The winning strategy of Tic Tac Toe Game model by using Theoretical Computer Science

Conference Paper · July 2017

DOI: 10.1109/COMPTELIX.2017.8003944

3 authors:

Sneha Garg
Malaviya National Institute of Technology Jaipur

Dalpat Songara
Govt. Women Engineering College Ajmer

Saurabh Maheshwari
Indian Institute of Technology Jodhpur

2017 International Conference on Computer, Communications and Electronics (Comptelix)
Manipal University Jaipur, Malaviya National Institute of Technology Jaipur & IRISWORLD, July 01-02, 2017

The Winning Strategy of Tic Tac Toe Game Model by using Theoretical Computer Science

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

Saurabh Maheshwari
Assistant Professor
Department of Computer Science and Engineering
Govt. Women Engineering College
Ajmer, India
dr.msaurabh@gmail.com

***Abstract*—Tic-Tac-Toe Game is a very popular game played by two participants on the grid of 3 by 3. A special symbol (X or O) is assigned to each participant to indicate that the slot is covered by the respective participant. The winner of the game is the participant who first cover a horizontal, vertical or diagonal row of the board having only their symbols. This paper proposed a winning strategy of Tic-Tac-Toe game and its computation is proved theoretically by the concepts of Theoretical Computer Science using multi-tape turing machine. This algorithm is designed for computer as a player in which computer act according to the intelligence of model to maximize the chances of success. The human player can makes its own choices. Any of the player can play first by their choice. The computation rules ensures selection of best slot for computer that will lead to win or prevent opponent to make a winning move. This is extended work of the paper *"The Winner Decision Model of Tic-Tac-Toe Game by using Multi-Tape Turing Machine".* The contribution of this work is to design a strategy to play Tic-Tac-Toe game in which computer will never lose.**

Keywords - Tic-Tac-Toe; Turing Machine; Algorithm; Winning Strategy;

I. INTRODUCTION

The Tic-Tac-Toe game is a thought-provoking game which is performed on a panel of 3 by 3 grid consisting 9 boxes as shown in fig.1. Each participant is assigned a special symbol i.e. either X or O to demonstrate that the particular box is filled by the particular participant. The empty boxes on the panel are chosen by the participants to mark their symbol on the game panel alternatively. It is a game of perfect information which illustrates that the previously occurred actions are known to each participant at the time of decision.

The main target of the participants in the game is to create a vertical, horizontal or a diagonal line on the game panel with assigned symbol. The participant who covers a row on the game panel with his assigned symbol first, will be declared as the winner. The game outcome is a draw when none of the boxes are empty and none of the sequences are formed i.e. vertical, horizontal or diagonal by any of the participant. The participants of this game have a special scheme to play it. However, the chances of win is more for the participant who has performed this earlier, in comparison with the new participant. If both of the participants perform smartly, then game outcome will be a draw [1].

X	O	O
O	X	
X		X

(a)

1	2	3
4	5	6
7	8	9

(b)

Figure 1. Tic-Tac-Toe Game Boards

An optimal strategy is required to play the game so that either the player win or game will be a draw and also play optimally with a sub-optimal opponent. The proposed strategy is implemented by using the concept of theoretical computer science. A mathematical model is designed using Multi-tape turing machine. There is a set of machines that read any possible input sequences by proposed model, process them as according to the logic design for machine and produce result accordingly. This algorithm is designed for computer as a player and human player can makes its own choices. Any of the player can become the first player. The computation rules helps computer in the selection of best slot that will lead to win or prevent opponent to make a winning move. The main objective of this research is to develop a strategy of Tic-Tac-Toe game so that player always win the game or game will be a draw.

The proposed strategy is designed by keeping in mind each and every case in which previous strategies are failed so that this strategy should not fail in any circumstances. The game is played many times by considering every configuration of the board which results in either winning of the machine or a draw. The algorithm also succeeds in playing with a sub-optimal player.

Theoretical Computation is used to express a problem in formal way by using mathematical models. These models divide the computational machinery for the problem and illustrate how ideally the problem can be resolved using an algorithm [2-3]. A Turing Machine (TM) is a machine having a tape to store input, which is divided into parts called as cells. The TM uses a read-write (R/W) head that can travel in any direction that can be left or right on the tape by reading or writing a symbol on each move [4-5].

