CS3191 Section 1

Introduction to Games

Andrea Schalk

Department of Computer Science, University of Manchester

Overview of the Course

This course consists of the following sections:

- Introduction. What is a game? Game trees, strategies, matrices, pay-off functions.

Overview of the Course

This course consists of the following sections:

- Introduction. What is a game? Game trees, strategies, matrices, pay-off functions.
- Small games. Optimal strategies in a game (‘solving games’), problems, size issues.

Overview of the Course

This course consists of the following sections:

- Introduction. What is a game? Game trees, strategies, matrices, pay-off functions.
- Small games. Optimal strategies in a game ('solving games'), problems, size issues.
- Medium games. Exploring games 'locally', minimax algorithm, alpha-beta algorithm.

Overview of the Course

This course consists of the following sections:

- Introduction. What is a game? Game trees, strategies, matrices, pay-off functions.
- Small games. Optimal strategies in a game (‘solving games’), problems, size issues.
- Medium games. Exploring games ‘locally’, minimax algorithm, alpha-beta algorithm.
- Large games. How game-playing programs work. History of Chess-playing programs.

Overview of the Course

This course consists of the following sections:

- Introduction. What is a game? Game trees, strategies, matrices, pay-off functions.
- Small games. Optimal strategies in a game ('solving games'), problems, size issues.
- Medium games. Exploring games 'locally', minimax algorithm, alpha-beta algorithm.
- Large games. How game-playing programs work. History of Chess-playing programs.
- Game models. Games used to model interactions.

Overview of the Course

This course consists of the following sections:

- Introduction. What is a game? Game trees, strategies, matrices, pay-off functions.
- Small games. Optimal strategies in a game (‘solving games’), problems, size issues.
- Medium games. Exploring games ‘locally’, minimax algorithm, alpha-beta algorithm.
- Large games. How game-playing programs work. History of Chess-playing programs.
- Game models. Games used to model interactions.
- Games and evolution. Games used to model the evolution of traits.

About this course

Elements of teaching on this course:

- Lectures.

About this course

Elements of teaching on this course:

- Lectures.
- Course notes.

About this course

Elements of teaching on this course:

- Lectures.
- Course notes.
- Exercises.

About this course

Elements of teaching on this course:

- Lectures.
- Course notes.
- Exercises.
- Examples classes.

About this course

Elements of teaching on this course:

- Lectures.
- Course notes.
- Exercises.
- Examples classes.
- Web page `http://www.cs.man.ac.uk/~schalk/3192/index.html`.

About this course

Elements of teaching on this course:

- Lectures.
- Course notes.
- Exercises.
- Examples classes.
- Web page `http://www.cs.man.ac.uk/~schalk/3192/index.html`.
- Literature.

About this course

Elements of teaching on this course:

- Lectures.
- Course notes.
- Exercises.
- Examples classes.
- Web page `http://www.cs.man.ac.uk/~schalk/3192/index.html`.
- Literature.
- The exam.

Introduction–Overview

In this section we give an introduction to what we mean in this course when we talk about games. It is mostly concerned with introducing a few basic notions, namely those of

- a game tree (required for the formal definition of game);

Introduction–Overview

In this section we give an introduction to what we mean in this course when we talk about games. It is mostly concerned with introducing a few basic notions, namely those of

- a game tree (required for the formal definition of game);
- a strategy;

Introduction–Overview

In this section we give an introduction to what we mean in this course when we talk about games. It is mostly concerned with introducing a few basic notions, namely those of

- a game tree (required for the formal definition of game);
- a strategy;
- a pay-off function;

Introduction–Overview

In this section we give an introduction to what we mean in this course when we talk about games. It is mostly concerned with introducing a few basic notions, namely those of

- a game tree (required for the formal definition of game);
- a strategy;
- a pay-off function;
- the normal form of a game.

What is a game?

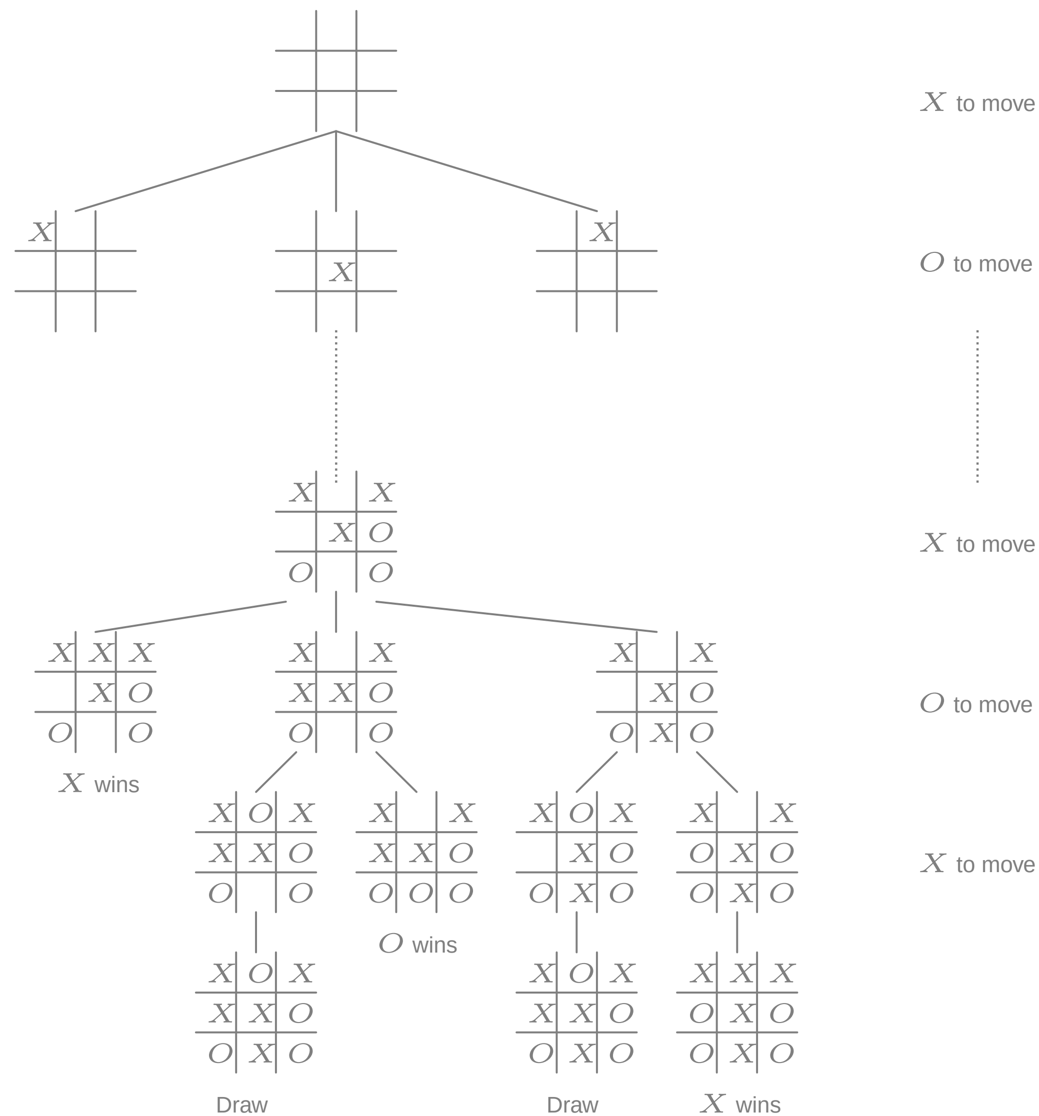
So, what is a game?

What is a game?

So, what is a game?

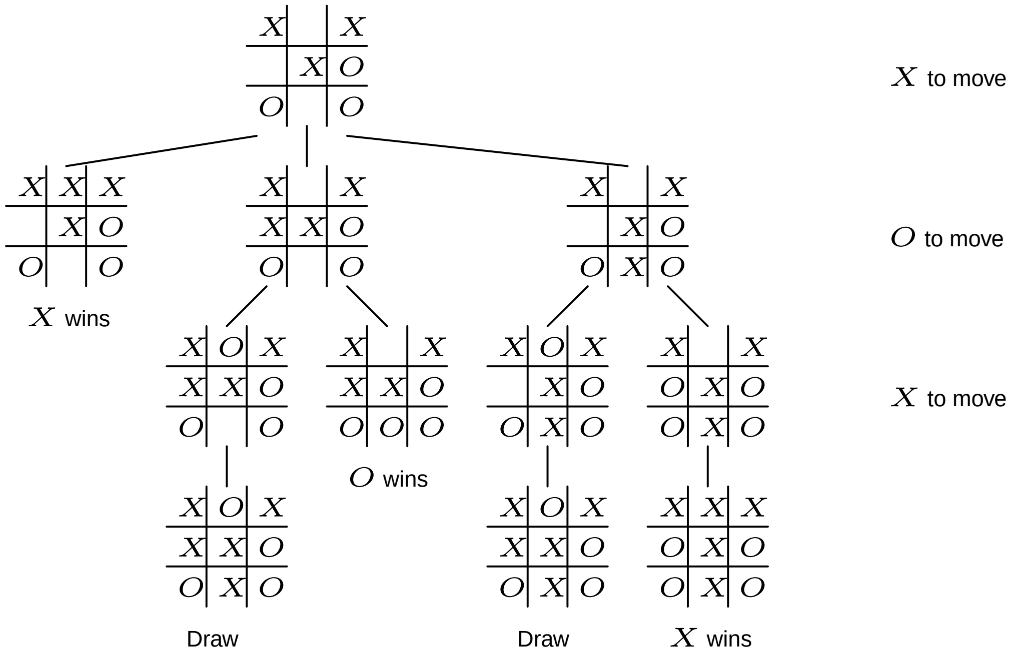
A game gives the rules for the interactions of the participating entities, the players.

Example



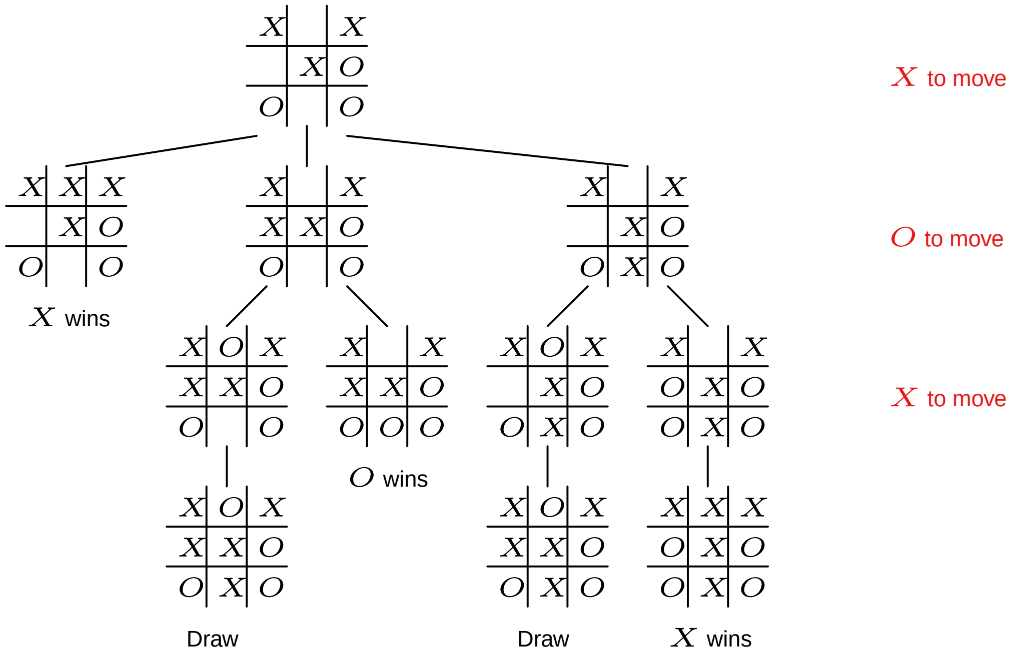
X to move
O to move
X to move
O to move
X to move
X wins
O wins
Draw
Draw
X wins

A game tree



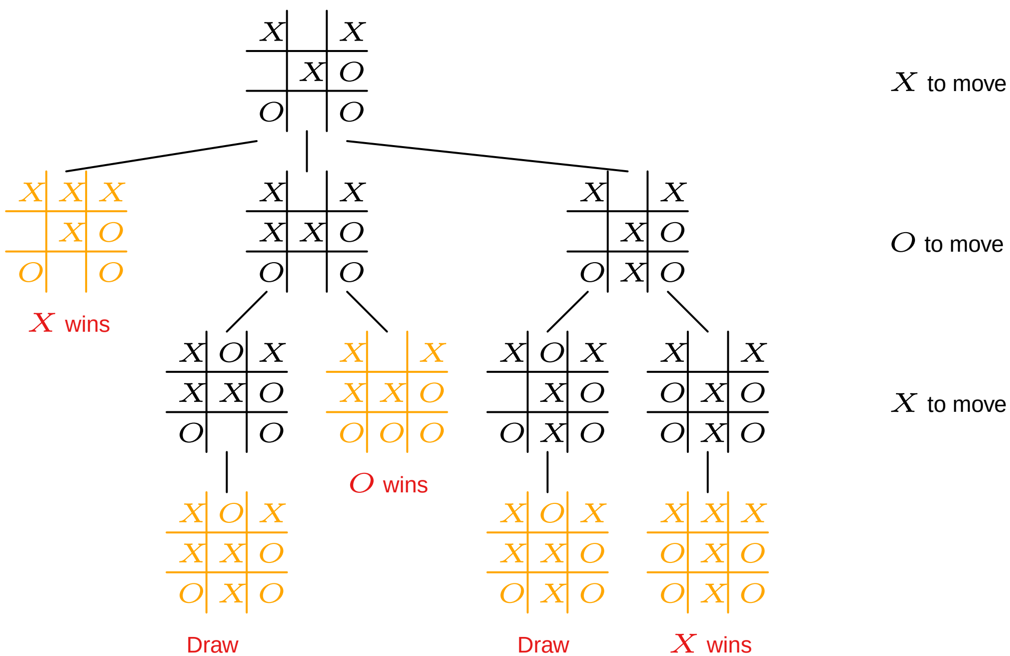
The nodes in such a tree are the positions of the game.

A game tree



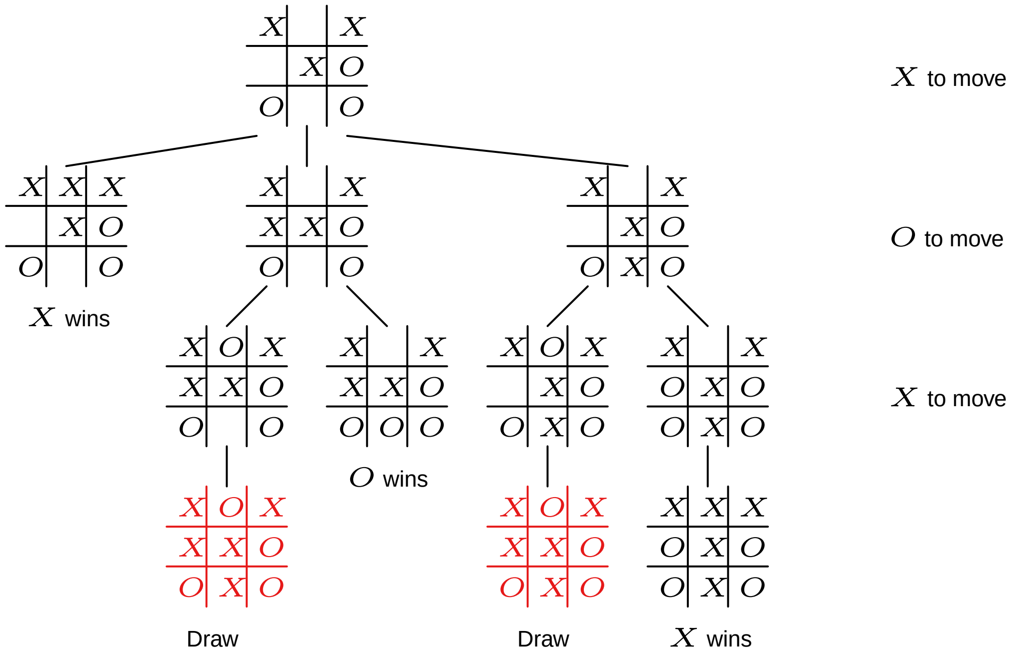
At each node it has to be clear whose **turn** it is.

A game tree



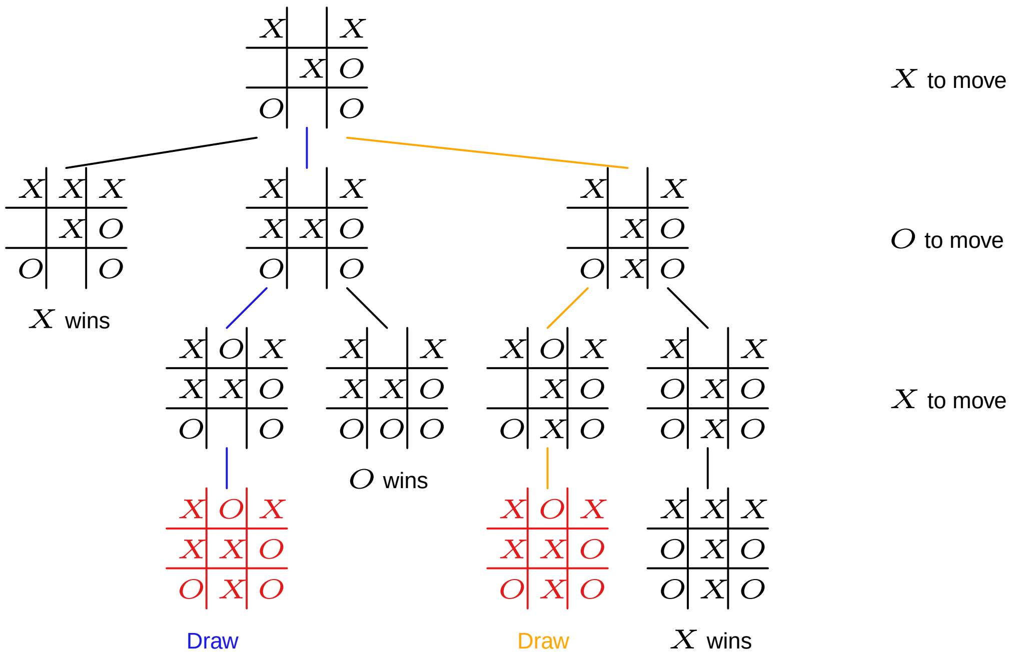
The bottom nodes in the tree, the **leaves**, are the **final positions** of the game. This is where we note the **result**.

A game tree



Note that different positions in the tree can correspond to the same constellation on the board.

A game tree



That is because in a tree, given a position we can reconstruct the path from the root leading to it (and thus the play of the game). Another path is shown. Taking trees instead of graphs makes our lives easier.

Questions about game trees

Question. Let us test the notion of a game tree.

Questions about game trees

Question. Let us test the notion of a game tree.

(a) Could you (in principle, don't mind the size) draw a game tree for Backgammon, or Snakes-and-Ladders? If not, why not?

Questions about game trees

Question. Let us test the notion of a game tree.

(a) Could you (in principle, don't mind the size) draw a game tree for Backgammon, or Snakes-and-Ladders? If not, why not?

(b) Could you draw a game tree for Paper-Stone-Scissors? If not, why not?

Questions about game trees

Question. Let us test the notion of a game tree.

(a) Could you (in principle, don't mind the size) draw a game tree for Backgammon, or Snakes-and-Ladders? If not, why not?

(b) Could you draw a game tree for Paper-Stone-Scissors? If not, why not?

(c) Consider the following simple game between two players: Player 1 has a coin which he hides under his hand, having first decided whether it should show head or tail. Player 2 guesses which of these has been chosen. If she guesses correctly, Player 1 pays her 1 quid, otherwise she has to pay the same amount to him. Could you draw a game tree for this game? If not why not?

More general games

The following are potential issues that can come up in games for which our current definition for ‘game tree’ doesn’t really work.

- Chance. A game might contain chance moves.

More general games

The following are potential issues that can come up in games for which our current definition for ‘game tree’ doesn’t really work.

- Chance. A game might contain chance moves.
- Imperfect information. At some time, the players might not know in which position of the game tree the game currently is. Examples are card games.

More general games

The following are potential issues that can come up in games for which our current definition for 'game tree' doesn't really work.

- Chance. A game might contain chance moves.
- Imperfect information. At some time, the players might not know in which position of the game tree the game currently is. Examples are card games.
- Simultaneous moves. Our notion of game tree doesn't seem to allow for several players to move simultaneously.

More general games

The following are potential issues that can come up in games for which our current definition for 'game tree' doesn't really work.

