a) This Turing Machine has 4 Tapes (Input Tape K_i , Player p 's Tape K_p , Player o 's Tape K_o , and Winning Sequences Tape K_w).

- Input tape K_i is used to keep track of the remaining input that can be chosen by both players.

- Player's individual tapes K_p *and* K_o are used to remember their covering slots on the board to check winning sequence by comparing with the sequences in K_w.
- Winning sequence tape K_w contains all the sequences that are winning sequences for tic tac toe, used for checking player's input K_p *and* K_o with it.

b) Tape contents of K_i , K_p , K_o and K_w are referred as, k_1, k_2 and w, respectively with values $1 to 9$.

c) The Tape K_w contain 8 winning sequences for 3 by 3 Grid Tic tac toe $(123,456,789,147,258,369,159,357)$ that are separated by T.

d) Either Player p or Player o can start the game, by choosing the input p and o , respectively.

e) After choosing input, Subroutine CR_TM is called to check the result each time.

- Cell content k_1 of Player's tape K_p is picked up and replaced by X in K_w. Similarly k_2 of K_o is picked up and replaced by O in K_w.
- If no winning sequence is matched out of the specified 8 sequences that are separated by T in K_w and also all cell contents from $1 to 9$ are replaced by either X or O, then result will be a draw at final state Q_d.
- If winning sequence is found for any of the player, then result is declared by moving to the final state either $W_p or W_o$ according to the respective player.
- Otherwise, return to M_TM and next player choose place to mark from input tape and subroutine CR_TM is called and so on, until final state is reached.

2) Transition Diagram and Transition Functions for M_TM

Transition Functions have 4 tape symbols and have this type

$$\delta(CS,\ \Gamma's\ \ of K_i,K_p,K_o,K_w)\rightarrow \begin{pmatrix} NS,\ replaced\,\Gamma's\ \ of K_i,K_p,K_o,K_w,\ \ L,R,S \\ move\ \ of\ \ R/W\ \ head\ \ K_i,K_p,K_o,K_w \end{pmatrix}$$

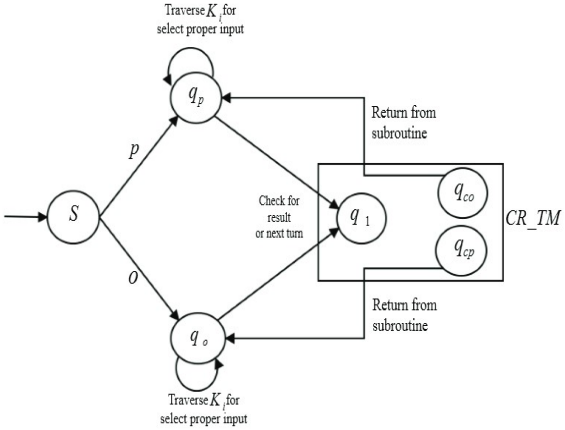

Figure. 7 Transition Diagram for M_TM

a) Rule I - Any player can start the game by choosing their respective symbol in fig. 6.

$$\delta_1(S,p,C,C,w)\rightarrow(q_p,C,C,C,w,R,R,R,R)$$
$$\delta_2(S,o,C,C,w)\rightarrow(q_o,C,C,C,w,R,R,R,R)$$

b) Rule II- Both player can choose input randomly, by moving left or right in K_i tape and replace it by b in K_i, and call subroutine CR_TM.

$$\delta_3(q_p,i,b,b,\{w,X,O\})\rightarrow(q_1,b,i,b,\{w,X,O\},R,S,S,S)$$
$$\delta_4(q_o,i,b,b,\{w,X,O\})\rightarrow(q_1,b,b,i,\{w,X,O\},R,S,S,S)$$
$$\delta_5(q_p,\$,b,b,\{w,X,O\})\rightarrow(q_p,\$,i,b,\{w,X,O\},L,S,S,S)$$
$$\delta_6(q_o,\$,b,b,\{w,X,O\})\rightarrow(q_o,\$,i,b,\{w,X,O\},L,S,S,S)$$
$$\delta_7(q_p,C,b,b,\{w,X,O\})\rightarrow(q_p,C,b,b,\{w,X,O\},R,S,S,S)$$
$$\delta_8(q_o,C,b,b,\{w,X,O\})\rightarrow(q_o,C,b,b,\{w,X,O\},R,S,S,S)$$
$$\delta_9(q_p,\{i,b\},b,b,\{w,X,O\})\rightarrow (q_p,\{i,b\},b,b,\{w,X,O\},\{L,R\},S,S,S)$$
$$\delta_{10}(q_o,\{i,b\},b,b,\{w,X,O\})\rightarrow (q_o,\{i,b\},b,b,\{w,X,O\},\{L,R\},S,S,S)$$

