Different Kinds of Proof

Henry Towsner

This presentation looks best in full-screen mode.

This presentation looks best in full-screen mode.

If you aren't sure how to do that, you can probably find a command named something like "Full Screen" or "Presentation" in the "View" menu.

Mathematicians are endlessly concerned with *proofs*.

Mathematicians are endlessly concerned with *proofs*.

The purpose of these slides is to provide an introduction to the idea that, even within mathematics, there are *qualitatively different* kinds of proofs.

Mathematicians are endlessly concerned with *proofs*.

The purpose of these slides is to provide an introduction to the idea that, even within mathematics, there are *qualitatively different* kinds of proofs. And perhaps to hint at why the branch of mathematics known as *Proof Theory* has something to say about these different kinds of proofs.

Mathematicians are endlessly concerned with *proofs*.

The purpose of these slides is to provide an introduction to the idea that, even within mathematics, there are *qualitatively different* kinds of proofs. And perhaps to hint at why the branch of mathematics known as *Proof Theory* has something to say about these different kinds of proofs.

It's helpful to introduce a *toy problem*

Mathematicians are endlessly concerned with *proofs*.

The purpose of these slides is to provide an introduction to the idea that, even within mathematics, there are *qualitatively different* kinds of proofs. And perhaps to hint at why the branch of mathematics known as *Proof Theory* has something to say about these different kinds of proofs.

It's helpful to introduce a *toy problem*, a simple example that still illustrates the main idea.

Mathematicians are endlessly concerned with *proofs*.

The purpose of these slides is to provide an introduction to the idea that, even within mathematics, there are *qualitatively different* kinds of proofs. And perhaps to hint at why the branch of mathematics known as *Proof Theory* has something to say about these different kinds of proofs.

It's helpful to introduce a *toy problem*, a simple example that still illustrates the main idea.

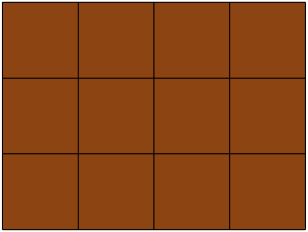
In this case the "toy" part is taken a bit literally.

The example is long, but interesting in its own right.

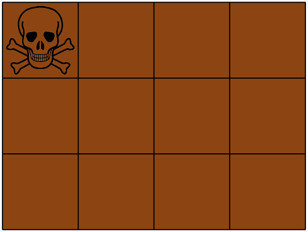
The example is long, but interesting in its own right. It'll take a while, but we'll return to proofs and Proof Theory in the end.

Chomp is a game played on a bar of chocolate

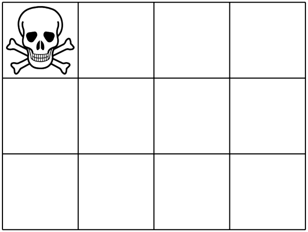
Chomp is a game played on a bar of chocolate that has been divided into a grid



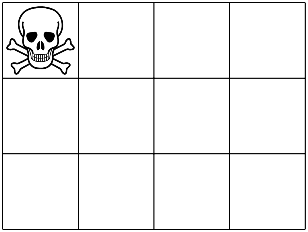
Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!



Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!

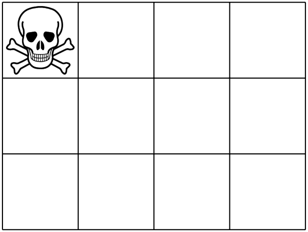


Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!



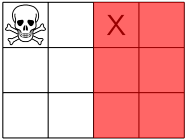
The game is played between two players, who we'll call Alice and Bob.

Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!



The game is played between two players, who we'll call Alice and Bob.
Alice and Bob take turns choosing a square to remove, together with the rectangle of squares *below* or *to the right*.

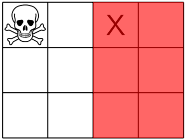
Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!



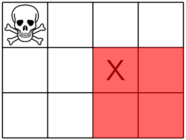
Like this

The game is played between two players, who we'll call Alice and Bob. Alice and Bob take turns choosing a square to remove, together with the rectangle of squares *below* or *to the right*.

Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!



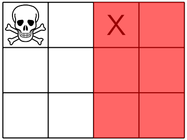
Like this



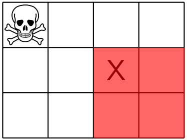
Or this

The game is played between two players, who we'll call Alice and Bob. Alice and Bob take turns choosing a square to remove, together with the rectangle of squares *below* or *to the right*.

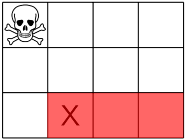
Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!



Like this



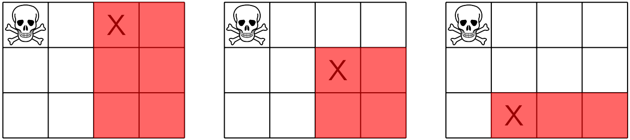
Or this



Or this.

The game is played between two players, who we'll call Alice and Bob. Alice and Bob take turns choosing a square to remove, together with the rectangle of squares *below* or *to the right*.

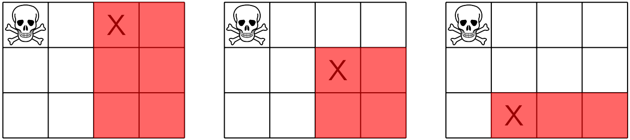
Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!



Like this Or this Or this.

The game is played between two players, who we'll call Alice and Bob. Alice and Bob take turns choosing a square to remove, together with the rectangle of squares *below* or *to the right*. Whichever player is forced to take the poisoned square loses.

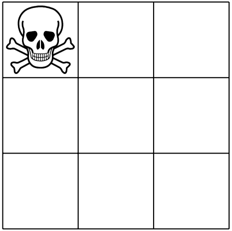
Chomp is a game played on a bar of chocolate that has been divided into a grid and had the square in the top left poisoned!



Like this   Or this   Or this.

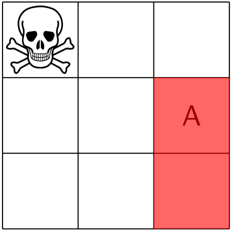
The game is played between two players, who we'll call Alice and Bob. Alice and Bob take turns choosing a square to remove, together with the rectangle of squares *below* or *to the right*. Whichever player is forced to take the poisoned square loses. The bar of chocolate can be divided into many grids of many different sizes, so there are many possible games of Chomp.

Here's an example game, starting with a 3x3 board.



Alice's turn to play

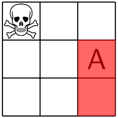
Here's an example game, starting with a 3x3 board.



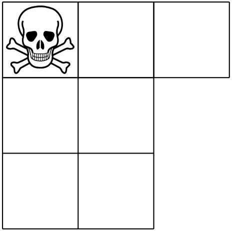
Alice's play

Here's an example game, starting with a 3x3 board.

Recap:



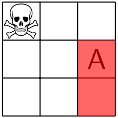
Alice's play



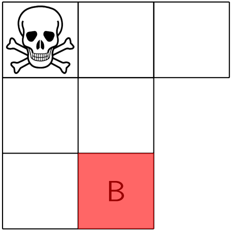
Bob's turn to play

Here's an example game, starting with a 3x3 board.

Recap:



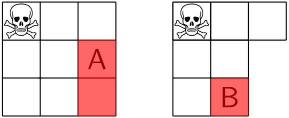
Alice's play



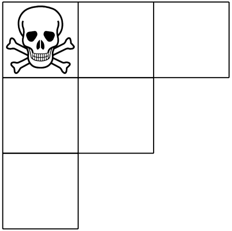
Bob's play

Here's an example game, starting with a 3x3 board.

Recap:



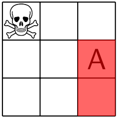
Alice's play Bob's play



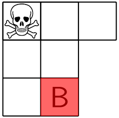
Alice's turn to play

Here's an example game, starting with a 3x3 board.

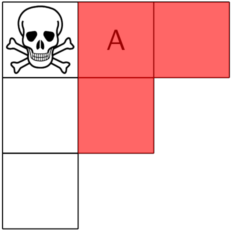
Recap:



Alice's play



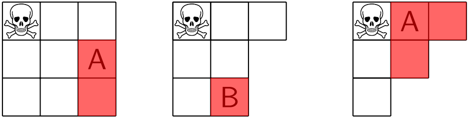
Bob's play



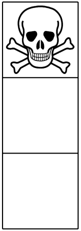
Alice's play

Here's an example game, starting with a 3x3 board.