- Chance. A game might contain chance moves.
- Imperfect information. At some time, the players might not know in which position of the game tree the game currently is. Examples are card games.
- Simultaneous moves. Our notion of game tree doesn't seem to allow for several players to move simultaneously.

We will see how to deal with these issues later.

More general games

The following are potential issues that can come up in games for which our current definition for ‘game tree’ doesn’t really work.

- Chance. A game might contain chance moves.
- Imperfect information. At some time, the players might not know in which position of the game tree the game currently is. Examples are card games.
- Simultaneous moves. Our notion of game tree doesn’t seem to allow for several players to move simultaneously.

We will see how to deal with these issues later. For now we assume that all our games are of complete (or perfect) information, that is, each player knows precisely where in the game tree they are.

Games–definition

Definition 1 *A* *game* *is given by*

- *a finite set of* *players*,

Games–definition

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*

Games–definition

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*
- *for each node of the tree, a player whose turn it is in that position and*

Games–definition

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*
- *for each node of the tree, a player whose turn it is in that position and*
- *for each final node and each player a pay-off function.*

Games–definition

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*
- *for each node of the tree, a player whose turn it is in that position and*
- *for each final node and each player a pay-off function. (We will ignore this part of the definition for the moment.)*

Chomp

Consider the game where two players have a bar of chocolate consisting of $(m \times n)$ squares. The square in the top left corner is known to be poisonous. The players take turns in making moves, and a valid move consists of

Chomp

Consider the game where two players have a bar of chocolate consisting of $(m \times n)$ squares. The square in the top left corner is known to be poisonous. The players take turns in making moves, and a valid move consists of

- choosing a square of chocolate and

Chomp

Consider the game where two players have a bar of chocolate consisting of $(m \times n)$ squares. The square in the top left corner is known to be poisonous. The players take turns in making moves, and a valid move consists of

- choosing a square of chocolate and
- eating all the squares which are in the lower right quadrant relative to that square (that is, 'to the right and below' the chosen square).

Chomp

Consider the game where two players have a bar of chocolate consisting of $(m \times n)$ squares. The square in the top left corner is known to be poisonous. The players take turns in making moves, and a valid move consists of

- choosing a square of chocolate and
- eating all the squares which are in the lower right quadrant relative to that square (that is, 'to the right and below' the chosen square).

The person who has to eat the poisonous square loses.

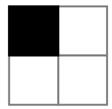
Chomp

Consider the game where two players have a bar of chocolate consisting of $(m \times n)$ squares. The square in the top left corner is known to be poisonous. The players take turns in making moves, and a valid move consists of

- choosing a square of chocolate and
- eating all the squares which are in the lower right quadrant relative to that square (that is, 'to the right and below' the chosen square).

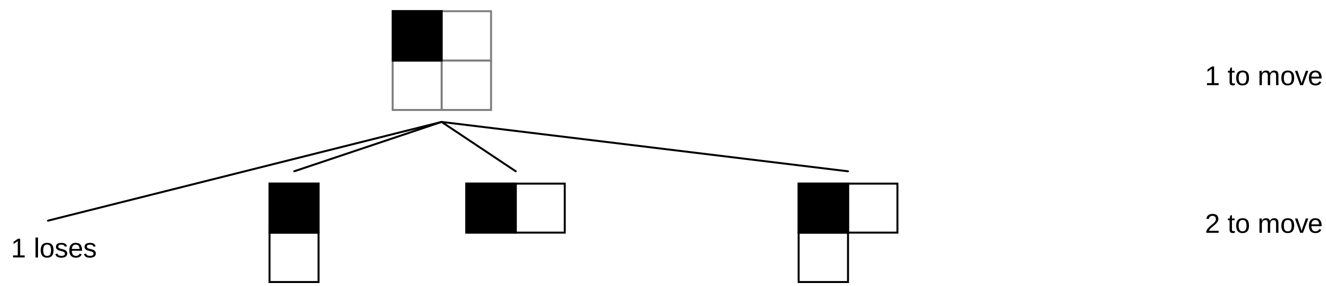
The person who has to eat the poisonous square loses. The resulting game is known as $(m \times n)$-Chomp.

(2×2)-Chomp

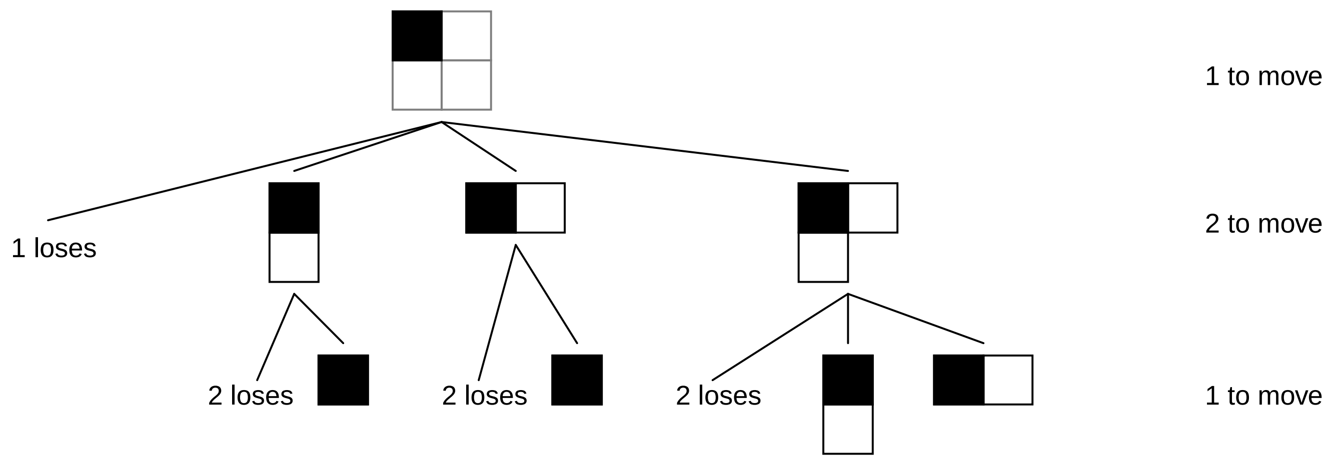


1 to move

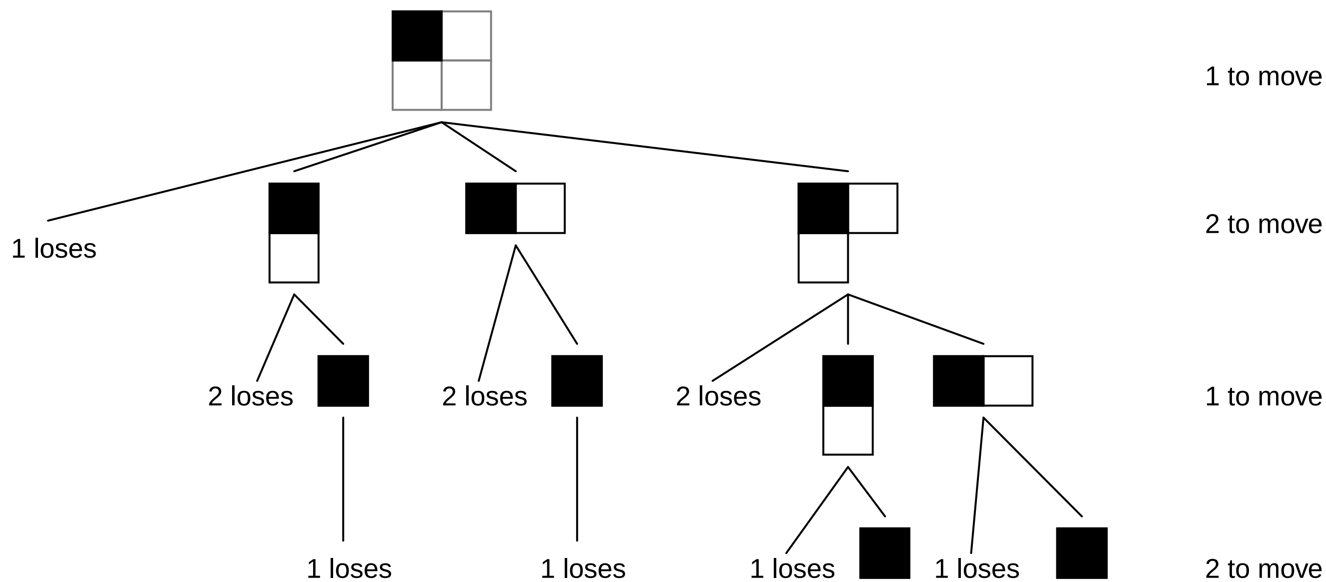
(2×2)-Chomp



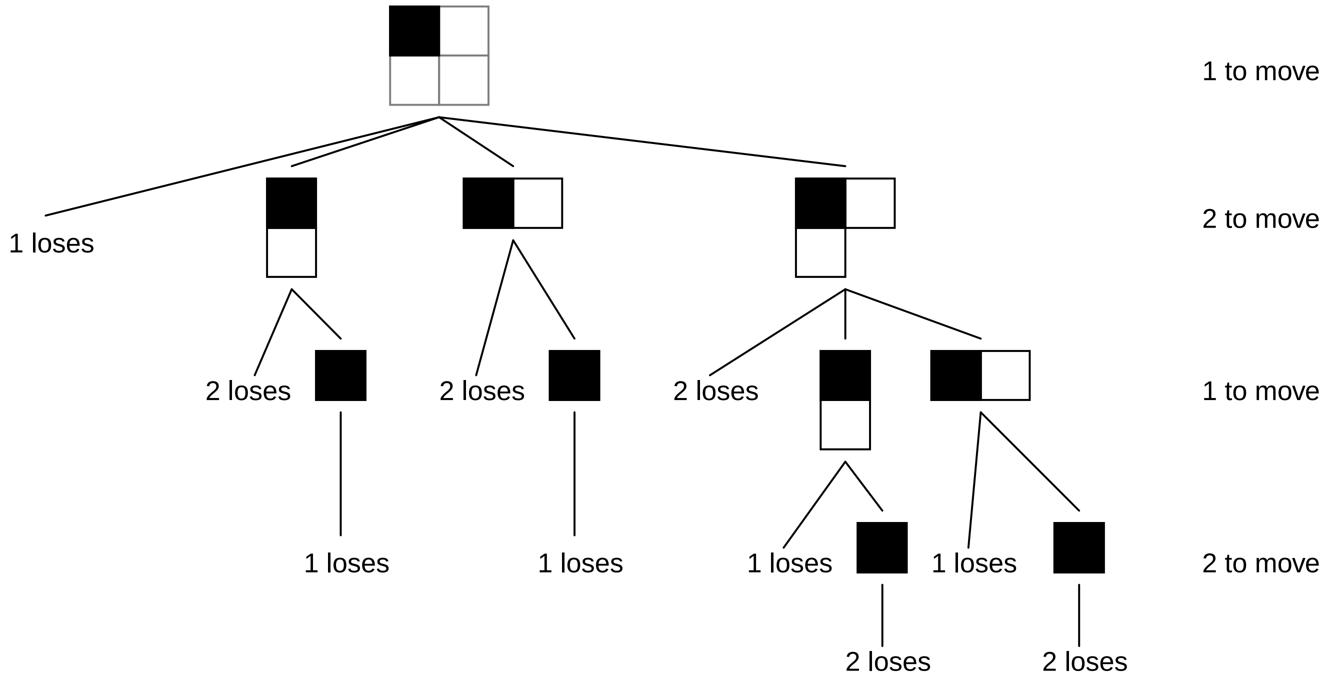
(2×2)-Chomp



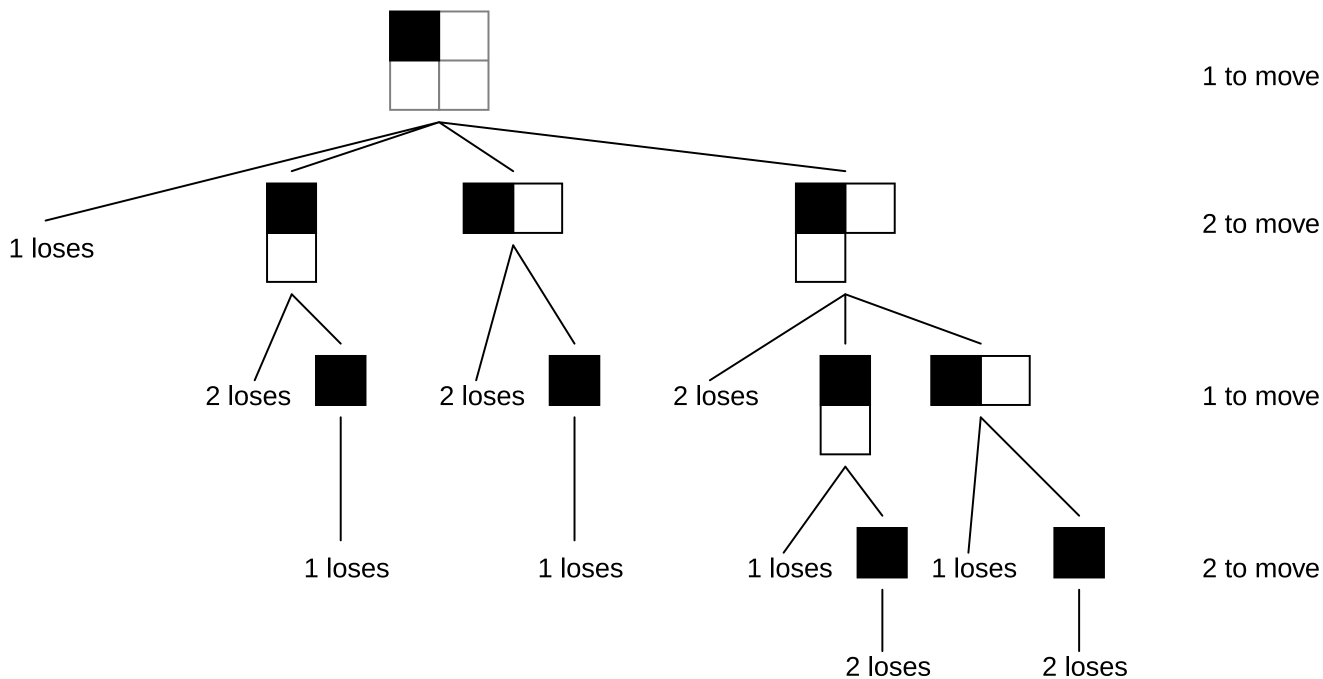
(2×2)-Chomp



(2×2)-Chomp



(2×2)-Chomp



Question. Which player would you rather be, 1 or 2?

Games–are they interesting?

Question. What do you think of the games we've been looking at?

Games–are they interesting?

Question. What do you think of the games we’ve been looking at?

Do you still think this course is going to be interesting?

Games–are they interesting?

Question. What do you think of the games we've been looking at?

Do you still think this course is going to be interesting?

What would make the games more exciting, and why aren't we looking at those issues?

Understanding games

We can think of the nodes of the game tree as decision points, where part of the information we are given is whose decision this is as well as what their options are.

Understanding games

We can think of the nodes of the game tree as decision points, where part of the information we are given is whose decision this is as well as what their options are.

Many games, such as Chess and Checkers, have game trees which are far too big to be drawn, or even to be fully held in a computer's memory at any given time.

Understanding games

We can think of the nodes of the game tree as decision points, where part of the information we are given is whose decision this is as well as what their options are.

Many games, such as Chess and Checkers, have game trees which are far too big to be drawn, or even to be fully held in a computer's memory at any given time. We can still play these games, because their rules tell us the information needed to play them, and the game tree can be constructed from those. As a result, we only need local information to know whose turn it is and what their moves are.

Understanding games

We can think of the nodes of the game tree as decision points, where part of the information we are given is whose decision this is as well as what their options are.

Many games, such as Chess and Checkers, have game trees which are far too big to be drawn, or even to be fully held in a computer's memory at any given time. We can still play these games, because their rules tell us the information needed to play them, and the game tree can be constructed from those. As a result, we only need local information to know whose turn it is and what their moves are.

Chess, for example, has twenty opening moves for White, and as many replies for Black. In other words, after only two moves there are already $20 \times 20 = 400$ different positions!

Understanding games

We can think of the nodes of the game tree as decision points, where part of the information we are given is whose decision this is as well as what their options are.

Many games, such as Chess and Checkers, have game trees which are far too big to be drawn, or even to be fully held in a computer's memory at any given time. We can still play these games, because their rules tell us the information needed to play them, and the game tree can be constructed from those. As a result, we only need local information to know whose turn it is and what their moves are.

Chess, for example, has twenty opening moves for White, and as many replies for Black. In other words, after only two moves there are already $20 \times 20 = 400$ different positions! We will talk about games with such big game trees later in the course, in Sections 3 and 4.

Understanding games

We can think of the nodes of the game tree as decision points, where part of the information we are given is whose decision this is as well as what their options are.

Many games, such as Chess and Checkers, have game trees which are far too big to be drawn, or even to be fully held in a computer's memory at any given time. We can still play these games, because their rules tell us the information needed to play them, and the game tree can be constructed from those. As a result, we only need local information to know whose turn it is and what their moves are.

Chess, for example, has twenty opening moves for White, and as many replies for Black. In other words, after only two moves there are already $20 \times 20 = 400$ different positions! We will talk about games with such big game trees later in the course, in Sections 3 and 4. While all the considerations we make in Section 2 are valid for such big games, the methods we introduce cannot be applied to them.

Plays

A play of a game is a path starting at the root of the game tree (that is, the start position). We say that a play is complete if it leads all the way to a final position.

Plays

A play of a game is a path starting at the root of the game tree (that is, the start position). We say that a play is complete if it leads all the way to a final position.

Question. How many plays are there for Noughts and Crosses? If you can't give the precise number, can you give an upper bound?

Plays

A play of a game is a path starting at the root of the game tree (that is, the start position). We say that a play is complete if it leads all the way to a final position.

Question. How many plays are there for Noughts and Crosses? If you can't give the precise number, can you give an upper bound?

The answer to this question depends on whether or not we want to consider symmetry consideration. A crude upper bound for the number of plays ignores these. Player 1 has 9 possibilities for placing his first mark, Player 2 has the remaining 8 squares for her first move, then Player 1 can choose among any of the remaining 7 fields, leading to

Plays

A play of a game is a path starting at the root of the game tree (that is, the start position). We say that a play is complete if it leads all the way to a final position.

Question. How many plays are there for Noughts and Crosses? If you can't give the precise number, can you give an upper bound?

The answer to this question depends on whether or not we want to consider symmetry consideration. A crude upper bound for the number of plays ignores these. Player 1 has 9 possibilities for placing his first mark, Player 2 has the remaining 8 squares for her first move, then Player 1 can choose among any of the remaining 7 fields, leading to

$$9 \times 8 \times \cdots \times 2 \times 1 = 9! = 362880$$

plays.

Adding Chance

In order to add the notion of chance moves to our game trees, we assume that there is somebody (often called Nature) who takes care of such moves. (Nature is not normally included among the ‘players’ of the game.)

Adding Chance

In order to add the notion of chance moves to our game trees, we assume that there is somebody (often called Nature) who takes care of such moves. (Nature is not normally included among the 'players' of the game.)

In the game tree, we add nodes where

Adding Chance

In order to add the notion of chance moves to our game trees, we assume that there is somebody (often called Nature) who takes care of such moves. (Nature is not normally included among the ‘players’ of the game.)

In the game tree, we add nodes where

- it is not the turn of any of the players and

Adding Chance

In order to add the notion of chance moves to our game trees, we assume that there is somebody (often called Nature) who takes care of such moves. (Nature is not normally included among the 'players' of the game.)

In the game tree, we add nodes where

- it is not the turn of any of the players and
- for each of the branches from such a node there is a label giving the probability of that move occurring.

Risk

In the board game Risk players have ‘armies’ which can defend or conquer territory on a map (which forms the board).

Risk

In the board game Risk players have ‘armies’ which can defend or conquer territory on a map (which forms the board).

A possibility in the game is to attack another country. Here we assume the choice is between attacking with one and attacking with two.

Risk

In the board game Risk players have ‘armies’ which can defend or conquer territory on a map (which forms the board).

A possibility in the game is to attack another country. Here we assume the choice is between attacking with one and attacking with two.

To make it simpler, we assume here that the defender has only one army to defend the country with.

Risk

In the board game Risk players have ‘armies’ which can defend or conquer territory on a map (which forms the board).

Both players then roll as many dice as they have armies in the bout (here, one or two). The result is evaluated as follows:

Risk

In the board game Risk players have ‘armies’ which can defend or conquer territory on a map (which forms the board).

Both players then roll as many dice as they have armies in the bout (here, one or two). The result is evaluated as follows:

- The attacker has to roll a higher number than the defender to win;

Risk

In the board game Risk players have ‘armies’ which can defend or conquer territory on a map (which forms the board).

