

Artificial Intelligence in Game Theory

An Expository Paper

P. Coulton *

Department of Mathematics
Eastern Illinois University

Abstract

We consider some primary forms of artificial intelligence as they apply to sophisticated games:

- memory retrieval
 - situation analysis and building the data base
 - maximizing expectation and minimizing risk
 - strategic functions
-

*Author, E-mail: cfprc@eiu.edu

1. Memory Retrieval

- Managing the Data Base

Some games are easily analyzed. These games usually have a small number of possible outcomes or the rules of the game allows one player to force a win or draw. Tic-Tac-Toe is one such game. It is well known that this game leads to a forced draw. Let the first player be denoted by X and the second player by O . There are basically three cases:

- player X begins in the center

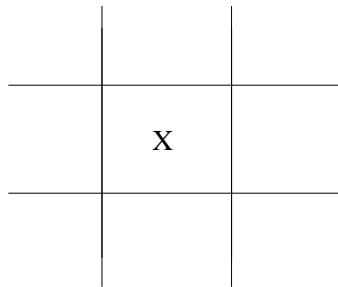


Fig.1 Case 1

In this case player O must choose one of the corners. If X does not choose a corner and O blocks then X is reduced to blocking wins for O . The second play for X should be in a corner. If the play is in the opposite corner from O , then O must again choose a corner.

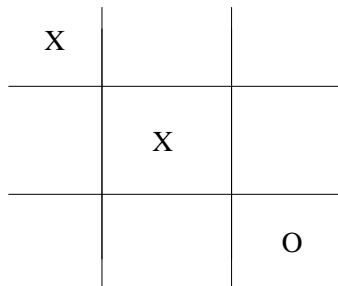


Fig.2 O to play

This forces X to block, which forces O to block. X has only one more possible win which is easily blocked win which O must block

X		O
O	X	X
		O

Fig.3 O to play

X has only one more possible win which is easily blocked.

X	X	O
O	X	X
	O	O

Fig.4 Classic Draw

If the second play for X is not in the opposite corner, O simply blocks all attempted wins.

- The first play is on a corner.

X		

Fig.5 Case 2

If O takes the middle and X plays the opposite corner then O must not play a corner.

X		O
	O	
		X

Fig.6 *X* to play and win by blocking

If *O* chooses a side, then a draw is forced by optimal play.

- The first play is on a side. *O* should take the middle. Every attempt for a win is easily blocked.

	X	

Fig.7 Case 3

X	X	O
	O	

Fig.8 Case 3

There are two straight forward ways to solve this as an algorithmic draw. One is to use the above analysis to write the algorithm that forces a draw. In addition, one could include algorithms that would take advantage of a win situation. This is essentially a situation analysis.

Another approach would be to represent all possible games with a tree diagram. The algorithm then selects the forced draw paths along the tree diagram. This second method is a memory or data base solution. Simple games are certainly prone to this sort of solution, but complex games can also make effective use of this approach.

We say that a game is finite state game if the rules of the game only provide for a finite number of final states. Chess is a finite state game provided that we include the three identical position rule. Without this condition, chess is not a finite state game, but we may qualify this by changing the rule such that a game with more that 100 moves (on each side) is called a draw. Under this rubric we may treat chess as a finite state game with somewhat simpler analysis.

It is clear that every finite state game can be analyzed using a tree diagram. The problem with many such games is that the tree diagram is so complex that it poses a problem with respect to memory allocation and retrieval.

In particular, many chess playing machines take advantage of huge data bases of the standard openings. Clearly, fast reliable methods of allocation and retrieval are critical to the work of artificial intelligence algorithms.

For instance, if one could map all possible games of chess, (actually it would probably only be necessary to map those games which are well played in some sense) it might be possible to determine the truth of the following conjectures:

1. White can force a win.
2. White can always force a draw.
3. There is no forced win for Black.

It seems unlikely that white could force a win, but it is clear that some standard positions are so well known and studied that a forced win is an accepted fact. A good defense would tend to avoid these positions. It is clearly possible to demonstrate that certain position (in particular, end game positions) lead to forced wins.