Recap:



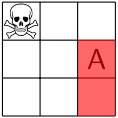
Alice's play Bob's play Alice's play



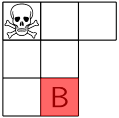
Bob's turn to play

Here's an example game, starting with a 3x3 board.

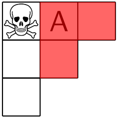
Recap:



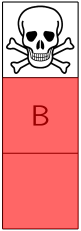
Alice's play



Bob's play



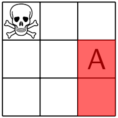
Alice's play



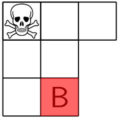
Bob's play

Here's an example game, starting with a 3x3 board.

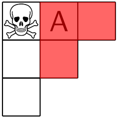
Recap:



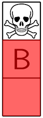
Alice's play



Bob's play



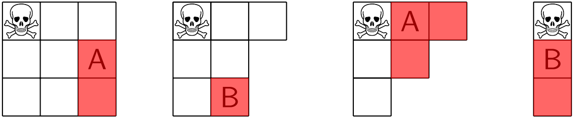
Alice's play



Bob's play

Alice's turn to play

Here's an example game, starting with a 3x3 board.
Recap:



Alice's play Bob's play Alice's play Bob's play

Alice's play

Alice has to move, so she's forced to take the poison piece. Bob wins this time!

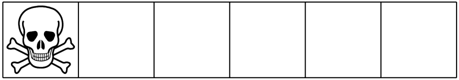
In some games, one player has a *winning strategy*. That means that there's a technique for playing which always works, no matter what the other player does.

In some games, one player has a *winning strategy*. That means that there's a technique for playing which always works, no matter what the other player does.

For example, when N is any integer larger than 1, when Chomp is played on a Nx1 board, the player who goes first always wins.

In some games, one player has a *winning strategy*. That means that there's a technique for playing which always works, no matter what the other player does.

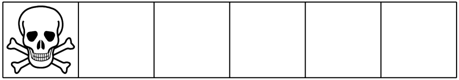
For example, when N is any integer larger than 1, when Chomp is played on a Nx1 board, the player who goes first always wins.



Alice's turn to play

In some games, one player has a *winning strategy*. That means that there's a technique for playing which always works, no matter what the other player does.

For example, when N is any integer larger than 1, when Chomp is played on a Nx1 board, the player who goes first always wins.

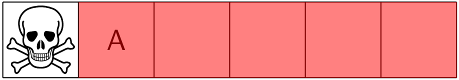


Alice's turn to play

Alice just takes everything other than the poison.

In some games, one player has a *winning strategy*. That means that there's a technique for playing which always works, no matter what the other player does.

For example, when N is any integer larger than 1, when Chomp is played on a Nx1 board, the player who goes first always wins.



Alice's play

Alice just takes everything other than the poison.

In some games, one player has a *winning strategy*. That means that there's a technique for playing which always works, no matter what the other player does.

For example, when N is any integer larger than 1, when Chomp is played on a Nx1 board, the player who goes first always wins.

Bob's turn to play

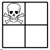
Alice just takes everything other than the poison, forcing Bob to lose on his very first move.

Usually, however, Bob will get a chance to respond to Alice by making his own moves.

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality.

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.



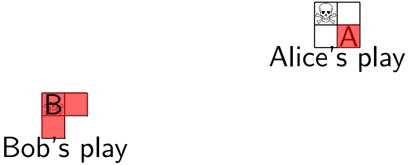
Alice's turn to play

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.

Alice's play

Alice's first move will be to nibble the lower right corner.

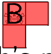
Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.



If Alice is lucky, Bob will eat the poison immediately, handing her the game. . .

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.

Alice's play

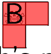


Bob's play

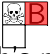
If Alice is lucky, Bob will eat the poison immediately, handing her the game. . .but that doesn't seem very likely.

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.

Alice's play



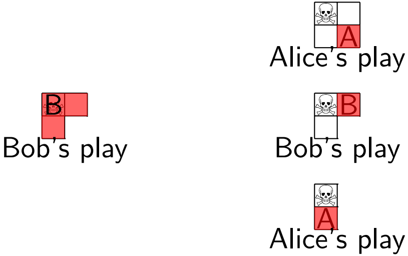
Bob's play



Bob's play

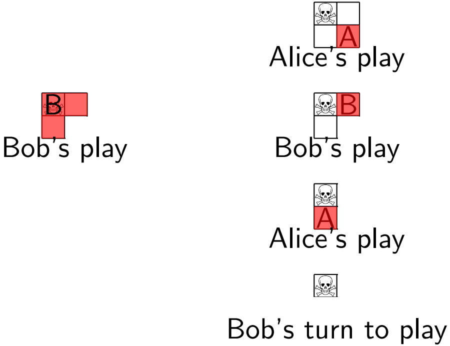
Instead, Bob might play like this.

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.



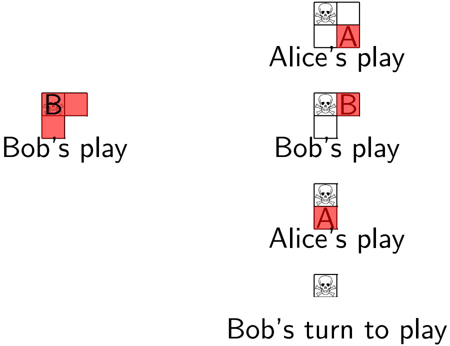
Instead, Bob might play like this. In which case Alice retaliates like this

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.



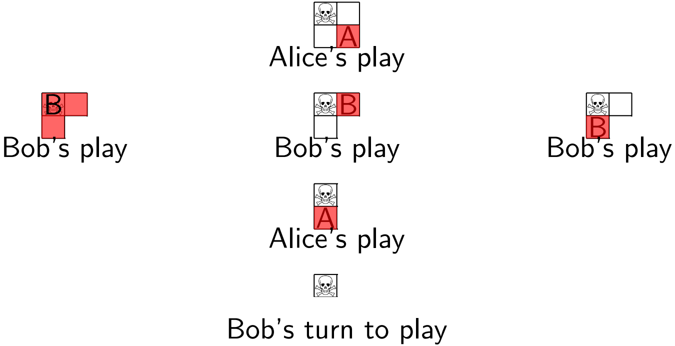
Instead, Bob might play like this. In which case Alice retaliates like this, and Bob is forced to eat the poison.

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.



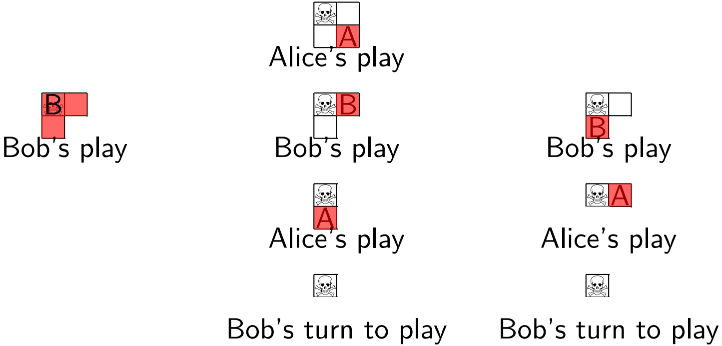
But Bob isn't out of options! Remember that Alice has to be able to defeat *anything* Bob does.

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.



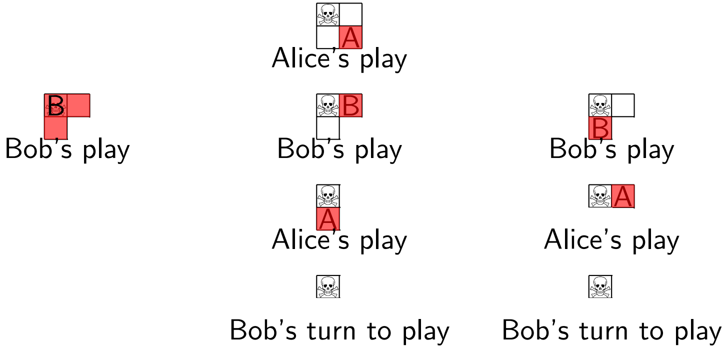
But Bob isn't out of options! Remember that Alice has to be able to defeat *anything* Bob does. What if, instead, Bob plays like this?

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.



But Bob isn't out of options! Remember that Alice has to be able to defeat *anything* Bob does. What if, instead, Bob plays like this? Then Alice responds like this, and once again Bob has to take the poison.

