Game Theory

The study of games goes back at least 300 years, though it became a major topic of study in mathematics only in the last century. The use of the word “games” can be misleading, as the term covers a wide variety of adversarial and cooperative relationships, as found in economic systems and international relations, for example. John Nash received the Nobel Prize in Economics for his work on game theory, and game theory has been heavily used by military strategists for the last 60 years. Recreational games have also been studied extensively, both as models for some other applications and because games are interesting and important in their own right.

Different situations and types of games require different approaches to analysis. We will see a few of these, but by no means all. We begin with some simple games that nevertheless give some insight.

There are 12 stones in a pile. Each of two players in turn may remove 1, 2, or 3 stones. The person who removes the last stone wins, or in an alternate version loses. The question: who wins, and how? Once we have figured this out, we broaden the question to allow any number of stones in the original pile.

How should we approach this problem? 12 is not a very big number, so you may be able to figure this out by trial and error, but we want to develop techniques of analysis that we will be able to use over and over on harder problems. One simple approach that is often very useful is to look at small versions of a problem and hope that a pattern emerges.

So let’s focus on the “last stone wins” rule, and start with very small piles. It’s easy to see that the first player wins when the original pile has 1, 2, or 3 stones, since he may simply take the whole pile. A little thought shows that the second player wins the game with 4 stones, since no matter what the first player does, the second player can take the whole pile and win. Note that no matter what the first player does, the game that remains looks exactly like a game that starts with 1, 2, or 3 stones, and we already know that the first player wins such a game. Suppose now that there are 5, 6, or 7 stones in the pile. The first player can take 1, 2, or 3 respectively to leave exactly 4 stones; we know that the first player in this game (who is really the second player) loses, so the first player wins. In the game with 8 stones, the second player can arrange to leave 4 stones for the first player, so the first player loses. Now the pattern is established: whenever the number of stones is a multiple of 4 the first player loses, and otherwise the first player wins.

To simplify the discussion of two person games, it is normal to refer to the first player as White and the second as Black. Of course, in many games (like the stones game) this has nothing to do with the game, but there are many games in which the players are White and Black and usually White goes first, as in chess.

What about the alternate stones game in which the player who takes the last stone loses? Again we start with small games and hope to spot a pattern. With 1 stone White loses, since White is forced to take the only stone. With 2, 3, or 4 stones White wins, since White can leave just one stone for Black. With 5 stones White loses again, since no matter how many White takes, Black can leave just one stone. The pattern is similar to the first version of the game, but now White loses when the number of stones is one more than a multiple of four and wins otherwise.

Here’s a game that is superficially very much like the first game. Start with a collection of stones in a line. Each player may take 1, 2, or 3 stones, but only if they are immediately adjacent to each other. For example, starting with 6 stones, White might take 3 stones from the middle, leaving 2 adjacent stones on one end and a single stone on the other end. Now Black can take 1 stone from either end or the 2 adjacent stones on one end, but Black cannot take one from each end or all three. Again we might stipulate either “last stone wins” or “last stone loses.”

In the case that last stone wins, this game is always won by White, using a simple strategy. On the first turn, White takes one or two stones from the middle, so that there are two identical groups remaining. On further moves, White does exactly what Black does, but in the other half of the game. No matter what Black does, White can always match the move, so that White must take the last stone. If last stone loses, this game is a little trickier, but essentially the same strategy can be used until the very end of the game, at which point a new strategy must be used to finish the game.

A strategy that consists of mimicking the opponent’s moves is often called a mirror strategy.

A game similar to the preceding two but more complicated is called Nim. Here the game starts with multiple piles of stones. On each turn a player may take any number of stones from a single pile. The most well-known starting configuration is four piles of sizes 7, 5, 3, 1. Again it turns out that “last stone wins” is somewhat simpler to analyze than “last stone loses,” though the strategies do turn out to be similar.

It is possible to examine a series of Nim games with different starting sizes and eventually spot a pattern that gives a winning strategy for all Nim games. It turns out that the final strategy is actually quite simple, so instead of going through the long analysis, we’ll just pull the strategy out of a hat and see that it works.

The strategy depends on binary notation for numbers, also known as base 2. We usually write numbers in decimal notation or base 10, in which the digits represent some multiple of different powers of 10, that is, the number 976 in base 10 means $9 \cdot 10^2 + 7 \cdot 10^1 + 6 \cdot 10^0 = 900 + 70 + 6$. In base two, each “digit,” now called a “bit” represents some multiple of a power of 2. In base 10 the individual digits only go up to 9 = 10 − 1, and in base 2 the bits only go up to $1 = 2 − 1$. So 1011001 in base 2 means $1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 = 64 + 32 + 8 + 1 = 101$. Although 1011001 is considerably longer than 89, it is in some ways easier to work with since only two different bits are allowed; in particular, it is easier to design machines to work with numbers in this form.