It also seems likely that a forced draw by White is tenable as a theory. Chess analysis generally looks for the losing move when White fails to win. The same is not true with respect to draws. The general feeling seems to be that White and Black drift into uninteresting draws.

Conjecture 3 is a corollary to Conjecture 2. This conjecture seems to be very likely true.

The analysis of complexity of the game of chess is generally beyond the ability of the unaided human mind. It seems to me that it is not beyond the ability of artificial intelligence. Some imagine that this means that computers are smarter than humans, but we must remember that the computer is the slave of the design of our algorithm.

2. Situation Analysis

- Building the Data Base

Assuming that our finite state game is too complicated to analyze using a tree diagram system, we must be able to write an algorithm which can evaluate the present state and nearby future states. The algorithm must be capable of determining the optimal future states and make a choice among those states. Let us return to the problem of the Tic-Tac-Toe game.

1	2	3
4	5	6
7	8	9

Figure 2.1 Positional choices

We may write a game out as follows:

(5, 1, 9, 3, 2, 8, 4, 6, 7)

The result looks like

o			o		o	o	x	o	o	x	o
	x			x			x		x	x	o
					x		o	x		o	x

Figure 2.2 A forced draw

Observe that a draw is forced by optimal play, i.e. blocking losses, after (5, 1, 9, 3). In this case, it is only necessary to memorize the sequence (5, 1, 9, 3) to know how to force the draw. Once this position is in memory it is only necessary to write the algorithm which blocks any wins for X . Now consider the game (5, 1, 3, 7, 4, 6, 2, 8, 9)

o			o		x	o		x	o	x	x
	x			x		x	x	o	x	x	o
			o			o			o	o	x

Figure 2.3 Another forced draw

Clearly, a draw is forced by optimal play after (5, 1, 3). Up to permutations, these two cases represent all drawing chances for O if X chooses position 5 on the first move. As an example (5, 2, 1) gives a forced win for X with optimal play. The play goes as (5, 2, 1, 9, 7) with two chances for X .

	o		x	o		x	o	
	x			x			x	
					o	x		o

Figure 2.3 A forced win after (5, 2, 1)

This is a complete situation analysis for drawing as O provided that the first move by X is position 5.

We conclude that for a machine playing O there are only two data sets required to draw if X plays the center on the first move.

Now suppose that X chooses a corner for the first play. In this case, O is forced to select the middle. Since $(1, 2, 5, 9, 7)$ is a forced win, $(1, 3, 7, 4, 9)$ is a forced win and $(1, 9, 3, 7)$ is a forced win. But $(1, 5, 9, 2), (1, 5, 7)$ and $(1, 5, 2)$ are forced draws by optimal play. There are three data sets necessary for drawing as O if the first move of X is in a corner.

If X chooses a side, then O may choose the center and force a draw by optimal play. For instance, $(2, 5, 1)$ is forced, $(2, 5, 6, 1)$ is forced, $(2, 5, 7, 6)$ is forced, $(2, 5, 8, 1)$ is forced. There are three data sets required if the first move of X is one a side.

In total, only eight data sets are required to program a machine to draw as O . All the other data sets can be generated using permutations and simple algorithms. The advantage of this sort of system is that allocation is minimized and retrieval time is minimized.

To build a competent chess playing machine, we would begin by placing all the standard openings and the losing or drawing moves for each opening. Addition important games with major variations could also be included. But after this we must apply principles of situation analysis. The Tic-Tac-Toe game is so simple that the general procedures of analysis can be ignored.

Chess is more complicated. Even a winning position, does not necessarily lead to a win. A calculation of the position by many standard techniques may not take into account the sharp line that must be followed to achieve the result. There are some examples of Queen sacrifices that would lead to casual calculation of a loss. But in some cases, Queen sacrifices lead to forced wins by the sacrificing player. In other cases the development of play indicates that the sacrifice leads to even or better odds for the sacrificing player.

As an example, we take the following values for pieces $Q = 10, R = 5, N = 3, B = 3, P = 1$. Given these values we may easily calculate what might be considered the winning side. In many case such a calculation will give correct results, but usually only if positions are relatively even.