Both players then roll as many dice as they have armies in the bout (here, one or two). The result is evaluated as follows:

- The attacker has to roll a higher number than the defender to win;
- if the number of dice thrown by each side are not the same, then the ‘best’ throws by the player who has more dice counts.

Risk

In the board game Risk players have ‘armies’ which can defend or conquer territory on a map (which forms the board).

Both players then roll as many dice as they have armies in the bout (here, one or two). The result is evaluated as follows:

- The attacker has to roll a higher number than the defender to win;
- if the number of dice thrown by each side are not the same, then the ‘best’ throws by the player who has more dice counts.

To keep the size of the game tree reasonable, we assume that instead of using ordinary dice we use ones which produce the numbers 1, 2 and 3 only, with equal probability.

Risk

In the board game Risk players have 'armies' which can defend or conquer territory on a map (which forms the board).

Both players then roll as many dice as they have armies in the bout (here, one or two). The result is evaluated as follows:

- The attacker has to roll a higher number than the defender to win;
- if the number of dice thrown by each side are not the same, then the 'best' throws by the player who has more dice counts.

To keep the size of the game tree reasonable, we assume that instead of using ordinary dice we use ones which produce the numbers 1, 2 and 3 only, with equal probability.

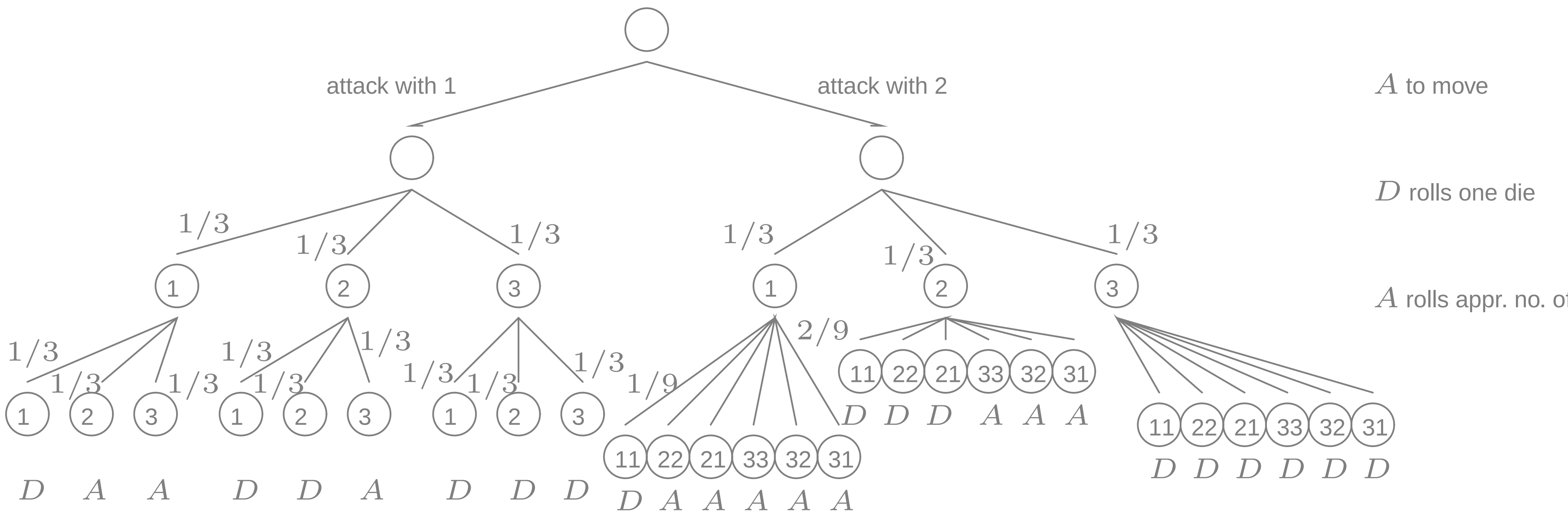
We want to find out whether it is better to attack with one or with two armies.

Risk

In many ways, the game tree is easier to understand than a description using words—at least it is unambiguous.

Risk

In many ways, the game tree is easier to understand than a description using words—at least it is unambiguous.



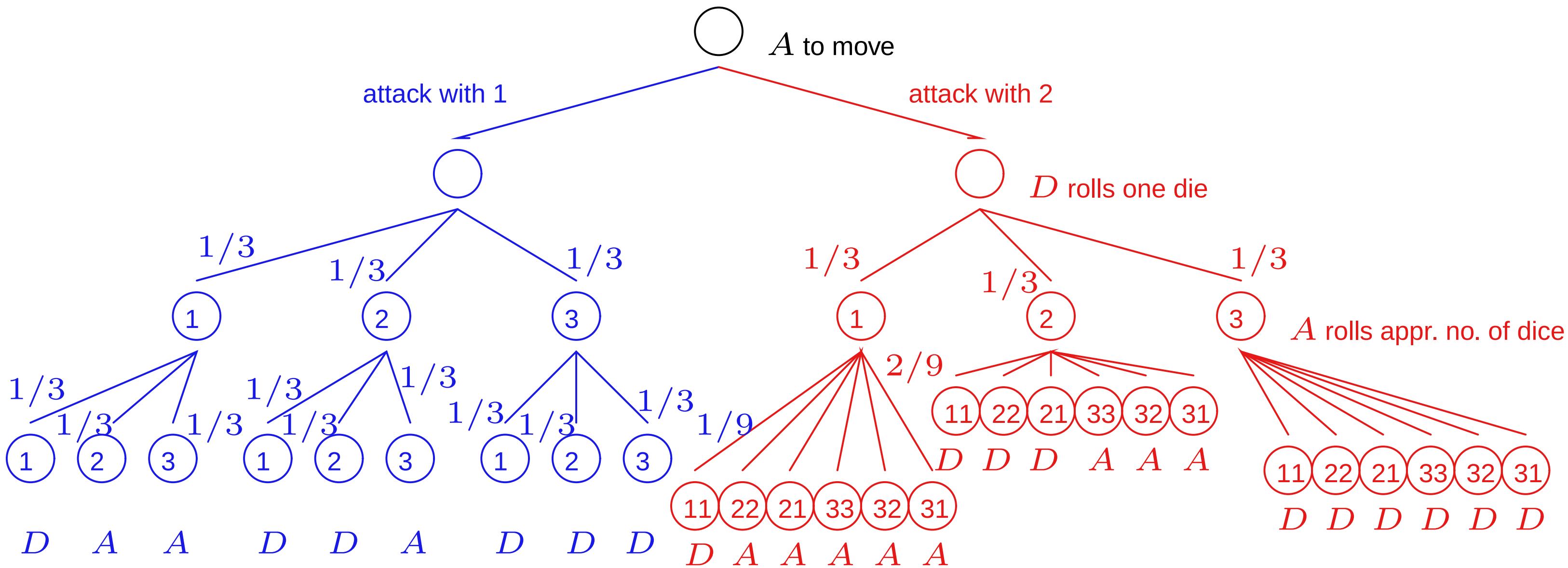
Probabilities for throwing two dice: $1/9$ for each branch where the two numbers agree, $2/9$ where they differ.

Risk–Which is better?

We will examine the two branches of the tree separately to find out whether the Attacker should attack with 1 or with 2 armies.

Risk–Which is better?

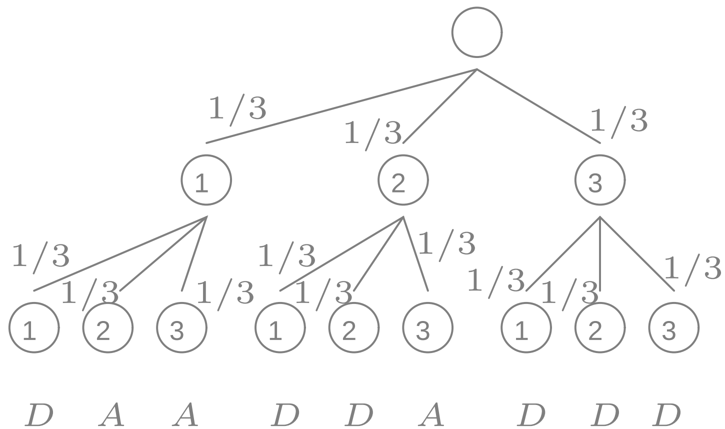
We will examine the two branches of the tree separately to find out whether the Attacker should attack with 1 or with 2 armies.



Probabilities for throwing two dice: $1/9$ for each branch where the two numbers agree, $2/9$ where they differ.

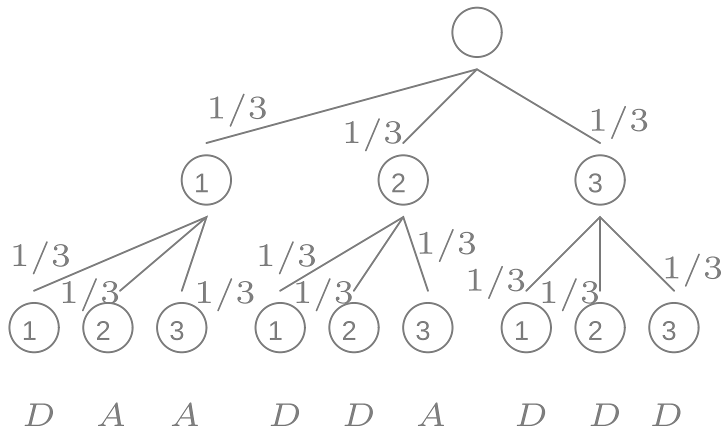
Risk–Attack with 1

Attack with 1



Risk–Attack with 1

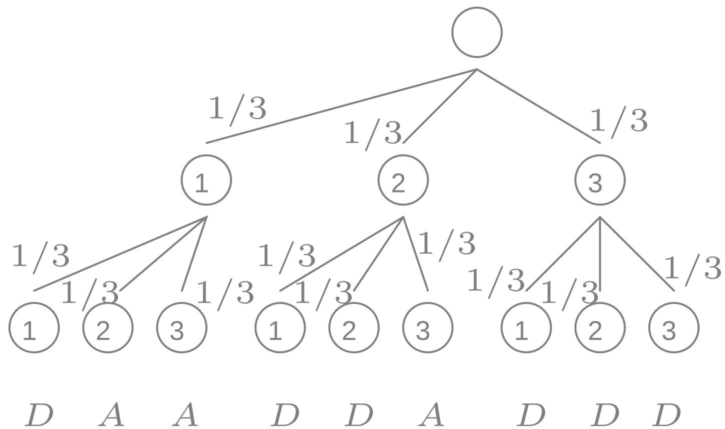
Attack with 1



We want to calculate with which probability A will win. For that, we first have to calculate the probability for each outcome (that is, each leaf) of the tree.

Risk–Attack with 1

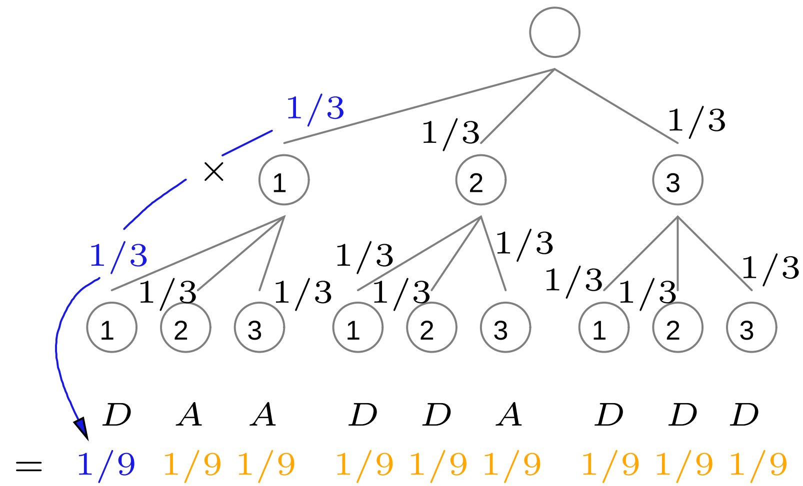
Attack with 1



We want to calculate with which probability A will win. For that, we first have to calculate the probability for each outcome (that is, each leaf) of the tree. For that, we have to multiply the probabilities along the path.

Risk–Attack with 1

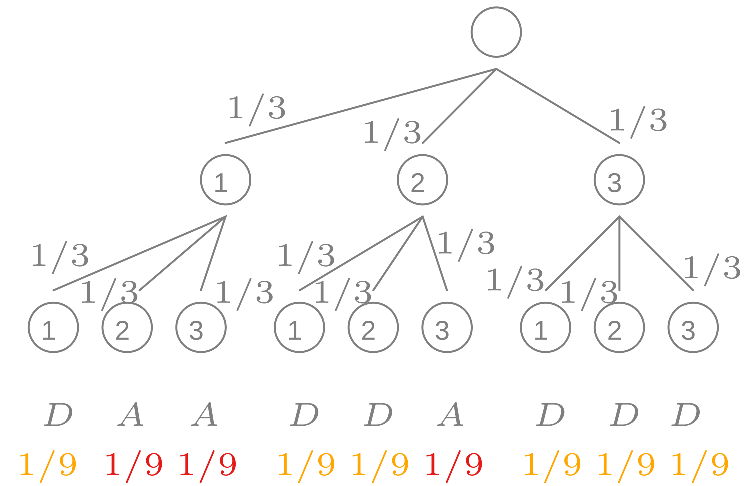
Attack with 1



We want to calculate with which probability A will win. For that, we first have to calculate the probability for each outcome (that is, each leaf) of the tree. For that, we have to multiply the probabilities along the path. So every branch has a probability of $1/9$.

Risk–Attack with 1

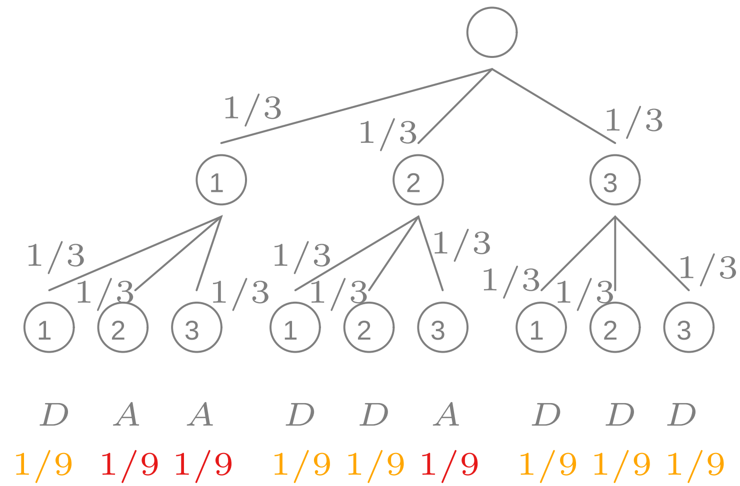
Attack with 1



We want to calculate with which probability A will win. For that, we first have to calculate the probability for each outcome (that is, each leaf) of the tree. For that, we have to multiply the probabilities along the path. So every branch has a probability of $1/9$. In order to find the probability for A to win, we have to add the probabilities for all the paths where A wins.

Risk–Attack with 1

Attack with 1



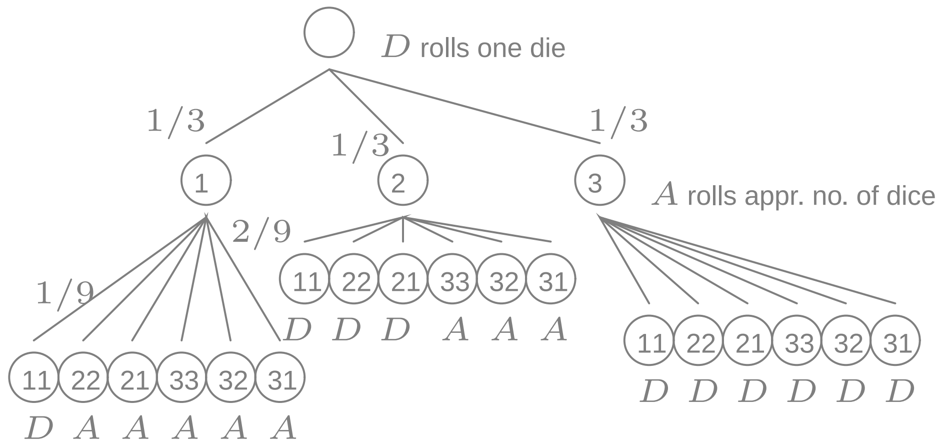
We want to calculate with which probability A will win. For that, we first have to calculate the probability for each outcome (that is, each leaf) of the tree. For that, we have to multiply the probabilities along the path. So every branch has a probability of $1/9$. In order to find the probability for A to win, we have to add the probabilities for all the paths where A wins. This gives

$$1/9 + 1/9 + 1/9 = 3/9 = 1/3,$$

so Attacker wins in $1/3$ of all cases.

Risk–Attack with 2

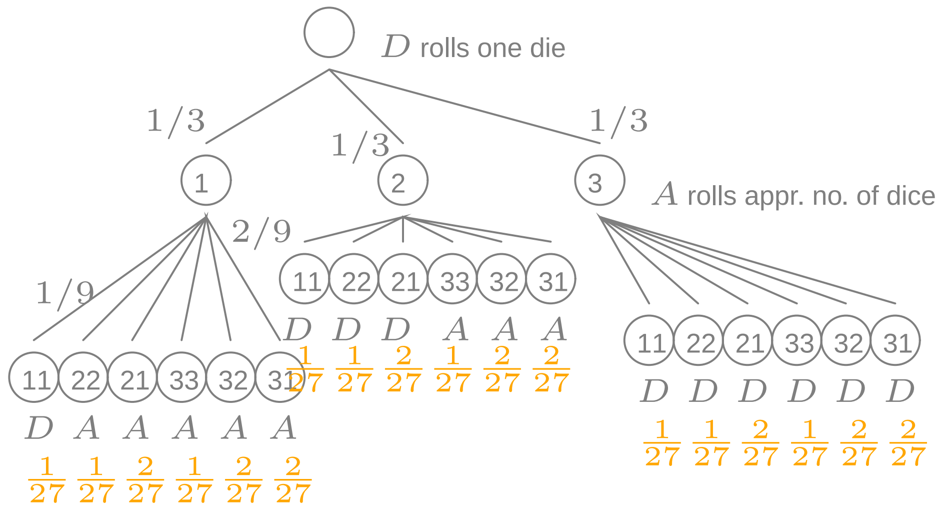
Attack with 2



We again calculate the probabilities for the leaves by multiplying the probabilities along the corresponding paths.

Risk–Attack with 2

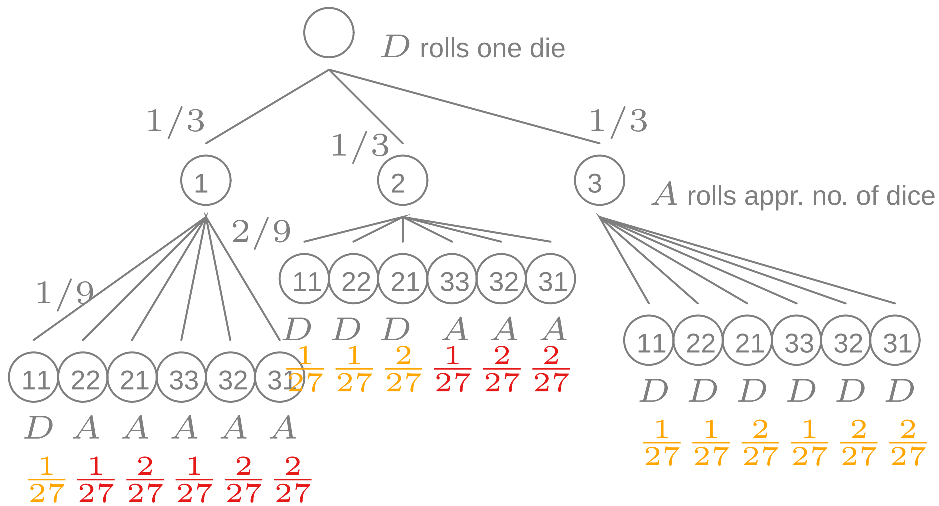
Attack with 2



We again calculate the probabilities for the leaves by multiplying the probabilities along the corresponding paths.