II. RELATED WORK

There are numerous techniques to play the game of Tic-Tac-Toe. The previous approaches for playing it are following:-

Jain *et al*. [6] designed tic-tac-toe game by using NXT Lego Mindstorms kit and use a microcontroller to solve it and its limitation is that it works only for the player who plays second, no strategy is defined for the player who plays first. Karamchandani *et al.* [7] proposed an algorithm to solve tic-tac-toe by considering some of board configurations and its limitation is that it was fail with a suboptimal player. Singh *et al.* [8] designed some objective functions which are used to select best slot in the tic-tac-toe. Ling *et al.* [9] solve tic-tac-toe game by neural network with double transfer functions. Procedures in [7-9] failed in cases where opponent plays first and one of the diagonal filled in first 3 steps. Sriram *et al.* [10] proposed a decision tree algorithm for the player of tic-tac-toe which play second. This algorithm have no idea to play the game for the player who start the game and also fork creation for opponent. Chakraborty [11] suggested artificial intelligence centered strategy using primacy based heuristics to win the tic-tac-toe game and failed in creating fork for the opponent player. Mohammadi *et al*. [12] and Bhatt *et al.* [13] use genetic programming to provide no loss strategies of tic-tac-toe. Abety *et al*. [14] explain an approach to build winning heuristic moves of the tic-tac-toe game by using simulation. A 3D tic-tac-toe using multilayer perceptron presented by Steeg *et al.* [15] with temporal difference knowledge. This approach use 5 different structures for multilayer perceptron that skilled over self-play and a stable opponent with which they verified.

Garg and Songara [1] proposed a winner decision model of Tic-Tac-Toe game by using multi-tape TM. This model declare outcome in which any of the player can start the game and select any empty slot from the input tape of TM and both players play alternatively.

As explained above the previous effort to design strategies for tic-tac-toe game are failed in some situations like fork creation for opponent player if possible and play optimally with a sub-optimal player. So an optimal strategy is required to play the game so that either the player win or game will be a draw and also play optimally with a sub-optimal opponent. The present paper represents future work of [1] in which a machine will play against a human player according to proposed algorithm. It describes a theoretical abstract model to play a game of Tic-Tac-Toe using a winning strategy.

III. PROPOSED WORK

A Multi-Tape TM is designed to formulate a winning strategy model for Tic-Tac-Toe. The proposed model consist of 6 tapes which store the symbols and 6 subroutines to choose empty slot appropriately. Now 6 rules are defined under the model for empty slot selection. There are 2 players described as p and o as Machine player and Human opponent, respectively. According to the proposed algorithm the machine player p will always choose the slot by which p will be the winner or game will be a draw.

A. Mathematical Model

The Proposed winning strategies for the game is designed using a Multi-Tape Turing Machine and represented as

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

Where Q is set of states, Σ is set of input symbols, Γ is set of stack symbols, F is set of final states, S represents initial state and b represents blank

1) Strategy and Procedure :

a) It have 6 tapes (input tape K_i, player $p's$ tape K_p, player $o's$ tape K_o, winning sequences tape K_w, tape K_c to store empty slots and tape K_R to store occurrences of empty slots). The block diagram for the model is shown in fig 2.

b) The game can be initiated by any of the player, by choosing an input p or o. Now each time call a subroutine CR_TM to check the result whenever an empty slot is selected [1].

c) The proposed strategy for choosing an empty slot when machine is at q_p state is as follows:

Rules:

- **Rule R1-** Under this rule, machine will analyze the immediate winning condition for player p that is 2 X and 1 empty slot w $\{XXw, wXX, XwX\}$ and choose that empty slot, otherwise check for rule R2 as shown in fig 3.
- **Rule R2-** Here the machine will try to find immediate winning condition for player o that is 2 O and 1 empty slot w $\{OOw, wOO, OwO\}$ and choose that empty slot to prevent player o from win, otherwise go for special case if first player is o or check for rule R4 and R5 if first player is p as shown in fig 3.
- **Special Case –** It states that machine will check for sequence OXO on any of the diagonal $\{159, 357\}$ and remaining slots are empty then choose any middle slot of the board, otherwise rule R3 will be imposed as shown in fig 4.
- **Rule R3-** Here the machine will check for sequences having 2 empty slots and 1 player $o's$ symbol O $\{Oww, wwO, wOw\}$ and keep these empty slots to do action A, as shown in fig 5.
- **Rule R4-** This rule is implement to check for sequences consisting 2 empty slots and 1 player $p's$ symbol X $\{Xww, wwX, wXw\}$ and keep these empty slots then rule R5 will be tested as shown in fig 6.