Now let’s consider the Nim game 14, 9, 3, 1. We begin by writing these numbers in
binary notation:

\[
\begin{align*}
14 & = 1 1 1 0 \\
9 & = 1 0 0 1 \\
3 & = 1 1 \\
1 & = 1 \\
\end{align*}
\]

Look at the columns in this array of zeros and ones. The first column contains an even number of ones, the second column contains an odd number of ones, the third column contains an even number of ones, and the fourth column contains an odd number of ones. Our goal is to make the number of ones even in each column by taking some number of stones from just one pile. First, find the first column with an odd number of ones; in this example it is the second column. Now pick a row that has a one in this column; in this example, there is only one such row, the first. Now change this row by replacing the 1 in the first column with an odd number of ones with a 0, so now the first row looks like this: 1 0 1 0. Now the first three columns have an even number of ones. To fix the fourth column, change the entry, in this case from 0 to 1. Now the array is

\[
\begin{align*}
11 & = 1 0 1 1 \\
9 & = 1 0 0 1 \\
3 & = 1 1 \\
1 & = 1 \\
\end{align*}
\]

So the actual move we make in the game is to take 3 stones from the pile of 14. Now the second player makes a move, and a little thought should convince you that no matter what the second player does, some column will end up with an odd number of 1s; in particular this means that the second player cannot take the last stone, since then every column would have zero 1s. The first player then follows the same strategy, leaving every column with an even number of 1s. Eventually the last stone must be taken, and it must be taken by the first player, who wins.

Let’s look at the standard Nim game 7, 5, 3, 1:

\[
\begin{align*}
7 & = 1 1 1 \\
5 & = 1 0 1 \\
3 & = 1 1 \\
1 & = 1 \\
\end{align*}
\]

Every column already has an even number of 1s, so in this case the second player can always win by using the same strategy: the first player must leave an odd number of 1s in some column, the second player can restore all columns to an even number of 1s, and so on.

Tree Games

The games we have looked at so far are examples of “tree games”, because they can all be viewed in a uniform way as a single game played on a “tree”. In this context a tree is something like a family tree. Here is a picture of a typical small tree:

Here’s how to play a game on this tree: Put a marker on the top box. Each player in turn moves the marker one level down, to one of the boxes connected by lines to the current box. The game is over when there are no boxes below the marker, when the box containing the marker is a “leaf node” in the tree. In this example the game is over after the second move unless the first player chooses the third move and the second player chooses the first move below that. Then the first player gets to move again, to one of the three boxes in the bottom row.

Who wins? To turn this into a game, each leaf node is marked W, L, or D. When the game is over, if the marker is in a box marked W then White has won; if the box is marked L then White has lost; if the box is marked D then the game is a draw. Thus the same tree can be reused for many different games, depending on how the leaf nodes are labeled.

Any game that can be translated to this format is a “tree game”. For example, consider the single-pile stones game with 5 stones, last stone wins. As a tree this game looks like this:

Here the boxes have been replaced by numbers indicating the number of stones left in the pile. At the first move, for example, White can take either 1, 2, or 3 stones leaving 4, 3, or 2 respectively. I have labeled the leaf nodes with W and L to indicate the winner.

Why would we want to recast a game as a game played on a tree? There are two reasons. If we discover some feature of a game played on a tree, then this feature must also be shared by any game that can be translated to be played on a tree. Thus, instead of proving something over and over about lots of different, specific games, we can do it just once. Second, not all
games have easy strategies. We don’t need the tree diagram to tell us how to play the 5-stones game, but some tree games are hard to analyze. We can sometimes figure out the best strategy for a game by examining the tree diagram. Perhaps more to the point, it is not hard to program a computer to analyze the tree diagram. This is how many game-playing computer programs work.

As an example, let’s pretend we don’t know the result of the 5-stones game and see how we can use the tree to discover who wins and how. Of course, this tree is so small that this can be done with a little thought and trial and error. But we want an approach that will clearly work with large trees—in essence, we want a “bookkeeping” method that will calculate the winning strategy if there is one.

We start with some important information already in the tree: we know exactly who has won at any leaf node. Now we can move up the tree one level at a time to figure out the whole game. That is, we will label every node, not just leaf nodes, with W, L, or D depending on the outcome once the game reaches that node.

Consider first the 1 in the next to last row. From this position there is only one possible move, and White wins. Therefore we can label the node containing 1 with W. Moving up one level, consider the 2. It has two child nodes, one labeled 1W and the other 0L. Now it is important to know whose turn it is, namely Black. Black clearly prefers the 0L node and can move there, so the 2-node should be labeled L, since Black definitely can win if the game ever gets to this point. There are three 1s in this same row, and both are easily seen to be properly labeled by L. Putting in the new labels, the tree looks like this:

Now consider the 3 at the beginning of the next row up. It has three child nodes labeled 2L, 1L, 0W. It is White’s turn to move here, so White can choose to move to the node labeled 0W; hence the 3-node should be labeled W. We can continue this purely mechanical process until the entire tree is labeled. In particular, once the top node gets labeled we will know who can win and how. Here is the final labeled tree:

Since the 5-node is labeled W we know White can win, and we know that a winning move for White is any child node labeled W. In this case there is only one of these, the 4-node, so we know that the only winning move for White is to take one stone.