Risk–Attack with 2

Attack with 2

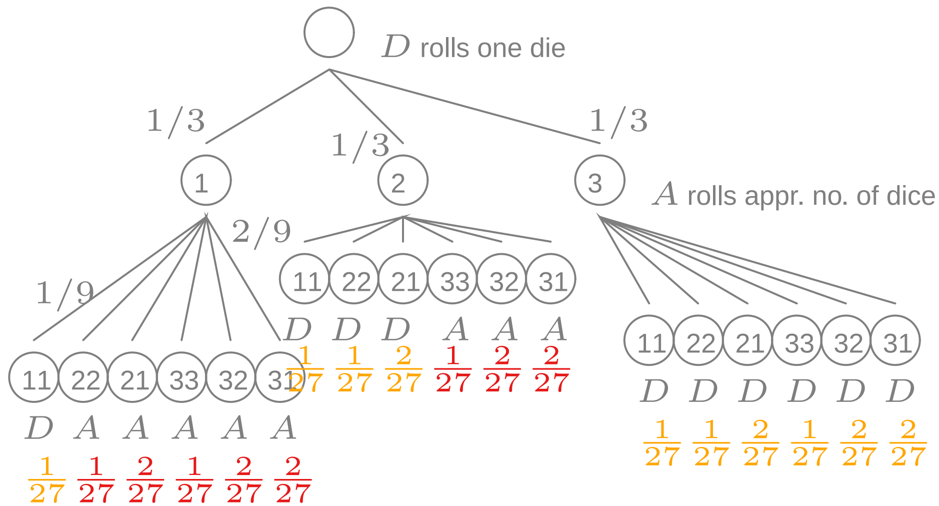


We again calculate the probabilities for the leaves by multiplying the probabilities along the corresponding paths. Again, we add up the probabilities for those occasions where A wins, and get

$$1/27 + 2/27 + 1/27 + 2/27 + 2/27 + 1/27 + 2/27 + 2/27 = 13/27 \approx 0.48.$$

Risk–Attack with 2

Attack with 2



We again calculate the probabilities for the leaves by multiplying the probabilities along the corresponding paths. Again, we add up the probabilities for those occasions where A wins, and get

$$1/27 + 2/27 + 1/27 + 2/27 + 2/27 + 1/27 + 2/27 + 2/27 = 13/27 \approx 0.48.$$

Since this is bigger than $1/3$, it is better for Attacker to attack with 2 armies. (And, in fact in general for Risk the chance of winning an encounter increases with the number of committed armies.)

Adding imperfect information

In order to describe card games, for example, we need to have a way of incorporating the idea of imperfect information into game trees.

Adding imperfect information

In order to describe card games, for example, we need to have a way of incorporating the idea of imperfect information into game trees.

The idea behind this is very simple: For each player, whenever it is his turn, we need to mark in the game tree which positions that player cannot distinguish between.

Adding imperfect information

In order to describe card games, for example, we need to have a way of incorporating the idea of imperfect information into game trees.

The idea behind this is very simple: For each player, whenever it is his turn, we need to mark in the game tree which positions that player cannot distinguish between.

But for this to make sense, the options for the player whose turn it is have to be the same for all the positions he might think he is in.

Adding imperfect information

In order to describe card games, for example, we need to have a way of incorporating the idea of imperfect information into game trees.

The idea behind this is very simple: For each player, whenever it is his turn, we need to mark in the game tree which positions that player cannot distinguish between.

But for this to make sense, the options for the player whose turn it is have to be the same for all the positions he might think he is in.

Question. Why do we make that a requirement?

Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.

Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.

We can draw a game tree for this kind of game by

Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.

We can draw a game tree for this kind of game by

- assuming that one player, say Player 1, moves first and that

Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.

We can draw a game tree for this kind of game by

- assuming that one player, say Player 1, moves first and that
- Player 2 does not know which move Player 1 has chosen.

Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.

We can draw a game tree for this kind of game by

- assuming that one player, say Player 1, moves first and that
- Player 2 does not know which move Player 1 has chosen.

In other words, we can cover simultaneous moves by using imperfect information.

Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.

Player 1 has the choice between three moves.

Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.

Player 1 has the choice between three moves.

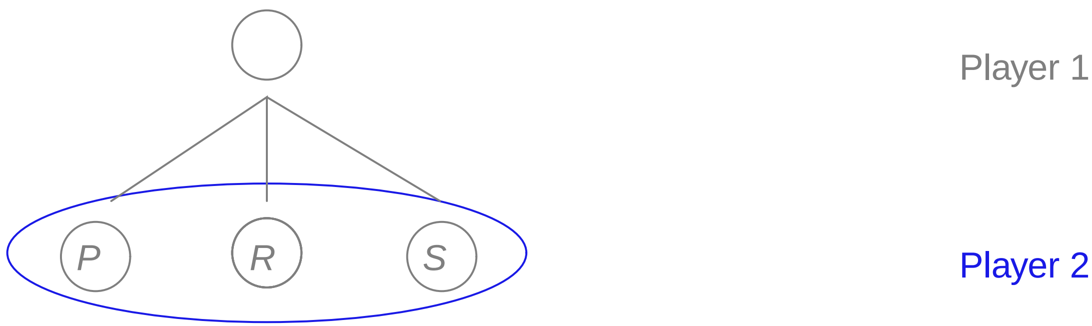
Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.

Player 1 has the choice between three moves. Player 2 cannot distinguish between those.

Paper-Stone-Scissors

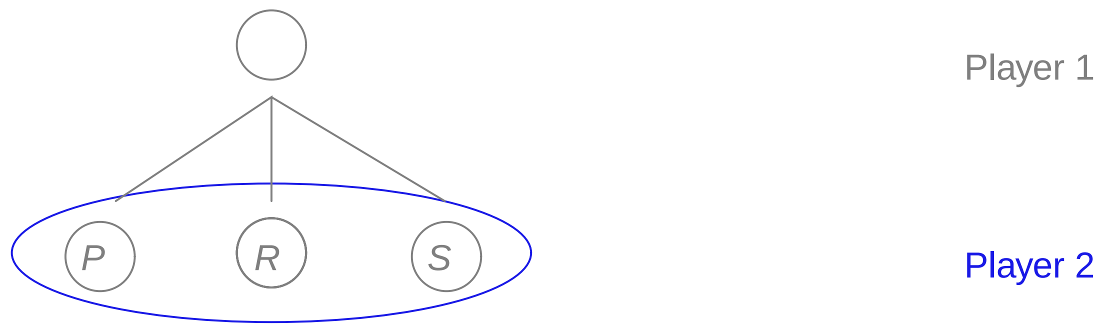
At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.



Player 1 has the choice between three moves. Player 2 cannot distinguish between those.

Paper-Stone-Scissors

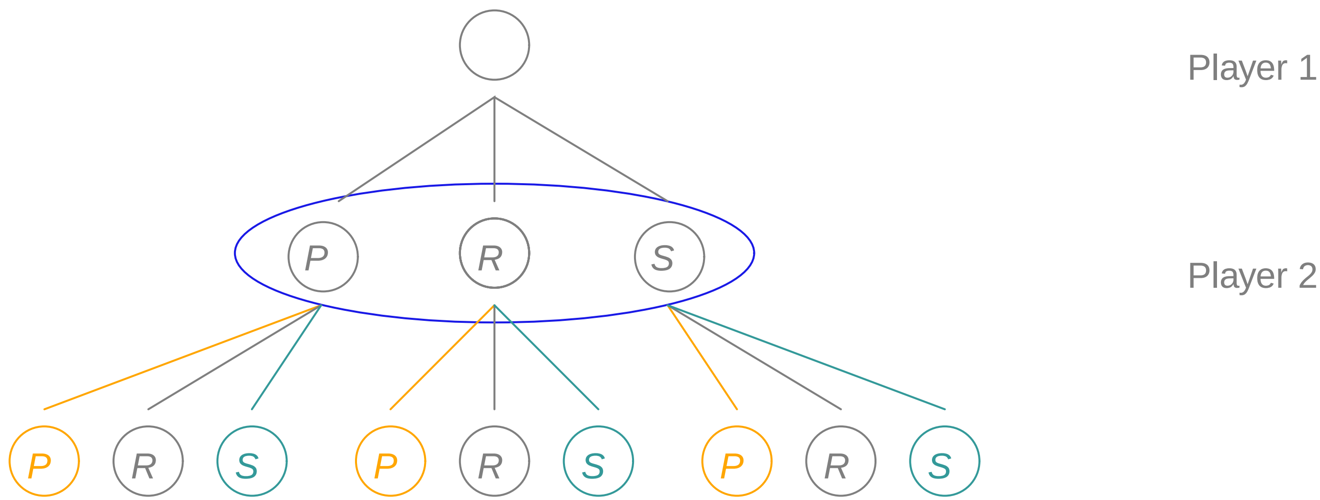
At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.



Player 1 has the choice between three moves. Player 2 cannot distinguish between those. Player 2 has three choices for each of these positions, but they are the same in each case.

Paper-Stone-Scissors

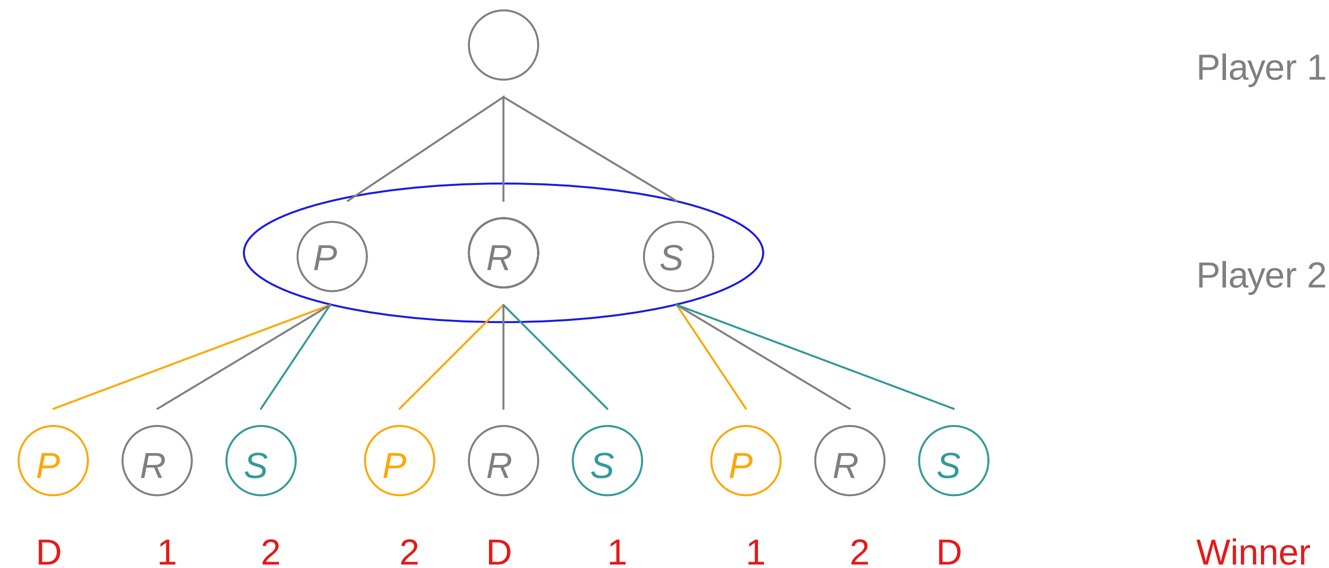
At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.



Player 1 has the choice between three moves. Player 2 cannot distinguish between those. Player 2 has three choices for each of these positions, but they are the same in each case.

Paper-Stone-Scissors

At a signal, two players simultaneously hold out their right hand in one of three ways, indicating whether they have chosen paper, stone or scissors. Paper beats stone beats scissors beats paper.



Player 1 has the choice between three moves. Player 2 cannot distinguish between those. Player 2 has three choices for each of these positions, but they are the same in each case. We mark the winner for each possible play.

Games: Addition to definition

Here is a reminder of our definition of game.

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*
- *for each node of the tree, a player whose turn it is in that position and*
- *for each final node and each player a pay-off function. (We will ignore this part of the definition for the moment.)*

Games: Addition to definition

Here is a reminder of our definition of game.

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*
- *for each node of the tree, a player whose turn it is in that position.*

As additional information, it is possible to indicate groups of nodes, so called information sets, that one player cannot distinguish between. The nodes have to have the property that

Games: Addition to definition

Here is a reminder of our definition of game.

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*
- *for each node of the tree, a player whose turn it is in that position.*

As additional information, it is possible to indicate groups of nodes, so called information sets, that one player cannot distinguish between. The nodes have to have the property that

- for all the nodes in an information set it is the same player's turn, and he is the one who cannot distinguish between them and

Games: Addition to definition

Here is a reminder of our definition of game.

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*
- *for each node of the tree, a player whose turn it is in that position.*

As additional information, it is possible to indicate groups of nodes, so called information sets, that one player cannot distinguish between. The nodes have to have the property that

- for all the nodes in an information set it is the same player's turn, and he is the one who cannot distinguish between them and
- the moves from one of the nodes in the information set are indistinguishable from the moves from any other such node.

Strategies

In every-day language, people often use the term ‘strategy’ when they mean ‘a general plan to proceed’. For our purposes, we will be looking at a much stricter notion. Assume we are a player in some game.

Strategies

In every-day language, people often use the term ‘strategy’ when they mean ‘a general plan to proceed’. For our purposes, we will be looking at a much stricter notion. Assume we are a player in some game.

Slogan: A strategy is a list of instructions which tells me which move to make in any position I may find myself in when it is my turn.

Strategies

In every-day language, people often use the term ‘strategy’ when they mean ‘a general plan to proceed’. For our purposes, we will be looking at a much stricter notion. Assume we are a player in some game.

Slogan: A strategy is a list of instructions which tells me which move to make in any position I may find myself in when it is my turn.

So this is stricter than having just a ‘plan’. However, I do not need to make a decision for positions that I cannot reach (typically because of earlier choices I have made).

Strategies

In every-day language, people often use the term ‘strategy’ when they mean ‘a general plan to proceed’. For our purposes, we will be looking at a much stricter notion. Assume we are a player in some game.

Slogan: A strategy is a list of instructions which tells me which move to make in any position I may find myself in when it is my turn.

So this is stricter than having just a ‘plan’. However, I do not need to make a decision for positions that I cannot reach (typically because of earlier choices I have made).

Note that nothing in this notion of strategy says that I have to follow some sort of principle!

Strategies

In every-day language, people often use the term ‘strategy’ when they mean ‘a general plan to proceed’. For our purposes, we will be looking at a much stricter notion. Assume we are a player in some game.

Slogan: A strategy is a list of instructions which tells me which move to make in any position I may find myself in when it is my turn.

So this is stricter than having just a ‘plan’. However, I do not need to make a decision for positions that I cannot reach (typically because of earlier choices I have made).

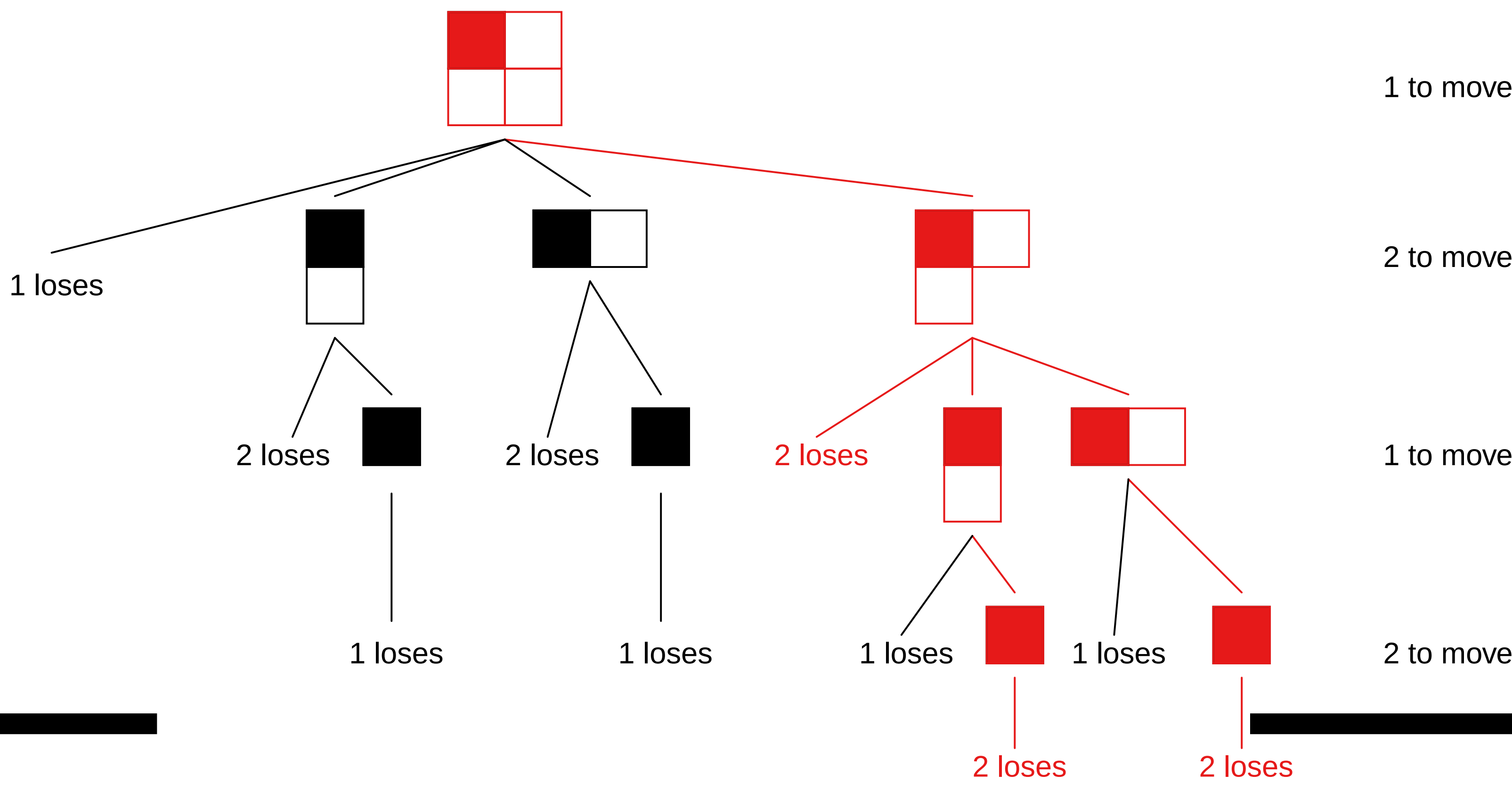
Note that nothing in this notion of strategy says that I have to follow some sort of principle! In particular I am allowed to choose different moves to make in positions which look the same on the board, but have different histories.

Example: Strategy on (2×2)-Chomp

Here is an example for a strategy for Player 1 in (2×2)-Chomp.

Example: Strategy on (2×2)-Chomp

Here is an example for a strategy for Player 1 in (2×2)-Chomp.

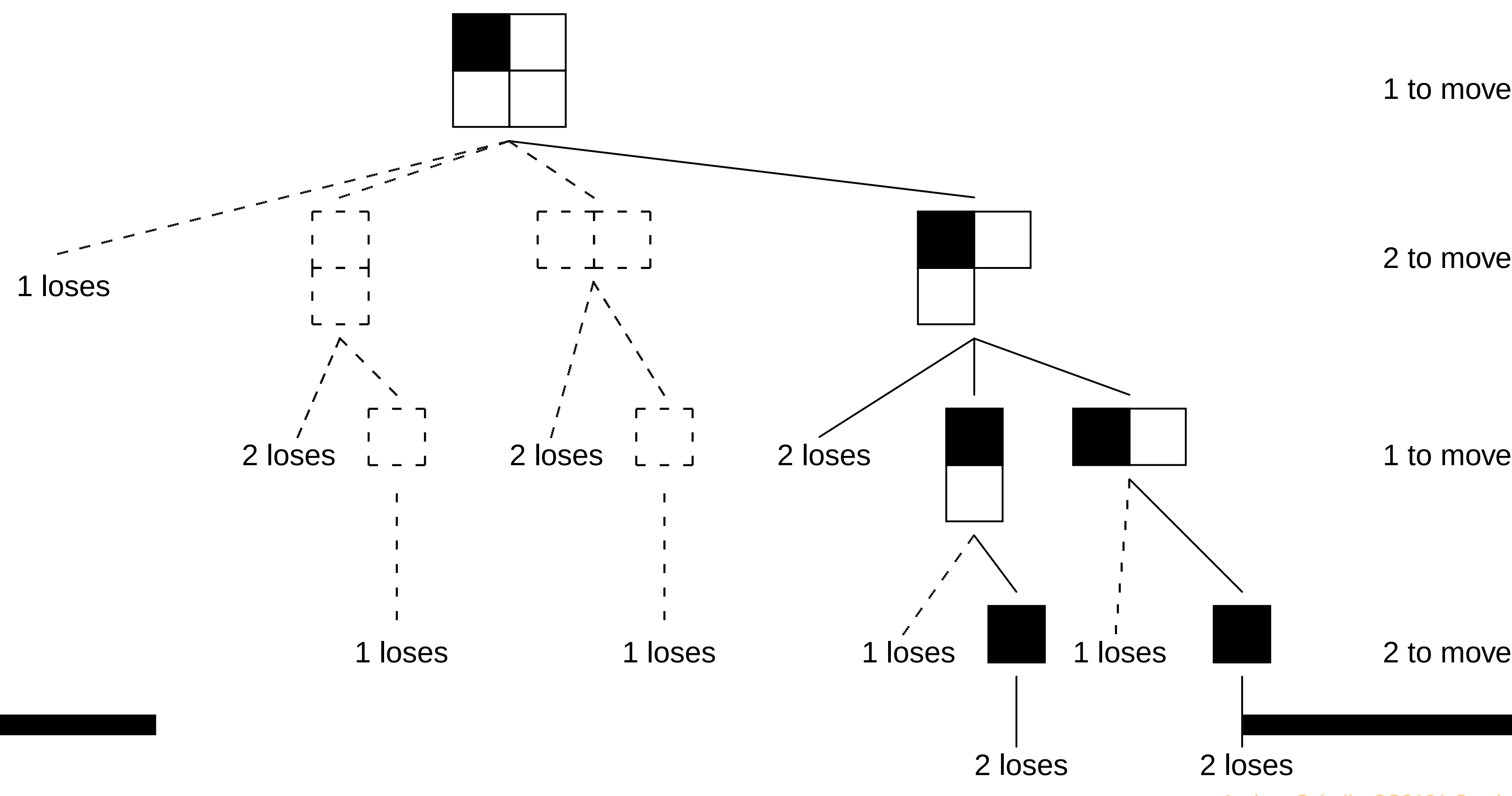


Example: Strategy on (2×2)-Chomp

In the notes, the same strategy is displayed something like this.

