Tic-Tac-Toe

Thesis · December 1975

1 author:

Peter Baum
Aesir Research

Tic-Tac-Toe

by Peter Baum
December 1975

Thesis for the Master of Science Degree

Computer Science Department
Southern Illinois University
Carbondale, Illinois 62901

Thesis Adviser: Professor Kenneth J. Danhof

TABLE OF CONTENTS

AESIR RESEARCH HOME PAGE

Last Modified: November 11, 2021

INTRODUCTION

Tic-Tac-Toe is a well known game (11,17,19,24) played by two persons who alternately place X's and O's upon a 3x3 playing field such as figure 1. The players first decide who will mark his moves with an X and who will go first. Play proceeds with the opponents alternately placing their marks in any unoccupied cell delineated by the figure. The object of the game is to be the first player with 3 marks in a row, where a row can be either vertical, horizontal, or diagonal. If all the cells become filled the game is a draw. A great number of isomorphs of this game are known, many of which are discussed by Herbert A. Simon in his monograph "Representations in Tic-Tac-Toe" (36).

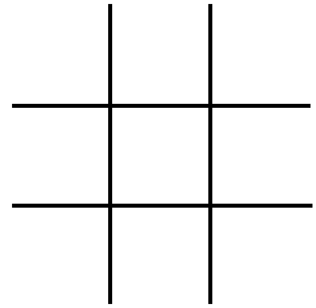


FIGURE 1

A variety of spellings for the standard game described above have been used. Throughout this paper, we shall use the abbreviation "TTT." The exact origin of TTT is unknown but variations such as "three men's morris" were popular in England in 1300 (24). In this game each player has exactly 3 counters to mark his moves. After your 3 counters have been placed on the board, successive moves will consist of moving one of your counters to an unoccupied cell. GO-MOKU is a variant of TTT played upon a 19x19 matrix where one must get exactly 5-in-a-row to win. It is an ancient Japanese game that is usually played with stones on the intersections of a GO board.

Most readers will be aware that there are only a few basic patterns of play that occur during a game of TTT because of its symmetrical nature. Once these patterns become familiar, the games result in a draw and the players apt to lose interest. There are however, many difficult and interesting questions to be considered. One class of such questions considers a particular arrangement of X's and O's within the game field and asks who went first or upon which cell the last move was made. Other subjects arise as a result of it being used as part of a magician's prediction routine (28). The problems dealt with in this paper will be limited to those relevant to questions of strategy.

The search for a good strategy for TTT may appear in the guise of circuit design considerations. Let's look at a few of the TTT playing machines that have actually been constructed. For about $45 (1971) one can build Don Lancaster's special purpose computer TIC-TAC-TRONIX (29) that uses 5 integrated circuits, 31 transistors, and a resistor-diode array for the logic element. Play is simplified by having the computer go first and make its first move in one of the corners. Strategy on the next two successive moves is determined by the opponent's moves. It should be pointed out that quite a bit of the circuitry here is involved with lamp drivers, power supply, and assuring that the opponent doesn't cheat. In addition to the usual controls, a hidden smart-dumb switch allows the computer to be occasionally beaten, a feature that may be considered by some as absolutely essential for any game playing machine (37)! The circuit

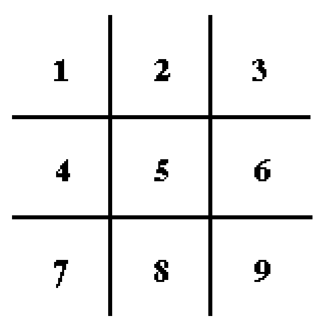


FIGURE 2

design here is much different from B. Bawer and W. J. Hawkins "Tick'Tack'Toe in a cigarette box" (27) which requires that the human go first and that this move is to one of the specified corners. If we number the cells as in figure 2, the circuit consists of switches that light the pairs given in figure 3.

Hendrix and Purcell (23) used tubes, relays, and 187 neon lamps in the late 1950's to create a machine as different in strategy from the preceding as it was in hardware requirements. The machine first looked to see if it had played in 2 cells along a row with the third cell empty. If not, it looked to see if its opponent had played 2 cells along a row with the third cell empty. If neither of these situations had arisen, it played the first vacant cell in the sequence 5,1,6,7,2,9,4,3,8 (see figure 2). As this strategy sometimes led to a loss, a special defense mode was in effect under certain "exceptional circumstances."

switch and light for human is in this cell	also lights "computer" response in this cell
1 = human's first move	5
2	3
3	2
4	7
6	8
7	4
8	6
9	8

FIGURE 3

Programs to play TTT written for general purpose computers are probably as varied as is their special purpose relatives. We suspect however, that most use the "many if statement" approach rather than an evaluation function (33). The problems of best representation and best algorithm are of major importance here. We can define the best algorithm as one whose computer program representation uses a minimal number of bits of machine code or minimizes the time needed to execute the algorithm on a particular machine.

Programs to play some of the more interesting generalizations seem to be becoming more popular as of late. The first North American Computer GO-MOKU Tournament is to be held on November 29th and 30th, 1975 at the University of Guelph in Guelph Ontario.

There is no nice mathematical theory into which we can place TTT. Indeed, it is seeing the elegant theory developed for NIM (1) that may cause us to wonder if we really understand TTT on anything but a superficial level. It is the author's prejudice that the creation of such a mathematical theory would help us answer some of these questions about circuit design and optimal strategies. If nothing else, it might simplify the proof that some particular strategy did indeed guarantee a win. Like TTT, machines have been designed to play the game of NIM and the interplay between theory and circuit design is quite evident (7,8,9,10). The theory also provides a nice platform from which a host of generalizations have been launched (2,3,4,5,6). One notices for example, that the strategies for the generalizations are usually very similar to that described in (1) and were undoubtably aided by this work.

Can we be more precise when we state that we wish to create a mathematical theory for the game of TTT? One possible meaning is that we are seeking a precise description of the strategy that guarantees a win for those games for which a win is possible, and one that is a great deal simpler than a complete search of the game tree. By "a great deal simpler," I refer the reader to the works of James R. Slagle (33) and Gregory Chaitin (34). It is the same kind of thing that is dealt with in other complex problems (33), chess for example (32). One must add however, that chess players seldom (35) prove explicitly that their heuristic or algorithm must inevitably lead to a win (solution). This then is the motivation for attempting to create a mathematical theory of TTT. An indication of just how difficult such an undertaking is, can be deduced by examining the meager results that have been obtained to date by such men as Paul Erdös (30) and Daniel Cohen (38).

After much fruitless effort, it appeared that the configuration of TTT was of such a special nature that perhaps a generalization of the game would allow one to see more clearly the essence of the required strategy. For this reason, the following generalization was made. Consider the alternative description of TTT as illustrated in figure 4. Here the moves are made by occupying each of the 9 points (indicated by the symbol "O" and the collection of positions that are 3-in-a-row (which constitute a possible winning sequence) are indicated by connecting the points with lines. We do not require that the lines be straight. Following the conventions of combinatorial mathematics, we define a block to be a collection of points, which in our case, will indicate the points needed to form a win. Generalized Tic-Tac-Toe will be a game played on any arbitrary collection of points and blocks. An example of such a game is that given by figure 5. Here every block contains exactly 3 points. To avoid confusion, we sometimes connect the points on a block with a " " as well as " ". To win the game, we must be the first player to occupy all the points on a single block.

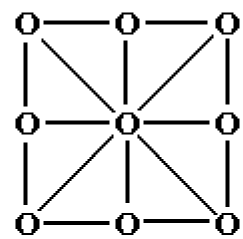


FIGURE 4

Another example of such a game is that played with four 4x4 fields stacked up to form a cube. In this game, it takes 4-in-a-row to win and is already so complex that it is not known whether or not the first player can play so as to always win. This game is sometimes sold under the name "Cubic" but we shall refer to it as the "4-cube" game. For those readers who have never played this game, may we suggest that the configuration of 64 points and 76 blocks is highly entertaining.

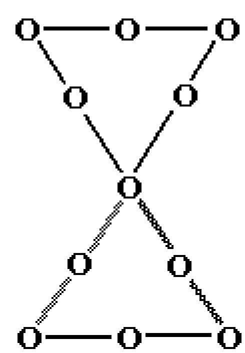


FIGURE 5

The structure of this paper is somewhat elucidated by figure 6. Here we see TTT situated along with NIM and NIM's generalizations as a 2-person game. Naturally, other descriptive arrangements are possible, this particular one being presented to stress the relationships relevant to the chosen presentation. The point/block representation of TTT provides the basis for the generalized version of TTT. The lambda referred to represents the maximum number of points that 2 blocks can have in common. For TTT and the 4-cube game it is equal to 1. For GO-MOKU, lambda = 4. Examples of games for which lambda is even larger are given in the paper by P. Erdös and J. L. Selfridge (30).