In general, the label D might also appear. What exactly are the rules for labeling the nodes? A little thought leads to these rules:

• At a node at which it is White’s turn to move, label the node W if some child node is labeled W. If there is no such child node but there is a node labeled D, then label the node D. If all the child nodes are labeled L, then label the node L.

• At a node at which it is Black’s turn to move, label the node L if some child node is labeled L. If there is no such child node but there is a node labeled D, then label the node D. If all the child nodes are labeled W, then label the node W.

As a simple example, consider the small tree from above with labels on the leaf nodes:

Filling in the rest of the boxes, bottom to top, gives this:
So White can win this game by first moving the marker to the rightmost node on level two.

A somewhat easier way to keep track of the labeling is to use 1, −1, 0 in place of W, L, D, respectively. Notice that in this scheme, bigger numbers are better for White and smaller numbers are better for Black. This simplifies the rules from above, which can be restated:

- At a node at which it is White’s turn to move, label the node with the largest value among the labels of its children.
- At a node at which it is Black’s turn to move, label the node with the smallest value among the labels of its children.

This is usually referred to as the mini-max procedure. Here’s the same example as above with the new numerical labels.

```
     1
    / \  \
   0   -1 1
  / \ / \  \\
1   0 -1 1 0 1 -1
```

So we see as before that White can win since the root is labeled 1, and that White should move to the node on the right, also labeled 1.

Another variation once the labels have been changed to numbers is to allow leaf nodes to be labeled with many numbers. A nice way to think about the meaning of a number is as a “payoff”. If the game ends on a leaf node labeled 10, that means White gets 10 dollars; if it ends on a leaf node labeled −7, White loses 7 dollars. The same rules as above can be applied to figure out perfect play in such a tree, that is, take the largest and smallest values at alternate levels. Here’s an example:

```
     8
    /  \
   6   -5 8
  / \ / \  \\
2  0 -5 10 3
```

So now White can win 8 dollars by moving to the right hand node. Then Black can either choose to give White 10 dollars or 8 dollars, and chooses the 8.

In practice, when real games are represented as trees they are enormous, too big even for computers to examine completely. Tic-tac-toe is a game that is too simple to be interesting; as a tree it is small enough for a computer to handle but certainly too big for a human to use. There is one top level node, then nine level two nodes. Each level two node has eight children, and so on. Of course, some of the lines of play end early, so not every game goes all the way to level 10. Still, it’s clear that there are too many nodes for a human to look at. The situation for interesting games is much worse, yet it turns out the basic idea is still useful when designing computer programs to play games.

If the tree is too big to examine, the program can examine just part of it. For example, we could think of the previous tree diagram as representing just part of the tree for some game, so that the nodes that appear to be leaf nodes aren’t really—under each apparent leaf node is a whole subtree of moves. If we can look at the position of the game at each apparent leaf node and estimate how good the position is for White, we can use the estimates to label the nodes. So for example the first leaf node is labeled 6 because we determine somehow that it looks pretty good for White; the next node is labeled 0 because it appears that White and Black are evenly matched at that position, and so on. Once all the apparent leaf nodes have numbers assigned, we use the same process as before to assign the number 8 to the root. The meaning of this 8 is different now—it means that we think there is a pretty good chance that White can win by moving to the right-hand node at the next level. This appears to be the best thing to do, but it relies on our estimate of the values of the positions at the apparent leaf nodes. If those estimates are not very good then neither is the value at the root. For real games like chess, it can be quite difficult to assign a numerical value to a position that accurately reflects the state of the game, but the best chess playing programs use quite sophisticated methods to do exactly that. It is still true that the more levels you can look at the better the final estimate attached to the root seems to be, so the best programs also look at many levels of the tree.

There is one important fact about tree games that really makes it worthwhile to think of them as trees: every tree game, when played without mistakes, always has the same outcome. That is, if each player makes no mistakes, either White wins every time, or Black wins every time, or the game ends in a tie every time. Tic-tac-toe, for example, ends in a tie every time. This is summarized by saying that every tree game has a natural outcome. By proving this fact for games played on trees, we instantly know it is true of any game that can be represented by a tree, like tic-tac-toe or chess, even if we have no idea how to play without making mistakes, as in the case of chess.

In some tree games a tie is impossible, so that we know from the outset that one player or the other can always win. This means we can conceivably show that, for example, Black cannot win, which implies that White always can win, without having any idea how White can win. We will see this for an interesting game called hex.

How could we possibly show that every tree game has a natural outcome? The first step, of course, is to think of every tree game as actually being played on a tree, so that we need not take into account arbitrary sets of rules. For games played on trees, we really already know why this is true: because we know how to label all the nodes with W, L, or D, and we know how to use the labels to play. If the top node is labeled W then we know White can win every time; if it is labeled L then Black wins every time; if it is labeled D, then the game is a draw.
In the example shown, the chosen black stone is marked by an arrow; the stone to its immediate right must be white. Now we simultaneously trace a chain of black stones and a chain of white stones. Imagine standing with your right foot on the black stone marked by the arrow and your left foot on the white stone to its right, so you are facing the center of the board. Now obey the following rule: if the stone immediately in front of you is black, put your right foot on it; if it is white put your left foot on it. By repeating this you trace a chain of black stones with your right foot and a chain of white stones with your left foot. Remember that the board is assumed to be full; in the diagram above only a small number of stones are shown. The white stones shown are the ones that your left foot will trace out as the beginning of the white chain.