- **Rule R5-** Under this rule, machine will try to find for sequences having 1 empty slot, 1 player $o's$ symbol O and 1 player $p's$ symbol X or 3 empty slots $\{XOw, OXw, wXO, wOX, XwO, OwX, www\}$ and keep these empty slots then proceed for action A as shown in fig 6.
- **Action A -** Find empty slot with maximum occurrence and if more than one slots found, then check for corner slot among them if available, otherwise any of them can be selected. This process is done by using subroutines CF_TM *and* MF_TM as shown in fig 7 and 8.

2) Proposed Algorithm for Tic-Tac-Toe Game (WST Algorithm)

The Proposed model have transaction diagrams as well as productions for each subroutine of turing machine. The game is played many times by considering each and every configuration of the board which results in either winning of the machine or a draw. The algorithm also succeeds in playing with a sub-optimal player.

In tape K_w, a cell not replaced with any symbol X *or* O but contains some value will be referred as an empty slot and represented by w.

Input and Output: There are 6 working tapes in the model of Tic-Tac-Toe Game which will act as input and output for the proposed algorithm. Here, subroutine M_TM contain 5 tapes $\{K_i, K_p, K_o, K_w, and\ K_c\}$ and CR_TM contain 4 tapes $\{K_i, K_p, K_o, and\ K_w\}$. Subroutines $R1\_R2\_TM$, SC_TM, $R3\_TM$ and $R4\_R5\_TM$ contain 2 Tapes $\{K_c\ and\ K_w\}$ each, whereas subroutines CF_TM and MF_TM contain 2 tapes $\{K_c\ and\ K_R\}$.

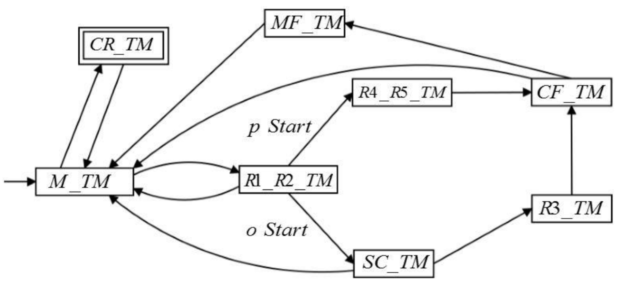

Figure 2. Block Diagram for proposed winning strategy of Tic-Tac-Toe Game

1. M_TM

 a. Initialize Tapes
  a) $K_i - 1\ to\ 9$
  b) K_w – All winning sequences separated by T
  c) $K_p\ and\ K_o - \ Blank\ b$

 b. ***If*** Player p plays as first player then
  a) Choose p from K_i to indicate player p is the first player and machine will choose slot 5 from K_i and write it into tape K_p

 c. ***Else*** game will be started by player o then
  a) Choose any of the slot according to his/her choice and write it into tape K_o

 d. ***Repeat Step 4*** till current state is not a final state
  a) ***Call*** CR_TM
  b) ***If*** player $p's$ turn
   - ***Call*** $R1\_R2\_TM$ and then player p can choose any slot from K_c & write it into tape K_o

  c) ***Else*** player o will choose any of the slot according to his/her choice and write it into tape K_o

2. CR_TM

 a. Read cell content of tape K_p and K_o
  a) ***If*** there is an input value at tape K_p and b is at tape K_o
   - search this input value in tape K_w
   - ***If*** the value is found then replace it with X

  b) ***Else If*** b is at tape K_p and an input value is at tape K_o
   - search this input value in tape K_w
   - ***If*** the value is found then replace it with O

 b. Traverse tape K_w from left to right
  a) ***If*** three X symbols found consecutively then player p will be the winner
  b) ***Else If*** three O symbols found consecutively then player o will be the winner // But this will never happen in this model as this is a winning strategy model for player p
  c) ***Else If*** all cells of tape K_w are replaced, hence there is no empty slot then game results as a draw
  d) ***Else Return*** to M_TM