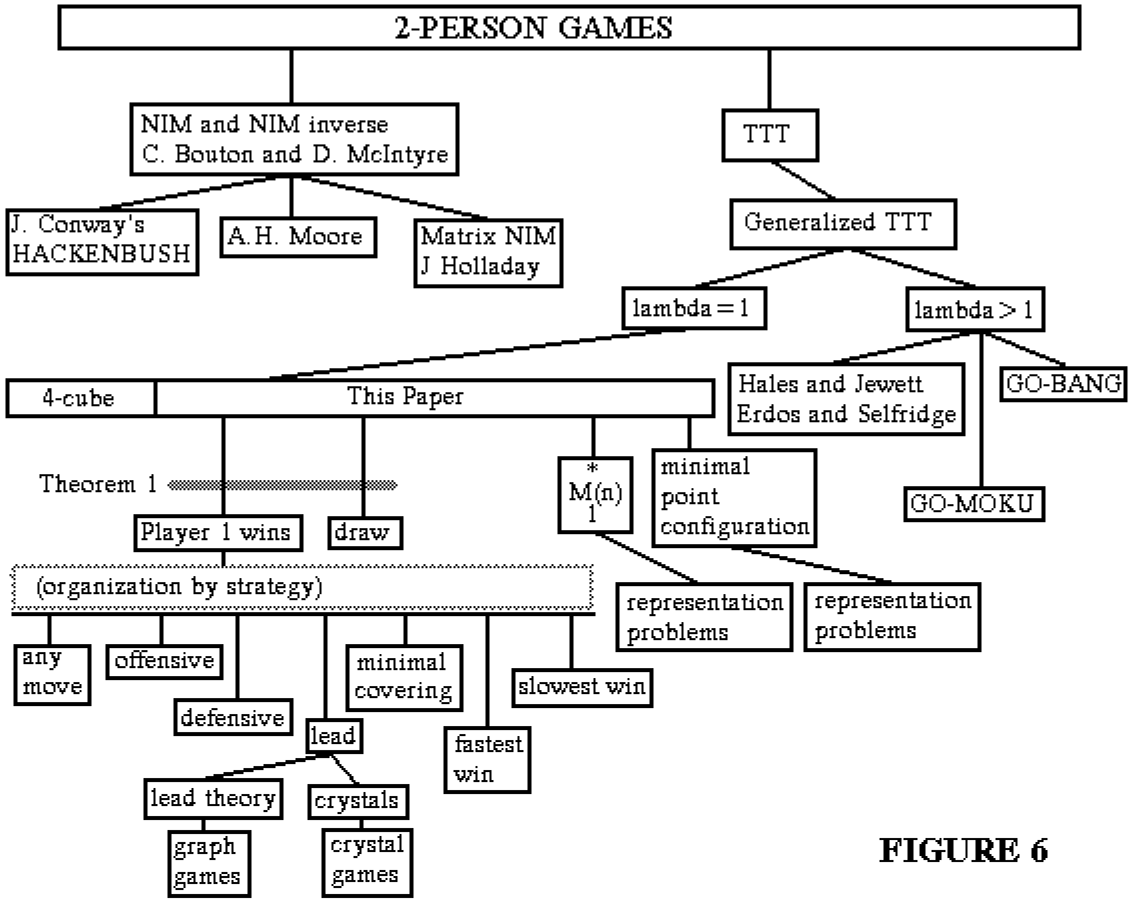


FIGURE 6

Given that lambda = 1, we may also consider the following questions. What is the minimal number of blocks for which a game exists where every block contains N points? This is sometimes expressed as "what is the value of $M_1^*(n)$?" The value for $M_1^*(3) = 6$, but the value for $M_1^*(4)$ is not known. Similarly, we can ask, "what is the minimal number of points for which a system of N block, K points to a block, gives a win to the first player?" After we have determined what these minimal number of blocks and points are, there remains the questions of representation for the various configurations.

The primary focus of this paper is on the generalized games of TTT for which lambda = 1. The reason that lambda was so chose is because it is not clear how such conditions affect various game strategies and one of the original objectives of this research was to deal with some unanswered questions about the 4-cube game. As previously explained, lambda is 1 for the 4-cube game. Theorem 1 will show that player 2 can never win against flawless play by the first player. Most of the results that follow deal with games for which the first player has a win. This is primarily because there are nice collections of such games that, under the assumption of perfect play, can be shown to be completed within a small number of moves. In general, games for which there is a draw will be less manageable.

CHAPTER 1 - FUNDAMENTAL THEOREM

First we shall prove a theorem which shows that for a large class of games the second player can never win against perfect play by the first player. It will be clear that our generalized TTT belongs to this class. This allows us to break such games into 2 groups, those which result in a win for player 1 and those which result in a draw. As previously explained, we shall deal primarily with the first class. For such games, every possible move for the first player can be considered either safe, that is, leading to a win, or unsafe and leading, if the second player is clever, to a draw or a loss. In chapter 2, we shall begin the search for a function which tells the first player whether or not various moves are safe or not.

Definition: A player has **a forced win** provided that he can win regardless of the moves taken by his opponent.

Theorem 1: If TTT is generalized to a game on an arbitrarily complex point/block structure where the object of the game is to occupy some points on one or more blocks (not to include games where your opponent must also occupy the block with a certain number of points), then the second player never has a forced win.

> PROOF: Assume player 2 has a forced win. As a particular game progresses, we denote the successive moves (points) taken by the first player as p(1), p(2), ..., p(n). Since play alternates between players, player 2 wins on move n. Denote the moves of player 2 by q(1), q(2), ..., q(n). Let P(i) be the set of elements p(1), p(2), ..., p(i) and Q(i) the set of elements q(1), q(2), ..., q(i). We also define P(0) = Q(0) = the empty set. Since player 2 has a winning strategy, we can represent this as a function that depends on the point/block structure, the points taken previously by player 1, and his own prior move. Functionally this is written as
>
> q(i) = S(P(i), Q(i-1))   for i=1, 2, ..., n
>
> where the 1st argument is the set of player 1's moves and the second argument is player 2's previous moves. It is understood that this function is over a fixed point/block structure. Consider the game played where player 2 plays any strategy (we may as well take it to be the winning strategy S) and player 1 plays strategy S' defined as follows:
>
> Let p(1) be any arbitrary but fixed move.
>
> Define a(0)=p(1)
>
> For i=2, 3, ..., N
> define p(i) = S(Q(i-1), P(i-1) - a(i-2))
>
> and
> for i=1, 2, ..., n-1
> a(i) =
> | a(i-1) if p(i) not equal to a(i-1)
> | any arbitrary point not yet taken otherwise.

We can think of the a's as being arbitrary points that player 1 "throws away" as he follows player 1's strategy S. First let us be sure that the definition makes sense. S is defined on arguments that are two mutually exclusive sets, the second with one fewer point than the first. Notice that a(n) is not defined, a matter that will be elucidated later.

Consider a game that is played to completion. Notice that if a new game is played but a(1) is taken to be the a(n-1) of the old game, that similar moves of player 2 will result in the exact same sequence of moves by player 1 since as far as the function S is concerned, the a's are not visible. Let us assume that this is the case to simplify the following description. The argument is not substantially changed if this assumption is not made.

Our new game sequence can be (relative to a sequence of p's and q's given for a game played using strategy S) described by

(i) a(1) p(1) q(1) p(2) q(2) p(3) q(3) ... q(n-1) p(n-1)

Now strategy S assures us that

p(1) q(1) p(2) q(2) p(3) q(3) ... q(n-1) p(n)q(n)

(q's generated by strategy S) is a winning sequence for the player placing down the q(n) provided that

1. a(1) does not "interfere" with the sequence in some way
2. q(n) can or has been played by player 1.

Now if we show statement 1 to be true, then if there are 2n points, a(1) occupies the only point left. If more than 2n points on the structure, then q(n) already occupied by a(1) or it is player 1's move and he can move there.

We now show statement 1. First notice that we need only consider effects of a(1) that are detrimental to player 2 since we are trying to show that contrary to our original assumption, player 1 has a forced win. Now a(1) can have an effect on the game in any of three possible ways:

3. effect on an element of the goal, which is to play points on a block
4. effect by reason of occupying a point, and thus preventing either player from again taking the point
5. the play affects the sequence of moves.

We consider these points individually:

6. If a(1) is on a block useful to player 2, it nullifies the block. If on a block useful to player 1, it helps, perhaps causing a win before move N. If eventually on a block occupied by both players, no harm is done.

7. By construction, such a move will not hurt player 1 since the effect is to play there and then make an additional move. Restricting the possible responses of your opponent, does no harm (assuming a faultless opponent!).
8. By (i) we see that the move sequence is not disrupted. In fact the only effect of a(1) is to limit the responses of player 2.

CHAPTER 2 - ATTACKING THE PROBLEM

Let us begin the search for some components out of which we can build a mathematical theory for our game on a point/block structure with lambda=1. We will be examining games for which the first player can always win and try to characterize both the games and the strategies required. Let me say as the outset, that no neat all encompassing theory has been discovered. It is hoped however, that the reader will find that the explorations in an attempt to reach such a goal are none the less of some value.

Perhaps one of the first questions one might ask is whether or not the possibility of a win is a strictly local phenomenon. Stated more precisely, can one always determine by examining the points and blocks near the winning blocks(s), if a win is always possible? We will see that the answer is no by looking at figure 7, which also shows that there are games with an arbitrarily large number of points and blocks for which the first player has a win.

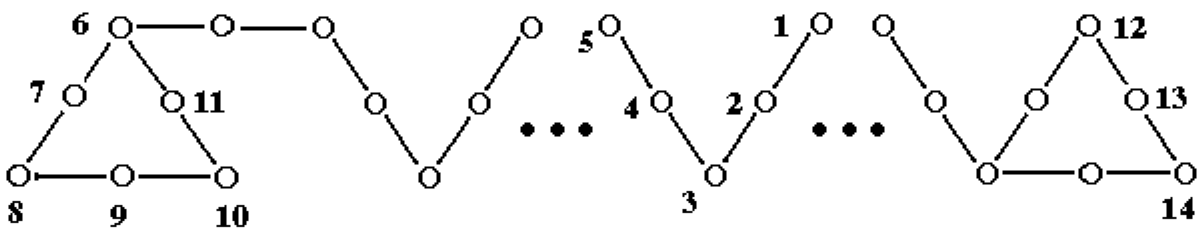


FIGURE 7

Player 1 begins by moving to point 1. Player 2 must make his move on the structure either to the right of point 1 or to the left. Let us assume without loss of generality that it is to the right. Player 1 then plays to point 3, forcing player 2 to point 2. Then play is made to point 5 forcing player 2 to point 4. This continues for an arbitrarily long number of moves until player 1 has moved to points 6 and 8. Player 2 then moves to point 7. Player 1's move to point 10 will force a win either to points 9 or 11, for a **fork** will have been set up. Notice that although our winning blocks are either (8,9,10) or (6,11,10) that without block (12,13,14), or (3,4,5), the point/block structure will not give a win every time for player 1. Thus the characteristic of being a game that the first player can always win on can not be determined by examining sections of the structure local to the winning block(s). Two other important ideas are also illustrated by this example: the creation of a fork that results in a win and the situation where one's opponent is forced to exactly one move on successive play. This second idea will be described by the term **lead** which is carefully defined in Chapter 4.