A second calculation must be based on the relative positions of pieces, including hanging pieces, potential pins, potential spears, and so forth. Then the question arises, what relative weight do we give to each calculation? Is the relative weight fixed or dependent on the number of pieces on the board?

Third, we must consider the calculation of a future position when there are forced moves.

3. Expectation and Risk Analysis

- Decision Theory

As an example of the importance of expectation in game theory, consider the following well-known problem: A game show host gives a contestant a choice of three doors. Behind two doors is a goat, and behind one door is a million dollars in prizes. After the contestant has chosen, the host opens a door with a goat and offers the contestant the opportunity to switch. Should the contestant switch?

This problem appeared in a column by Marilyn Vos Savant. But her solution caused some controversy with several different opinions by mathematicians as to the correct solution. In part, the dilemma lies in the fact that the problem is not well-posed. This is actually a common difficulty in mathematics texts. Often the problem is posed in such a way that the conditions of the problem can be interpreted in a variety of ways.

The unstated assumption in the solution given by Vos Savant is that the host will always show a goat. But other possibilities are that the host may choose to open one of the doors at random or that the host has some other strategy. In fact, on the original *Let's Make a Deal* show the host would use mixed strategies, sometimes enticing the contestant to switch away from a winning door.

A complete solution to this problem must take into account that the host has a consistent strategy. Such a strategy might be understood by considering the history of the host in making such deals. The game to be solved, then, is to maximize the contestant's expectation given some strategy by the host. Let us assume that the host opens the door with a goat with probability $P(G) \geq 1/3$ and opens the door with one million dollars with probability $1 - P(G)$. Let $P(W)$ denote the probability that the contestant has chosen the door with the money and $P(W^c)$ denote the complement. Let $P(M|W^c)$ denote the probability that the host shows the money when the contestant has not chosen the winning door. Then the probability, $P(G)$, that he shows a goat is

$$P(G) = P(W^c)(1 - P(M|W^c)) + \frac{1}{3}, \quad \text{and,} \quad P(M|W^c) = \frac{3}{2}(1 - P(G)).$$

The probability that the host shows a goat when the contestant has not

chosen the winning doors is

$$P(G|W^c) = 1 - P(M|W^c) = \frac{3P(G) - 1}{2}.$$

Then after the door has been opened, the probability that the million dollars is behind the contestant's door is

$$P(W^c|G) = \frac{P(G|W^c)P(W^c)}{P(G)} = 1 - \frac{1}{3P(G)}.$$

Thus

$$P(W|G) = \frac{1}{3P(G)}.$$

The expectation, when the contestant holds onto the door, after a goat is shown, is

$$E_h = \frac{1000000}{3P(G)}$$

and if the contestant switches then the expectation is

$$E_s = 1000000 \left(\frac{3P(G) - 1}{3P(G)} \right).$$

From this analysis, we have that the expectation equalizes when $P(G) = 2/3$. In other words, the contestant should switch if the hosts shows a goat with greater probability than $2/3$. Now consider the contestant with mixed strategy such that the contestant switches with probability q . Then the expectation is

$$E(q) = 1000000 \left(\frac{1 + q(3P(G) - 2)}{3P(G)} \right).$$

If $P(G) > 2/3$, then the expectation is maximized for $q = 1$. If $P(G) < 2/3$, then the expectation is maximized when $q = 0$. We conclude that mixed strategies do not work in this case.

Let us return to our consideration of our Tic-Tac-Toe game. Let us assume that we wish to analyze the expectation for X , given that we know that O chooses play at random. If X plays position 5 on the first move, then any play into a corner keeps draw possibilities alive. Thus the probability of O choosing correctly is $1/2$.

On the other hand, if X plays a corner, say position 1, on the first move, then O must play position 5 to maintain drawing chances. The probability of O choosing correctly is $1/8$.

The question naturally arises: which strategy for X maximizes expectation?