3. $R1\_R2\_TM$

 a. Initialize tapes
  a) $K_c\ and\ K_R - \ Blank\ b$

 b. Traverse tape K_w from left to right
  a) ***If*** a sequence as like rule R1 found, write the empty slot value in tape K_c and ***return*** to M_TM
  b) ***Else If*** sequences as like rule R2, R3, R4, R5 and R6 found, ignore them and keep traversing K_w

 c. ***If*** R/W head reached at end of tape K_w traverse tape K_w from left to right again
  a) ***If*** a sequence as like rule R2 found, write the empty slot value in tape K_c and ***return*** to M_TM

b) ***Else If*** sequences as like rule R3, R4, R5 and R6 found, ignore them and keep traversing K_w

d. ***Else If*** R/W head reached at end of tape K_w and sequences as like R1 and R2 not found then

a) ***Call*** $R4\_R5\_TM$, if player p is a first player

b) ***Call*** SC_TM , if player o is a first player

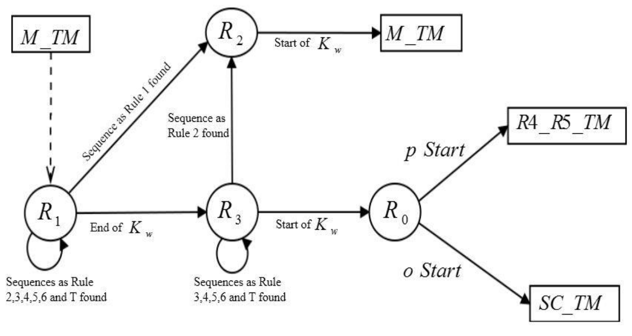

Figure 3. Transition Diagram of $R1\_R2\_TM$ for Tic-Tac-Toe Game

4. SC_TM

a. Traverse tape K_w from left to right to find sequence OXO on any diagonal of board that is 159 *and* 357 . This is possible only if first three sequences of K_w must be $\{either\ \ wwO\ , wXw\ , Oww\ \ \ or\ \ \ Oww\ , wXw\ , wwo\ \}$

a) ***If*** any of the above sequences found, write $2,4,6,8$ in tape K_c and ***return*** to M_TM

b) ***Else If*** any other sequences found in K_w , ***Call*** $R3\_TM$

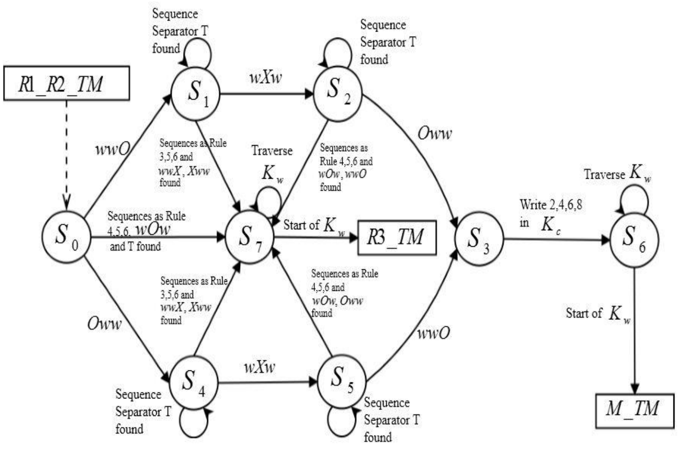

Figure 4. Transition Diagram of SC_TM for Tic-Tac-Toe Game

5. $R3\_TM$

a. Traverse tape K_w from left to right till the end of K_w

a. ***If*** sequence as like rule R3 is found, write all the empty slots in tape K_c

b. ***Else If*** sequences as like rule R4, R5 and R6 found ignore them and keep traversing K_w

b. At the end of tape K_w , ***Call*** CF_TM

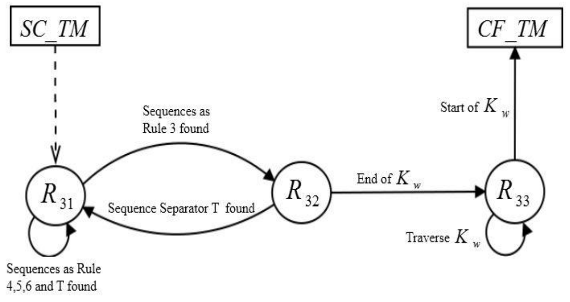

Figure 5. Transition Diagram of $R3\_TM$ for Tic-Tac-Toe Game

6. $R4\_R5\_TM$

a. Traverse tape K_w from left to right till end of K_w

a) ***If*** sequences as like rule R4 and R5 found, write all the empty slots in tape K_c