Figure 8 shows that we should not assume that the best move is to a point on many blocks. In that example, every block contains 3 points so that point 1 is on 6 blocks but it is not a winning move for player one. On the other hand, point 2 which is on only 2 blocks is an excellent choice for player 1's first move.

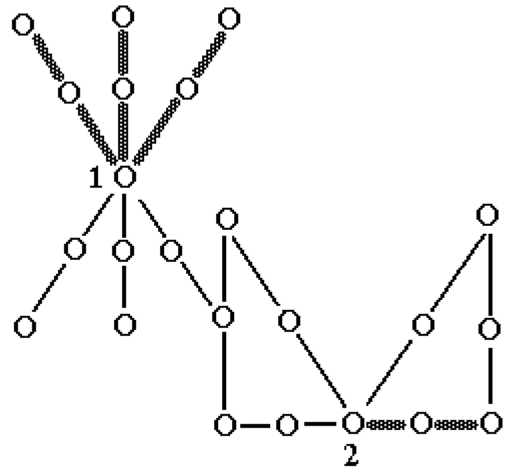


FIGURE 8

<u>**Conjecture**</u>- If the first player's first move is to a point on a single block, then the second player need not lose.

Perhaps one of the most natural approaches is to try to determine the possible moves prior to a fork, their antecedent moves, and thus working backwards, determine the kind of point/block structure that was present to begin with. It was this approach that resulted in much of the development found in Chapter 4.

Another idea is to try to find some basic point/block structures upon which the strategy for a win is evident. One would then hope to show that adding blocks does not have any affect on the required strategy. Figure 9 shows that by adding a block we can completely change the property of a point/block structure giving a win for the first player. Player 1 can force a win on structure 9a but not 9b which contains an additional block. Notice that in this example, we have not even added any additional points to the structure.

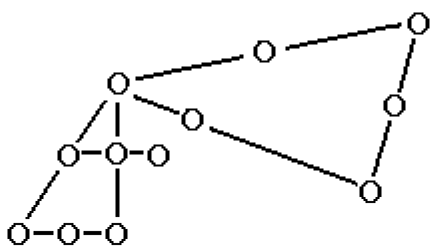


FIGURE 9a

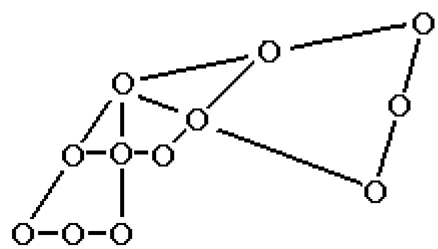


FIGURE 9b

Another means of attacking the problem of discovering the elements of a mathematical theory for generalized TTT is to examine some strategies for playing the game and try to determine the class of games for which the strategy is successful. One notes here that in October of 1899, Mr. Paul E. More showed Charles Bouton a method for winning at NIM but was unable to give a proof for his rule. Bouton's classic paper, "Nim, a Game with a Complete Mathematical Theory" (3), was a result of the examination of this rule.

One common classification of strategies is by their offensive or defensive nature: are we looking for forks that we can create, or trying to prevent our opponent from creating forks? Although such a classification may not be of fundamental relevance, it is difficult to proceed with an investigation such as this without using the terms to express intuitive ideas about what is occurring.

Similar strategies can appear in different forms. For example, we can assign weights to the blocks according to the number of points taken by each of the two opponents. We may then choose points that maximize or minimize a function over the values of all the blocks of the game. Alternately, this entire situation can be simply described in terms of choosing the points so that various blocks will have some specified characteristics such as 3 X's in a row. By using a system of weights for the moves and an arithmetic function, we can sometimes simplify our algorithm by taking advantage of the symmetrical nature of the game and the computer's ability to perform certain arithmetic calculations. Typical values for such a function are given in figure 10. Here we shall be considering games for which every block has exactly 3 points. We let the points of the block be x, y, and z. These variables are 0 if unoccupied, 1 if occupied by an X, and -1 if occupied by an O.

x	y	z	$F(x,y,z)=(x+y+z)^3 + (x*y*z)$
0	0	0	0
+1	0	0	+1
-1	0	0	-1
-1	-1	0	-8
+1	+1	0	+8
-1	+1	0	0
+1	-1	+1	0
-1	+1	-1	0
+1	+1	+1	+28
-1	-1	-1	-28
.	.	.	.
.	.	.	.
.	.	.	.

FIGURE 10

This function has some nice properties. First of all the function value doesn't depend upon the order of its arguments. Notice also that the larger the absolute value of F(x,y,z) is, the more significant the block probably is as a game is actually being played. Let

X(1), X(2), X(3), ..., X(i)

be the blocks of the point/block structure at some point in a game. Let F(X(j)) be F applied to the points of block X(j). We now define G.

$$G = \sum_{i} F(X(i))$$

(the sum is over all blocks of the game)

3	2	3
2	4	2
3	2	3

1	2	1
2	X	2
1	2	1

0	1	X
1	-1	1
0	1	0

-1	X	-1
0	-2	0
-1	0	-1

7	X	7
0	Ø	0
1	-1	1

FIGURE 11

If player X plays a point so that the resulting G is maximal (or player O plays so G is minimal), we have a fairly good strategy for TTT. Figure 11 gives the values of G for the points on some familiar TTT patterns. We will sometimes use the symbol Ø in place of the symbol O (oh) to distinguish it from 0 (zero). Assume in the figures that player X moves first. The value in a cell is the value of G if a move is made to that cell.

The only problem occurs for the situation given in figure 12. Here our strategy of picking a point that maximizes G will lead to a fork such as shown in figure 13 and we lose.

X	-7	-8
-7	Ø	-7
-8	-7	X

FIGURE 12

X		X
	Ø	
Ø		X

FIGURE 13

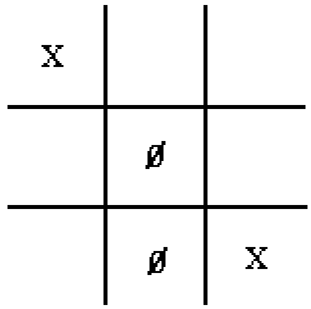


FIGURE 14a

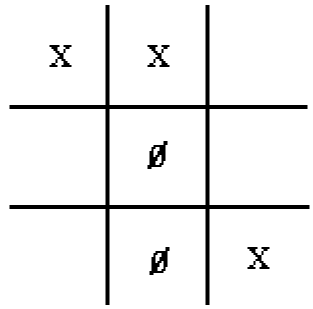


FIGURE 14b

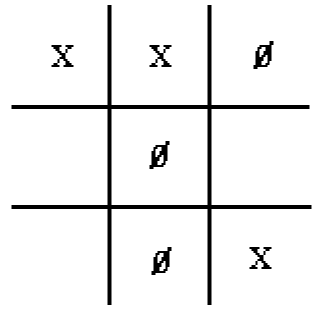


FIGURE 14c

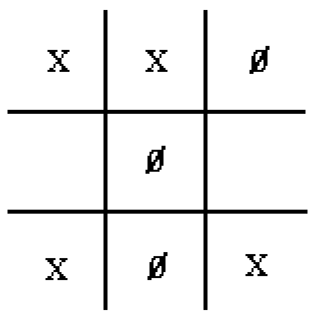


FIGURE 14d

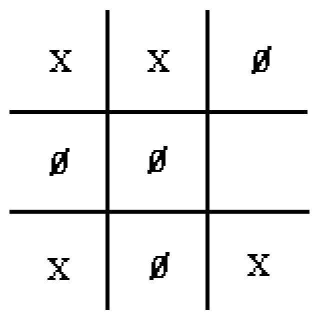


FIGURE 14e

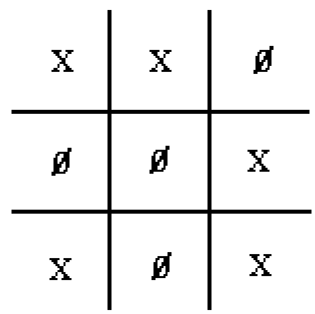


FIGURE 14f

A correct move is shown in figure 14a and results in a draw. Now it may be possible to define a new G that uses function F and does not have this defect. What is important however, is that although it is true that player X is better off in general by maximizing the function G, he may be forcing his opponent to make a good move. Another way to look at this situation is to notice that although the move to position 8 in figure 14a (numbering as given by figure 2) still allows player 1 to form a fork by moving to cell 3, player 1 must respond to the threat and ignore the highly prized fork that is still possible. In the case of the sequence shown in figures 14a thru 14f, the board fills up and a draw results. We see that this forcing of one's opponent, and being forced in turn, can nullify completely what generally speaking are strong strategic elements. The idea of forcing one's opponent and being forced in turn is examined more closely in the following chapter. A modification of our function G that does result in a good strategy for TTT is to play through any forced moves before the function F is applied. Such a strategy does not in general guarantee a win for an arbitrary point/block structure as can be seen by examining figure 8.