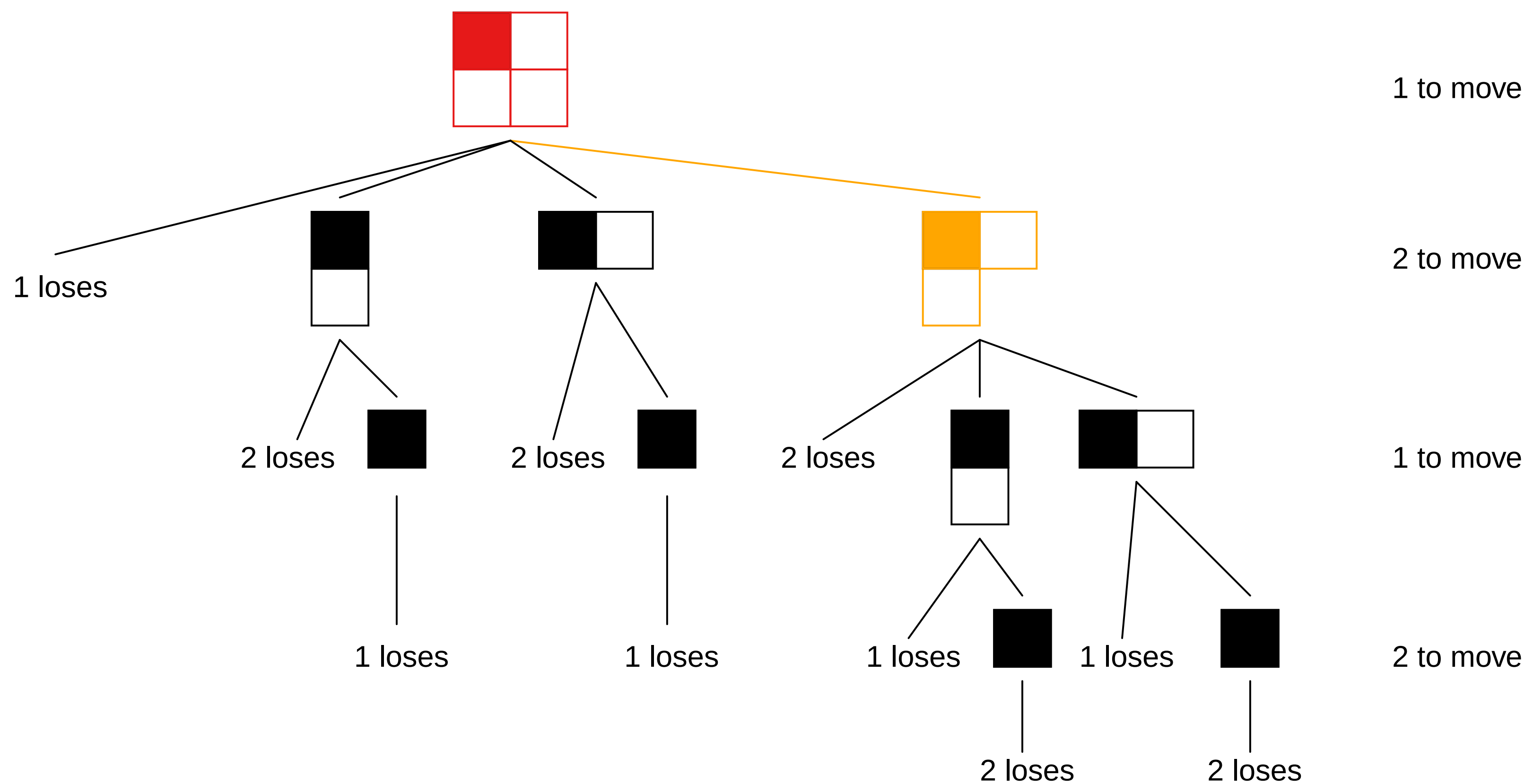
Example: Strategy on (2×2)-Chomp

In the notes, the same strategy is displayed something like this.



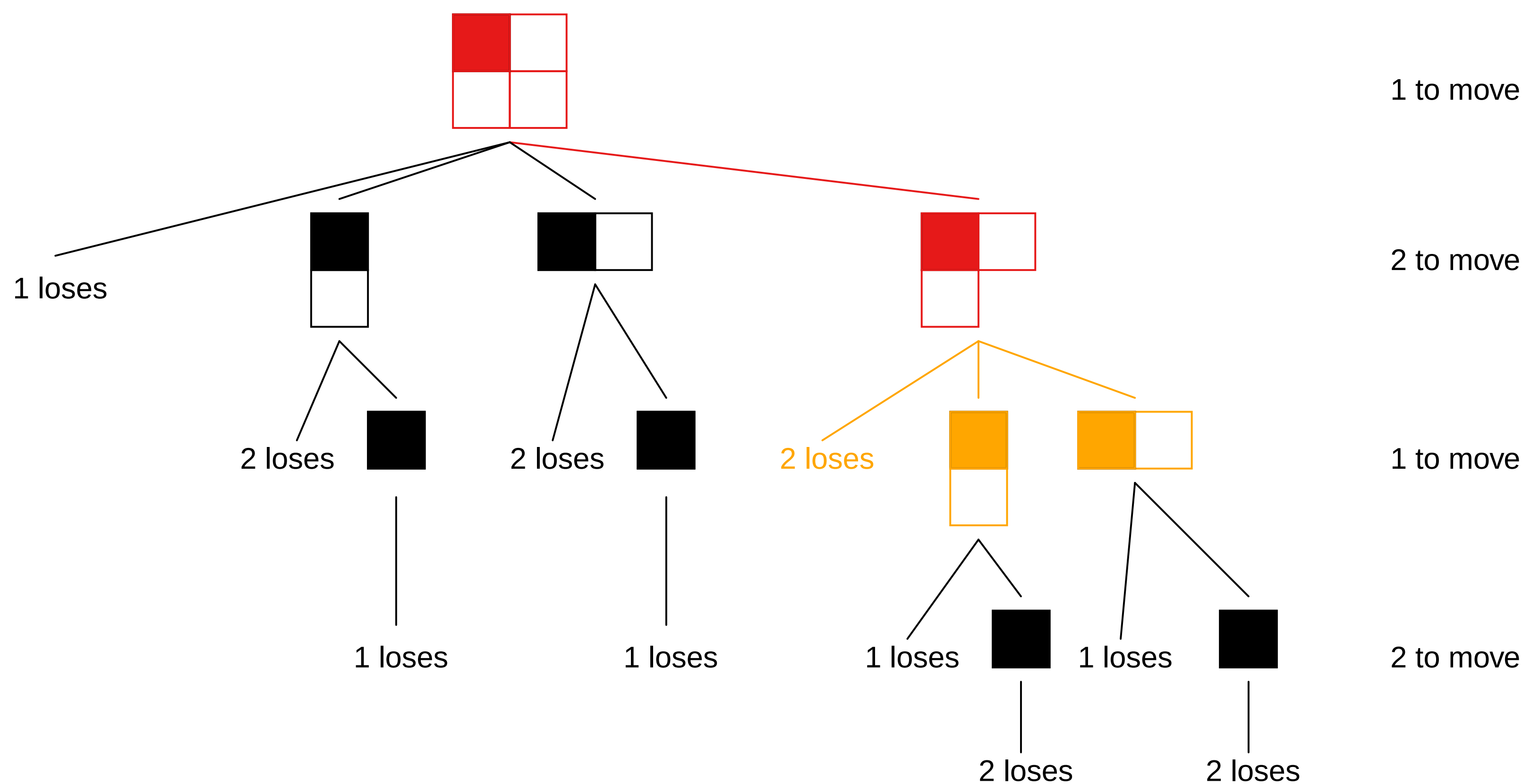
Example: Strategy on (2×2)-Chomp

It is Player 1's turn at the start, so he has to choose a move.



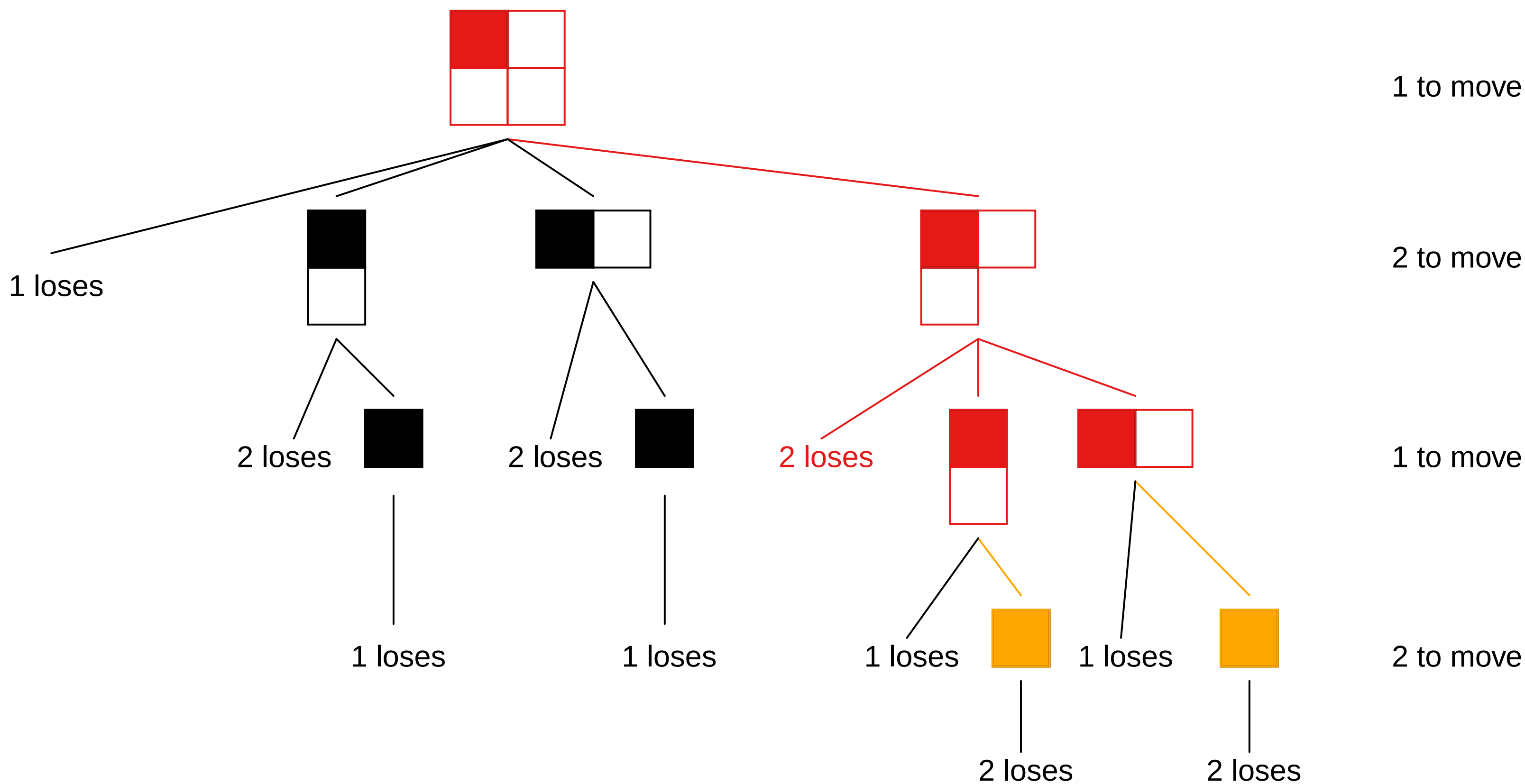
Example: Strategy on (2×2)-Chomp

It is Player 1's turn at the start, so he has to choose a move. It is Player 2's turn next, so Player 1 must allow for all of her possibilities.



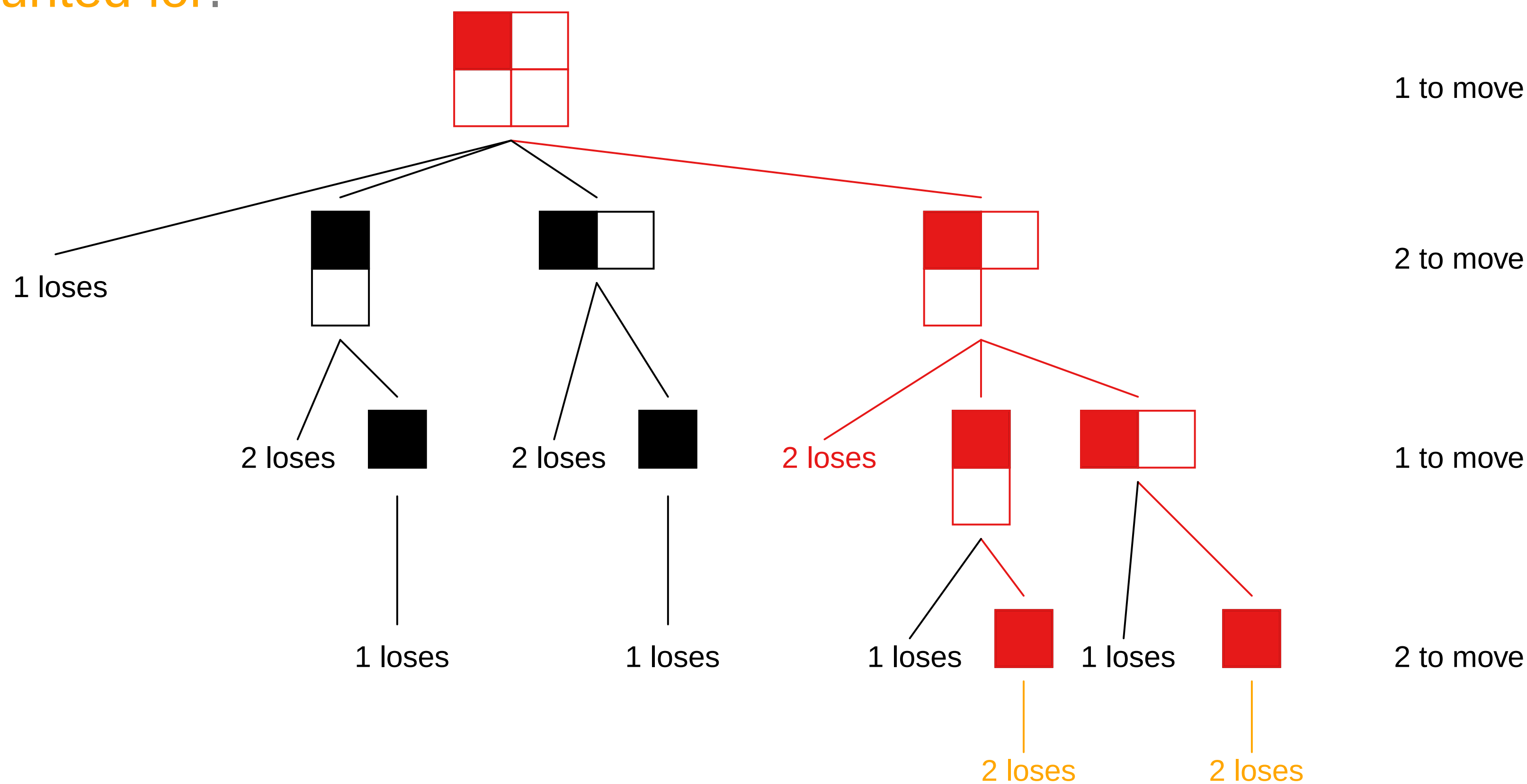
Example: Strategy on (2×2)-Chomp

It is Player 1's turn at the start, so he has to choose a move. It is Player 2's turn next, so Player 1 must allow for all of her possibilities. Again, Player 1 must make a choice for each position he might be in after move 2.



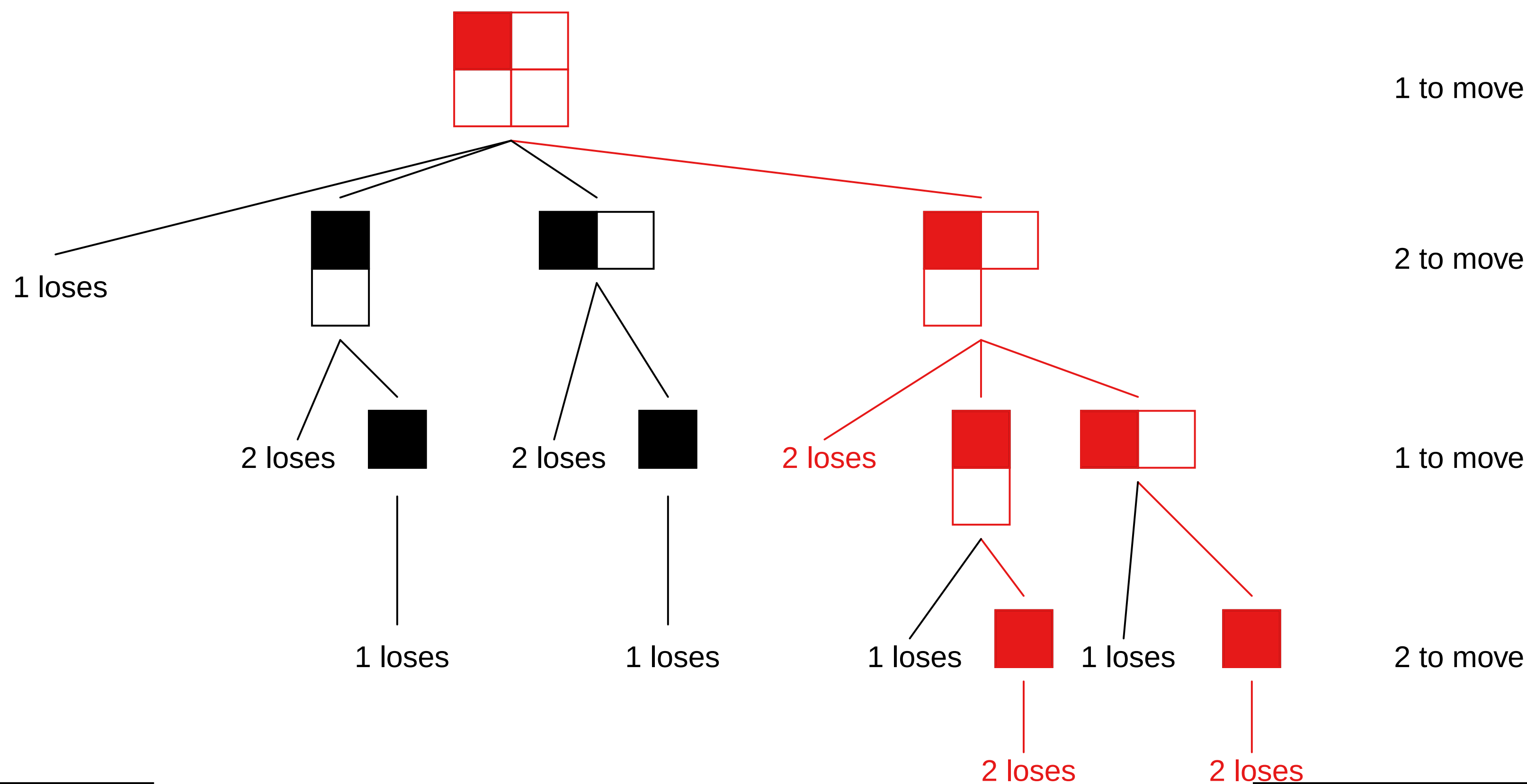
Example: Strategy on (2×2)-Chomp

It is Player 1's turn at the start, so he has to choose a move. It is Player 2's turn next, so Player 1 must allow for all of her possibilities. Again, Player 1 must make a choice for each position he might be in after 2 moves. And finally, Player 2's last potential move has to be accounted for.