b) ***Else If*** sequences as like rule R3 and R6 found, ignore them and keep traversing K_w

b. At the end of tape K_w , ***Call*** CF_TM

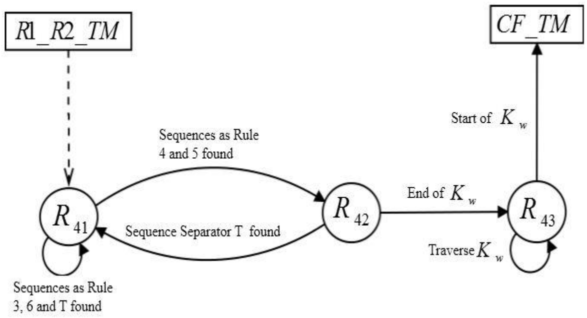

Figure 6. Transition Diagram of $R4\_R5\_TM$ for Tic-Tac-Toe Game

7. CF_TM

a. Traverse tape K_c from left to right till all cells of tape K_c are not counted

a) Write cell content of K_c into tape K_R and check for the next occurrence of the cell content in tape K_c and count it

b) ***If*** cell content is 5

- Initialize tape K_c *and* K_R with blanks and write 5 in tape K_c and ***Return*** to M_TM

c) At the end of tape K_c , write counter to tape K_R and then symbol T to separate the next pair.

d) Move to step (1.a) to count occurrence of next slot of tape K_c

b. After counting occurrences for all cells of tape K_c

a) Initialize tape K_c with blanks and ***Call*** MF_TM

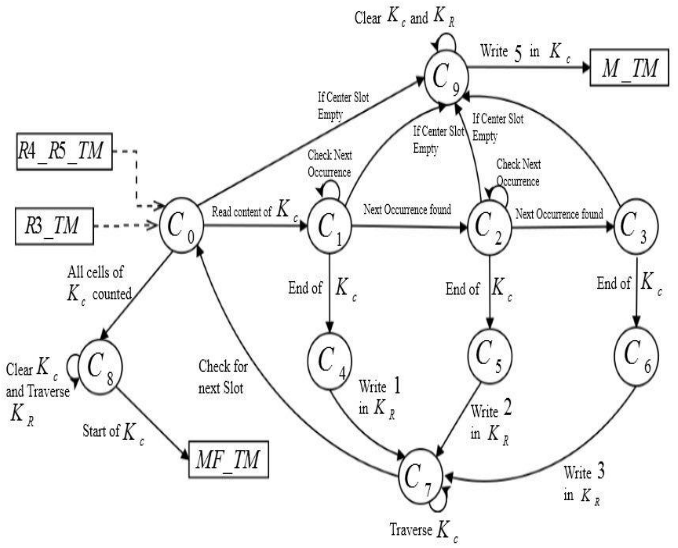

Figure 7. Transition Diagram of CF_TM for Tic-Tac-Toe Game

8. MF_TM

a. Repeat till the end of tape K_R
   a) Read cell content from K_R, write it in K_c and read its counter from K_R
      i. ***If*** counter is 1
         - ***If*** previous cell has counter 2 or 3, erase the cell from K_c and check next cell
         - ***Else*** do nothing and check next cell
      ii. ***Else If*** counter is 2
         - ***If*** previous cell has counter 1, erase all the previous cells from K_c and write this one
         - ***Else If*** previous cell has counter 2, do nothing and check next cell
         - ***Else If*** previous cell has counter 3, erase the cell from K_c and check next cell
      iii. ***Else If*** counter is 3
         - ***If*** previous cell has counter 1 or 2, Erase all the previous cells from K_c and write this one
         - ***Else If*** previous cell has counter 3, do nothing and check next cell

b. ***If*** R/W head arrived at end of tape K_R through the slot having counter 3 then clear K_R by initializing it with blanks and ***return*** to M_TM

c. ***Else If*** R/W head arrived at end of tape K_R through the slot having counter 1 or 2 then traverse tape K_c from left to right to find corner slots $(1,3,5,7)$ and clear tape K_R
   a) ***If*** middle slots found in K_c then write them in tape K_R and check for next cell
   b) ***Else If*** corner slot found
      i. Clear tape K_R, write the corner slot in K_R
      ii. Traverse tape K_c starting from the next cell, left to right
         - ***If*** it is corner slot then write it in K_R
         - ***Else If*** it is a middle slot just ignore it and check the next cell
   c) When all cells of K_R traversed
      i. Write content of tape K_R in tape K_c and ***Return*** to M_TM