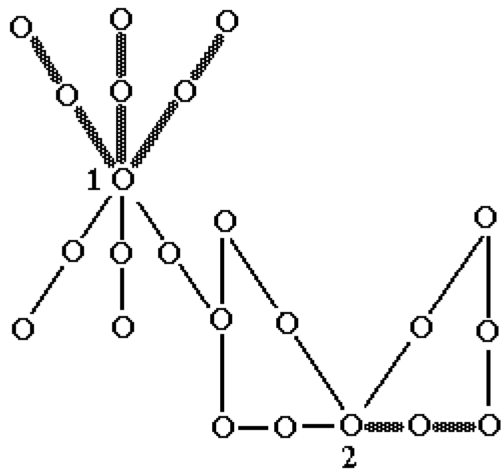


FIGURE 8

As stated previously this paper deals primarily with games for which lambda=1. We will not hesitate however, to remove this restriction in order to obtain useful ideas. In general, when lambda is greater than 1, we have lots of blocks around on the points we take and the second player has a harder time pulling off a draw game. The most extreme case occurs when we have n points, k points to a block, and the blocks are the n!/((n-k)!k!) subsets of the n points that contain exactly k points. Here the first player wins in k moves regardless of the points chosen so that the strategy is trivial.

P. Erdos and J. L. Selfridge (30) consider games such that every block contains k points, lambda unrestricted, and the first player can always win under the assumption of perfect play. They showed that such games with the fewest number of blocks contain exactly 2^{k-1} blocks. Figure 15 gives an enumeration of the blocks for such a game when k=4. Notice that every point is on exactly half of the blocks except for point number 1. The winning strategy is as follows. Player 1 first takes point 1. Thereafter, if player 2 takes point A, player 1 tapes point A+3. The pairs of points A and A+3 for A=2, 3, and 4 cover all the blocks. Since player 2 can then only block half of the remaining blocks unoccupied by one of his points, the 4th move of player 1 results in a win. In chapter 4 we shall develop this idea of situations evolving so that a player is unable to block all the critical portions of a game.

BLOCK NUMBER	**POINTS TO FORM BLOCK**			
1	**1**	**2**	**3**	**4**
2	**1**	**2**	**3**	**7**
3	**1**	**2**	**6**	**4**
4	**1**	**2**	**6**	**7**
5	**1**	**5**	**3**	**4**
6	**1**	**5**	**3**	**7**
7	**1**	**5**	**6**	**4**
8	**1**	**5**	**6**	**7**

FIGURE 15

CHAPTER 3 - CRYSTALS

In view of the sequence given by figure 14a-14f we make the following definitions:

<u>**Definition**</u>: A **move is forced** if the player must move there to prevent his opponent from winning the game on the next move.

<u>**Definition**</u>: During play of a generalized TTT game, we call a sequence of moves a **crystal** provided each successive move of the sequence is forced.

<u>**Definition**</u>: An **X-tal** is a generalized TTT game that has progressed to such a point that an understood move(s) to some point(s) will result in a crystal

<u>**Definition**</u>: The point where a move will generate a crystal is called a **germ** of the crystal. Making a move to such a point is called **germination**.

<u>**Definition**</u>: The collection of moves prior to germination are called a **seed**.

Our purpose here is to examine crystals more closely in an attempt to understand this very important strategic element. Ideally one would like a simple arithmetic formula which would give the final game state after a crystal has been generated from any specified seed and germ. It would also be nice to completely characterize the structures on which crystals are used by the first player to win a game. Although not able to reach these objectives, we will show how to generate an infinite class of games whose primary component is a crystal.

<u>**Definition**</u>: If the number of points per block is always K, then we way that the **molecular length** of the game is K.

<u>**Definition**</u>: The X's and O's which constitute a move are called **atoms**.

The simplest infinite (non-trivial) X-tal has molecular length 3 and is given in figure 16.

··· –Ø—O—X—O—Ø—O—X—O—Ø—O—X—O—Ø– ···

FIGURE 16

Germination takes place with either type of atom placed at any of the open positions (indicated by O).

Figure 17 shows three seeds that when germinated properly, completely fill the point/block structure. We call such seeds, **perfect seeds** and the generated crystal, a **perfect crystal**.

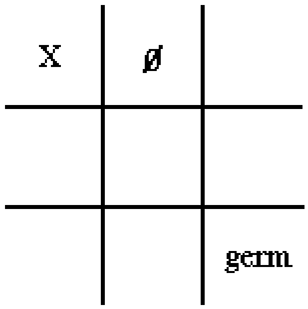


FIGURE 17a

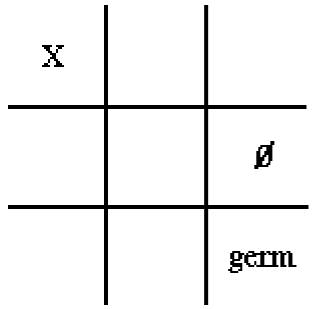


FIGURE 17b

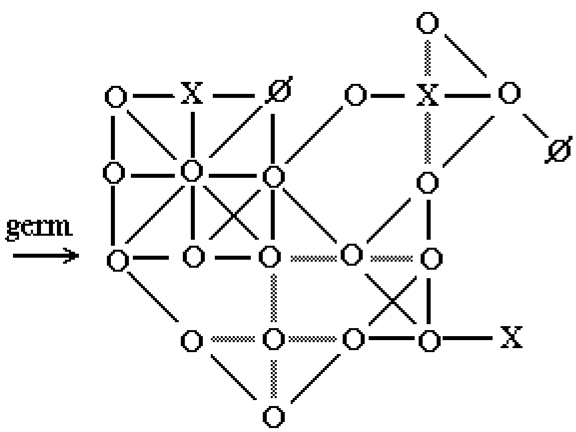


FIGURE 17c

Here as in figure 18, the molecular length is 3. Figure 18 has an interesting growth pattern that the reader is invited to discover.

Two questions may have already occurred to the reader:

1. Given a particular point/block structure, how do we find the smallest (fewest atoms) perfect seed?
2. Does a finite seed exist which generates an infinite crystal?

The first question is probably very difficult. The answer to the second question is yes. Figure 19 shows how this can be done. Here again the molecular length is 3. The block structure is perhaps more clearly described by an enumeration of the blocks:

(1 3 4) X on point 1, germinate at 3
(2 4 5) Ø on point 2
(3 5 6)
(4 6 7)
(5 7 8)
(6 8 9)
(7 9 10)
(8 10 11)
(9 11 12)
.
.
.

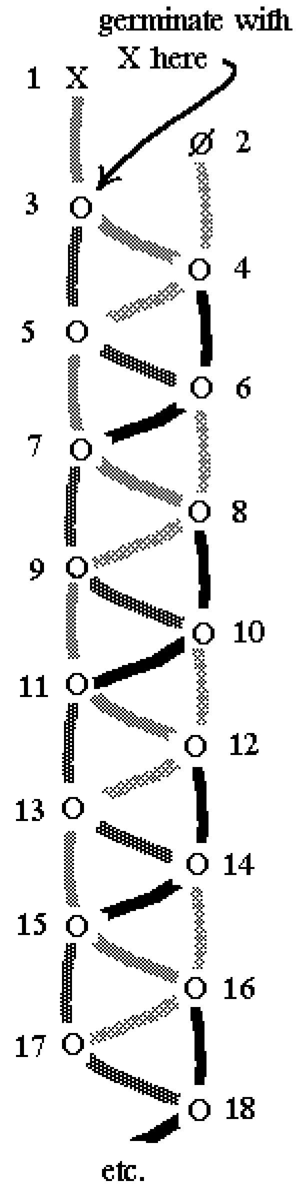


FIGURE 19

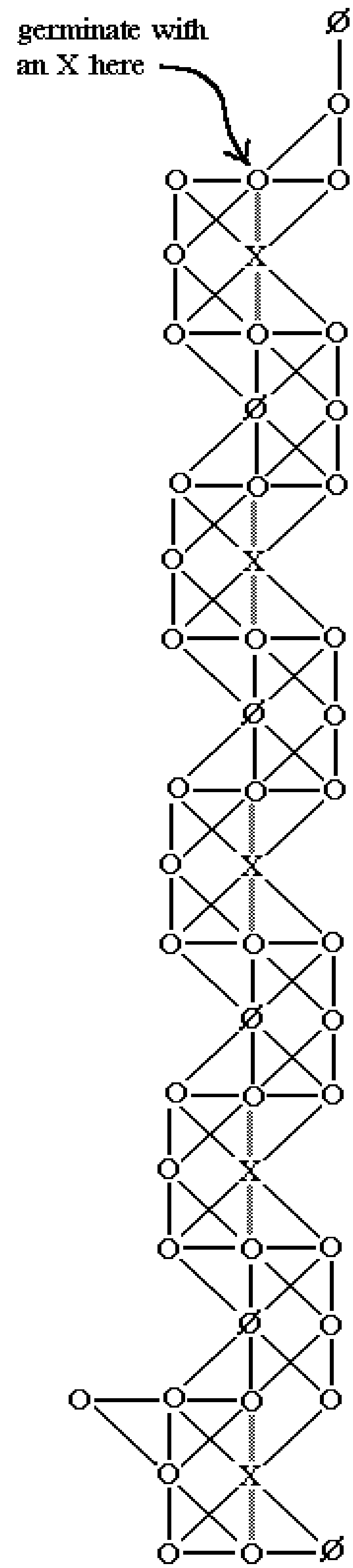


FIGURE 18

The reader will find that we can represent the blocks by points on straight line segments but that the segments will get larger or smaller as the figure progresses. The start of such a representation is given by figure 20.