Usually, however, Bob will get a chance to respond to Alice by making his own moves. In order to have a winning strategy, Alice has to be prepared for every eventuality. For instance, on a 2x2 board, Alice still has a winning strategy.



No matter what Bob does, Alice has a response. So Alice has a winning strategy: a rule for playing that guarantees her victory against any opponent.

In order to demonstrate that the player who went first had a winning strategy on a 2x2 board, we exhaustively listed all ways the game could play.

In order to demonstrate that the player who went first had a winning strategy on a 2x2 board, we exhaustively listed all ways the game could play.

An extremely literal person might insist that this is the only way to discuss a winning strategy: that a proper winning strategy must consist of a complete list of all possible moves and the correct response to them.

In order to demonstrate that the player who went first had a winning strategy on a 2x2 board, we exhaustively listed all ways the game could play.

An extremely literal person might insist that this is the only way to discuss a winning strategy: that a proper winning strategy must consist of a complete list of all possible moves and the correct response to them.

That's fine for a very small game, but when we think about bigger boards, that's going to become unwieldy very quickly.

In order to demonstrate that the player who went first had a winning strategy on a 2x2 board, we exhaustively listed all ways the game could play.

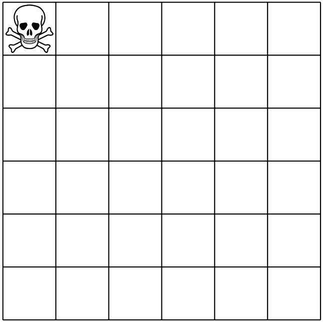
An extremely literal person might insist that this is the only way to discuss a winning strategy: that a proper winning strategy must consist of a complete list of all possible moves and the correct response to them.

That's fine for a very small game, but when we think about bigger boards, that's going to become unwieldy very quickly.

As mathematicians, we're prepared to accept less than that. We'll accept a *proof* that someone has a winning strategy—that is, a rigorous argument that the person can always win—in place of a giant book of moves.

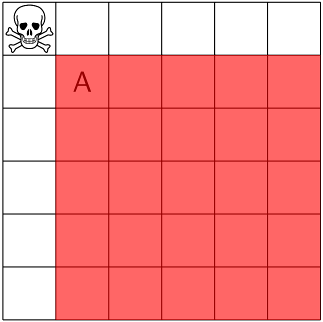
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.

Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Alice's turn to play

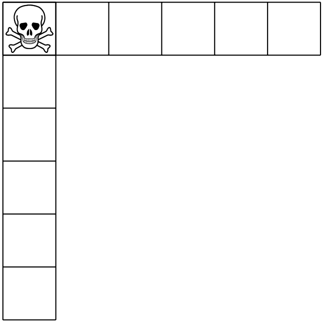
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Alice's play

The first thing to do is to grab the space just below and right of the poison.

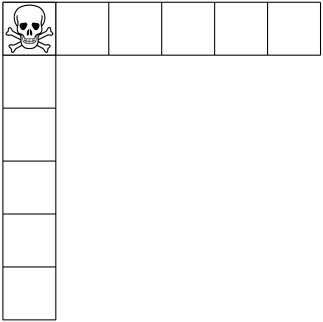
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Bob's turn to play

The first thing to do is to grab the space just below and right of the poison. Bob is left with two long skinny legs to play in.

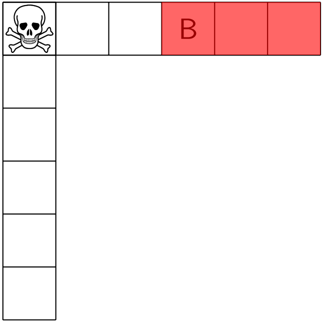
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Bob's turn to play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it.

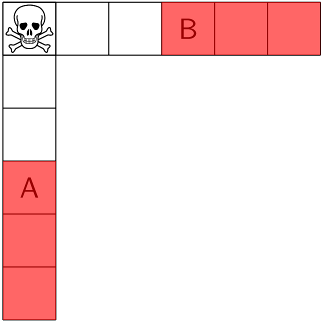
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Bob's play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it. No matter what he does in one leg...

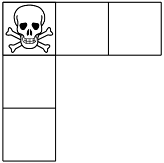
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Alice's play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it. No matter what he does in one leg. . . Alice just copies it in the other leg.

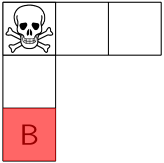
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Bob's turn to play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it. No matter what he does in one leg... Alice just copies it in the other leg.

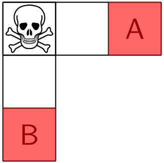
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Bob's play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it. No matter what he does in one leg... Alice just copies it in the other leg.

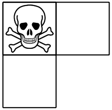
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Alice's play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it. No matter what he does in one leg... Alice just copies it in the other leg.

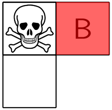
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Bob's turn to play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it. No matter what he does in one leg... Alice just copies it in the other leg.

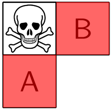
Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Bob's play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it. No matter what he does in one leg. . . Alice just copies it in the other leg.

Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.



Alice's play

The only way to affect both legs is to take the poison, so Bob has to pick one leg to play it. No matter what he does in one leg... Alice just copies it in the other leg.

Let's show that the player who goes first has a winning strategy in every game of Chomp on a *square* board bigger than 1x1.

Bob's turn to play

Alice ensures that both legs get used up at the same time, with Bob stuck taking the poisoned piece in the middle.

So Alice has a winning strategy on every square board.

So Alice has a winning strategy on every square board.

We didn't actually list all the possible moves, but we did something better: we gave a description of Alice's strategy, and (briefly) explained why it always works.

Does Bob ever get to win?

Does Bob ever get to win?

Well, Bob wins on the 1x1 board:

Alice's turn to play

Does Bob ever get to win?
Well, Bob wins on the 1x1 board:

Alice's turn to play

But that doesn't exactly count.

Does Bob ever get to win?
Well, Bob wins on the 1x1 board:

Alice's turn to play

But that doesn't exactly count.
Other than that, the answer is no: there is no size board other that 1x1 where the second player has a winning strategy.

Theorem

There is no size board other that 1x1 where the second player has a winning strategy.

Theorem

There is no size board other that 1x1 where the second player has a winning strategy.

To prove this, we'll use a proof by *contradiction*.

Theorem

There is no size board other that 1x1 where the second player has a winning strategy.

To prove this, we'll use a proof by *contradiction*.

We'll pretend that the second player *does* have a winning strategy, and show that it doesn't work.

Theorem

There is no size board other that 1x1 where the second player has a winning strategy.

To prove this, we'll use a proof by *contradiction*.

We'll pretend that the second player *does* have a winning strategy, and show that it doesn't work.

So suppose Bob announces that he has uncovered a masterful technique that allows him to win all games on, say, a 7x4 board, when he goes second.

Theorem

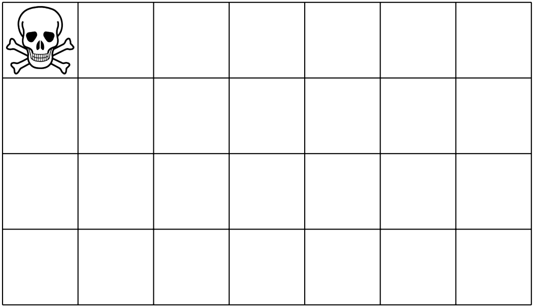
There is no size board other that 1x1 where the second player has a winning strategy.

To prove this, we'll use a proof by *contradiction*.

We'll pretend that the second player *does* have a winning strategy, and show that it doesn't work.

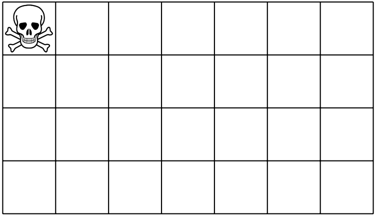
So suppose Bob announces that he has uncovered a masterful technique that allows him to win all games on, say, a 7x4 board, when he goes second.

(The same method will work with any other size board—except for the 1x1.)

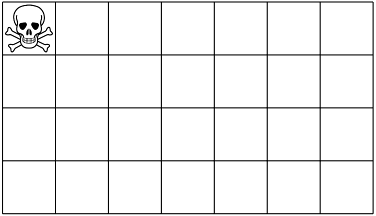


Alice's turn to play

Alice agrees to play against Bob's marvelous strategy.