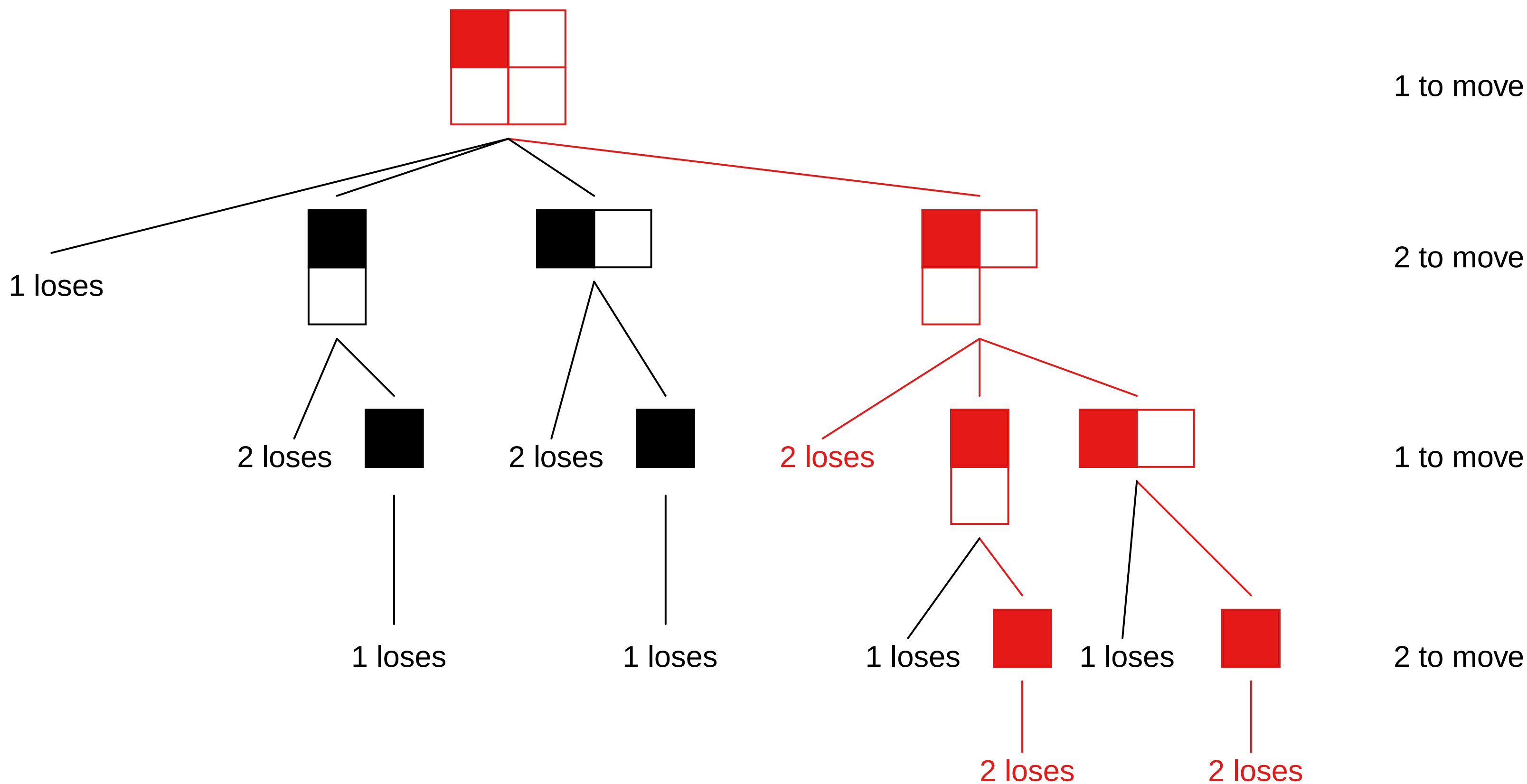
Example: Strategy on (2×2)-Chomp

It is Player 1's turn at the start, so he has to choose a move. It is Player 2's turn next, so Player 1 must allow for all of her possibilities. Again, Player 1 must make a choice for each position he might be in after 2 moves. And finally, Player 2's last potential move has to be accounted for.



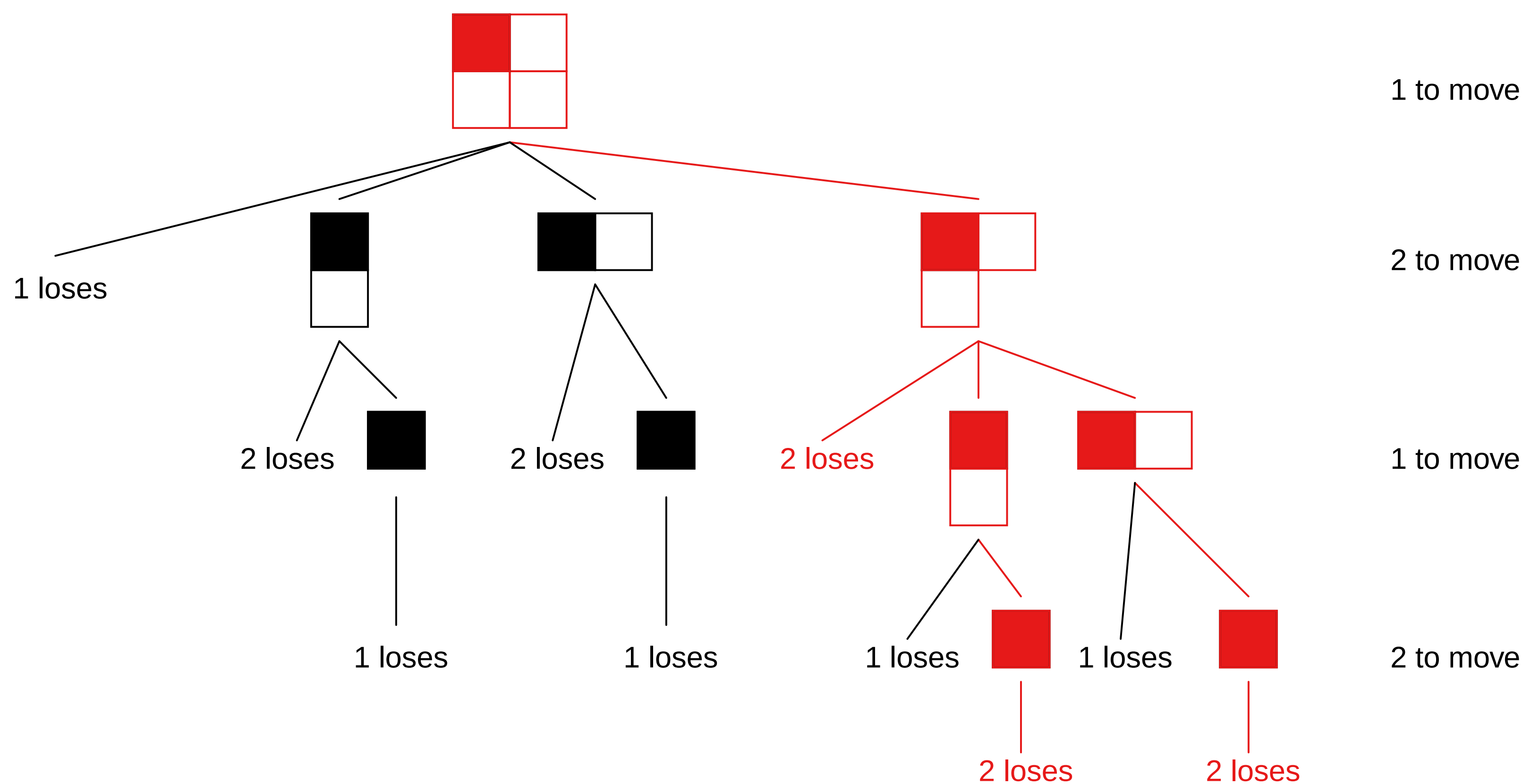
Example

Question. How many possible outcomes (final positions) does playing in accord with this strategy have?



Example

Question. How many possible outcomes (final positions) does playing in accord with this strategy have? How many are advantageous to Player 1?



Strategies: towards a definition

Closer inspection shows that a strategy can be described as subtree of the game tree satisfying the following conditions. Assume we have chosen a player for whom we want to give a strategy. Then we can build a subtree, starting from the root, where when we reach a new node,

Strategies: towards a definition

Closer inspection shows that a strategy can be described as subtree of the game tree satisfying the following conditions. Assume we have chosen a player for whom we want to give a strategy. Then we can build a subtree, starting from the root, where when we reach a new node,

- if it is the chosen player's turn, precisely one of the available moves is chosen;

Strategies: towards a definition

Closer inspection shows that a strategy can be described as subtree of the game tree satisfying the following conditions. Assume we have chosen a player for whom we want to give a strategy. Then we can build a subtree, starting from the root, where when we reach a new node,

- if it is the chosen player's turn, precisely one of the available moves is chosen;
- if it is not the chosen player's turn, all available moves are chosen.

Strategies: towards a definition

Closer inspection shows that a strategy can be described as subtree of the game tree satisfying the following conditions. Assume we have chosen a player for whom we want to give a strategy. Then we can build a subtree, starting from the root, where when we reach a new node,

- if it is the chosen player's turn, precisely one of the available moves is chosen;
- if it is not the chosen player's turn, all available moves are chosen.

Note that we do not worry about positions we can never reach!

Strategies: towards a definition

Closer inspection shows that a strategy can be described as subtree of the game tree satisfying the following conditions. Assume we have chosen a player for whom we want to give a strategy. Then we can build a subtree, starting from the root, where when we reach a new node,

- if it is the chosen player's turn, precisely one of the available moves is chosen;
- if it is not the chosen player's turn, all available moves are chosen.

Note that we do not worry about positions we can never reach!

Also note that we force a strategy to make a choice of a move if it is the chosen player's turn—we do not allow 'give up by refusing to make a move'. If we want giving up to be an option, we have to make it a move in the game tree.

Strategies: definition

We can turn this observation into a definition.

Definition 2 *A strategy for player X is a subtree of a game tree which satisfies the following conditions.*

Strategies: definition

We can turn this observation into a definition.

Definition 2 *A strategy for player X is a subtree of a game tree which satisfies the following conditions.*

- *It is rooted at the root of the game tree;*

Strategies: definition

We can turn this observation into a definition.

Definition 2 *A strategy for player X is a subtree of a game tree which satisfies the following conditions.*

- *It is rooted at the root of the game tree;*
- *whenever it is player X's turn at a node that belongs to the subtree, exactly one of the available moves (and so exactly one successor node) belongs to the subtree;*

Strategies: definition

We can turn this observation into a definition.

Definition 2 *A* *strategy for player* X *is a subtree of a game tree which satisfies the following conditions.*

- *It is rooted at the root of the game tree;*
- *whenever it is player* X*'s turn at a node that belongs to the subtree, exactly one of the available moves (and so exactly one successor node) belongs to the subtree;*
- *whenever it is not player* X*'s turn at a node that belongs to the subtree, all of the available moves (and so all successor nodes) belong to the subtree.*

Strategies: definition

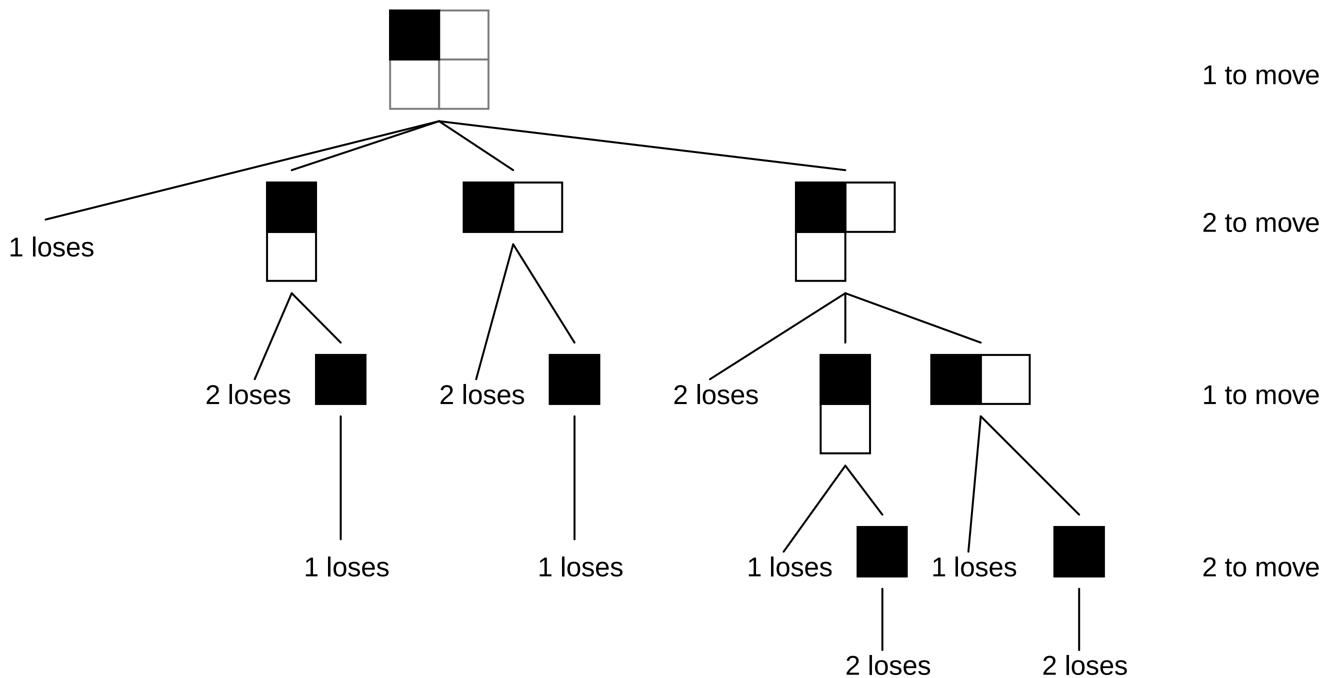
We can turn this observation into a definition.

Definition 2 *A strategy for player X is a subtree of a game tree which satisfies the following conditions.*

- *It is rooted at the root of the game tree;*
- *whenever it is player X's turn at a node that belongs to the subtree, exactly one of the available moves (and so exactly one successor node) belongs to the subtree;*
- *whenever it is not player X's turn at a node that belongs to the subtree, all of the available moves (and so all successor nodes) belong to the subtree.*
- *For all nodes in the same information set for Player X, the move chosen by the strategy is the same.*

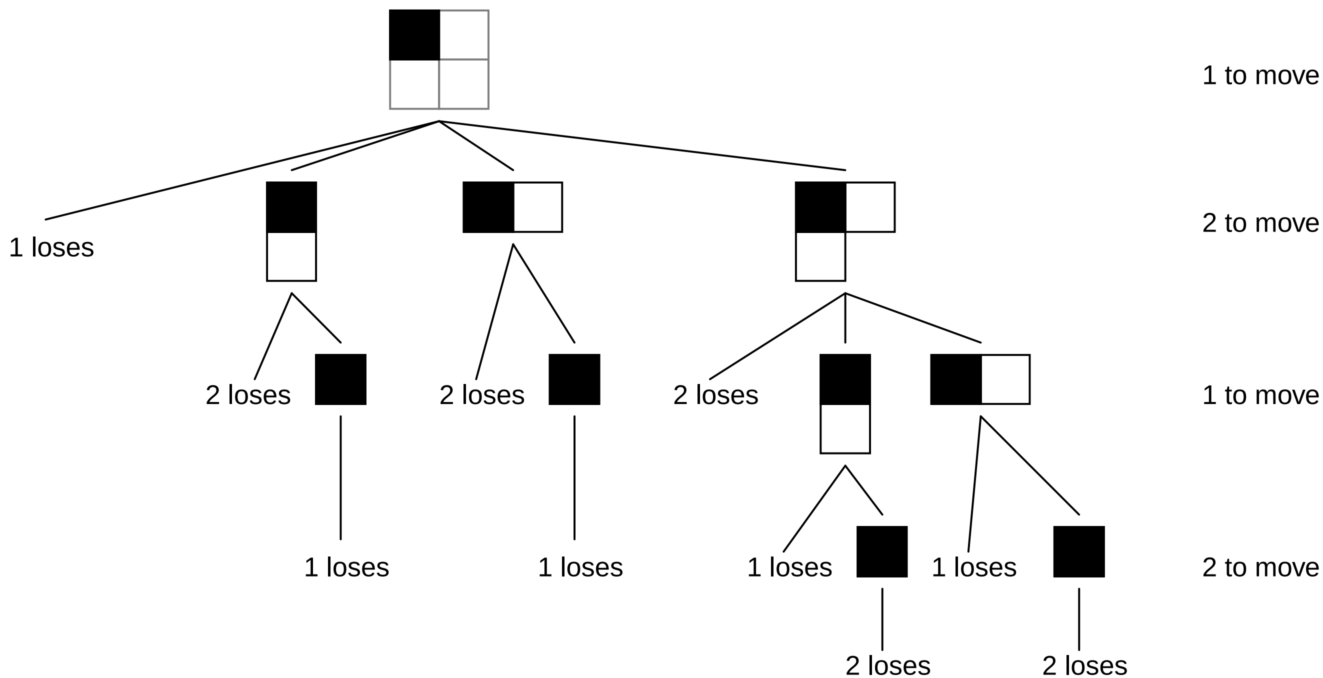
Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp?



Different strategies

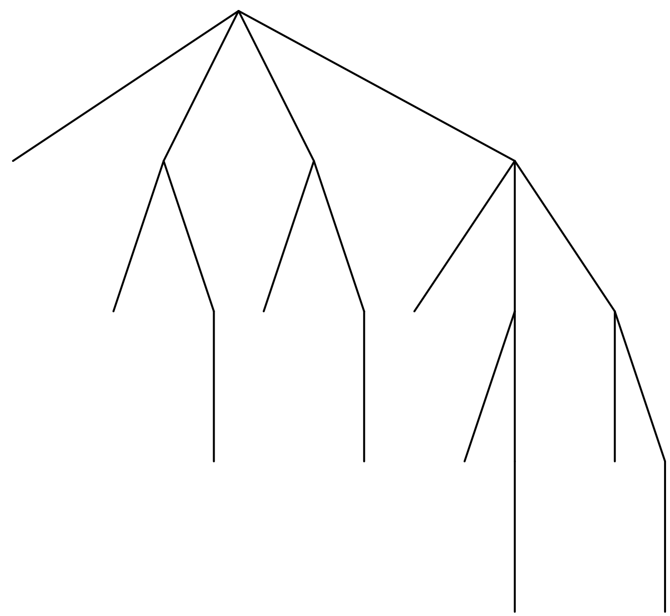
How many strategies are there for Player 1 in (2×2)-Chomp? What are they?



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

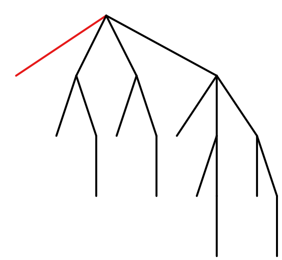
If we remember that Player 1 starts the game and they then move alternatingly, we can think of the game tree for (2×2)-Chomp like this:



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

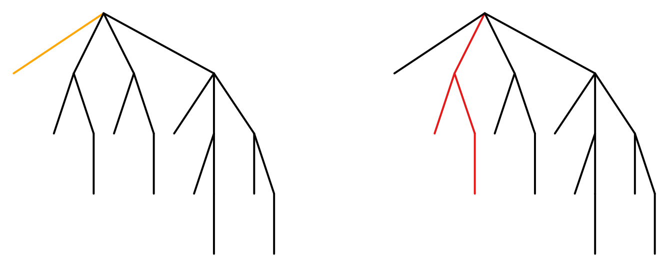
Player 1 chooses the left-most move.



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

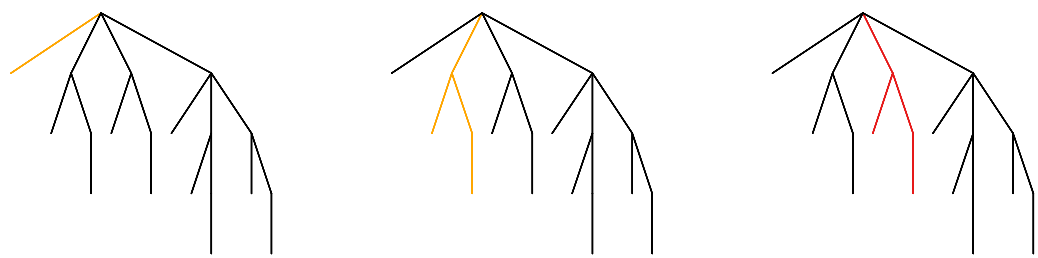
Player 1 chooses the second move from the left.