What happens? As long as there is a stone in front of you, you can take a step according to the rule, so the only way to get stuck is to end up on an edge facing out. The only alternative is to repeat your steps after some point, but this is impossible—it is not possible to end up with your feet on the same two stones in two different ways. So in fact you eventually get stuck. If you get stuck on the bottom edge then you have discovered a winning white chain, so we need to show that you can’t get stuck anywhere else.

Since we have ruled out the alternatives, you must get stuck on the bottom, and White has won.

Now that we know the outcome of hex is a win for one player or the other, we want to show that in fact White can win. We can do this by showing that Black cannot win.

Hex

Hex is a very interesting game for a number of reasons. It was created in the mid-twentieth century by two different people independently: Piet Hein and John Nash. It is complicated enough to be interesting to play, but simple enough that some analysis is possible. In particular, it is known that White can always win, but it is not known how White can win, except in very small examples of the game.

Hex is a board game, played on a field of hexagons that looks like this:

In the example shown, the chosen black stone is marked by an arrow; the stone to its immediate right must be white. Now we simultaneously trace a chain of black stones and a chain of white stones. Imagine standing with your right foot on the black stone marked by the arrow and your left foot on the white stone to its right, so you are facing the center of the board. Now obey the following rule: if the stone immediately in front of you is black, put your right foot on it; if it is white put your left foot on it. By repeating this you trace a chain of black stones with your right foot and a chain of white stones with your left foot. Remember that the board is assumed to be full; in the diagram above only a small number of stones are shown. The white stones shown are the ones that your left foot will trace out as the beginning of the white chain.

What happens? As long as there is a stone in front of you, you can take a step according to the rule, so the only way to get stuck is to end up on an edge facing out. The only alternative is to repeat your steps after some point, but this is impossible—it is not possible to end up with your feet on the same two stones in two different ways. So in fact you eventually get stuck. If you get stuck on the bottom edge then you have discovered a winning white chain, so we need to show that you can’t get stuck anywhere else.

You can’t get stuck on the left edge, since then both feet would be on black stones. If you get stuck on the right edge then there would be a black chain from the left side to the right side, but in fact there is not. Suppose you get stuck on the top edge, to the left of your original starting point. Since you will be facing out, you will have discovered a white chain from your starting point to an ending point to the left of the black stone you started on, but this can’t happen, since the white chain would have to cross the black chain from the left side to your starting black stone. So to be stuck on the top edge you must end up to the right of your starting position. But then your right foot will have found a black stone in the top row that is farther to the right than your starting black stone, but still connected to the left side by a black chain, and that is impossible. So you can’t get stuck on the top edge either.

Since we have ruled out the alternatives, you must get stuck on the bottom, and White has won.

Now that we know the outcome of hex is a win for one player or the other, we want to show that in fact White can win. We can do this by showing that Black cannot win.
Suppose Black could win. There would be a “strategy”, which you can think of as a huge book of diagrams showing all possible distributions of stones when it is Black’s turn, and the move that Black should make to win in each case. If Black always makes the move that the book says to make, Black wins.

We will describe how White can use this book to win also. But since Black and White cannot both win, this implies that there can be no such book—that is, that there cannot be any winning strategy for Black. Since hex is a tree game that doesn’t end in a draw and doesn’t end in a win for Black, it must end in a win for White.

So how does White use the book designed for Black? White places a stone on the board and then pretends it is not there, so when Black moves White views it as a first move. One way to think of this is that White is playing a “phantom” game in which White is playing second, that is, White is playing the black stones in the phantom game. This way White can use Black’s strategy book to determine a move, then translate that move into the real game. To make the games match up properly, the phantom board is obtained from the real board by flipping it across a diagonal, removing one white stone, and interchanging the colors of the stones. For example, after White and Black have each moved once, the two boards might look like this:

![Real Game vs Phantom Game](image)

The white stone in the phantom game corresponds to Black’s move in the real game. In the phantom game, White looks up the correct response to the opening move in the Black strategy book. White makes the recommended move in the phantom game, then places a white stone in the corresponding place in the real game. For example, after White’s second move in the real game, the two boards might look like this:

![Real Game vs Phantom Game](image)

At some point, the strategy might tell White to place a stone on the hexagon that contains White’s first move, which until now White has been pretending is not there. White can’t place another stone there, since Black would object, so White stops ignoring the stone, pretending to have just placed it there. White still must make a move, so White places a white stone on some open hexagon, and pretends that it isn’t there.

Could the phantom game end before the real game, or even at the same time? That is, could White ever think the phantom game is over? In the phantom game, White is guaranteed to win (playing the black stones) since White is using the winning strategy book. So if the phantom game ends, White sees a winning black chain, which in the real game must be a winning white chain; the “extra” white stone that White has been ignoring has no effect. But this can’t be the case, since in the real game Black has been playing a winning strategy, and so can’t lose.