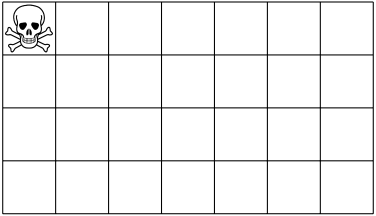


Alice's turn to play

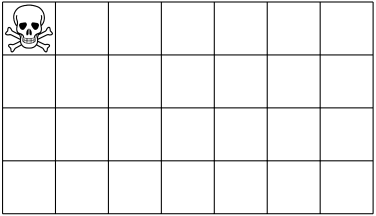


Alice's turn to play

Alice agrees to play against Bob's marvelous strategy. But she knows a trick—she insists on playing two games at once!

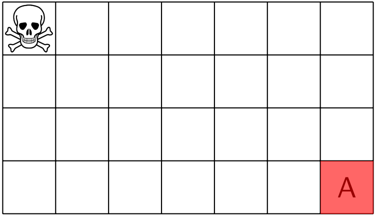


Alice's turn to play

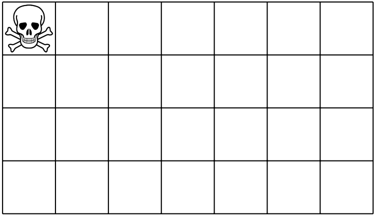


Alice's turn to play

Alice agrees to play against Bob's marvelous strategy. But she knows a trick—she insists on playing two games at once! After all, if Bob's strategy wins every game, it shouldn't be any harder for him to beat her in two games at once than it is to beat her in one.

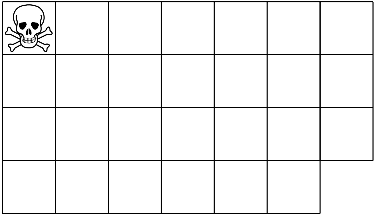


Alice's play

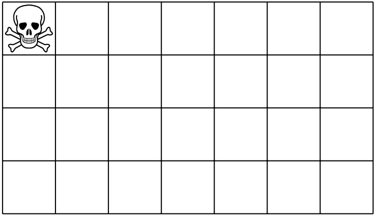


Alice's turn to play

Alice's first move is to nibble the lower right corner in *one* game.

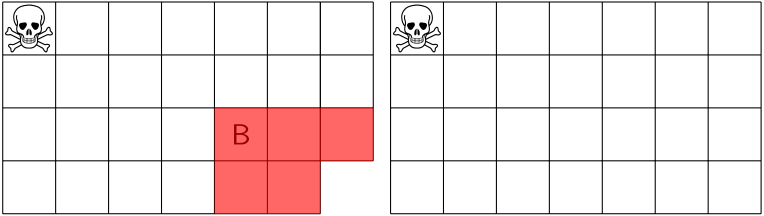


Bob's turn to play



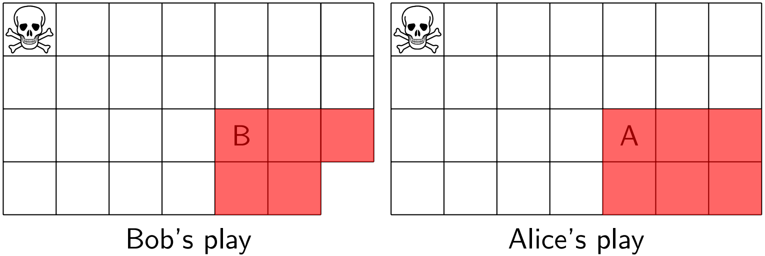
Alice's turn to play

Alice's first move is to nibble the lower right corner in *one* game. She tells Bob to go ahead and make his move in the first game, because she's still thinking about the second game.

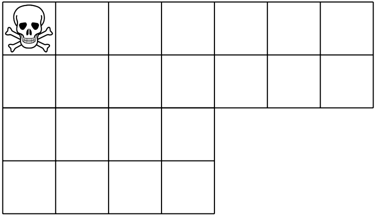


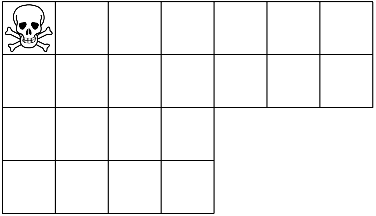
Bob's play Alice's turn to play

Alice's first move is to nibble the lower right corner in *one* game. She tells Bob to go ahead and make his move in the first game, because she's still thinking about the second game. Bob has his master strategy to use, so he ignores the second game and makes his "perfect response" to Alice's move in the first game.

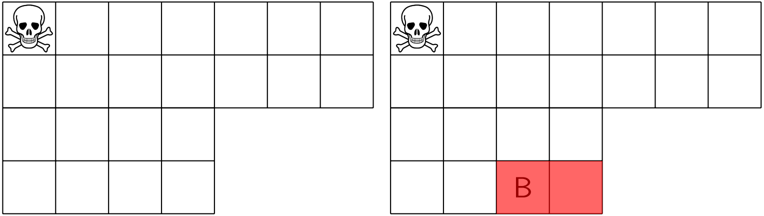


Alice thinks that move looks pretty good, so she makes the *same* move in the *second game*.





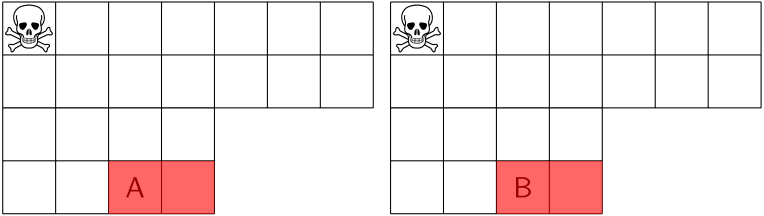
Alice thinks that move looks pretty good, so she makes the *same* move in the *second game*. Now she waits for Bob's response in the second game.



Alice's turn to play

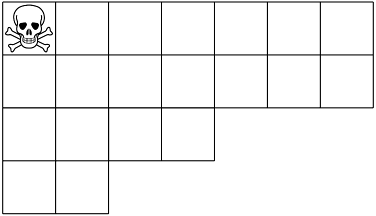
Bob's play

Alice thinks that move looks pretty good, so she makes the *same* move in the *second game*. Now she waits for Bob's response in the second game. And once she gets it,

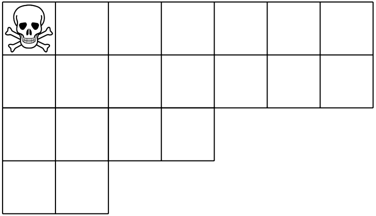


Alice's play Bob's play

Alice thinks that move looks pretty good, so she makes the *same* move in the *second game*. Now she waits for Bob's response in the second game. And once she gets it,She copies it back to the first game.

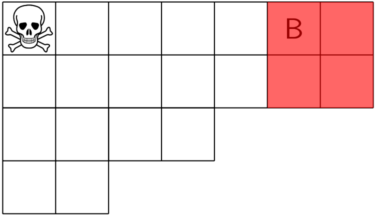


Bob's turn to play

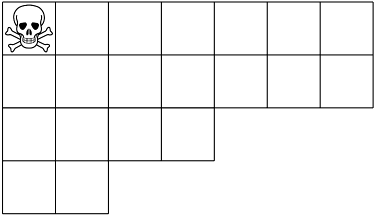


Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other.

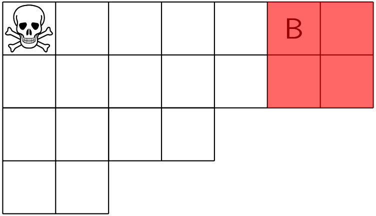


Bob's play

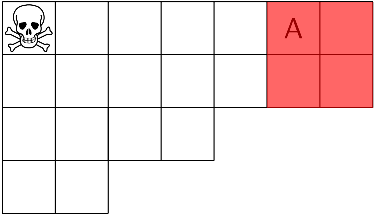


Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other.

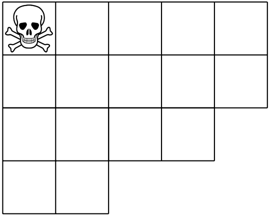


Bob's play

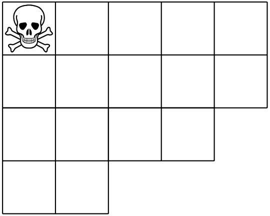


Alice's play

Alice just keeps copying Bob's moves from one game to the other.