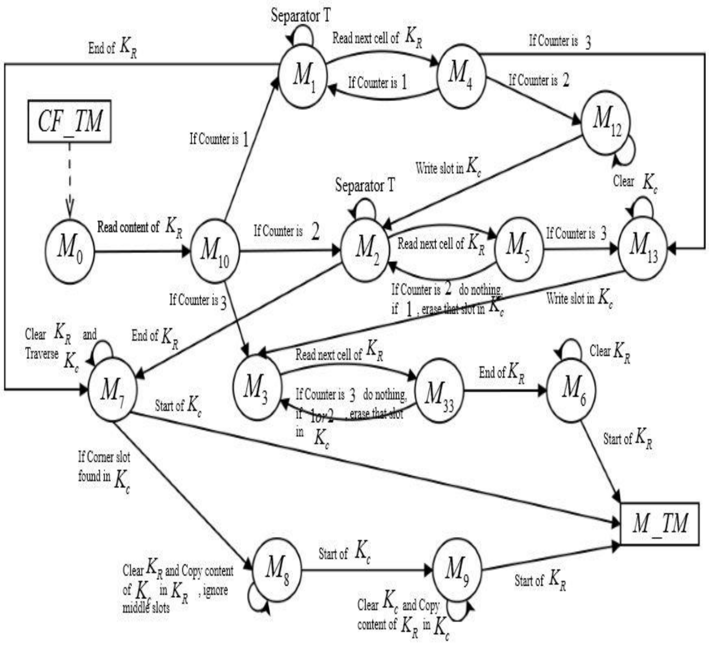

Figure 8. Transition Diagram of MF_TM for Tic-Tac-Toe Game

B. Working of Machine with an Example

To understand the working of the machine an example of the tic-tac-toe game is explained. Let in this example player p (computer) plays first and player o (human) plays second.

Tapes representation

$$K_i = [C,\ p\ /\ o,1,2,3,4,5,6,7,8,9,\$]$$
$$K_p = [C,b,b,b,b,b,\$]$$
$$K_o = [C,b,b,b,b,b,\$]$$
$$K_w = \begin{bmatrix} C,1,2,3,T,4,5,6,T,7,8,9,T,1,4,7,T, \\ 2,5,8,T,3,6,9,T,1,5,9,T,3,5,7,\$ \end{bmatrix}$$
$$K_C = [C,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,,\$]$$
$$K_R = [C,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,b,\$]$$

Step 1- In M_TM player o first chooses p that indicate player p plays first. According to algorithm it chooses slot 5 and then CR_TM replaces all 5's in K_w to player p's symbol X.

$$K_i = [C,\ p,1,2,3,4,b,6,7,8,9,\$]$$
$$K_p = [C,5,b,b,b,b,\$]$$
$$K_w = \begin{bmatrix} C,1,2,3,T,4,X,6,T,7,8,9,T,1,4,7,T, \\ 2,X,8,T,3,6,9,T,1,X,9,T,3,X,7,\$ \end{bmatrix}$$

Step 2- In M_TM , let player o chooses slot 1 then CR_TM replaces all 1's in K_w to player o's symbol O.

$$K_i = [C, p, b, 2, 3, 4, b, 6, 7, 8, 9, \$]$$
$$K_o = [C, 1, b, b, b, b, \$]$$
$$K_w = \begin{bmatrix} C, O, 2, 3, T, 4, X, 6, T, 7, 8, 9, T, O, 4, 7, T, \\ 2, X, 8, T, 3, 6, 9, T, O, X, 9, T, 3, X, 7, \$ \end{bmatrix}$$

Step 3- According to algorithm, first $R1\_R2\_TM$ then $R4\_R5\_TM$ called (if player p plays first). Empty slots under rule R4 and R5 stored in tape Kc and their occurrences are counted by CF_TM and slot with maximum occurrence found by MF_TM that is 9. So CR_TM replaces all 9's in K_w to player p's symbol X.