3) Transition Diagram and Transition Functions for CR_TM

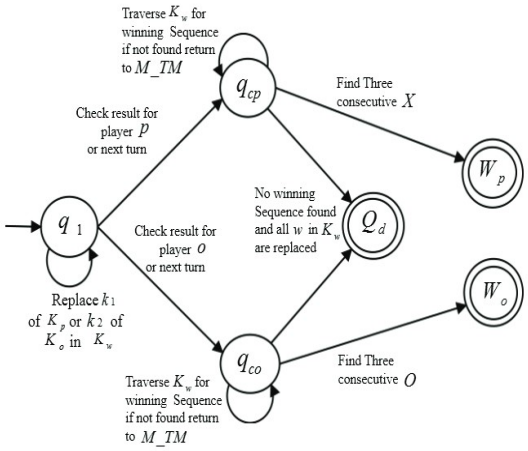

Figure. 8 Transition Diagram for CR_TM

The Subroutine CR_TM is called to check the result each time shown in fig. 8.

a) Rule III- Marking of Cell content k_1 or k_2 of Player's tape K_p, K_o in K_w by replacing them X or O, respectively.

$$\delta_{12}(q_1,\{i,b,\$\},k_1,b,k_1)\rightarrow(q_1,\{i,b,\$\},k_1,b,X,S,S,S,R)$$
$$\delta_{13}(q_1,\{i,b,\$\},b,k_2,k_2)\rightarrow(q_1,\{i,b,\$\},b,k_2,O,S,S,S,R)$$
$$\delta_{14}(q_1,\{i,b,\$\},k_1,b,\{w,X,O,T\})\rightarrow (q_1,\{i,b,\$\},k_1,b,\{w,X,O,T\},S,S,S,R)$$
$$\delta_{15}(q_1,\{i,b,\$\},b,k_2,\{w,X,O,T\})\rightarrow (q_1,\{i,b,\$\},b,k_2,\{w,X,O,T\},S,S,S,R)$$

Rule IV- After reaching right end of K_w check for result or next turn of other player.

$$\delta_{16}(q_1,\{i,b,\$\},k_1,b,\$)\rightarrow(q_{cp},\{i,b,\$\},C,b,\$,S,R,S,L)$$
$$\delta_{17}(q_1,\{i,b,\$\},b,k_2,\$)\rightarrow(q_{co},\{i,b,\$\},b,C,\$,S,S,R,L)$$

b) Declare winner of the game.

Rule V- Traverse K_w for player p to find 3 consecutive X and move to final state W_p that represent that player p won the game.

$$\delta_{18}(q_{cp},\{i,b,\$\},\{b,\$\},b,XXX)\rightarrow (W_p,\{i,b,\$\},\{b,\$\},b,XXX,S,S,S,S)$$
$$\delta_{19}(q_{cp},\{i,b,\$\},\{b,\$\},b,\{w,O,T\})\rightarrow (q_{cp},\{i,b,\$\},\{b,\$\},b,\{w,O,T\},S,S,S,L)$$

Rule VI- Traverse K_w for player o to find 3 consecutive O and move to final state W_o that represent that player o won the game.

$$\delta_{20}(q_{co},\{i,b,\$\},b,\{b,\$\},OOO)\rightarrow (W_o,\{i,b,\$\},b,\{b,\$\},OOO,S,S,S,S)$$
$$\delta_{21}(q_{co},\{i,b,\$\},b,\{b,\$\},\{w,X,T\})\rightarrow (q_{co},\{i,b,\$\},b,\{b,\$\},\{w,X,T\},S,S,S,L)$$

c) Rule VII- After reaching at left end of K_w if both K_p and K_o have b at their head means still input is remaining so no need to check K_w for draw and return to M_TM for next input from other player.

$$\delta_{22}(q_{cp},\{i,b,\$\},b,b,C)\rightarrow(q_o,\{i,b,\$\},b,b,C,S,S,S,R)$$
$$\delta_{23}(q_{co},\{i,b,\$\},b,b,C)\rightarrow(q_p,\{i,b,\$\},b,b,C,S,S,S,R)$$

Rule VIII- After reaching at left end of K_w if any tape K_p and K_o have $\$$ means all 9 inputs are marked and no winning sequence found so game results in a draw.

$$\delta_{24}(q_{cp},\{b,\$\},\$,b,C)\rightarrow(Q_d,\{b,\$\},\$,b,C,S,S,S,S)$$
$$\delta_{25}(q_{co},\{b,\$\},b,\$,C)\rightarrow(Q_d,\{b,\$\},b,\$,C,S,S,S,S)$$

Infinite Roaming- When a player moves back and forth in input tape K_i without making any selection, this condition is called infinite roaming.