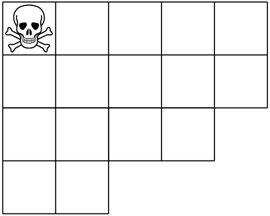


Alice's turn to play

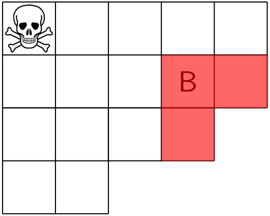


Bob's turn to play

Alice just keeps copying Bob's moves from one game to the other.

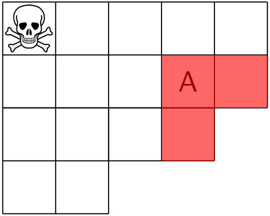


Alice's turn to play

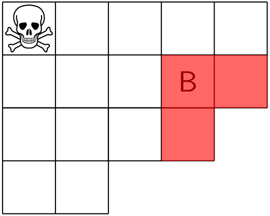


Bob's play

Alice just keeps copying Bob's moves from one game to the other.

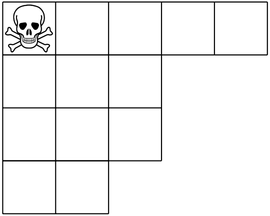


Alice's play

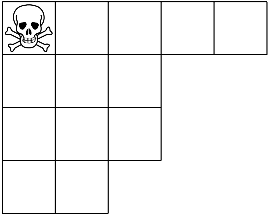


Bob's play

Alice just keeps copying Bob's moves from one game to the other.

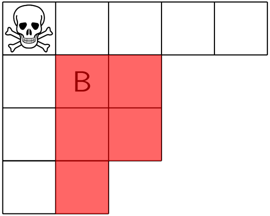


Bob’s turn to play

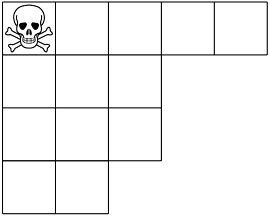


Alice’s turn to play

Alice just keeps copying Bob’s moves from one game to the other.

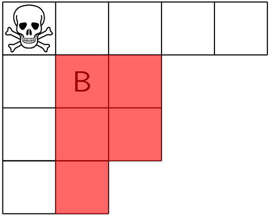


Bob's play

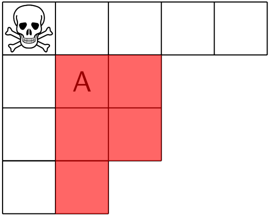


Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

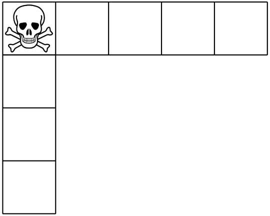


Bob's play

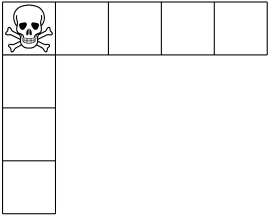


Alice's play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

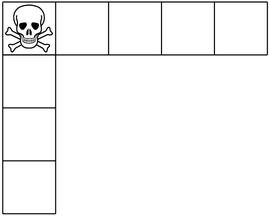


Alice's turn to play

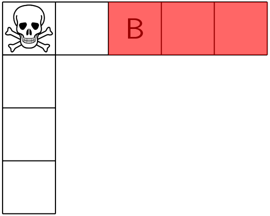


Bob's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

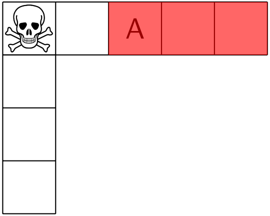


Alice's turn to play

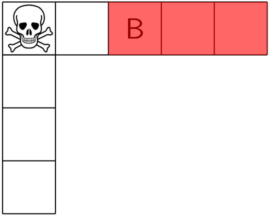


Bob's play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

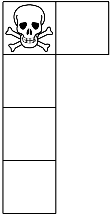


Alice's play

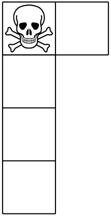


Bob's play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

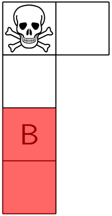


Bob's turn to play

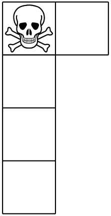


Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

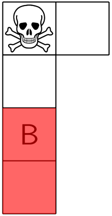


Bob's play

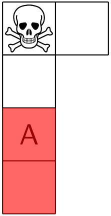


Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

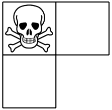


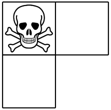
Bob's play



Alice's play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

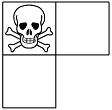


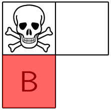


Alice's turn to play

Bob's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

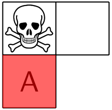




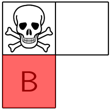
Alice's turn to play

Bob's play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.



Alice's play



Bob's play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

Bob's turn to play

Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

Bob's play

Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

Alice's turn to play

Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy.

Alice's play

Alice's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy. So even though Bob is sure to win in one game...

Bob wins

Alice's play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy. So even though Bob is sure to win in one game. . .

Bob wins

Bob's turn to play

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy. So even though Bob is sure to win in one game. . .

Bob wins

Alice wins

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy. So even though Bob is sure to win in one game. . . he's sure to lose in the other!

Alice just keeps copying Bob's moves from one game to the other. Alice has tricked Bob into playing against his own strategy. So even though Bob is sure to win in one game. . . he's sure to lose in the other!
So Bob's "winning strategy" doesn't always win after all.

This shows that that the second player can't have a winning strategy in any game other than the 1x1.

This shows that that the second player can't have a winning strategy in any game other than the 1x1. A winning strategy should be able to win every single game, and we can guarantee that the first player will win *one* out of a pair of games.

This shows that that the second player can't have a winning strategy in any game other than the 1x1. A winning strategy should be able to win every single game, and we can guarantee that the first player will win *one* out of a pair of games.

Does that mean that the first player always has a winning strategy?

This shows that that the second player can't have a winning strategy in any game other than the 1x1. A winning strategy should be able to win every single game, and we can guarantee that the first player will win *one* out of a pair of games.

Does that mean that the first player always has a winning strategy? The only other possibility is that *neither* player has a winning strategy.

This shows that that the second player can't have a winning strategy in any game other than the 1x1. A winning strategy should be able to win every single game, and we can guarantee that the first player will win *one* out of a pair of games.

Does that mean that the first player always has a winning strategy? The only other possibility is that *neither* player has a winning strategy. Is it possible that on some boards neither player has a winning strategy?

This shows that that the second player can't have a winning strategy in any game other than the 1x1. A winning strategy should be able to win every single game, and we can guarantee that the first player will win *one* out of a pair of games.

Does that mean that the first player always has a winning strategy? The only other possibility is that *neither* player has a winning strategy. Is it possible that on some boards neither player has a winning strategy?

No!

Theorem

Every game of Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Theorem

Every game of Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

The proof of this theorem involves a small detour, so I'll leave it out.

Theorem

Every game of Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

The proof of this theorem involves a small detour, so I'll leave it out. (For completeness, it's included in an appendix at the end of the slides.)

So we now know that:

- In every game of Chomp, either the first player or the second player has a winning strategy, and

So we now know that:

- In every game of Chomp, either the first player or the second player has a winning strategy, and
- Except for the 1x1 board, the second player does not have a winning strategy.

So we now know that:

- In every game of Chomp, either the first player or the second player has a winning strategy, and
- Except for the 1x1 board, the second player does not have a winning strategy.

So the first player must always have a winning strategy!

So we now know that:

- In every game of Chomp, either the first player or the second player has a winning strategy, and
- Except for the 1x1 board, the second player does not have a winning strategy.

So the first player must always have a winning strategy!

What is it?

So we now know that:

- In every game of Chomp, either the first player or the second player has a winning strategy, and
- Except for the 1x1 board, the second player does not have a winning strategy.

So the first player must always have a winning strategy!

What is it?

We proved that there *is* a winning strategy, but we didn't find out what it was!

We've seen two different proofs that Alice has winning strategies for various games of Chomp.

We've seen two different proofs that Alice has winning strategies for various games of Chomp.

For square boards, we actually explained what Alice's strategy was.

We've seen two different proofs that Alice has winning strategies for various games of Chomp.

For square boards, we actually explained what Alice's strategy was. This is called an *effective* proof.

We've seen two different proofs that Alice has winning strategies for various games of Chomp.

For square boards, we actually explained what Alice's strategy was. This is called an *effective* proof.