Assuming that X always plays to stop possible wins by O , then the best expectation

$$E_1 = 383/384$$

is obtained by playing the strategy (1, 5, 2, 3, 7, 4, 6). In this case, one can easily construct the best possible machine. In fact, for this strategy the player wins 191/192 games and draws the other game. We remark that by starting play in position 5 the maximum expectation decreases to

$$E_5 = 382/384$$

with an optimal strategy given by (5, 1, 2, 8, 4, 6, 3, 7).

A game like chess is considered too complicated to calculate all possible expectations. At the same time it should be possible to calculate expectations for many simple situations and it appears to be only a question of time before analysis solves the problem of chess for some of the standard openings. I would venture to claim that the analysis we have provided for Tic-Tac-Toe can be eventually applied to give a complete analysis of chess, even though there are far too many cases to analyze it appears that many of these cases can be eliminated from considerations.

One might begin by giving an analysis of positions commonly held to be draws. The next step would be to analyze one of the standard variations such as the Ruy Lopez. Of course, once this program is carried out for White, it should be extremely difficult for any human to win against a machine playing White.

As our analysis of Tic-Tac-Toe shows it is not necessary to consider all possible variations to find an extremely successful strategy.

4. Strategic Functions

As we have seen before, it is possible to construct machine that appear to think and respond to a given situation with a rational and appropriate decision.

Most of the intelligent tasks that we would like computers to be able to handle can be put in the form of a game. The question is: can a computer

respond in some non-rational ways which could act to open possibilities that the programmer could not foresee in the predetermined algorithms?

One important approach is the concept of a learning machine. As an example of this, consider a machine that is given the rules for the Tic-Tac-Toe game and plays both X and O initially using random placement in each case. The machine learns to play the game by keeping an empirical record of strategies along with the win percentage. For this game there are exactly 9 factorial possible games if we do not take symmetries into account. Even so, this will easily lead to a machine that is very efficient in winning. But to accomplish this, the machine must build an accurate empirical record. This is essentially how humans learn, though in general they do not have an exact empirical record, they can quickly learn which opening moves lead to losses.

For chess players such empirical methods are used in skittles games. The complexity of chess with respect to standard memory constraints seems to reduce the general applicability of methods of this sort for solving the strategy problem with current technology. On the other hand, it is certainly possible to use this approach during a game to explore multiple variations from a fixed position. Such strategies may appear to give the machine the ability to choose irrational moves, even though it has calculated an expectation several moves ahead that will justify the move. This is precisely the case when a major piece is sacrificed to obtain a forced check mate.

Another task that appears to be non-algorithmic in nature is the ability to complete mathematical proofs. Much work has been done in this direction to dispel this prejudice. Let us take a simple example of the algorithmic nature of certain proofs. Suppose we wish to prove the set identity

$$(A \cup B)^c = A^c \cap B^c,$$

where A^c denotes the set compliment. This statement is logically equivalent to

$$\langle \text{not} \rangle (p \langle \text{OR} \rangle q) = \langle \text{not} \rangle (p) \langle \text{AND} \rangle \langle \text{not} \rangle (q).$$

Each logical statement can be realized as function

$$f_i : GF(2) \times GF(2) \rightarrow GF(2), i = 1, 2$$

where $GF(2) = \{0, 1\}$ is the field of order 2. We will identify 0 with false and 1 with true. Then the statements are logically equivalent if the functions are equal. We will define the operation $\langle \text{OR} \rangle$ operation by

$$p \langle \text{OR} \rangle q = p + q + p \cdot q.$$

and the $\langle AND \rangle$ operation by

$$p \langle AND \rangle q = p \cdot q.$$

We will identify the operation $\langle not \rangle$ with complementation in $GF(2)$.