So in fact the real game must end before the phantom game, and it must end with a winning black chain, since Black has an unbeatable strategy. But this winning black chain will appear in the imaginary game to be a winning white chain, since White is not ignoring any black stones in the real game—they all appear as white stones in the phantom game. But in the phantom game, White has been playing the winning Black strategy, so there cannot be a winning white chain. Thus, the real game cannot be over.

So by assuming that Black has a winning strategy, we find ourselves in a logical bind—none of the possible outcomes can occur. The only way to resolve this is to conclude that Black does not in fact have a winning strategy; then the whole problem disappears.

So there is a winning strategy for White, though we have no idea what it is (except on small boards, where it can be discovered by trying essentially all possibilities). You might expect that the same argument we used above will show that Black can use White’s winning strategy to win, putting us right back in the same bind as before, but in fact this doesn’t work out. Black would have to pretend to be going first, by ignoring White’s first move. But while it was harmless for White to ignore a White move, since that ignored move can’t hurt White, it is not harmless for Black to ignore a White move—the ignored white stone could be part of a winning chain for White that Black wouldn’t acknowledge, but it would be a win for White in reality.

If you are not careful, you might interpret the argument above to mean that White can win with any first move, but this is false. Under the assumption that Black has a winning strategy we proved that White could win with any first move, but that assumption turned out to be false, so any conclusions based on it are false as well. In fact there are two known first moves for White that give the game to Black: the two acute corners of the board. The proof of this is very much like the one above: we assume that White can win after making one of these moves, and we show that Black also can win. Since as before this is impossible, it must be that White cannot win and therefore Black can win.

Suppose White moves to an acute corner, and has a strategy to win with this move. We will show how Black can use the same strategy to win, which as before will imply that White does not have a winning strategy after all.

Black starts by placing the black stone shown in the diagram of the real game below and then ignoring it. Black also plays a phantom game, in which the board has been flipped and the colors interchanged as before, with one exception: White’s original corner move in the real game will also be white in the phantom game. The two games look like this after each player has moved once:
Now whenever White moves in the real game, Black records the move with a corresponding black stone in the phantom game. Black then uses White’s strategy to pick a white move in the phantom game, and plays by putting a black stone on the real board. After one more move by each player, the boards might look like this:

As before, if the strategy ever tells Black to move to the hexagon that contains the ignored black stone, Black stops ignoring it and puts a black stone somewhere else. For reference, here again is the board with the first two moves, and one additional hexagon marked with “∗”.

Suppose at some point Black thinks the phantom game is over, in which case Black has won, since Black is using a guaranteed winning strategy. Then Black sees a white chain from top to bottom in the phantom game, which is really a black chain from left to right, except that the original white stone might be used. In fact, the original white stone must be used, since otherwise there is a real winning black chain, but White has been using a strategy guaranteed to win so this is impossible. But any black chain that uses the original white stone goes through the hexagon marked “∗”, and can use the first black stone played instead of the corner white stone, so again there is a real winning black chain, but that is impossible, since White wins the real game. The upshot is that the phantom game cannot be over. Here is an example of this situation, in which there is a winning white chain in the phantom game and a winning black chain in the real game:

The alternative is that the real game is over but the phantom game is not. This means that White has a winning chain in the real game that Black doesn’t see in the phantom game, so the winning white chain must use the original White move. But because of Black’s first move, this winning white chain must use the hexagon marked “∗”, which means there is a winning white chain that does not use the original corner move, so this looks like a winning chain in the phantom game too, but that is impossible.

We conclude as before that the presumed winning strategy for White does not exist, so if White moves first to an acute corner, Black can win. As before, we don’t know how to do this on large boards.

As before, it may be tempting to conclude that if White moves to the acute corner, Black should move to the hexagon adjacent to it along the black side, but again this is a good move only under the false assumption that White can win. In fact, on the five by five board, this is a losing move for Black—if Black moves next to White then White regains the advantage. It is not hard to see that if White moves to an acute corner, Black should respond by moving to the center hexagon on the five by five board.

**Matrix Games**

Here’s a game, called Matching Pennies. Two players simultaneously put down a penny each; if they are both heads, Player A wins and takes both pennies; otherwise Player B gets the pennies. What can we say about this game? It is certainly not a tree game, since the players go simultaneously, and there is no “best move” in any given circumstance. In fact, we clearly have no hope of giving a strategy that will guarantee a win in any particular game.

Let’s introduce a diagram to represent the game:

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>−1</td>
</tr>
<tr>
<td>T</td>
<td>−1</td>
<td>1</td>
</tr>
</tbody>
</table>

In tree games we adopted the convention that outcomes of games were always from White’s point of view: W meant White wins, L meant White loses. Likewise, in matrix games, we agree that outcomes are from Player A’s point of view. Thus, a 1 in the table means that Player A wins a penny, and −1 means Player B wins a penny.
Suppose \( p \). Note that when \( p = 1 \) we have to assume that A can do a fair flip and that B does not see the flip before moving.