For arbitrary boards, we proved that Alice has a winning strategy, but we have no idea what it is.

We've seen two different proofs that Alice has winning strategies for various games of Chomp.

For square boards, we actually explained what Alice's strategy was. This is called an *effective* proof.

For arbitrary boards, we proved that Alice has a winning strategy, but we have no idea what it is. This is a *non-effective* proof.

We've seen two different proofs that Alice has winning strategies for various games of Chomp.

For square boards, we actually explained what Alice's strategy was. This is called an *effective* proof.

For arbitrary boards, we proved that Alice has a winning strategy, but we have no idea what it is. This is a *non-effective* proof.

If all we care about is whether or not we should spend our time trying to find a winning strategy for Bob, the difference may not matter.

We've seen two different proofs that Alice has winning strategies for various games of Chomp.

For square boards, we actually explained what Alice's strategy was. This is called an *effective* proof.

For arbitrary boards, we proved that Alice has a winning strategy, but we have no idea what it is. This is a *non-effective* proof.

If all we care about is whether or not we should spend our time trying to find a winning strategy for Bob, the difference may not matter. But if we're about to go play in the National Chomp Championships, knowing there is a winning strategy doesn't do us much good if we don't know what that strategy is.

Non-effective proofs first appeared in mathematics towards the end of the 1800's.

Non-effective proofs first appeared in mathematics towards the end of the 1800's.

When they first appeared, non-effective proofs were *controversial*. Some mathematicians argued that non-effective proofs weren't proofs at all.

Today, almost all mathematicians accept non-effective proofs as mathematically valid.

Today, almost all mathematicians accept non-effective proofs as mathematically valid.

But even if a non-effective proof convices us that something is true, sometimes we'd like an effective proof anyway.

Today, almost all mathematicians accept non-effective proofs as mathematically valid.

But even if a non-effective proof convices us that something is true, sometimes we'd like an effective proof anyway.

Like in the case of Chomp:

Today, almost all mathematicians accept non-effective proofs as mathematically valid.

But even if a non-effective proof convices us that something is true, sometimes we'd like an effective proof anyway.

Like in the case of Chomp:knowing that there exists some winning strategy in principle doesn't give us any clue how to win the game if we actually play it.

The field of mathematics known as *Proof Theory* investigates what we can expect to say about the nature of mathematical proof itself.

The field of mathematics known as *Proof Theory* investigates what we can expect to say about the nature of mathematical proof itself.

Some typical proof theoretic questions we might ask, in light of the examples we have just seen:

The field of mathematics known as *Proof Theory* investigates what we can expect to say about the nature of mathematical proof itself.

Some typical proof theoretic questions we might ask, in light of the examples we have just seen:

- Are there theorems which don't have *any* effective proof?

The field of mathematics known as *Proof Theory* investigates what we can expect to say about the nature of mathematical proof itself.

Some typical proof theoretic questions we might ask, in light of the examples we have just seen:

- Are there theorems which don't have *any* effective proof?
- Alternatively, is there a *systematic way* to turn every non-effective proof into an effective one?

The field of mathematics known as *Proof Theory* investigates what we can expect to say about the nature of mathematical proof itself.

Some typical proof theoretic questions we might ask, in light of the examples we have just seen:

- Are there theorems which don't have *any* effective proof?
- Alternatively, is there a *systematic way* to turn every non-effective proof into an effective one?

In order to answer these sorts of questions, we need to pin down formally exactly what it means to be a proof, and what it means for a proof to be effective.

The field of mathematics known as *Proof Theory* investigates what we can expect to say about the nature of mathematical proof itself.

Some typical proof theoretic questions we might ask, in light of the examples we have just seen:

- Are there theorems which don't have *any* effective proof?
- Alternatively, is there a *systematic way* to turn every non-effective proof into an effective one?

In order to answer these sorts of questions, we need to pin down formally exactly what it means to be a proof, and what it means for a proof to be effective. This is a problem that involves both mathematics—to make the definition precise, and to use it to prove interesting things—and philosophy, to determine whether the formal notions we come up with actually encompass the informal ideas we started with.

The field of mathematics known as *Proof Theory* investigates what we can expect to say about the nature of mathematical proof itself.

Some typical proof theoretic questions we might ask, in light of the examples we have just seen:

- Are there theorems which don't have *any* effective proof?
- Alternatively, is there a *systematic way* to turn every non-effective proof into an effective one?

For instance, it turns out that we could mean several different things by an "effective" proof, and the answers to the questions above depend on which one we mean.

Let's consider three possible ways to try to pin down what makes an effective proof that Alice has a winning strategy different from a non-effective one.

Let's consider three possible ways to try to pin down what makes an effective proof that Alice has a winning strategy different from a non-effective one. An effective proof should always give Alice a method to determine, at each step, what her next move should be. The problem is that we have to decide what counts as a method.

Let's consider three possible ways to try to pin down what makes an effective proof that Alice has a winning strategy different from a non-effective one. An effective proof should always give Alice a method to determine, at each step, what her next move should be. The problem is that we have to decide what counts as a method.

Possibility 1: A "method" is just a computer program: there should be some way to instruct a computer so that Alice can type in the size of the board and what Bob's move is, and the computer will output a winning response for her.

Let's consider three possible ways to try to pin down what makes an effective proof that Alice has a winning strategy different from a non-effective one. An effective proof should always give Alice a method to determine, at each step, what her next move should be. The problem is that we have to decide what counts as a method.

Possibility 1: A "method" is just a computer program: there should be some way to instruct a computer so that Alice can type in the size of the board and what Bob's move is, and the computer will output a winning response for her. This has been the standard interpretation of what a "method" means in mathematics since about the 1930's. (Rather before mathematicians—or anyone—had computers to try this out on.)

Let's consider three possible ways to try to pin down what makes an effective proof that Alice has a winning strategy different from a non-effective one. An effective proof should always give Alice a method to determine, at each step, what her next move should be. The problem is that we have to decide what counts as a method.

Possibility 1: A "method" is just a computer program: there should be some way to instruct a computer so that Alice can type in the size of the board and what Bob's move is, and the computer will output a winning response for her. This has been the standard interpretation of what a "method" means in mathematics since about the 1930's. (Rather before mathematicians—or anyone—had computers to try this out on.)

Let's call this the *computability* interpretation of effectiveness.

Working with the computability interpretation of effectiveness is at the heart of the area of mathematics known as *proof theory*, and the source of its interaction with the area known, naturally, as *computability theory*.

Working with the computability interpretation of effectiveness is at the heart of the area of mathematics known as *proof theory*, and the source of its interaction with the area known, naturally, as *computability theory*.

For example:

- There are theorems which have no proofs which are computably effective, but

Working with the computability interpretation of effectiveness is at the heart of the area of mathematics known as *proof theory*, and the source of its interaction with the area known, naturally, as *computability theory*.

For example:

- There are theorems which have no proofs which are computably effective, but
- There are *kinds* of theorems which always have computably effective proofs, and

Working with the computability interpretation of effectiveness is at the heart of the area of mathematics known as *proof theory*, and the source of its interaction with the area known, naturally, as *computability theory*.

For example:

- There are theorems which have no proofs which are computably effective, but
- There are *kinds* of theorems which always have computably effective proofs, and
- For those kinds of theorems, there is a systematic way of turning non-effective proofs into effective ones, known as *proof mining*.

Working with the computability interpretation of effectiveness is at the heart of the area of mathematics known as *proof theory*, and the source of its interaction with the area known, naturally, as *computability theory*.

For example:

- There are theorems which have no proofs which are computably effective, but
- There are *kinds* of theorems which always have computably effective proofs, and
- For those kinds of theorems, there is a systematic way of turning non-effective proofs into effective ones, known as *proof mining*.

Sadly, there's a catch.

Our non-effective proof that Alice has a winning strategy gives an example.

Our non-effective proof that Alice has a winning strategy gives an example.

The theorem that Alice has a winning strategy for Chomp is the kind of theorem which always has an effective proof.

Our non-effective proof that Alice has a winning strategy gives an example.

The theorem that Alice has a winning strategy for Chomp is the kind of theorem which always has an effective proof.

Proof mining assures us that the proof gives Alice the following winning strategy:

Our non-effective proof that Alice has a winning strategy gives an example.

The theorem that Alice has a winning strategy for Chomp is the kind of theorem which always has an effective proof.

Proof mining assures us that the proof gives Alice the following winning strategy:

- List all possible games of Chomp on the given size board.