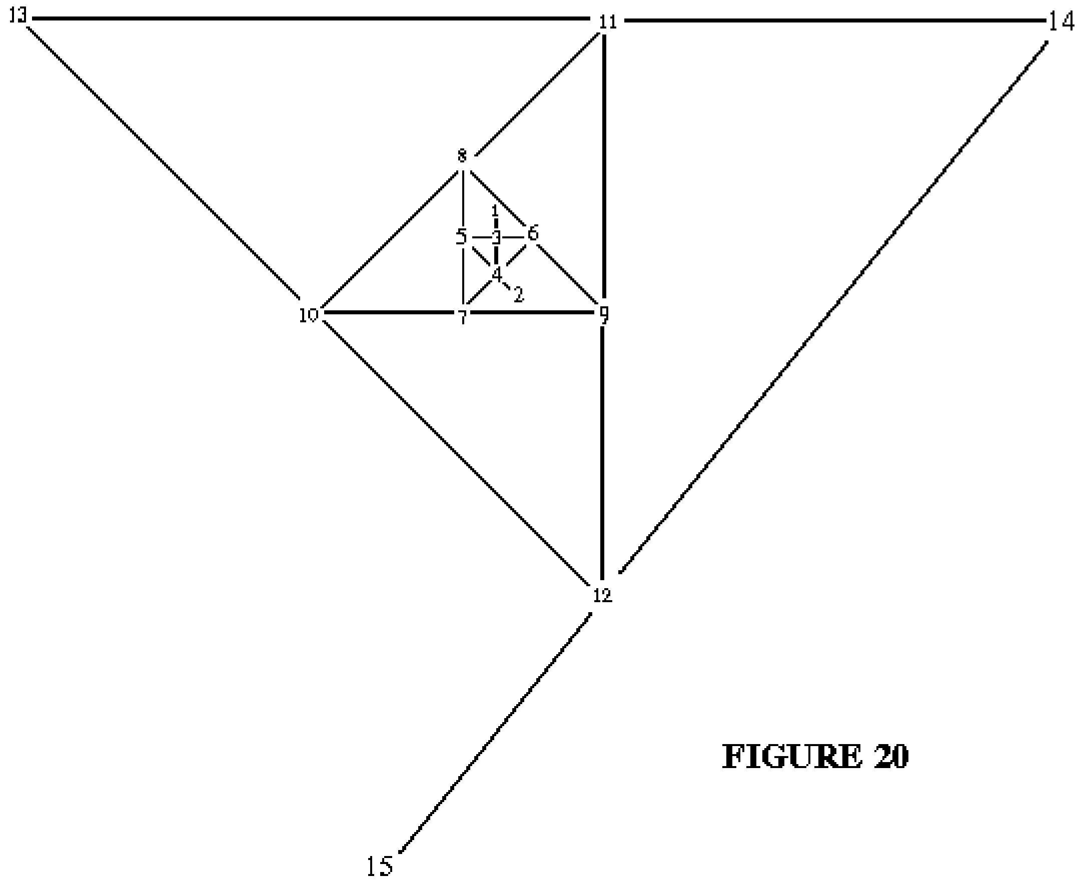


FIGURE 20

One can also take a finite section of the X-tal shown in figure 19 and bend it to form a circular shaped structure that generates a perfect crystal. One must however, join the ends together properly.

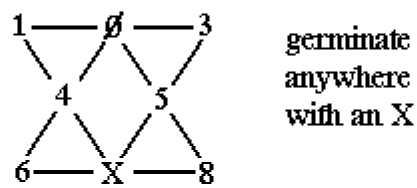


FIGURE 21a

Let us see how to create a game where the first player has a win by using a crystal. We start with an X-tal such as that shown in figure 21a. Assume player 1 moves to point 7. Below we give the proper response to each possible move by player 2.

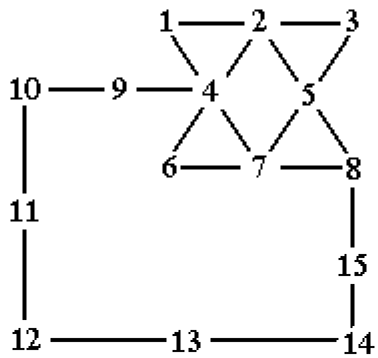
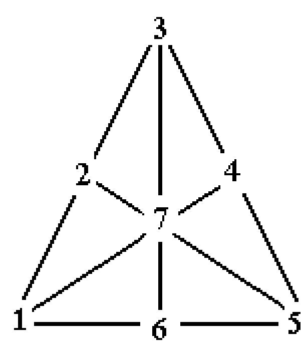


FIGURE 21b

Player 2 goes to point	**Player 1's response**
4,9,10,11,12,13,14, or 15	5, fork at 8
1	8, fork at 5
3	6, fork at 4
5	4, fork at 6
6	3, fork at 1
8	1, fork at 3

If the second player moves to point 2, player 1 moves to point 4, a crystal develops, and then the blocks added to figure 21a to get figure 21b are used by player 1 to win.

The general procedure is to find an X-tal where player 2 is limited by the structure to moves which form a seed. We run through the crystal and then add blocks to still give player 1 a win. The thing that is difficult about this procedure is that we must take great care when we add blocks (and perhaps points) to assure that the seed or the crystal isn't adversely affected. Figure 22 gives the block structure for a crystal with 7 points and 6 blocks. If player 1 moves to point 1, player 2 must move to point 3 or 5 to avoid some obvious forks. On the other hand, if for instance player 2 moves to point 3, player 1 moves to point 4 forming a crystal. By adding a block (2,4,6) we can form a winning game. This 7 point, 7 block structure is a well known figure called the **Fano Plane**. Actually, the X-tal of figure 22 is already a point/block structure on which player 1 can always win!
It is figures such as this and our previous figure 5 that motivated the development seen in Chapter 4.



(1,2,3)
(3,4,5)
(1,6,5)
(1,7,4)
(3,7,6)
(2,7,5)

FIGURE 22

The following are the blocks of a 10 point X-tal that can also be transformed into the required crystal game without the addition of any points:

Block Number	points on block
1	(2,4,3)
2	(1,3,6)
3	(4,6,5)
4	(1,5,8)
5	(2,8,7)
6	(1,7, 10)
7	(2,10,9)
8	(3,9,8)

It is easy (although tedious) to show that by adding the block (6, 8, 10) or the blocks (6,8,10) and (3,5,9) we form a winning game for player 1 provided his first move is to point 2.

Finally we show that there are infinitely many such crystal games by showing explicitly how to construct arbitrarily long structures.

<u>Theorem 2</u>: The structure consisting of B blocks where B is even and $B \leq 8$ is a crystal game generator where block b is (b+1 - 2*E(b), b+3 - 2*E(b), b+2 + 2*E(b)) and E(x) is defined to be 0 if x is odd, 1 otherwise. The last three blocks are exceptions and have values (B-3,B-1,2), (B,2,1), and (B-1, 1,4).

> PROOF: Let us first enumerate some of the blocks: The X-tal is a finite piece of our old friend figure 19, bent to form a circle and joined properly so as to be perfectly symmetric. Let us assume that player 1 moves to point 4. If player 2 moves to any points other than 1, 2, 3, 5, or 6, then player 1 to point 3, player 2 to point 2 (or loses), and player 1 forks with point 1. It is easy to see that player 1 can also win if player 2 moves to points 2, 5, or 6. If player 2 moves to points 1 or 3, player 1 can generate a perfect crystal and this crystal has the property that player 1 moves to every even numbered point.

Block Number	points on block
1	(2,4,3)
2	(1,3,6)
3	(4,6,5)
4	(3,5,8)
5	(6,8,7)
6	(5,7,10)
7	(8,10,9)
8	(7,9,12)
9	(10,12,11)
.	.
.	.
.	.
B-3	.
B-2	(B-3,B-1,2)
B-1	(B,2,1)
B	(B-1,1,4)

To form a game from this X-tal such that player 1 can always win, we simply add a block on 3 of the even numbered points. To be safe, we can place this block away from point 4 so as not to interfere with things unless a crystal develops. Actually, it is not very difficult (a long case analysis) to show that we can add blocks to all the even numbered points and no problems arise for player 1.

We have seen that crystals are often a subtle but intrinsic element in a large class of generalized TTT games for which player 1 has a win. This element has algorithmic significance since once the crystal begins, it is very easy to perform the calculations which give the resulting game pattern. From a theoretic point of view, we know almost nothing in general about the form of structures which allow such crystals to develop. This is especially true when the number of points per block is greater than 3 as seed generation becomes a very complex matter.

CHAPTER 4 - LEADS

We have seen previously, situations arise during play of a generalized TTT game where a player forces every move of his opponent. This strategic element was especially important in some games such as those played upon figure 7.

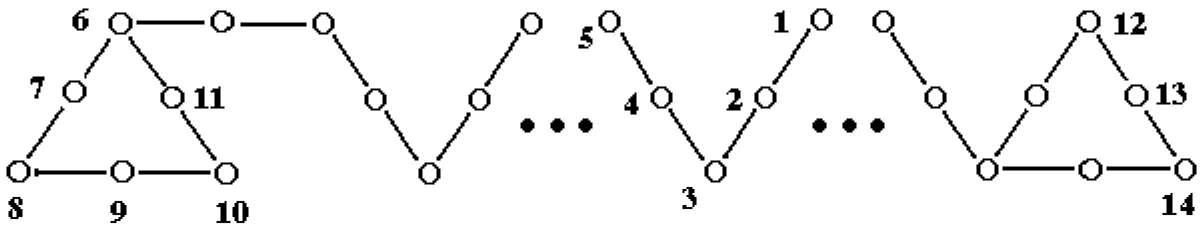


FIGURE 7

<u>Definition</u>: A **lead** is a sequence of moves where one player forces every move of his opponent.