Now suppose the players start a long series of games. Whenever B plays H, B is betting that A will play T. But in fact, looking at all the times B plays H, A will play H about half the time and T about half the time, so the results of the games in which B plays H will be about evenly split. Likewise, the results of the games in which B plays T will be about evenly split. Hence, overall the outcomes will be about evenly split and each player will break even.

Can Player A do better? No. Since Player B can also adopt the flip method, Player B can insure that she will at least break even in the long run, so there can be no strategy that will guarantee Player A a positive payoff in the long run. If both players choose moves by coin flipping, then the game will end up in each of the four boxes about a quarter of the time, and both players break even.

Now let’s change the game a bit. Here’s a new diagram:

<table>
<thead>
<tr>
<th>Player B</th>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>1</td>
<td>−2</td>
</tr>
<tr>
<td>T</td>
<td>−2</td>
<td>3</td>
</tr>
</tbody>
</table>

Now what happens if A moves by flipping the coin? Looking at all the times B plays H, A gets 1 about half the time and loses 2 about half the time, so in the long run A loses 1 for every two plays, or loses 1/2 per play. Looking at the times that B plays T, A gets 3 about half the time and loses 2 about half the time, so in the long run A gains 1 for every two plays, or gains 1/2 per play. Since B has a choice between these, the worst A will do with this strategy is lose 1/2 per play. If B is playing H, A wants to favor H as well. So what if A starts playing H 1/2 of the time?

Now when B plays H, in the long run about 3 times out of 5 A gains 1, and 2 times out of 5 A loses 2. So on average in 5 plays, the result for A is \( 1 + 1 + 1 - 2 = 1 \), so A loses 1/2 per play, better than the 1/2 A was losing before. When B plays T, then in the long run about 3 times out of 5 A loses 2, and 2 times out of 5 A gains 3. On average in 5 plays A gets \( -2 - 2 - 2 + 3 + 3 = 0 \). B is now faced with the choice of breaking even by playing T or gaining 1/2 per play by playing H, so B still prefers to play T, but the difference is much smaller than before.

Suppose A starts playing H even more often, say 3/4 of the time. When B plays H, A sees on average \( 1 + 1 + 1 - 2 - 1 = 1 \) on every 4 plays, or a gain of 1/4 per play. When B plays T, A sees \( -2 - 2 - 2 + 3 - 3 = -3 \) on every 4 plays, or −3/4 per play. Now B prefers to play T, and gains 1/4 per play.

It looks like somewhere between 1/2 and 3/4, the gain or loss to B for the two choices H and T came together and then reversed places. Can we find the fraction at which H and T give B the same result?

Suppose \( p \) is a number between 0 and 1, and A plays H that fraction of the time; we also say that A plays H with probability \( p \). Note that when \( p = 1/2 \) the prospective payoffs to A in the long run were \( p(1 + (1-p)(-2) = (1/2)(1) + (1/2)(-2) = -1/2 \) and \( p(-2) + (1 - p)(3) = (1/2)(-2) + (1/2)(3) = 1/2; when \( p = 3/5 \) the payoffs were \( p(1) + (1-p)(-2) = (3/5)(1) + (2/5)(-2) = -1/5 \) and \( p(-2) + (1 - p)(3) = (3/5)(-2) + (2/5)(3) = 0; and when \( p = 3/4 \) the payoffs were \( p(1) + (1-p)(-2) = (3/4)(1) + (1/4)(-2) = 1/4 \) and \( p(-2) + (1 - p)(3) = (3/4)(-2) + (1/4)(3) = -3/4 \). In general, no matter what \( p \) is, the payoffs to A are \( p(1) + (1-p)(-2) \) and \( p(-2) + (1 - p)(3) \) when B plays H and T respectively. To see when these are equal, we can simply set them equal and solve for \( p \):

\[
\begin{align*}
p(1) + (1-p)(-2) &= p(-2) + (1-p)(3) \\
p - 2 + 2p &= -2p + 3 - 3p \\
3p - 2 &= 3 - 5p \\
8p &= 5 \\
p &= 5/8
\end{align*}
\]

So if A plays H 5/8 of the time, then the payoffs to A are \( (5/8)(1) + (3/8)(-2) = -1/8 \) when B plays H and \( (5/8)(-2) + (3/8)(3) = -1/8 \) when B plays T— it doesn’t matter what B does, the long term payoff to A is \(-1/8\), and this is the best so far for A.

Can A do any better? As it turns out, no. B can do a similar analysis. If B plays H with probability \( r \), what are the payoffs? When A plays H the long term payoff is \( r(1) + (1-r)(-2) \) and when A plays T the long term payoff is \( r(-2) + (1-r)(3) \). For which value of \( r \) are these the same? Although this is not always true, notice that in this case the only difference between these payoffs and the ones above is the \( r \) in place of \( p \). So without doing the calculation over again, we know that if B plays H 5/8 of the time, the long term payoff will be \(-1/8\) no matter what A does. This means that B can force A to give up \(-1/8\) per play, while A can force B to take no more than \(-1/8\) per play. We say that the value of this game is \(-1/8\), since that is the average long term payoff if both players adopt this optimal strategy.