TM always halt for any input sequence, ie. a machine always reaches the final state(either winner or draw as accepting state).If the proposed model is used for other grid sizes (like 4 by 4, 5 by 5 or any other) then input and winning sequences are also increased. Therefore tape cells are increased for storing them. There is a simple way to draw Tic tac toe because more choices will be available to choose and also each row and column are needed to be covered to prevent the opponent player from winning. This process might take long time to complete the game although if game will be drawn then it can be predicted at 10th step for 4 by 4 grid.

4) Time and Space Complexity

Time complexity is measured by counting the number of times the tape head moves when the machine is started on some input. Space complexity is measured by counting the total number of cells used that the tape heads ever reads, including cells read or written to that contain blanks [19].

a) Space Complexity - If Tic tac toe grid size is n by n then total number of cells required in this model $\theta(n^2)$. Space Complexity of this model is $S(n^2)$.

When n is Odd-

Input Tape K_i require n^2+3 cells

Player's tape K_p *and* K_o require n^2+5 cells

Winning Sequence tape K_w requires $(2n+2)n$ cells to store winning sequences and $(2n+3)$ cells to separate winning sequences.

So total number of cells are

$$(n^2+3)+(n^2+5)+(2n+2)n+(2n+3)$$

$$\Rightarrow (4n^2+4n+11)$$

When n is Even-

Input Tape K_i requires n^2+3 cells

Player's tape K_p *and* K_o requires n^2+4 cells

Winning Sequence tape K_w requires $(2n)n$ cells to store winning sequences (no diagonal sequences) and $(2n+3)$ cells to separate winning sequences.

So total number of cells are

$$(n^2+3)+(n^2+4)+(2n)n+(2n+3)$$

$$\Rightarrow (4n^2+2n+8)$$

b) Time Complexity - If Tic tac toe grid size is n *by* n then total number of times tape head moves in this model is $\theta(n^4)$. Time Complexity of this model is $T(n^4)$.

When n is Odd-

Tape head moves to choose input- (n^2+2)

Tape head moves to processing after choosing input $2[(2n+2)n+(2n+3)]$

The above tape head moves are for each input that is n^2

So total number of tape head moves are

$$n^2[(n^2+2)+2\{(2n+2)n+(2n+3)\}]$$

$$\Rightarrow 5n^4+8n^3+8n^2$$

When n is Even –

Tape head moves to choose input- (n^2+2)

Tape head moves to processing after choosing input $2[(2n)n+(2n+3)]$

The above tape head moves are for each input that is n^2

So total number of tape head moves are

$$n^2[(n^2+2)+2\{(2n)n+(2n+3)\}]$$

$$\Rightarrow 5n^4+4n^3+8n^2$$

IV. Conclusion and Future Scope

Automata Theory and Formal Languages are used to formally define a problem. This paper describes a Winner Decision Model of Tic tac toe that declares either winner of the game or a draw. Automaton of the game covers infinitely many possibilities of input strings (Sequence of moves to mark symbols). These strings are provided as an input to the automaton and the result is declared. Automata theory can also be used to solve the other games like ultimate Tic tac toe, minesweeper, reverse free. Many approaches like artificial Intelligence, Genetic Algorithm, Heuristic Learning or decision tree algorithm are used to generate optimal solution for Tic tac toe. Automata Theory and Formal are also applied to solve Tic tac toe. These methods can also be used to solve other perfect information board games. The Tic tac toe with 4 by 4 or 5 by 5 grid can also be solved using this model with some modifications. Usually these games results in a draw due to covering a box in every row and column by any player.

References

[1] Amarjeet Singh, Kusum Deep and Atulya Nagar "A 'Never-loose' Strategy to Play the Game of Tic-tac-toe" International Conference on Soft Computing & Machine Intelligence ISBN 978-1-4799-4231-2 Page no. 1 – 4, 2014.