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

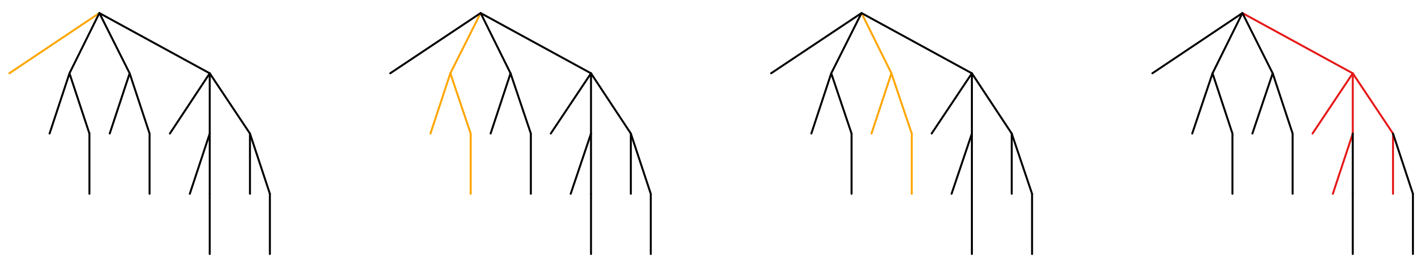
Player 1 chooses the third move from the left.



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

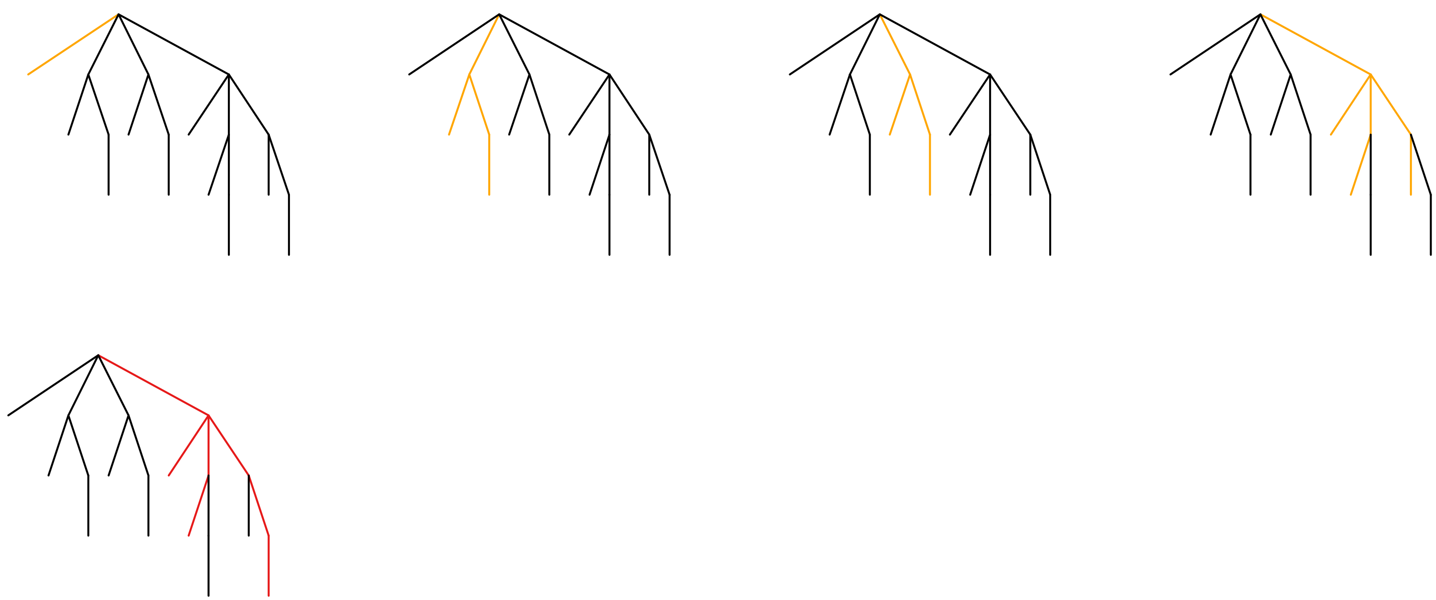
Player 1 chooses the right-most move. But that means there are further choices down the line! Two decision points with two choices each, that gives $2 \times 2 = 4$ strategies.



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

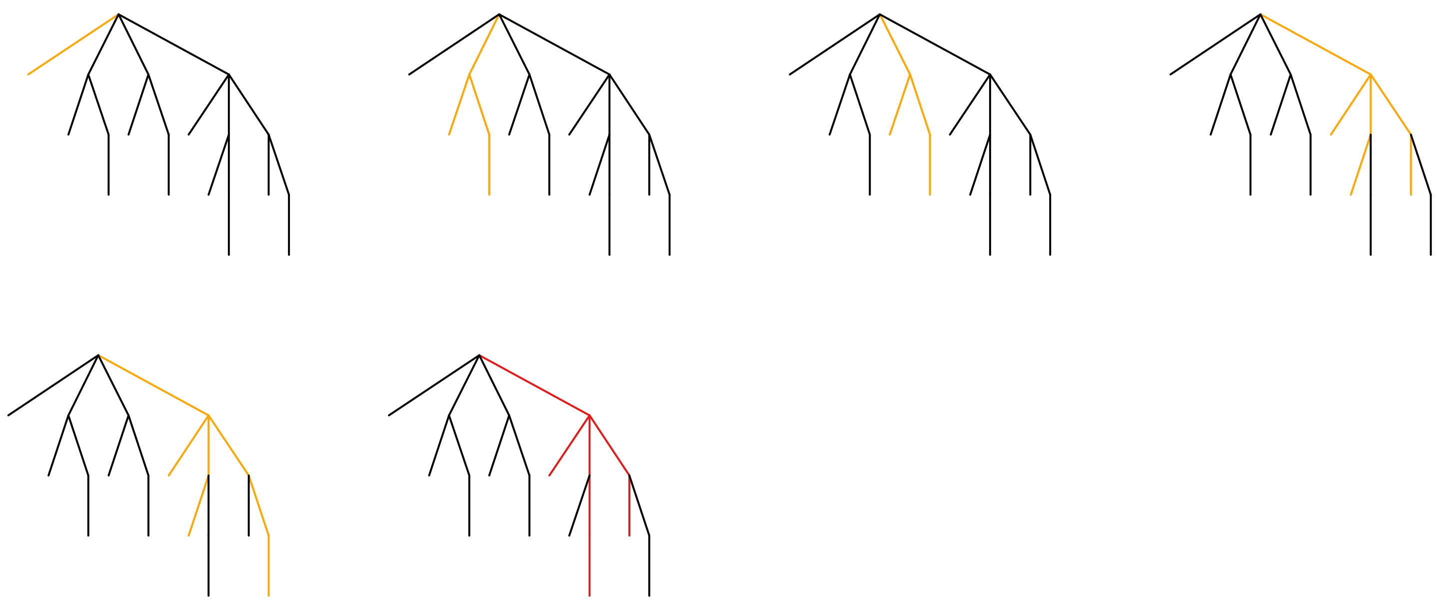
Player 1 chooses the right-most move. But that means there are further choices down the line! Two decision points with two choices each, that gives $2 \times 2 = 4$ strategies.



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

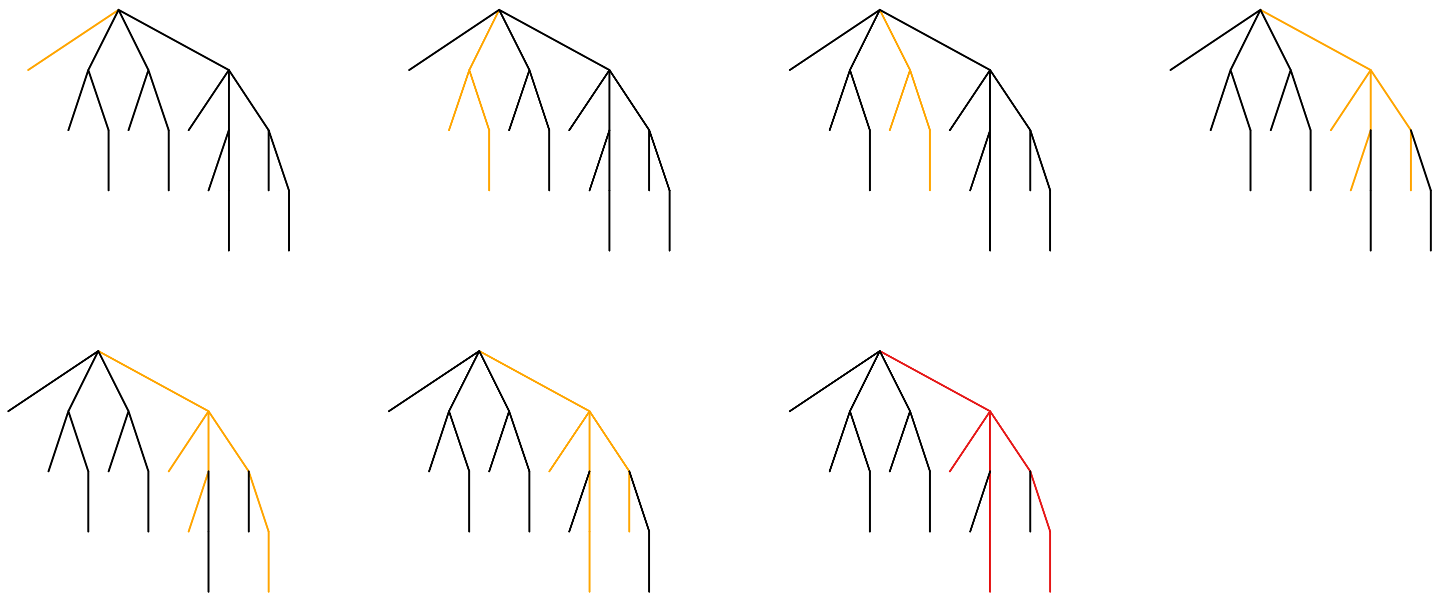
Player 1 chooses the right-most move. But that means there are further choices down the line! Two decision points with two choices each, that gives $2 \times 2 = 4$ strategies.



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

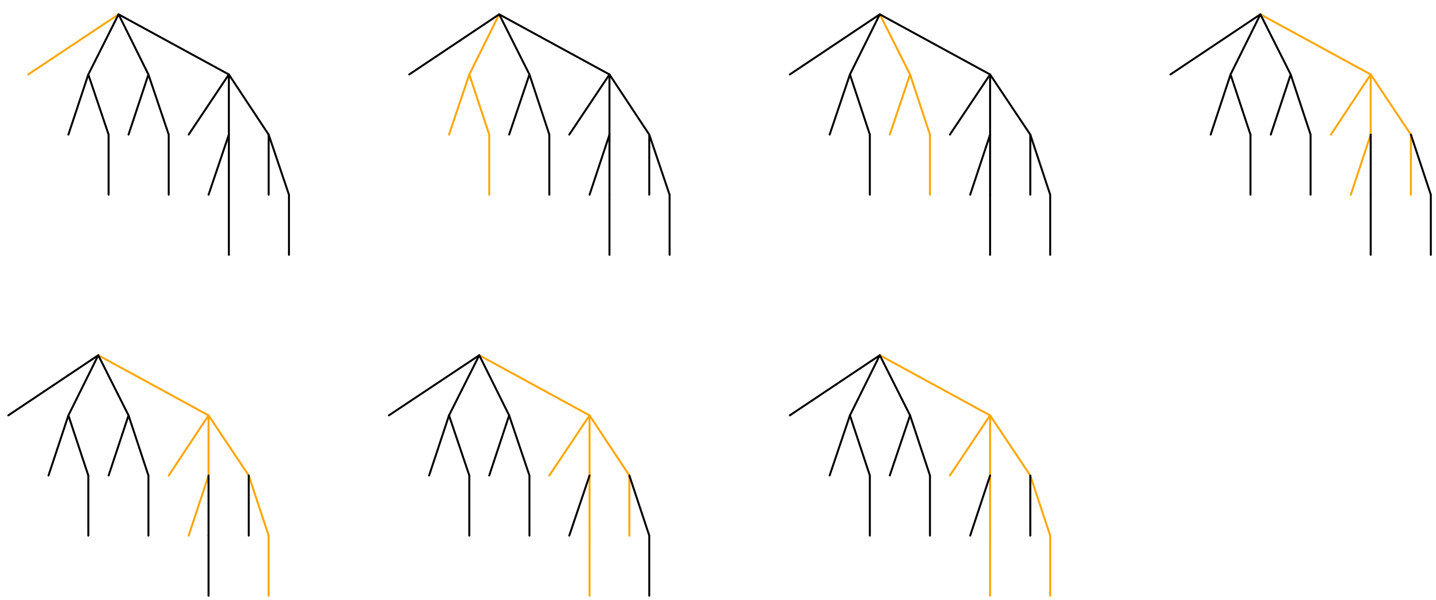
Player 1 chooses the right-most move. But that means there are further choices down the line! Two decision points with two choices each, that gives $2 \times 2 = 4$ strategies.



Different strategies

How many strategies are there for Player 1 in (2×2)-Chomp? What are they?

Player 1 chooses the right-most move. But that means there are further choices down the line! Two decision points with two choices each, that gives $2 \times 2 = 4$ strategies.



That makes seven strategies altogether.

Strategies and imperfect information.

Recall that in our definition of strategy, we stipulated that for all nodes in the same information set for Player X, the move chosen by the strategy is the same.

Strategies and imperfect information.

Recall that in our definition of strategy, we stipulated that for all nodes in the same information set for Player X, the move chosen by the strategy is the same.

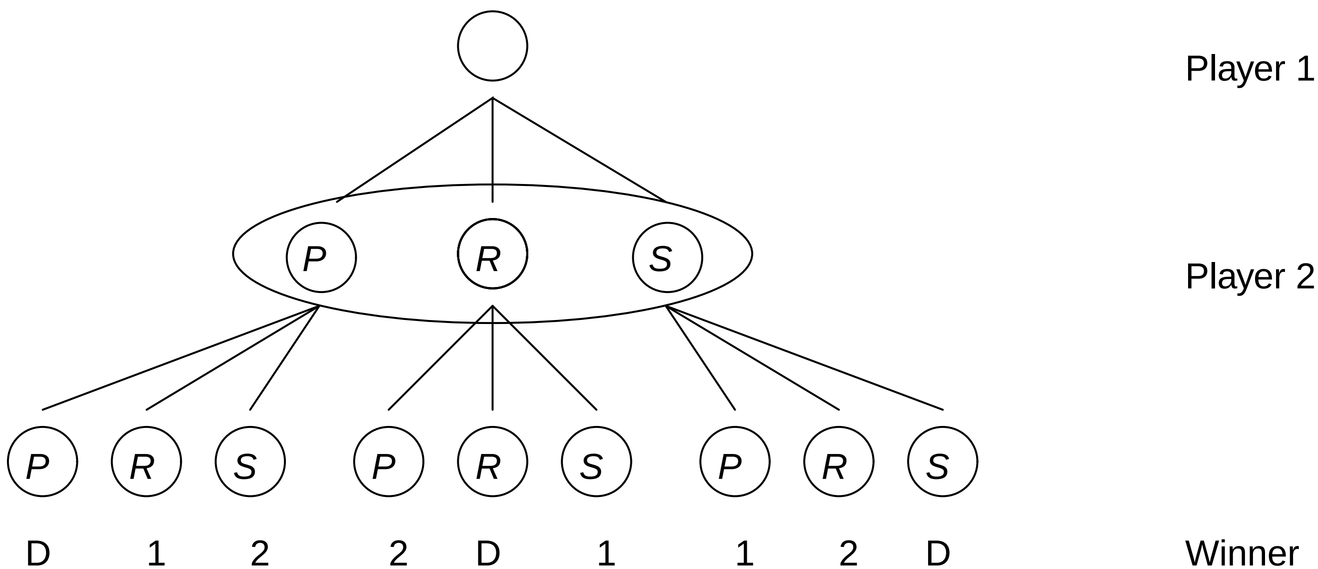
This is because Player X is not allowed to make decisions in his strategy based on information which he doesn't have.

Strategies and imperfect information.

Recall that in our definition of strategy, we stipulated that for all nodes in the same information set for Player X, the move chosen by the strategy is the same.

This is because Player X is not allowed to make decisions in his strategy based on information which he doesn't have.

We consider strategies for Player 2 of Paper-Stone-Scissors.

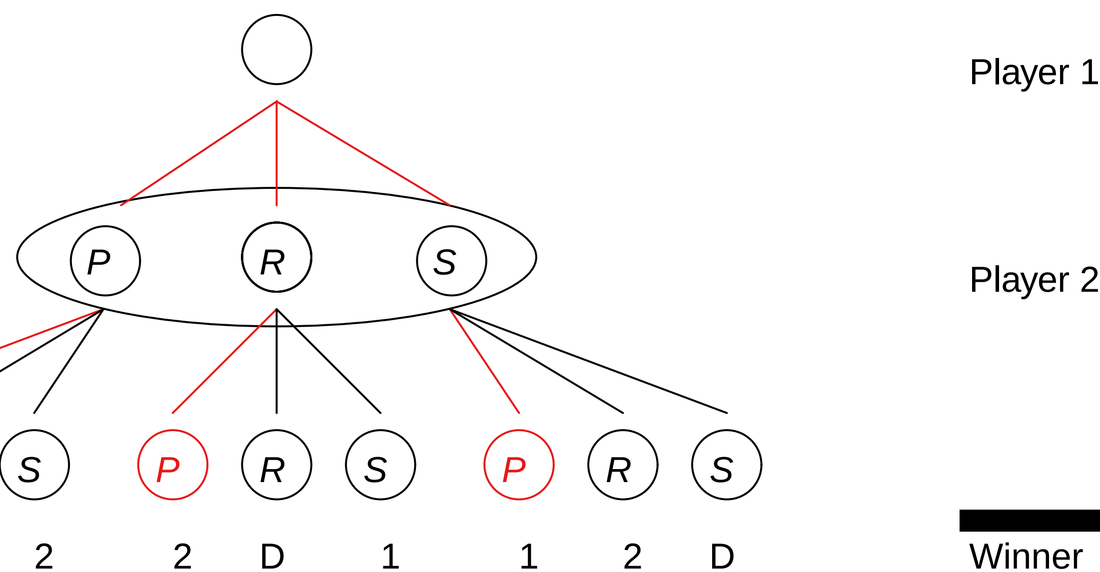


Strategies and imperfect information.

Recall that in our definition of strategy, we stipulated that for all nodes in the same information set for Player X, the move chosen by the strategy is the same.

This is because Player X is not allowed to make decisions in his strategy based on information which he doesn't have.

We consider strategies for Player 2 of Paper-Stone-Scissors. Player 2 might decide to show P for paper. She will have to do that for all the nodes in her information set.

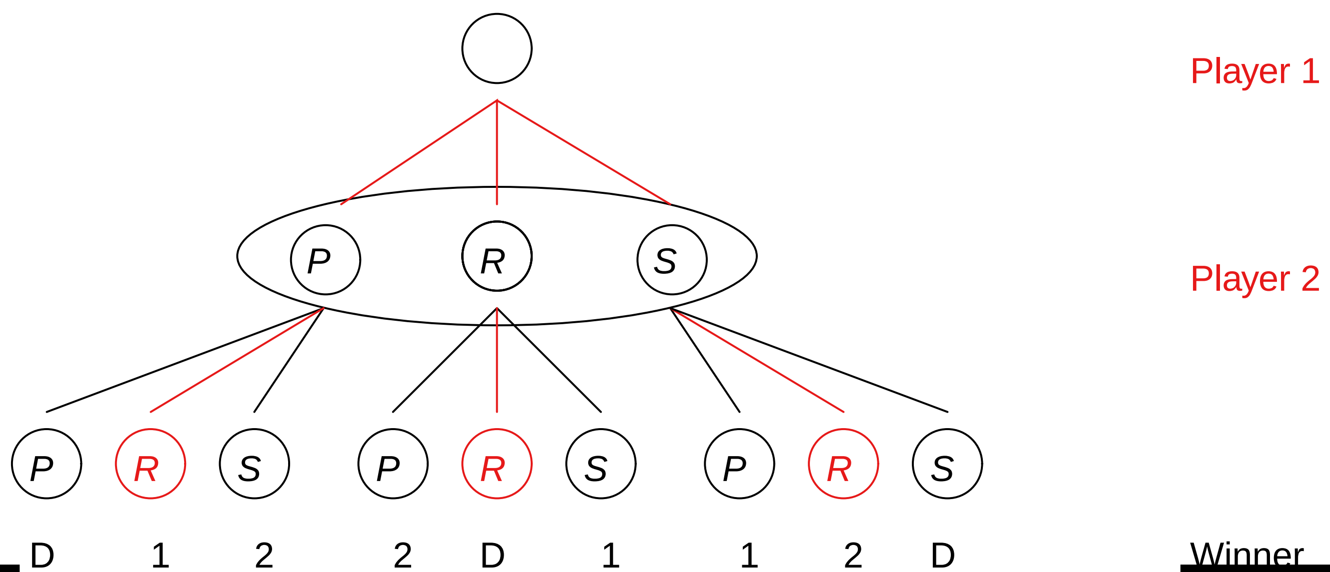


Strategies and imperfect information.