Our non-effective proof that Alice has a winning strategy gives an example.

The theorem that Alice has a winning strategy for Chomp is the kind of theorem which always has an effective proof.

Proof mining assures us that the proof gives Alice the following winning strategy:

- List all possible games of Chomp on the given size board.
- The game Chomp is finite, so there are only finitely many possible games, and it is possible to simply write down every possibility.

Our non-effective proof that Alice has a winning strategy gives an example.

The theorem that Alice has a winning strategy for Chomp is the kind of theorem which always has an effective proof.

Proof mining assures us that the proof gives Alice the following winning strategy:

- List all possible games of Chomp on the given size board.
- The game Chomp is finite, so there are only finitely many possible games, and it is possible to simply write down every possibility.
- These moves can be arranged in a "tree", listing all Alice's moves, then for each of her moves, all of Bob's possible responses, and so on. From this tree, it is possible to read off a winning strategy.

That fits the definition of what a computer can do, but it's not very satisfying in this case—the strategy it gives us is "play out all possible future games and pick a move which guarantees victory in all of them". Maybe we should change what counts as a method for purposes of being an effective proof.

That fits the definition of what a computer can do, but it's not very satisfying in this case—the strategy it gives us is "play out all possible future games and pick a move which guarantees victory in all of them". Maybe we should change what counts as a method for purposes of being an effective proof.

Possibility 2: A method for Alice to determine her next move should be an *insightful strategy* which makes use of an actual understanding of the game.

That fits the definition of what a computer can do, but it's not very satisfying in this case—the strategy it gives us is "play out all possible future games and pick a move which guarantees victory in all of them". Maybe we should change what counts as a method for purposes of being an effective proof.

Possibility 2: A method for Alice to determine her next move should be an *insightful strategy* which makes use of an actual understanding of the game.

Let's call this the *insightful* interpretation of effectiveness.

Certainly an insightful strategy is what we'd really like to have.

Certainly an insightful strategy is what we'd really like to have.

But that's not really a definition, since now we have to decide what counts as insightful, which just pushes the problem off one more step.

Certainly an insightful strategy is what we'd really like to have.

But that's not really a definition, since now we have to decide what counts as insightful, which just pushes the problem off one more step. Despite plenty of effort, we don't know any way to formalize what counts as an insightful proof, and it's not clear that we ever will.

Certainly an insightful strategy is what we'd really like to have.

But that's not really a definition, since now we have to decide what counts as insightful, which just pushes the problem off one more step. Despite plenty of effort, we don't know any way to formalize what counts as an insightful proof, and it's not clear that we ever will.

Without a formal definition, we can be happy when we have insightful strategies, but we can't hope to prove abstract theorems about all possible insightful proofs.

One more attempt at formalizing "effective":

One more attempt at formalizing "effective":

Possibility 3: Alice's method should be given by an *efficient* computer program.

One more attempt at formalizing "effective":

Possibility 3: Alice's method should be given by an *efficient* computer program.

Let's call this the *efficiency* interpretation of effectiveness.

Once again, we've just pushed the problem off onto characterizing efficiency.

Once again, we've just pushed the problem off onto characterizing efficiency.

Fortunately, there is a precise notion of what it means for a computer program to be efficient (actually, there are lots of such notions).

Once again, we've just pushed the problem off onto characterizing efficiency.

Fortunately, there is a precise notion of what it means for a computer program to be efficient (actually, there are lots of such notions). Once we fill in which kind of efficiency we mean, we have a notion of effectiveness.

The study of what it means for a computer program to be efficient is the branch of computer science known as *computational complexity*. The common interest in effective proofs leads to an overlap between this area of computer science and proof theory.

The study of what it means for a computer program to be efficient is the branch of computer science known as *computational complexity*. The common interest in effective proofs leads to an overlap between this area of computer science and proof theory.

Unfortunately, the efficiency interpretation of effectiveness doesn't have the same nice properties as the computability interpretation:

The study of what it means for a computer program to be efficient is the branch of computer science known as *computational complexity*. The common interest in effective proofs leads to an overlap between this area of computer science and proof theory.

Unfortunately, the efficiency interpretation of effectiveness doesn't have the same nice properties as the computability interpretation:

- There are theorems which have no efficiently effective proofs, and

The study of what it means for a computer program to be efficient is the branch of computer science known as *computational complexity*. The common interest in effective proofs leads to an overlap between this area of computer science and proof theory.

Unfortunately, the efficiency interpretation of effectiveness doesn't have the same nice properties as the computability interpretation:

- There are theorems which have no efficiently effective proofs, and
- There is no way to reliably tell what they are, and

The study of what it means for a computer program to be efficient is the branch of computer science known as *computational complexity*. The common interest in effective proofs leads to an overlap between this area of computer science and proof theory.

Unfortunately, the efficiency interpretation of effectiveness doesn't have the same nice properties as the computability interpretation:

- There are theorems which have no efficiently effective proofs, and
- There is no way to reliably tell what they are, and
- Even if we think a theorem should have an efficiently effective proof, there's no reliable way to find it.

All three notions of effectiveness are being actively studied.

All three notions of effectiveness are being actively studied.

- The computability interpretation is mostly studied by mathematicians,

All three notions of effectiveness are being actively studied.

- The computability interpretation is mostly studied by mathematicians,
- The insightful interpretation is mostly studied by philosophers,

All three notions of effectiveness are being actively studied.

- The computability interpretation is mostly studied by mathematicians,
- The insightful interpretation is mostly studied by philosophers,
- And the efficient interpretation is mostly studied by computer scientists.

All three notions of effectiveness are being actively studied.

- The computability interpretation is mostly studied by mathematicians,
- The insightful interpretation is mostly studied by philosophers,
- And the efficient interpretation is mostly studied by computer scientists.

The borders among these three areas are soft, though, and all three interpretations have been studied in all three fields.

All three notions of effectiveness are being actively studied.

- The computability interpretation is mostly studied by mathematicians,
- The insightful interpretation is mostly studied by philosophers,
- And the efficient interpretation is mostly studied by computer scientists.

The borders among these three areas are soft, though, and all three interpretations have been studied in all three fields.

As a mathematician, I mostly investigate the computability interpretation of effectiveness.

My own work is in the area of proof mining.

My own work is in the area of proof mining.

Typically, I work on projects like the following:

My own work is in the area of proof mining.

Typically, I work on projects like the following:

- Taking a non-effective proof for which a computably effective proof should exist, and determining what the computably effective proof is

My own work is in the area of proof mining.

Typically, I work on projects like the following:

- Taking a non-effective proof for which a computably effective proof should exist, and determining what the computably effective proof is (in a situation where either we don't care that the result is inefficient—for instance, because it's a theoretical result where there are no efficiency concerns—or where I suspect that the result will happen to be efficient, even though in general it doesn't have to be),

My own work is in the area of proof mining.

Typically, I work on projects like the following:

- Identifying new kinds of theorems and proofs which are guaranteed to have computably effective proofs, and new ways of finding them,

My own work is in the area of proof mining.

Typically, I work on projects like the following:

- Actually, my favorite thing to do is the previous two together: find a theorem which I think should have a computably effective proof, but where the existing techniques don't work, and simultaneously figure out a general method for extracting a computably effective proof while applying it to the particular example,

My own work is in the area of proof mining.

Typically, I work on projects like the following:

- Alternatively, sometimes I examine very complicated effective proofs, and look for non-effective proofs which are easier to understand.

My own work is in the area of proof mining.

Typically, I work on projects like the following:

- Alternatively, sometimes I examine very complicated effective proofs, and look for non-effective proofs which are easier to understand. This can happen because effective proofs might require lots of calculations, while in a non-effective proof we can sometimes replace very big numbers with "infinity" or very small numbers with 0 (in a mathematically rigorous way) to give a simpler non-effective proof.

My own work is in the area of proof mining.

Typically, I work on projects like the following:

- Alternatively, sometimes I examine very complicated effective proofs, and look for non-effective proofs which are easier to understand. This can happen because effective proofs might require lots of calculations, while in a non-effective proof we can sometimes replace very big numbers with "infinity" or very small numbers with 0 (in a mathematically rigorous way) to give a simpler non-effective proof. In addition, sometimes it turns out that the non-effective proof can be generalized to new results more easily than the original proof.

The study of games like Chomp is part of the branch of mathematics known as *combinatorial game theory*.

Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning ways for your mathematical plays*. Second. Natick, MA: A K Peters Ltd., 2001, pp. xx+276. ISBN: 1-56881-130-6.