[2] Peter Linz, University of California at Davis, "An Introduction to Formal Languages and Automata", ISBN 0-7637-1422, Jones and Bartlett Publishers, Canada.

[3] Anil Maheshwari, Michiel Smid, School of Computer Science Carleton University Ottawa Canada, "Introduction to Theory of Computation" , {anil,michiel}@scs.carleton.ca September 25, 2014.

[4] Vijay D'Silva "A Survey of Automated Techniques for Formal Software Verification", Ieee Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 27, No. 7, July 2008.

[5] Tribikram Pradhan, " Enhancement of Turing Machine to Universal Turing Machine to Halt for Recursive Enumerable Language and its JFLAP Simulation" International Journal of Hybrid Information Technology Vol.8, No.1 (2015), pp.193-202.

[6] Noman Sohaib Qureshi, Hassan Mushtaq, Muhammad Shehzad Aslam, Muhammad Ahsan, Mohsin Ali and Muhammad Aqib Atta, "Computing Game Design with Automata Theory", International Journal Of Multidisciplinary Sciences And Engineering, Vol. 3, No. 5, May 2012.

[7] Noman Sohaib Qureshi, Zahid Abbas, Muhammad Sohaib, Muhammad Arshad, Rizwan Ali Sabir, Asma Maqsood, "A Roller Coaster Game Design using Automata Theory", International Journal Of Multidisciplinary Sciences And Engineering, vol. 3, no. 5, may 2012

[8] Thomas Abtey and SUNY Oswego "A Tic tac toe Learning Machine Involving the Automatic Generation and Application of Heuristics".

[9] Sivaraman Sriram, Rajkumar Vijayarangan, Saaisree Raghuraman and Xiaobu Yuan "Implementing a No-Loss State in the Game of Tic-Tac-Toe using a Customized Decision Tree Algorithm" International Conference on Information and Automation, ISBN 978-1-4244-3608-8 Page no. 1211 – 1216, 2009.

[10] Bhatt, P. Varshney, and K. Deb, "In search of no-loss strategies for the game of tic-tac-toe using a customized genetic algorithm." In Proceedings of Genetic and Evolutionary Computation conference (GECCO-2008), (Atlanta, USA), pp. 889-896, 2008.

[11] H. Mohammadi, N.P.A. Browne, A.N. Venetsanopoulos, and M.V. Santos, "Evolving Tic-Tac-Toe Playing Algorithms Using Co-Evolution, Interactive Fitness and Genetic Programming." International Journal of Computer Theory and Engineering, Vol 5:5, pp. 797-801, 2013.

[12] Sunil Karamchandani , Parth Gandhi, Omkar Pawar and Shruti Pawaskar "A Simple Algorithm For Designing An Artificial Intelligence Based Tic tac toe Game" International Conference on Pervasive Computing ISBN 978-1-4799-6272-3 Page no. 1– 4, 2015

[13] Pinaki chakraborty, "Artificial Intelligence Based Strategies to Play the Tic-Tac-Toe Game" Journal of Technology and Engineering Sciences, Vol 1, No. 1 January –June 2009.

[14] Shaila Parekh, Shashank parbhu " Tic tac toe with A new Aspect" International Conference on Reliability InfoCom Technologies and Optimization ISBN 978-1-4799-6895-4 Page No. 1-4, 2014.

[15] S. Jain and N. Khera, "An intelligent method for solving TIC-TAC-TOE problem," Computing, Communication & Automation (ICCCA), 2015 International Conference on, Noida, 2015, pp. 181-184.

[16] Sai Ho Ling, Hak Keung Lam "Playing Tic-Tac-Toe Using Genetic Neural Network with Double Transfer Functions" Journal of Intelligent Learning Systems and Applications, 2011, 3, 37-44.

[17] N. F. Rajani, G. Dar, R. Biswas and C. K. Ramesha, "Solution to the Tic-Tac-Toe Problem Using Hamming Distance Approach in a Neural Network," 2011 Second International Conference on Intelligent Systems, Modelling and Simulation, Kuala Lumpur, 2011, pp. 3-6.

[18] M. V. D. Steeg, M. M. Drugan and M. Wiering, "Temporal Difference Learning for the Game Tic-Tac-Toe 3D: Applying Structure to Neural Networks," *Computational Intelligence, 2015 IEEE Symposium Series on*, Cape Town, 2015, pp. 564-570.

[19] John C. Martin, *North Dakota State University* "Introduction to Languages and The Theory of Computation",ISBN 978–0–07–319146–1.