Note that every crystal is a lead but not every lead a crystal. The games described in Theorem 2 shows that the special kind of lead known as a crystal is essential to the win on some games and a strategy can not be developed which allows us to only use leads which are not crystals. However, from an algorithmic point of view, we may be able to ignore the fact that one's own moves are being forced as the lead progresses. Generally speaking, when we have 3 points to a block, it is easy to replace pieces of a lead with finite pieces of the crystal given in figure 19. The important features of the lead are:

1. The end point of the lead is reached by a sequence of moves such that the opponent's moves were harmless.
2. Certain important points are taken as the game progressed. Sometimes these points are useful later in the game in setting up other leads or for the creation of forks.

With lambda restricted to 1, the simplest lead structure appears in figure 23. To start the lead we need to have 2-in-a-row if every block has 3 points. Player 1 moves to point 6. There exists two duplicate structures, the (1,2,3), (2,5,6), (1,4,6) blocks and the (6,7,8), (8,9,11), (6,10,11) blocks. Player 1 gets that "extra move" to create his lead since player 2 must move to one of the duplicate structures, leaving one of them free. The lead is of length 1 and then a fork can be made. Figure 24 shows that we must be careful in how we combine those triangular shaped duplicate structures to form a win. Player 1 can not force a win on the game given by figure 24.

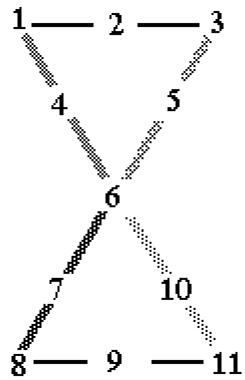


FIGURE 23

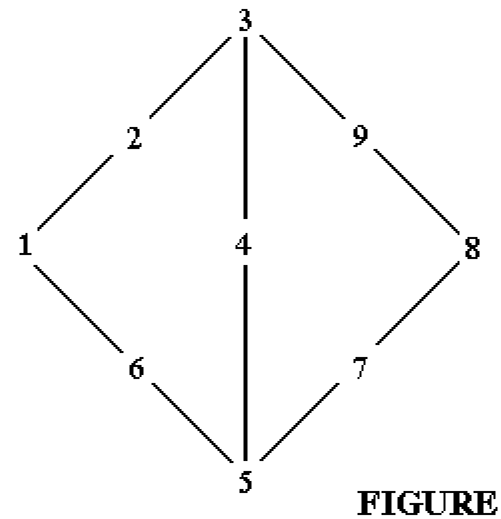


FIGURE 24

One would like to be able to say that we need only look for leads during the play of a generalized TTT game to assure a win. For one thing, they greatly restrict the size of the game tree that needs to be searched in order to show that player 1 has a win. In general, it is the fact that one's opponent has many alternative responses that prevents us from proving that the first player can always win a particular game. This is why, for instance, it is not known if the first player has a win on the 4-cube game. Unfortunately, we still have to contend with seed generation, a very non-trivial problem for situations where the number of points per block is greater than 3. Even if seed generation were not a problem, we have seen games such as that described by figure 15 where the duplicate structures alone seem to be the most significant element of the strategy and the leads don't really enter into the picture at all. Also, if we define the duplicate structures to be a set of blocks upon which the second player must make a move or lose, then leads can be thought of as duplicate structures with one of the structures a single block. Still, leads are an important strategic element from the practical point of view and we shall pursuit the matter a little further, if only primarily to indicate some limitations even for the case where every block contains exactly 3 points.

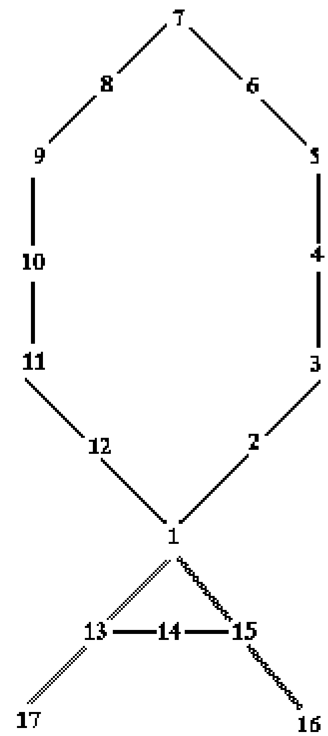


FIGURE 25

Figure 25 shows that even for situations where we can win using a lead, it is not necessarily the fastest win, that is, the win that requires the fewest number of moves. Suppose player 1 goes first to point 1. Now if player 2 wishes to delay the game as long as possible he will move to one of the points 13, 14, 15, 16, or 17 since otherwise he will lose on a game that takes exactly 7 moves. Suppose player 1 leads player 2 using the sequence 3, 5, 7, 9, and 11. This game takes 13 moves to complete. On the other hand, if he moves to point 7, he threatens on the structure (1,2,3), (3,4,5), (5,6,7) and the structure (7,8,9), (9,10,11), (11,12,1). Player 2 may force player 1 to block him on the structure that uses points 13, 14, 15, 16, and 17 but this delays the game only 2 moves and player 2 must eventually play to the right or the left of a line between points 1 and 7. Thereafter, player 1 uses a lead to win the game. In such a case, the total number of moves taken is only 11. This proves that

Proposition: If player 1 tries to win as quickly as possible and player 2 delays his loss as long as possible, and the game can be won using leads to reach a fork, then using such a strategic element does not necessarily guarantee the fastest win.

Next we show that even worse, we can relinquish certain victory by trying to use leads instead of giving one's opponent great freedom in choosing his next point.

Proposition: Generalized TTT games exist with the property that the only way to win the game is to not play any of the leads that exist.

Figure 26 gives an example which proves the proposition. Here player 1 must move to point 1 in order to win. If player 2 is playing to delay the loss, he moves to one of the points 11, 12, 13, 14, or 15. If player 1 starts a lead using 7, 8, 4, or 6 he can not force a win. The correct move is to point 2 followed by a fork at point 8 or point 4.

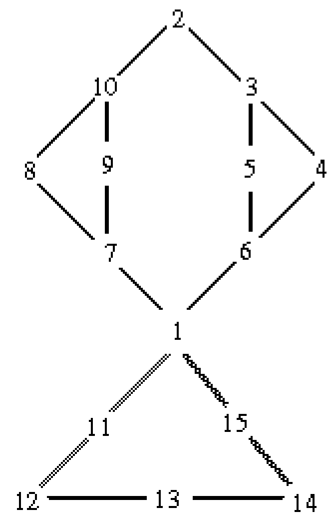


FIGURE 26

Finally, we show that leads are an essential element of an infinite class of generalized TTT games.

Definition: Let the points of a lead be labeled 1, 2, ..., n. A **linear lead** is a lead that for each j = 1, 3, 5, 7, ..., n-1, point j was used with point j+2 to force the other player to move to point j+1.

Definition: A **path between two points** is a collection of blocks on which a linear lead between the points can be made.

Definition: A **multiple path between a point and a non-empty set of points** exists provided two distinct paths exist between the point and (not necessarily distinct) members of the set, i.e. one path contains at least one block that is not on the other path.

Definition: A game after the i^{th} move of player 2 (move 2i of the game) has **property U(i)** or **(property uni-connected after i)** provided that no free point exists, which when played by player 1, would give a multiple path to any other points player 1 has already played.

Definition: A **graph game** is a generalized TTT game with the following properties:

1. lambda = 1
2. molecular length =3
3. every block contains a point that is on no other block

Theorem 3: Player 1 has a win on a graph-game G if and only if G does not have property U(1).

PROOF: Suppose a game does not have property U(1). Then player 1 uses a linear lead to get to that point. Since a multiple path existed, either a fork exists or else he can play a lead so as to get near one of the points on the first lead and form a fork.

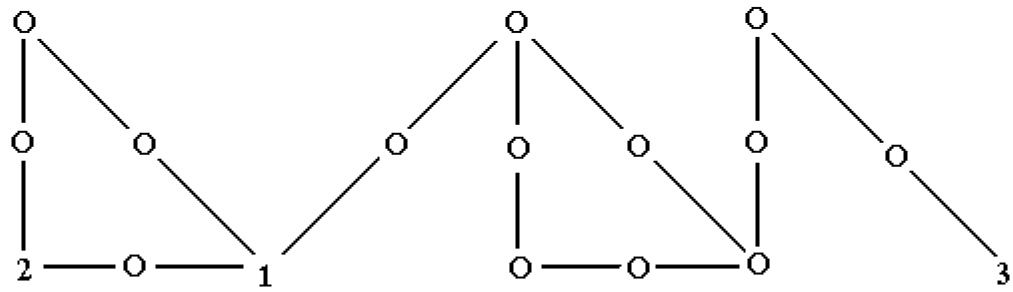


FIGURE 27

In figure 27, player 1 goes to point 1. Even after player 2 goes to point 2, there is a multiple path for player 1 from point 3 to point 1. Note that such play does not give the fastest possible win.

To show the converse, we prove the contra-positive, that is, if G does have U(1) then no win exists. Let the moves of player 1 be denoted by p(1), p(2), ..., p(n) and the moves of player 2 be denoted by q(1), q(2), ..., q(n-1). Since a game with a fork does not have property U and every winning game ends in a fork (if player 2 delays as long as possible), we need only show that U(1) implies U(i) for i=1, 2, ..., n-2.