$$K_i = [C, p, b, 2, 3, 4, b, 6, 7, 8, b, \$]$$
$$K_p = [C, 5, 9, b, b, b, \$]$$
$$K_C = [C, 4, 6, 7, 8, 9, 2, 8, 3, 6, 9, 3, 7, \$]$$
$$K_R = [C, 4, 1, T, 6, 2, T, 7, 2, T, 8, 2, T, 9, 3, T, 2, 1, T, 3, 2, \$]$$
$$K_C = [C, 9, \$] \; Maximum \quad occurrence$$
$$K_w = \begin{bmatrix} C, O, 2, 3, T, 4, X, 6, T, 7, 8, X, T, O, 4, 7, T, \\ 2, X, 8, T, 3, 6, X, T, O, X, X, T, 3, X, 7, \$ \end{bmatrix}$$

Step 4- In M_TM , let player o chooses slot 6 then CR_TM replaces all 6's in K_w to player o's symbol O.

$$K_i = [C, p, b, 2, 3, 4, b, b, 7, 8, 9, \$]$$
$$K_o = [C, 1, 6, b, b, b, \$]$$
$$K_w = \begin{bmatrix} C, O, 2, 3, T, 4, X, O, T, 7, 8, X, T, O, 4, 7, T, \\ 2, X, 8, T, 3, O, X, T, O, X, X, T, 3, X, 7, \$ \end{bmatrix}$$

Step 5- According to algorithm, first $R1\_R2\_TM$ then $R4\_R5\_TM$ called, same as step 3. Slot 7 is best slot. So CR_TM replaces all 7's in K_w to player p's symbol X.

$$K_i = [C, p, b, 2, 3, 4, b, b, b, 8, b, \$]$$
$$K_p = [C, 5, 9, 7, b, b, \$]$$
$$K_C = [C, 4, 7, 8, 2, 8, 3, 3, 7, \$]$$
$$K_R = [C, 4, 1, T, 7, 2, T, 8, 2, T, 2, 1, T, 3, 2, \$]$$
$$K_C = [C, 7, 3, \$] \; Maximum \quad occurrence \quad with \quad corner \quad slots$$
$$K_w = \begin{bmatrix} C, O, 2, 3, T, 4, X, O, T, X, 8, X, T, O, 4, X, T, \\ 2, X, 8, T, 3, O, X, T, O, X, X, T, 3, X, X, \$ \end{bmatrix}$$

Step 6- In M_TM, let player o chooses slot 3 then CR_TM replaces all 3's in K_w to player o's symbol O.

$$K_i = [C, p, b, 2, b, 4, b, b, b, 8, b, \$]$$
$$K_o = [C, 1, 6, 3, b, b, \$]$$
$$K_w = \begin{bmatrix} C, O, 2, O, T, 4, X, O, T, X, 8, X, T, O, 4, X, T, \\ 2, X, 8, T, O, O, X, T, O, X, X, T, O, X, X, \$ \end{bmatrix}$$

Step 7- According to algorithm, first $R1\_R2\_TM$ called and sequence under rule R1 found with empty slot 8. So CR_TM replaces all 8's in K_w to player p's symbol X and declare player p is the winner of the game. Fig 8. Represents board configurations at each steps.

$$K_i = [C, p, b, 2, b, 4, b, b, b, 8, b, \$]$$
$$K_p = [C, 5, 9, 7, 8, b, \$]$$
$$K_C = [C, 8, \$]$$
$$K_w = \begin{bmatrix} C, O, 2, O, T, 4, X, O, T, X, X, X, T, O, 4, X, T, \\ 2, X, X, T, O, O, X, T, O, X, X, T, O, X, X, \$ \end{bmatrix}$$

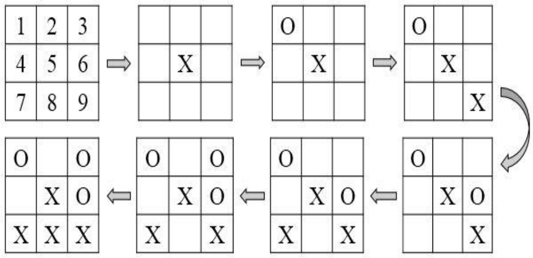

Figure 9. Example of Tic-Tac-Toe Game

C. Comparison of proposed algorithm with previous approaches

There are many approaches to play this game. These approaches use techniques like artificial intelligence, objective functions, neural-network, decision tree algorithm, heuristic learning and genetic algorithm. There are some circumstances where these approaches failed to follow the never lose idea or in creating fork for opponent player, whereas a chance to create fork is possible here so that machine player definitely win the game. The proposed algorithm emphasis not only on never lose strategy but also on the winning strategy. Comparisons among these algorithms is as follows in table I