Richard J. Nowakowski, ed. *Games of no chance*. Vol. 29. Mathematical Sciences Research Institute Publications. Papers from the Combinatorial Games Workshop held in Berkeley, CA, July 11–21, 1994. Cambridge: Cambridge University Press, 1996, pp. xii+537. ISBN: 0-521-57411-0.

Chomp itself has been studied in detail, but efficient or insightful winning strategies are known only for a few special cases.

- Eric J. Friedman and Adam Scott Landsberg. “Scaling, renormalization, and universality in combinatorial games: the geometry of Chomp”. In: *Combinatorial optimization and applications*. Vol. 4616. Lecture Notes in Comput. Sci. Berlin: Springer, 2007, pp. 200–207. DOI: 10.1007/978-3-540-73556-4_23. URL: http://dx.doi.org/10.1007/978-3-540-73556-4_23.
- Xinyu Sun. “Improvements on Chomp”. In: *Integers* 2 (2002), G1, 8. ISSN: 1867-0652.
- Doron Zeilberger. “Three-rowed Chomp”. In: *Adv. in Appl. Math.* 26.2 (2001), pp. 168–179. ISSN: 0196-8858. DOI: 10.1006/aama.2000.0714. URL: http://dx.doi.org/10.1006/aama.2000.0714.

Proof theory is a branch of logic focusing on the studying the intrinsic properties of proofs. Computably effective proofs play a central role in the field.

Samuel R. Buss. “An introduction to proof theory”. In: *Handbook of proof theory*. Vol. 137. Stud. Logic Found. Math. Amsterdam: North-Holland, 1998, pp. 1–78. DOI: 10.1016/S0049-237X(98)80016-5. URL: http://dx.doi.org/10.1016/S0049-237X(98)80016-5.

Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. Vol. 7. Cambridge Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press, 1989, pp. xii+176. ISBN: 0-521-37181-3.

Computational complexity investigates different notions of efficient computation and their relationships.

Erwin Engeler. *Introduction to the theory of computation*. Computer Science and Applied Mathematics. New York: Academic Press, 1973, pp. viii+231.

Christos H. Papadimitriou. *Computational complexity*. Reading, MA: Addison-Wesley Publishing Company, 1994, pp. xvi+523. ISBN: 0-201-53082-1.

The End

Theorem

Every game of Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Theorem

Every game of Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Unfortunately, this is a bit hard to prove

Theorem

Every game of Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Unfortunately, this is a bit hard to prove, so we'll use a standard math trick

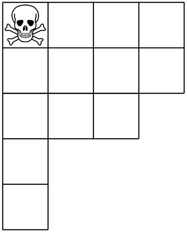
Theorem

Every game of Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Unfortunately, this is a bit hard to prove, so we'll use a standard math trick: faced with a problem too hard to solve, make it harder.

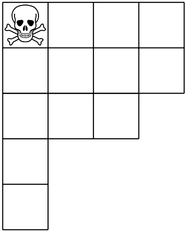
The game of Jagged Chomp is just like the game of Chomp, except the board doesn't have to be rectangular.

The game of Jagged Chomp is just like the game of Chomp, except the board doesn't have to be rectangular. So Jagged Chomp can be played on boards

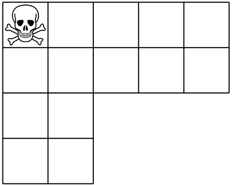


Like this

The game of Jagged Chomp is just like the game of Chomp, except the board doesn't have to be rectangular. So Jagged Chomp can be played on boards

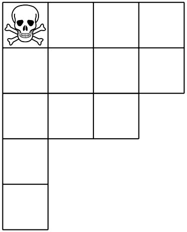


Like this

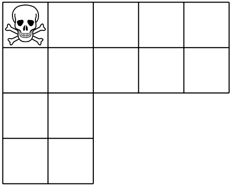


and this

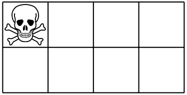
The game of Jagged Chomp is just like the game of Chomp, except the board doesn't have to be rectangular. So Jagged Chomp can be played on boards



Like this

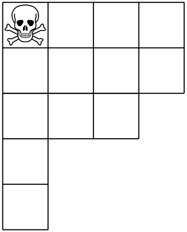


and this

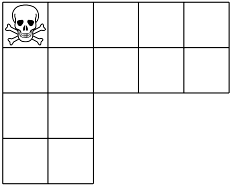


and this.

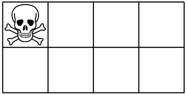
The game of Jagged Chomp is just like the game of Chomp, except the board doesn't have to be rectangular. So Jagged Chomp can be played on boards



Like this



and this



and this.

Because the board *can* be rectangular, it just doesn't have to be.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Again, we'll prove this by contradiction.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Again, we'll prove this by contradiction. Suppose *not*, so there *are* undetermined games of Jagged Chomp.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Again, we'll prove this by contradiction. Suppose *not*, so there *are* undetermined games of Jagged Chomp. Every game of Jagged Chomp has a board size: the number of squares on the board.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Again, we'll prove this by contradiction. Suppose *not*, so there *are* undetermined games of Jagged Chomp. Every game of Jagged Chomp has a board size: the number of squares on the board.

There's only one game whose board has only 1 square: the 1x1 game.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Again, we'll prove this by contradiction. Suppose *not*, so there *are* undetermined games of Jagged Chomp. Every game of Jagged Chomp has a board size: the number of squares on the board.

There's only one game whose board has only 1 square: the 1x1 game. The 1x1 game is certainly determined

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Again, we'll prove this by contradiction. Suppose *not*, so there *are* undetermined games of Jagged Chomp. Every game of Jagged Chomp has a board size: the number of squares on the board.

There's only one game whose board has only 1 square: the 1x1 game. The 1x1 game is certainly determined: the second player always wins, just by waiting for the first player to make a move and immediately lose.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

There must be some smallest size which is big enough for a board to be undetermined.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

There must be some smallest size which is big enough for a board to be undetermined. Consider some undetermined board of minimal size.

Theorem

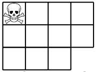
Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

There must be some smallest size which is big enough for a board to be undetermined. Consider some undetermined board of minimal size. So all boards with fewer squares *are* determined.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

For example, suppose this is the board.

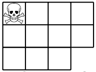


Alice's turn to play

Theorem

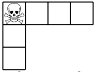
Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

For example, suppose this is the board. After Alice's move, what's left will be a smaller board.



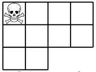
Alice's turn to play

becomes



Bob's turn to play

or



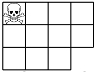
Bob's turn to play

or another option

Theorem

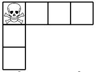
Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

For example, suppose this is the board. After Alice's move, what's left will be a smaller board. Alice will be the *second player* to move on these smaller boards.



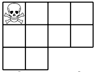
Alice's turn to play

becomes



Bob's turn to play

or



Bob's turn to play

or another option

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

For example, suppose this is the board. After Alice's move, what's left will be a smaller board. Alice will be the *second player* to move on these smaller boards.

Remember that we started with the *smallest* undetermined board.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Remember that we started with the *smallest* undetermined board. All the smaller boards *are* determined.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

Remember that we started with the *smallest* undetermined board. All the smaller boards *are* determined. That means on each of these smaller boards, there is either a winning strategy for the first player or a winning strategy for the second player.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

If the second player has a winning strategy on *one* of these smaller boards, Alice has a winning strategy.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

If the second player has a winning strategy on *one* of these smaller boards, Alice has a winning strategy. She first makes the choice that leads to a board where the second player has a winning strategy, and then once there, uses that strategy to win.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

If the second player *never* has a winning strategy on these smaller boards, the first player must have a winning strategy on all the smaller boards.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

If the second player *never* has a winning strategy on these smaller boards, the first player must have a winning strategy on all the smaller boards. Then Bob *does* have a winning strategy on the original board

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

If the second player *never* has a winning strategy on these smaller boards, the first player must have a winning strategy on all the smaller boards. Then Bob *does* have a winning strategy on the original board: wait for Alice to pick a board, then use the winning strategy for the board Alice picks.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

This is a contradiction: we started with a board that wasn't supposed to have a winning strategy, and showed that it had one anyway.

Theorem

Every game of Jagged Chomp is determined: either the first player has a winning strategy, or the second player has a winning strategy.

This is a contradiction: we started with a board that wasn't supposed to have a winning strategy, and showed that it had one anyway. So *every* game of Jagged Chomp, and in particular every game of Chomp, is determined.