Suppose U(j) for all j $\leq$k and we add p(k+1). We consider two cases:

1. The multiple path involves only p(k+1) and not p(i) for i = 1, 2, ..., k. If a multiple path exists after q(k+1) then if point p(k+1) played for p(1) at the beginning of the game, this would show that G did not have U(1).
2. The multiple path from point r involves p(k+1) and some p(i) for i = 1, 2, ..., k. Since U(k), we do not have a multiple path from p(k+1) to p(j) for some j $\leq$ k. Let q(k+1) be made to the single path between p(k+1) and p(j) if such a path exists. If another path exists from p(k+1) to r then a multiple path from p(k+1) to p(j) violating U(j) for all j $\leq$ k. Thus we have no path to p(k+1) after q(k+1) moved and thus no multiple path from point r and we have U(k+1). Figure 28 helps to graphically depict the second case.

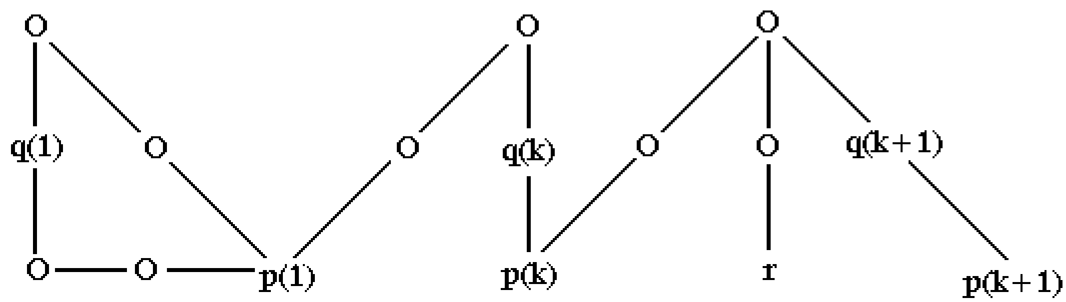


FIGURE 28

The strategy for graph games then, is as follows: we must search all points so that having made a move to p(1) we can find a multiple path to that point regardless of player 2's next move. Next we connect that point with P(1) using a linear lead. Finally, if a fork has not been created, we play on that part of the second path that is distinct from the first part and possibly with the aid of a linear lead, we form a fork.

CHAPTER 5 - DUPLICATE STRUCTURES AND MINIMAL COVERINGS

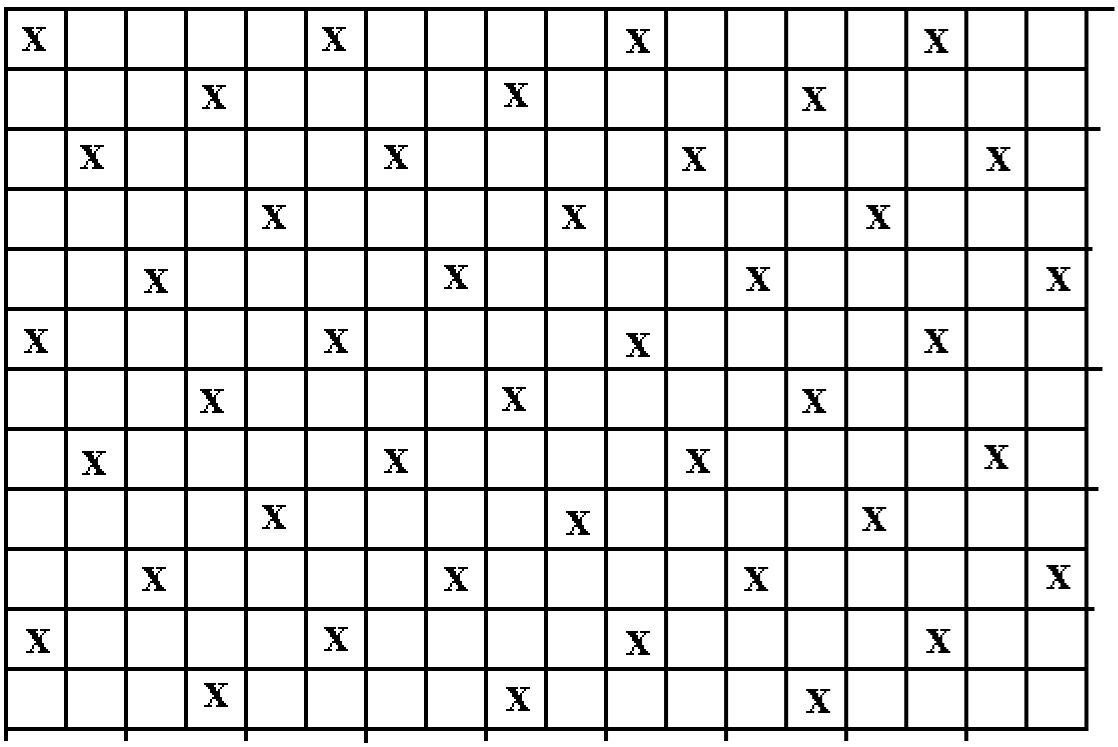
In the previous chapter, we saw that in a graph game, our first point had to be on two collections of blocks so that if our opponent plays on one collection, we can still move to the other one. Then with two points already played on this structure we used leads to force a win. These two structures will be referred to as **duplicate structures**. We attempt to abstract from this observation another strategic element. When we play on duplicate structures, we are in a certain sense, offensively making a (few) point(s) and threatening on some critical collection of blocks. When our blocks contain 3 or more elements, our offensive threats can consist of one or more elements per block. Defensively, we need only one move to nullify the block as a potential win for our opponent. Consider the strategy then, that attempts to occupy the fewest points such that every point is in one of the blocks. Now such a strategy, will not in general provide the best strategy. When however, the situation is very complex and there are many blocks per point (as is often the case when lambda is greater than 1) it can very well happen that our opponent threatens on many fewer blocks than we do so that his chance of finding a winning lead is decreased. Forming a **minimal covering** is then a heuristic that might help us toward a win.

First let's give some minimal coverings for some familiar point/block structures.

Proposition: The minimal covering for TTT is 3.

The validity of this proposition is easy to see once we note that the 3 horizontal blocks have no points in common. Thus the minimal covering is at least three and in fact the 3 points on either diagonal creates such a covering. From a strategic point of view, the value of such a covering is not clear as it can not be obtained without also being a win.

The minimal covering for the 4-cube game is at least 16 since there are 16 non-intersecting blocks. Perhaps a reader with access to a physical model can determine exactly what the covering is. The minimal covering for GO-MOKU is given in figure 29 which shows the portion of an actual board.



Portion of GO-MOKU Board FIGURE 29

<u>Theorem 4</u>: The minimal covering for GO-MOKU is 72.

> PROOF: First notice that every X is exactly 5 points from its neighbor. This distance can not be any greater as 5-in-a-row is the requirement for a win. If another configuration exists which contains fewer X's, then it must look exactly like this one except for possible rotations, reflections, or translations. By examining a matrix completely filled with the pattern indicated by figure 29, you will find that only rotations of 90 degree multiples are possible. You will also discover that translations, reflections, and rotations will not change the number of X's on the playing field.

For GO-MOKU one may get an intuitive feeling that the application of a minimal covering is especially promising because there are many blocks on each point and there is a regularity of structure except at the boarder of the playing field. This strategy was used to great success by the author against other novices. As play progresses, you try to occupy the points that will give a minimal covering. One's opponent often lumps his points together while you create your covering in the vicinity of his moves. As you begin occupying more and more area, you will often find the opportunity to place 2 moves together and begin your offensive attack. One has to have a good repertoire of leads after having occupied 2 adjacent points. These leads can sometimes generate many points for your opponent so you have to be very careful. Even without the advantage of being familiar with many lead plays, a few losses will give one a feeling of when to start the attack.

There are certain problems in applying this strategy. For one thing, after losing several times, one's opponent usually becomes more defensive. Since the minimal covering is strictly determined after 2 moves, there is a grater chance that one's opponent will play to one of the points that your require for your cover. Even worse, he may begin playing these points with a vengeance. In practice this doesn't seem to be much of a problem. Now it could be argued that any defensive sequence would produce this kind of result for novices like myself and it remains to be seen just how good such a heuristic is. There are also certain subtleties that any implementation on a computer will have to deal with.

The preceding gave a possible use for the minimal covering of a game as a strategic element of a heuristic approach. We can use the idea of duplicate structures to find an upper bound on the minimal number of blocks required for a game with the following properties.

1. lambda = 1
2. molecular length = 4

Proposition: $M_1^*(4) \leq 40$

No one has been able to establish the exact minimum number of blocks required (30).