The question is, does this always happen? We saw that when A chose \( p \) so that B’s play didn’t matter, that was also the best payoff for A. Likewise when B chooses \( r \) so that A’s play doesn’t matter, that gives the best payoff for B—that is, the best payoff that B can force. Moreover, the best that A can force and the best that B can force are the same outcome. It turns out that none of this is coincidental. No matter what numbers we put in the game squares, it will always be true that there is a single number that is both the best A can force and the best B can force, and that number occurs when they each play the strategy that makes the opponent’s choice irrelevant.
Here’s a third example:

<table>
<thead>
<tr>
<th>Player B</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
</tr>
<tr>
<td>H</td>
</tr>
<tr>
<td>T</td>
</tr>
</tbody>
</table>

If A plays H with probability \( p \) the payoffs are \( p(1) + (1 - p)(-2) \) when B plays H and \( p(-1) + (1 - p)(3) \) when B plays T. If we set these equal to each other and solve for \( p \) we get:

\[
p(1) + (1 - p)(-2) = p(-1) + (1 - p)(3) \\
3p - 2 = 3 - 4p \\
7p = 5 \\
p = 5/7
\]

The long term payoff is \((5/7)(1) + (2/7)(-2) = 1/7\) when A plays H \( \frac{5}{7} \) of the time.

If B plays H with probability \( r \) the payoffs are \( r(1) + (1 - r)(-1) \) when A plays H and \( r(-2) + (1 - r)(3) \) when A plays T. Now we get

\[
r(1) + (1 - r)(-1) = r(-2) + (1 - r)(3) \\
2r - 1 = 3 - 5r \\
7r = 4 \\
r = 4/7
\]

The long term payoff is \((4/7)(1) + (3/7)(-1) = 1/7\) when B plays H \( \frac{4}{7} \) of the time. Even though A and B have different strategies, the payoff that both can force is exactly the same.

We can picture what is going on with some simple graphs. When A plays H with probability \( p \), A is giving B a choice between two payoffs: \( 3p - 2 \) if B plays H and \( 3 - 4p \) if B plays T. We can graph these payoffs. If B plays H then this graph shows the payoff to B for different values of \( p \):

\[
\begin{align*}
0 & \quad 1 \\
2 & \quad 3 \\
-2 & \quad p
\end{align*}
\]

If B plays T then this graph shows the payoff to B for different values of \( p \):

\[
\begin{align*}
0 & \quad 1 \\
2 & \quad 3 \\
-2 & \quad p
\end{align*}
\]

When A chooses a value of \( p \), B will choose the lower line in this picture, which is sometimes the line H and sometimes T, depending on the value of \( p \). In choosing \( p \) A’s goal is to make the lower of the two lines as high as possible. This clearly happens at the point where the two lines come together, which is why we can find this best value for A by finding the intersection point of the lines.

If we do the same for B’s strategy, we plot \( 2r - 1 \) and \( 3 - 5r \). This time, B knows that A will prefer the upper of the two lines, so B wants to make the upper of the two as low as possible, which again occurs at the intersection.

\[
\begin{align*}
0 & \quad 1 \\
2 & \quad 3 \\
-2 & \quad r
\end{align*}
\]

Remarkably, although the intersection points occur at different places, the height of the two intersection points is the same.
As before, in the first picture B will always choose the lower value, and the largest of those, the highest point on the thick line, occurs when $p = 1$. In the second picture, A will always choose the higher value, and the smallest of those, the lowest point on the thick line, occurs not at the intersection point but at the left end, when $r = 0$. In other words, the point of intersection is not always the lowest point on the upper line or the highest point on the lower line. What is true is that the only time the intersection point fails to be the “right” point to look at is when there is domination: either one column dominates the other or one row dominates the other.

Since the domination case is so easy, we will deal only with the no-domination case. If there is no domination, we want to show that the intersection point in each graph always gives the best strategy, and that the height of the intersection point is the same in both graphs. Here is a completely generic game:

<table>
<thead>
<tr>
<th>Player A</th>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>T</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

It’s quite easy to see the outcome of this game without doing any calculation. Notice that in each row the first number is larger than the second number, $3 > 2$ and $4 > 1$. Since lower numbers are better from B’s point of view, there is no reason for B to even consider playing H; B will play T every time. This means that A is faced with a simple choice: play H and get 2 or T and get 1. Clearly, A plays H all the time. Thus the value of this game is 2 because the outcome on every play is 2. The official phrase for what’s going on here is that “column two dominates column one”.

Suppose we ignore this easy analysis, and try the algebraic analysis we’ve been using. If A plays H with probability $p$ the payoffs are $p(3) + (1 - p)(4)$ when B plays H and $p(2) + (1 - p)(1)$ when B plays T. If we set these equal to each other and solve for $p$ we get:

\[
p(3) + (1 - p)(4) = p(2) + (1 - p)(1)
\]

\[
4 - p = p + 1
\]

\[
3 = 2p
\]

\[
3/2 = p
\]

This doesn’t even make sense: A can’t play H “three halves” of the time.