Then

$$f_1(0,0) = (0 + 0 + 0)^c = 1, \quad f_1(0,1) = (0 + 1 + 0)^c = 0,$$

$$f_1(1,0) = (1 + 0 + 0)^c = 0, \quad f_1(1,1) = (1 + 1 + 1)^c = 0.$$

On the other hand

$$f_2(0,0) = (0)^c \cdot (0)^c = 1, \quad f_2(0,1) = (0)^c \cdot (1)^c = 0,$$

$$f_2(1,0) = (1)^c \cdot (0)^c = 0, \quad f_2(1,1) = (1)^c \cdot (1)^c = 0.$$

In other words we have proved the algebraic identity

$$(*) \quad (p + q + p \cdot q)^c = p^c \cdot q^c$$

on $GF(2)$. We can see that a simple set of calculations prove the theorem. On the other hand there are theorems that rely on sophisticated analysis to obtain the proof. Consider Urysohn's Lemma which requires the construction of a continuous function using limit supremums and limit infimums.

Now since there are 4 elements in the domain of a function on two truth variables, there are exactly 2^4 distinct associated functions for these variables. In general, there are 2^n elements in the domain a function on n truth variables and so there are 2^{2^n} associated functions for these variables. It is easy to prove all possible results for logical questions involving n truth variables provided that n is sufficiently small to make the calculation in a reasonable time.

Once we have an idea of the proof of the theorem on complements for finitely many sets we also have an idea of how it may be proved by induction.

Namely, let

$$f_1^n(p_1, \dots, p_n) = f_1^{n-1}(p_1, \dots, p_{n-1}) + p_n + f_1^{n-1}(p_1, \dots, p_{n-1}) \cdot p_n,$$

and let

$$f_2^n(p_1, \dots, p_n) = f_1^{n-1}(p_1, \dots, p_{n-1})^c \cdot p_n^c.$$

The theorem follows by induction, if

$$f_1^n(p_1, \dots, p_n)^c = f_2^n(p_1, \dots, p_n).$$

But this is simply the algebraic identity (*) given above.

Another example might involve proving the identity

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}.$$

If $g(n) = 1 + \dots + n$ and $f(n) = n(n+1)/2$ and it is known that $g(n) = f(n)$, for a fixed n then we must show that

$$\frac{n(n+1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2},$$

But this is a simple polynomial that is easily verified.

This raises a further question: How can we write an algorithm which discovers and proves such theorems?

Consider the following problem: Find the formula for

$$\sum_{i=1}^n i^2,$$

in terms of n . We begin by assuming that the formula is a polynomial $h(n)$ in n . We compute the following points

$$h(1) = 1, \quad h(2) = 5, \quad h(3) = 14, \quad h(4) = 30.$$

We know that three points determine a quadratic form and so we start with the first three points:

$$h_2(x) = 1 \frac{x(x-2)}{(1-0)(1-2)} + 5 \frac{x(x-1)}{2(2-1)} = \frac{x^2 - x}{2}.$$

On the other hand, if we take the first four points we have

$$\begin{aligned} h_3(x) &= 1 \frac{x(x-2)(x-3)}{(1-0)(1-2)(1-3)} + 5 \frac{x(x-1)(x-3)}{2(2-1)(2-3)} + 14 \frac{x(x-1)(x-2)}{3(3-1)(3-2)} \\ &= \frac{x^3}{3} + \frac{x^2}{2} + \frac{x}{6}, \end{aligned}$$

which gives the correct relation.

In general, to obtain the correct relation for

$$\sum_{i=1}^n i^k$$

we must calculate the polynomial through $(0, 0)$ and the next k points. A proof of the theorem is then obtained by a verification of the induction hypothesis on this polynomial. An induction proof works if the relation following relation holds for all $n \geq 1$:

$$f(n) = g(n, f(n - 1)),$$

for some choice of $g(x, y)$ and $f(x)$. A classification of many of these relations leads to a list of induction proofs. As we have just seen, the algebraic relation

$$(p + q + p \cdot q)^c + p^c \cdot q^c = 0,$$

or

$$(p + q + p \cdot q + 1) + (p + 1) \cdot (q + 1) = 0,$$

on $GF(2)$ leads to DeMorgan's Law for finite collections of sets. This last relation is a clear tautology.

It is natural to ask the following question: What other simple tautological algebraic relations on $GF(2)$ lead to set theoretic results? Another question: What can be said about the case of DeMorgan's Law for infinitely many sets?