TABLE I. PREVIOUS STRATEGIES V/S PROPOSED STRATEGY

Parameters / **Strategies**	**Who can start the game**	**If a diagonal is filled in first 3 steps (if opponent starts the game)**		**Sub-optimal player**	**Fork creation for Opponent if there is a chance (if Computer starts the game)**
		O_1 X_5 O_9	***O_5 X_1 O_9***		
Proposed Algorithm	Any Player	Succeed	Succeed	Succeed	Yes
Paper [6]	Only Opponent	Succeed	Succeed	Succeed	----
Paper [7]	Any Player	Fails	Succeed	Can fail	Sometimes
Paper [8]	Any Player	Fails	Succeed	Succeed	No
Paper [9]	Any Player	Fails	Fails	Succeed	No
Paper [10]	Only Opponent	Succeed	Fails	Succeed	----
Paper [11]	Any Player	Succeed	Succeed	Succeed	No

Advantage of proposed strategy over other approaches-

The proposed work have strategy to play for both the players either player plays first or second but algorithms described in [6, 10] works only for the player who play second, no strategy was defined for the player who plays first. The proposed strategy succeed in playing with a suboptimal opponent player of the game but solution provided in [7] can fail sometime with a suboptimal player. The proposed strategy succeed to follow never lose idea in cases where opponent plays first and one of the diagonal filled in first 3 steps but strategies explained in [7-9] fails in same. The proposed strategy succeed in fork creation for opponent player if possible but strategies explained in [8-9, 11] fails in situations explained above sometimes. Procedures in [12-13] shows only best sequences to play among all the sequences that are generated using genetic programming. Method described in [14] can fail in never lose idea because machine will play either according to the stored sequences if match found for current sequence or randomly.

IV. CONCLUSION AND FUTURE SCOPE

The proposed algorithm provide a winning strategy for Tic-Tac-Toe so that machine player will be the winner or game outcome will be a draw but never lose. It has been solved previously using genetic operators and algorithm, heuristic learning or decision tree algorithm but these strategies failed in some situations like fork creation for opponent player if possible and play optimally with a sub-optimal player. The proposed WST algorithm is successful in the same. This paper focus not only on never lose strategy but also on the winning strategy for Tic-Tac-Toe game. Result prediction model for the Tic-Tac-Toe game can also be designed using theoretical computation. These methods can also be used to solve other perfect information board games like Minesweeper, Reverse-free.

REFERENCES

[1] S. Garg and D. Songara, "The winner decision model of Tic-Tac-Toe game by using multi-tape turing machine," *International Conference on Advances in Computing, Communications and Informatics*, Sep 2016, pp. 573-579.

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

[7] S. Karamchandani, P. Gandhi, O. Pawar and S. Pawaskar “A Simple Algorithm for Designing an Artificial Intelligence Based Tic-Tac-Toe Game”, *International Conference on Pervasive Computing*, pp1-4, 2015.

[8] A. Singh, K. Deep and A. Nagar “A ‘Never-loose’ Strategy to Play the Game of Tic-Tac-Toe”, *International Conference on Soft Computing & Machine Intelligence*, pp. 1-4, 2014.

[9] S. H. Ling and H. K. Lam “Playing Tic-Tac-Toe Using Genetic Neural Network with Double Transfer Functions” *Journal of Intelligent Learning Systems and Applications*, vol 3, pp. 37-44, Feburary 2011.

[10] S. Sriram, R. Vijayarangan, S. Raghuraman and X. 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*, pp. 1211 – 1216, June 2009.

[11] P. chakraborty, “Artificial Intelligence Based Strategies to Play the Tic-Tac-Toe Game” *Journal of Technology and Engineering Sciences*, Vol 1, No. 1 January 2009.

[12] 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 no. 5, pp. 797-801, 2013.

[13] A. 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*, pp. 889-896, July 2008.

[14] Thomas Abtey and SUNY Oswego “A Tic-Tac-Toe Learning Machine Involving the Automatic Generation and Application of Heuristics”.

[15] 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," *IEEE Symposium Series on Computational Intelligence*, pp. 564-570, 2015.