PROOF OF PROPOSITION: Figure 30 gives half of the blocks for the required structure. To obtain the other blocks, add 38 to all points except 1. This is the first set of duplicate structures and without loss of generality, we say player 1 moves to points 1 and 2. The next duplicate structure consists of blocks 1-10 and 11-20. Without loss of generality, we let player 1 move to point 3. The next pair of duplicate structures are the blocks (1,2,6,7,8) and (3,4,5,9,10). Without loss of generality, let player 2 make a move on the second structure. Then player 1 moves to point 5. Since point 17 is a potential fork, we assume that player 2's next move is there. Player 1 then uses points 15 followed by 20 to create a fork. The structure has 77 points and 40 blocks. After verifying the lambda condition, the proposition is proved.

block number	points
1	(1,5,16,17)
2	(2,3,17,18)
3	(1,4,6,7)
4	(2,7,8,9)
5	(3,4,9,10)
6	(1,20,19,15)
7	(3,5,15,14)
8	(2,5,21,20)
9	(2,4,11,12)
10	(1,3,12,13)
11	(1,23,34,35)
12	(2,21,35,36)
13	(1,22,24,25)
14	(2,25,26,27)
15	(21,22,27,28)
16	(1,38,37,33)
17	(21,23,33,32)
18	(2,23,39,38)
19	(2,22,29,30)
20	(1,21,30,31)

FIGURE 30

Another point/block structure with molecular length 4 is the 4-cube game with 64 points and 76 blocks. We can not use 64 however as an upper limit on a minimal point game (lambda = 1) since it is not known if the second player can never prevent a loss under the assumptions of perfect play. Perhaps the reader can find structures which lower either the number of points or number of blocks required to create a game where player 1 has a win and lambda = 1. Another interesting side issue is that of representations for the various structures.

EXERCISE: Let lambda = 1 and player 1 have a win on a 6 block structure. Give representations for such a structure involving 7, 8, 9, 10, and 11 points. Note that every point should be on at least one block.

CHAPTER 6 - CONCLUDING REMARKS

In the preceding chapters, we have seen how the basic strategic elements of crystals, leads, duplicate structures, and minimal coverings can be used in strategies for playing generalized TTT. Crystals and leads allow for deep game tree probes when trying to prove that a sure win exists but did not always give the required strategy for such a win. Duplicate structures are so general that they are probably only useful as a step toward finding other strategic elements. Minimal coverings are a promising direction in some games but very little is known about the class of games for which this heuristic is useful. It should also be noted that it may be very difficult to determine what a minimal covering is for an arbitrary point/block structure.

One common means of measuring the complexity of a class of problems is by determining the relationship between the size of a member of the class and the time (number of instructions) or space needed by a computer to solve the problem. For example, to play a crystal game with n blocks (where n is even and n $\geq$ 8) we make an initial move and then determine if player 2 has responded to a nearby point. If not, our win takes place quickly on one of a very small number of blocks near our initial move and the number of such blocks is independent of the value of n. If player 2 plays elsewhere, we can generate a crystal to win. Although we must be able to respond to the points involved as the crystal proceeds, we need only recognize blocks with 2 and 3 in a row. Thus the time needed to calculate our response will involve the execution of C*n instructions where C is some constant. Since C*n is a polynomial in n, we say that our procedure for calculating a winning strategy on a crystal game is a polynomial time algorithm.

Although more complex, graph games also turn out to have a winning strategy that can be calculated in polynomial time (i.e. belongs to the class P). There is another class of problems (the class NP) that are much more difficult to solve because the function that relates problem size and solution time grows at least exponentially. It is clear that P $\leq$ NP and it is generally felt that the converse, NP $\leq$ P does not hold. In fact, this is a well-known unsolved problem. A problem Q in NP is called NP-Complete if P=NP would follow from Q being a member of P. Cook (31) showed that the satisfiability problem for Boolean formulas was NP-Complete. NP-Complete problems are generally considered to be computationally intractable.

We feel that with any significant relaxation of the constraints on graph games, the resulting class of TTT type games becomes NP-Complete, i.e. the question of whether or not player 1 can win in such a game and the determination of his strategy if he can win are NP-Complete problems. As evidence of this, we prove the following theorem:

Theorem 5: If lambda is unrestricted, then there is a class of generalized TTT games that are NP-Complete.

> PROOF: We show equivalence to the problem of a Boolean expression being satisfiable. For every Boolean variable, we create a 2 point block with points to correspond to the variables of the expression and their negations. We form more blocks by taking all combinations of one variable from each of the factors.

> Suppose a win exists. We can assume it is not on one of the first collection of blocks since the win is immediate and we assume player 2 tries to delay his loss as long as possible. If all points on a block taken, then this gives the values of the variables that will satisfy the Boolean expression.
>
> Conversely, if a Boolean expression has a solution, we play the 2 element blocks to get a representation of the values for the variable and then this combination that satisfies the expression will produce a win on the block that represents that choice of values for the variable.

As a result of the preceding theorem, we make the conjecture that the complexity of generalized TTT games is not decreased by requiring lambda to be 1. One might even suspect that the above theorem precludes the possibility that a simple mathematical theory will ever be discovered for such games. If this is indeed the case, then computer programs that play generalized TTT games will probably have to rely heavily on heuristics. We view much of the material found in chapter 2 through 5 as a basis for the design of such heuristics.

BIBLIOGRAPHY

NIM - THEORETIC

1. "Nim, a game with a complete mathematical theory" by Charles L. Bouton in Annals of Mathematics, Series 2, Volume 3, pages 35-39, 1901-1902
2. "A generalization of the Game called NIM" by S. H. Moore in Annals of Mathematics Series 2 11, pages 93-104, 1910
3. "A new System for Playing the Game of Nim" by D. P. McIntyre in the American Mathematical Monthly Volume 49, pages 44-46, 1942
4. "Matrix Nim" by John Holladay in American Mathematical Monthly, Volume 65, Number 2, pages 107-109
5. "Compound Games with Counters" by Cedric Smith in Journal of Recreational Mathematics, Volume 1, number 2, pages 67-77, April 1968
6. "How to triumph at Nim by playing safe, and John Horton Conway's game 'Hackenbush'" by Martin Gardner in Scientific American 226; pages 104-107, January 1972

NIM - MECHANIZATION

7. "The Nimatron", E. U. Condon in American Mathematical Monthly, Volume 49, number 5, pages 330-332, May 1942
8. "A machine for playing the game NIM" by Raymond Redheffer in American Mathematical Monthly, Volume 55, number 6, pages 343-350, June-July 1948
9. "Digital Computer plays NIM", Herbert Koppel in Electronics November 1952
10. "Win at Nim with Debicon", Harvey Pollack in Popular Electronics January 1958

TIC-TAC-TOE

11. "Tic-Tat-To" by Alain C. White in British Chess Magazine, July 1919
12. "GO-BANG" by Professor Hoffman (Angelo Lewis) in the Book of Table Games, pages 599-603, George Routledge and Sons. Ltd., 1894
13. "Hyper-Spatial Tit-Tat-Toe or Tit-Tat-Toe in Four Dimensions" by William Funkenbusch and Edwin Eagle in National Mathematics Magazine, Volume 19, number 3, pages 119-122, December 1944
14. "Design of a Tit-Tat-Toe Machine" by R. Haufe in Electrical Engineering, Volume 68, page 885, October 1949
15. "The game of Tick-Tack-Toe" by Harry D. Ruderman in Mathematics Teacher, Volume 44, pages 344-346, 1951
16. "Tick-Tack-Toe Computer" by Edward McCormick in Electronics, Volume 25, number 8, pages 154-162, August 1952
17. "Games of Alignment and Configuration" by M. J. R. Murray in A History of Board Games Other Than Chess, Chapter 3, Oxford University Press, 1952
18. "Ticktack Automaton" in Life 32:76 May 5, 1952
19. "Board Games" by Geoffrey Mott-Smith in Mathematical Puzzles, Chapter 13, Dover Publications Inc. 1954
20. "Scarne on Teeko" by John Scarne, Crown Publishers Inc. 1955

21. "Relay Moe Plays Tick Tack Toe" by Edmund C. Berkeley in Radio-Electronics, December 1956
22. "Tic-Tac-Toe Mate" by David D. Lockhart in Popular Electronics, November 1958
23. "Neon lamp logic gates play tic-tack-toe" by C. E. Hendrix and R. B. Purcell in Electronics 31: pages 68-69, June 20, 1958
24. "Ticktacktoe" by Martin Gardner in Scientific American Book of Mathematical Puzzles and Diversions, Simon and Schuster, New York, 1959
25. "Tic-Tac-Toe with an adult twist" by G. G. Dahlem in Recreation 54:523, December 1961
26. "Regularity and positional games", by A. W. Hales and R. I. Jewett in Transactions of the American Math. Soc., pages 222-229, 1963
27. "Electronic Tick-Tack-Toe in a Cigarette Box" by B. Bawer and W. J. Hawkins in Popular Science 197, pages 78-79, December 1970
28. "Tictacktoe and its complications" by Martin Gardner in Scientific American 225, pages 102-104, August 1971
29. "Can you beat tic-tac-tronix" by Don Lancaster in Radio-Electronics, 42: pages 32-35+, December 1971
30. "On a Combinatorial Game" by Paul Erdös and J. L. Selfridge in Journal of Combinatorial Theory (B) 14, pages 298-301

MISCELLANEOUS

31. "The design and analysis of computer algorithms" by Aho, Hopcroft and Ullman, Addison-Wesley, 1974
32. "Lasker's How to Play Chess" by Emanuel Lasker, Bell Publishing Company, Drexel Hill, Pa. Pages 25-29
33. "Artificial Intelligence: The Heuristic Programming Approach" by James R. Slagle, McGraw Hill Book Company, 1971
34. "Randomness and Mathematical Proof" by Gregory J. Chaitin in Scientific American Volume 232, number 5, pages 47-52, May 1975
35. "A new sub-field in Computer Chess" by Hans Berliner in SIGART Newsletter, August 1975
36. "Isomorphs of Tic-Tac-Toe" by Herbert A. Simon, CIP paper #90, Carnegie-Mellon University, June 10, 1966
37. "Jonathan Livingston Human" by Richard Bach in New York Times, page 31, July 18, 1975
38. "The solution of a Simple Game" by Daniel Cohen, in Mathematics Magazine, pages 213-216, September-October 1972

Last Modified: April 7, 2005