Observe that for the case of infinitely many sets we consider the corresponding Venn diagram. We may decompose the Venn diagram into countably many disjoint sets representing the contiguous region. These regions partition the space. Each of the regions corresponds to a basis element for the vector space of $GF(2)$ -valued functions over the independent variables associated to the given sets. As an example, observe that for three sets there are eighth distinct regions that determine a basis for the vector space of $GF(2)$ -valued functions.

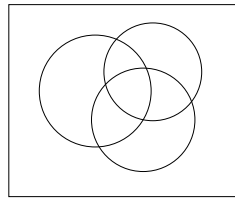


Fig.4.1 Eight distinct regions

For the case of three sets the basis elements are

$$A \cap B \cap C, \quad (A \cup B \cup C)^c, \quad A - B - C, \quad B - A - C,$$

$$C - A - B, \quad (A \cap B) - C, \quad (B \cap C) - A, \quad (A \cap C) - B.$$

In terms of the variables p, q, r , this basis is

$$\begin{aligned} p \cdot q \cdot r, \quad p + q + r + p \cdot q + r \cdot q + r \cdot p + 1, \quad p \cdot (q + 1) \cdot (r + 1), \\ q \cdot (p + 1) \cdot (r + 1), \quad r \cdot (p + 1) \cdot (q + 1), \quad p \cdot q \cdot (r + 1) \\ p \cdot (q + 1) \cdot r, \quad (p + 1) \cdot q \cdot r. \end{aligned}$$

A similar basis can be constructed in the case of an countable number of sets. It follows that there is a one to one correspondence between the $GF(2)$ -valued functions and the real numbers between 0 and 1, even though representations that have repeating zeros at the end and repeating ones at the end give two representations of the same number but are different functions.

Evidently, it is virtually impossible to check all possible functions for solutions to identities. In the case of DeMorgan's Law, it is possible to solve the countable problem using only two cases.

Case 1: Assume that all the variables take the value zero, then we have

$$0 + 1 = \prod_{i=1}^{\infty} 1$$

and the equality holds.

Case 2: Assume that at least one variable, p_1 is 1, then

$$1 + 1 \cdot f_1(p_2, \dots) + f_1(p_2, \dots) + 1 = 0 \cdot (f_2(p_2, \dots) + 1)$$

which holds for all values of the functions.

On the other hand, a proof of DeMorgan's Law only requires that if $A = B$, then a point $x \in A$ if and only if $x \in B$. In other words, we need only check the case that both sides take the value 1. If this is the case, it seems reasonable to accept the result for any collection of sets.

At the same time, it should be clear that in the case of at least countably many sets, one could construct very complicated theorems that could require a case by case approach making it impossible to decide the issue in a finite time.

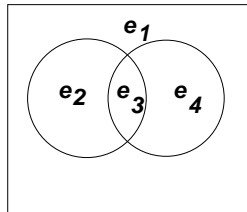


Fig.4.1 Four elements in the basis

Observe that the standard geometry on four point is generated by the set of functions

$$\{e_i\}_{i=1}^4 \cup \{e_i + e_j\}_{i \neq j, i, j=1}^4.$$

Apparently, all the theorems of the geometry can be stated in terms of these functions. The axioms are

- there exist four distinct regions, i.e., e_1, e_2, e_3, e_4 .
- Every pair of regions determine a unique line, i.e. $e_i + e_j$ such that $i \neq j$.
- every line is related to exactly two regions (as above).

We say that two lines $e_i + e_j$ and $e_k + e_r$ are parallel if

$$e_i + e_j + e_k + e_r = 1.$$

Equivalently, we could define two lines as parallel if the sums of the lines is not a line. In this case, the value one represents the fact that a point is in the universal set if and only if it is in the given set.

Theorem 4.1 *For any given line there exists a unique parallel line.*

Proof: Observe that the unique parallel line is determined by

$$e_k + e_r = e_1 + e_2 + e_3 + e_4 + e_i + e_j.$$

This completes the proof.

Clearly, as we move to geometries with a countable number of points, the ability to check the functional relations becomes more difficult. Though in some cases, a strategy is implied by the finite case.

Mathematics and Computer Science
Eastern Illinois University