If B plays H with probability $r$ the payoffs are $r(3) + (1 - r)(2)$ when A plays H and $r(4) + (1 - r)(1)$ when A plays T. Now we get

\[
r(3) + (1 - r)(2) = r(4) + (1 - r)(1)
\]

\[
r + 2 = 3r + 1
\]

\[
1 = 2r
\]

\[
1/2 = r
\]

So this appears to say that B should play H half the time. In that case the payoff would be $(1/2)(3) + (1/2)(2) = 2.5$, but we know B can do better than this by playing T all the time. So our method has failed. Why? It is instructive to look at the graphs. Here they are, from A’s point of view on the left and B’s point of view on the right:

As before, in the first picture B will always choose the lower value, and the largest of those, the highest point on the thick line, occurs when $p = 1$. In the second picture, A will always choose the higher value, and the smallest of those, the lowest point on the thick line, occurs not at the intersection point but at the left end, when $r = 0$. In other words, the point of intersection is not always the lowest point on the upper line or the highest point on the lower line. What is true is that the only time the intersection point fails to be the “right” point to look at is when there is domination: either one column dominates the other or one row dominates the other.

Since the domination case is so easy, we will deal only with the no-domination case. If there is no domination, we want to show that the intersection point in each graph always gives the best strategy, and that the height of the intersection point is the same in both graphs. Here is a completely generic game:

<table>
<thead>
<tr>
<th>Player A</th>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>w</td>
<td>x</td>
</tr>
<tr>
<td>T</td>
<td>z</td>
<td>y</td>
</tr>
</tbody>
</table>

If there is no domination, either $w > x$ and $z < y$ or vice versa. Since the two cases will work out in the same way, we only need to analyze the case $w > x, z < y$. It can’t be the case that $z > w$, because then $y > z > w > x$, so $y > x$, so row two dominates row one. Thus, we know that $z < w$ and $x < y$.

Consider what happens when A plays H with probability $p$. When B plays H the payoff is $pw + (1 - p)z = (w - z)p + z$, and when B plays T the payoff is $px + (1 - p)y = (x - y)p + y$. The first line $(w - z)p + z$ goes from $z$ when $p = 0$ to $w$ when $p = 1$, so it slopes up; the second line $(x - y)p + y$ goes from $y$ to $x$, so it slopes down. Since $z < y$ and $w > x$, the two lines cross, and the intersection point is the highest place on the lower “vec”, so it gives the best value of $p$ for A. Here’s a typical picture:

Suppose the intersection point is at $p = p_0$, that is, $p_0$ is A’s best choice. The payoff to A is $p_A = (w - z)p_0 + z = (x - y)p_0 + y$ no matter what B does. Suppose for example that B plays an $r, 1 - r$ strategy. Then we know that the payoff is also given by

\[
p_0w + p_0(1 - r)x + (1 - p_0)rz + (1 - p_0)(1 - r)y
\]
which must also be \( P_A \). In fact, we can verify this as follows:

\[
\begin{align*}
prw + p_0(1-r)x + (1-p_0)rzz + (1-p_0)(1-r)y \\
&= r(p_0w + z - p_0z) + (1-r)(xp_0 + y - p_0y) \\
&= r((w-z)p_0 + z) + (1-r)((x-y)p_0 + y) \\
&= rP_A + (1-r)P_A = (r + 1 - r)P_A = P_A
\end{align*}
\]

It is important to understand that this is true no matter what \( r \) is.

Now we turn to B’s point of view. B will play H with probability \( p \). When A plays H the payoff is \( w + (1-r)x = (w-x)r + x \) and when A plays T the payoff is \( rz + (1-r)y = (z-y)r + y \). If we graph these we again see that they cross because of the relative values of \( w, x, y, z \).

So the payoff is \( P_B = (w-x)r_0 + x = (z-y)r_0 + y \) no matter what A does. If A plays a \( p \), \( 1-p \) strategy, then the payoff is

\[
prw + p(1-r_0)x + (1-p)r_0z + (1-p)(1-r_0)y
\]

which must also be \( P_B \). As before:

\[
\begin{align*}
prw + p(1-r_0)x + (1-p)r_0z + (1-p)(1-r_0)y \\
&= p(r_0w + x - r_0x) + (1-p)(r_0z + y - r_0y) \\
&= p((w-x)r_0 + x) + (1-p)((z-y)r_0 + y) \\
&= pP_B + (1-p)P_B = (p + 1 - p)P_B = P_B
\end{align*}
\]

This is true no matter what \( p \) is.

So when A plays the \( p_0 \) strategy and B plays the \( r_0 \) strategy, we know that

\[
pr_0w + p_0(1-r_0)x + (1-p_0)r_0z + (1-p_0)(1-r_0)y = P_A
\]

and

\[
p_0r_0w + p_0(1-r_0)x + (1-p_0)r_0z + (1-p_0)(1-r_0)y = P_B
\]

so \( P_A = P_B \), which is exactly what we wanted to know.