Recall that in our definition of strategy, we stipulated that for all nodes in the same information set for Player X, the move chosen by the strategy is the same.

This is because Player X is not allowed to make decisions in his strategy based on information which he doesn't have.

We consider strategies for Player 2 of Paper-Stone-Scissors. Similarly she may decide to show R for stone.

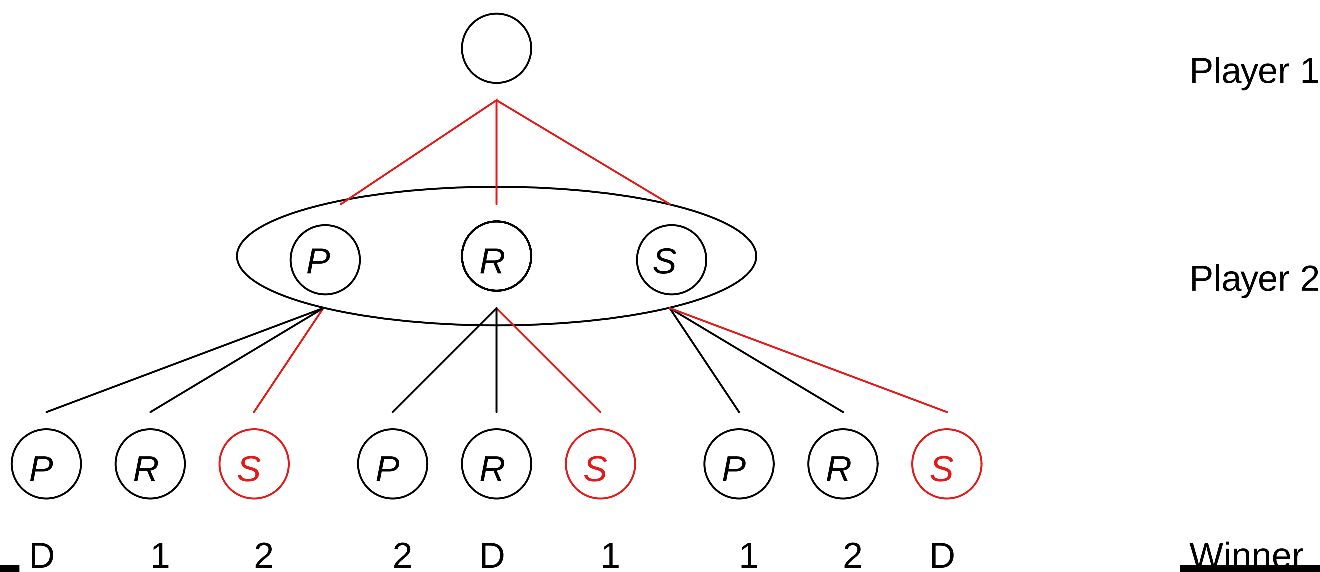


Strategies and imperfect information.

Recall that in our definition of strategy, we stipulated that for all nodes in the same information set for Player X, the move chosen by the strategy is the same.

This is because Player X is not allowed to make decisions in his strategy based on information which he doesn't have.

We consider strategies for Player 2 of Paper-Stone-Scissors. Lastly she may choose to show S for scissors.

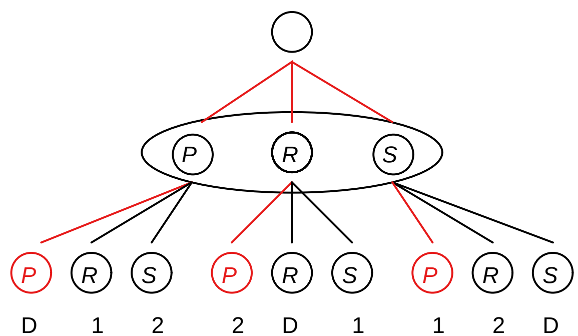


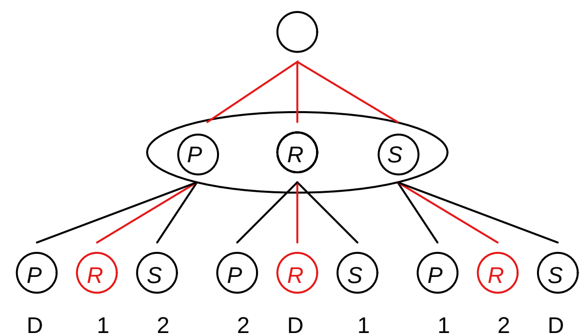
Strategies and imperfect information.

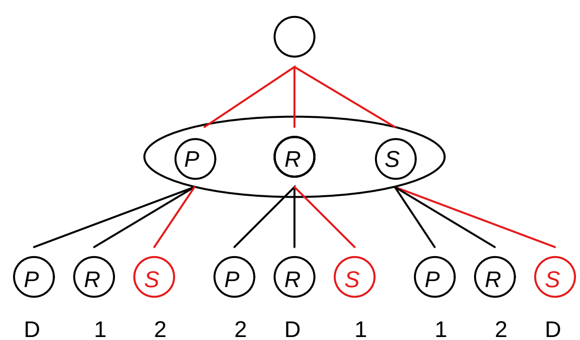
These are all the strategies for Player 2 in Paper-Stone-Scissors. Again they show that Player 2 cannot use information she does not have; her moves for two nodes in the same information set are the same.

Strategies and imperfect information.

These are all the strategies for Player 2 in Paper-Stone-Scissors.Again they show that Player 2 cannot use information she does not have; her moves for two nodes in the same information set are the same.

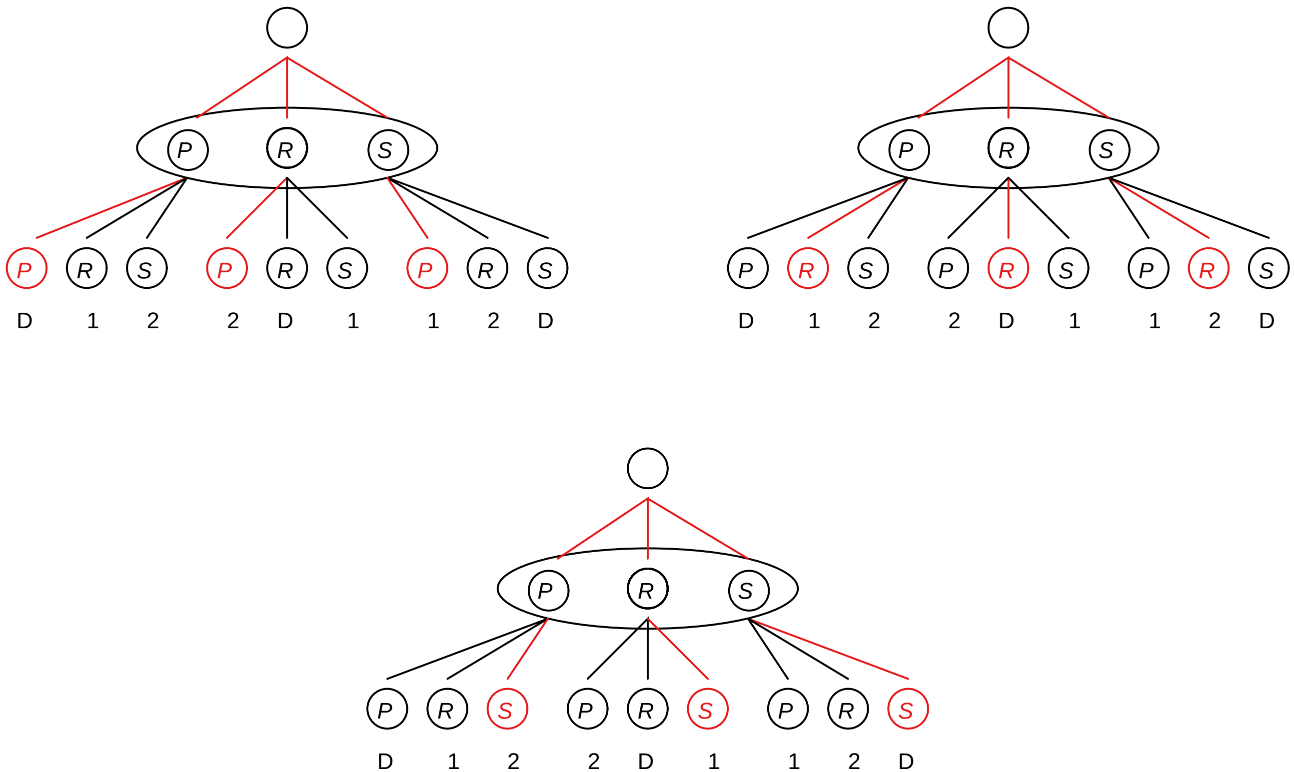






Strategies and imperfect information.

These are all the strategies for Player 2 in Paper-Stone-Scissors.



So she has the same choices as Player 1, preserving the symmetry of the game as it is defined informally. So we really have not changed the game by using *via* a game tree.

Generating all strategies

Let's assume we are given a game tree t and are trying to generate all strategies for Player X. This can be done recursively. (Note that when the game is not of perfect information then the following considerations have to be changed accordingly.)

Generating all strategies

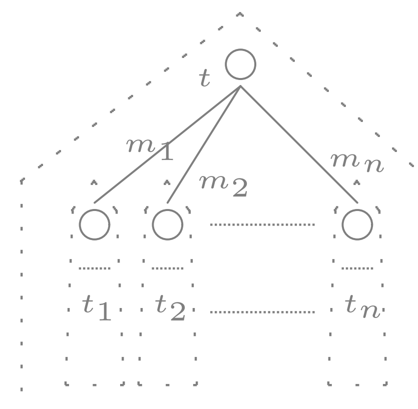
Let's assume we are given a game tree t and are trying to generate all strategies for Player X. This can be done recursively.

- There are no moves: There is only the 'do nothing' strategy for each player.

Generating all strategies

Let's assume we are given a game tree t and are trying to generate all strategies for Player X. This can be done recursively.

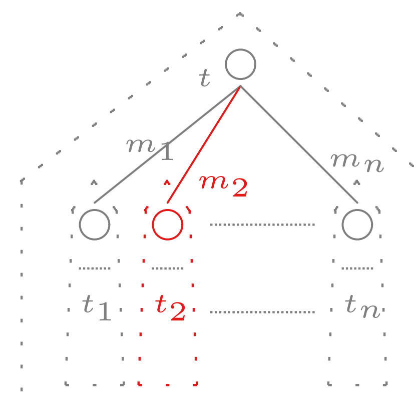
- There are no moves:
- The first move is Player X's: He has an according number of choices (m_1 to m_n). After that move is made, we get a new game tree, one of t_1 to t_n, and Player X can now use any of his strategies for the game following his chosen first move.



Generating all strategies

Let's assume we are given a game tree t and are trying to generate all strategies for Player X. This can be done recursively.

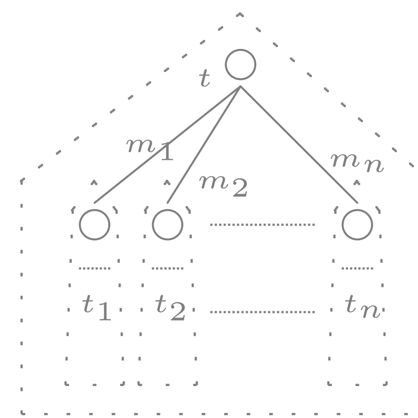
- There are no moves:
- The first move is Player X's: He has an according number of choices (m_1 to m_n). After that move is made, we get a new game tree, one of t_1 to t_n, and Player X can now use any of his strategies for the game following his chosen first move. Giving a strategy for t amounts to choosing one of m_i and then one of the strategies on t_i.



Generating all strategies

Let's assume we are given a game tree t and are trying to generate all strategies for Player X. This can be done recursively.

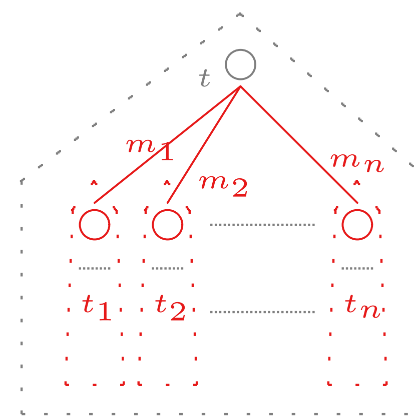
- There are no moves:
- The first move is Player X's:
- The first move is not Player X's: Player X has to choose a strategy in every possible subtree, t_1 to t_n, since he has to prepare an answer for every of the possible first moves.



Generating all strategies

Let's assume we are given a game tree t and are trying to generate all strategies for Player X. This can be done recursively.

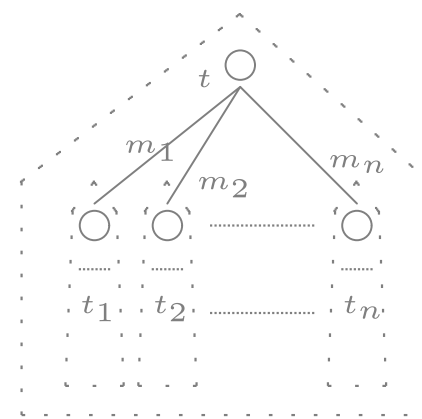
- There are no moves:
- The first move is Player X's:
- The first move is not Player X's: Player X has to choose a strategy in every possible subtree, t_1 to t_n, since he has to prepare an answer for every of the possible first moves. Giving a strategy amounts to choosing one strategy for each of the t_i.



Generating all strategies

Let's assume we are given a game tree t and are trying to generate all strategies for Player X. This can be done recursively.

- There are no moves:
- The first move is Player X's:
- The first move is not Player X's: Player X has to choose a strategy in every possible subtree, t_1 to t_n, since he has to prepare an answer for every of the possible first moves. Giving a strategy amounts to choosing one strategy for each of the t_i.



Note that this does account for chance moves.

Counting all strategies

Let's assume we are are now trying to count all strategies for Player X for game t. This can also be done recursively, with the number of strategies for Player X on game t being denoted by $N_X(t)$. (Note that when the game is not of perfect information then once again the following considerations have to be changed accordingly.)

Counting all strategies

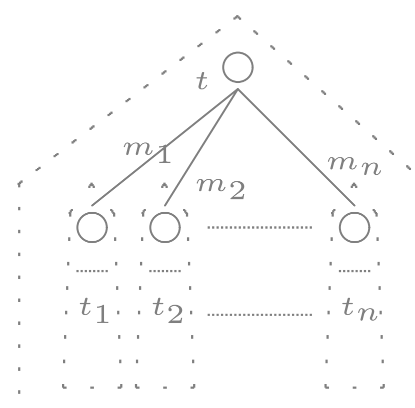
Let's assume we are are now trying to count all strategies for Player X for game t. This can also be done recursively, with the number of strategies for Player X on game t being denoted by $N_X(t)$.

- There are no moves: It has one strategy for each player.

Counting all strategies

Let's assume we are are now trying to count all strategies for Player X for game t. This can also be done recursively, with the number of strategies for Player X on game t being denoted by $N_X(t)$.

- There are no moves:
- The first move is Player X's: He has an according number of choices (m_1 to m_n). After that move is made, we get a new game tree, one of t_1 to t_n, and Player X can now use any of his strategies for the game following his chosen first move.

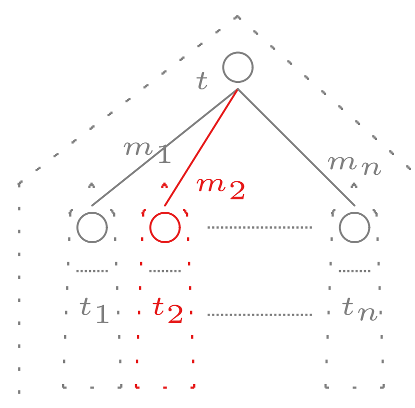


Counting all strategies

Let's assume we are are now trying to count all strategies for Player X for game t. This can also be done recursively, with the number of strategies for Player X on game t being denoted by $N_X(t)$.

- There are no moves:
- The first move is Player X's: He has an according number of choices (m_1 to m_n). After that move is made, we get a new game tree, one of t_1 to t_n, and Player X can now use any of his strategies for the game following his chosen first move. Then

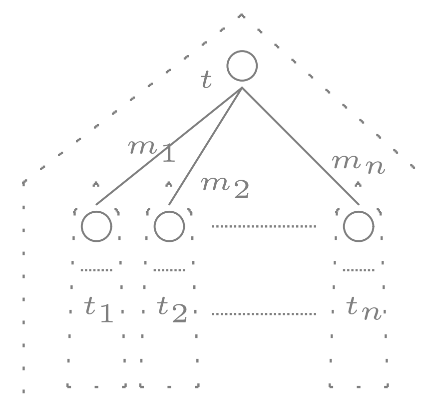
$$N_X(t) = N_X(t_1) + N_X(t_2) + \cdots + N_X(t_n).$$



Counting all strategies

Let's assume we are are now trying to count all strategies for Player X for game t. This can also be done recursively, with the number of strategies for Player X on game t being denoted by $N_X(t)$.

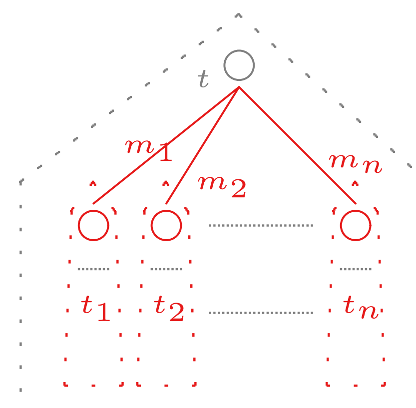
- There are no moves:
- The first move is Player X's:
- The first move is not Player X's: Player X has to choose a strategy in every possible subtree, t_1 to t_n, since he has to prepare an answer for every of the possible first moves.



Counting all strategies

Let's assume we are are now trying to count all strategies for Player X for game t. This can also be done recursively, with the number of strategies for Player X on game t being denoted by $N_X(t)$.

- There are no moves:
- The first move is Player X's:
- The first move is not Player X's: Player X has to choose a strategy in every possible subtree, t_1 to t_n, since he has to prepare an answer for every of the possible first moves. Then



$$N_X(t) = N_X(t_1) \times N_X(t_2) \times \cdots \times N_X(t_n).$$

Playing games *via* strategies

Once we know all the strategies for all the players of a game we can simplify playing it considerably.

Playing games *via* strategies

Once we know all the strategies for all the players of a game we can simplify playing it considerably.

We just let every player choose one of his or her available strategies.

Playing games *via* strategies

Once we know all the strategies for all the players of a game we can simplify playing it considerably.

We just let every player choose one of his or her available strategies.

We then go through the game tree as follows: For every node we reach, starting with the root, we ask the player whose turn it is which move his or her strategy chooses. If there are no chance moves then we thus follow a uniquely determined path through the tree ending at a leaf. That leaf gives us the outcome of playing all these strategies against each other.

Playing games *via* strategies

Once we know all the strategies for all the players of a game we can simplify playing it considerably.

We just let every player choose one of his or her available strategies.

We then go through the game tree as follows: For every node we reach, starting with the root, we ask the player whose turn it is which move his or her strategy chooses. If there are no chance moves then we thus follow a uniquely determined path through the tree ending at a leaf. That leaf gives us the outcome of playing all these strategies against each other.

If there are chance moves involved, then we get several paths ending in several leaves, together with a probability for each leaf that it will occur.

Playing games *via* strategies

Once we know all the strategies for all the players of a game we can simplify playing it considerably.

We just let every player choose one of his or her available strategies.

We then go through the game tree as follows: For every node we reach, starting with the root, we ask the player whose turn it is which move his or her strategy chooses. If there are no chance moves then we thus follow a uniquely determined path through the tree ending at a leaf. That leaf gives us the outcome of playing all these strategies against each other.

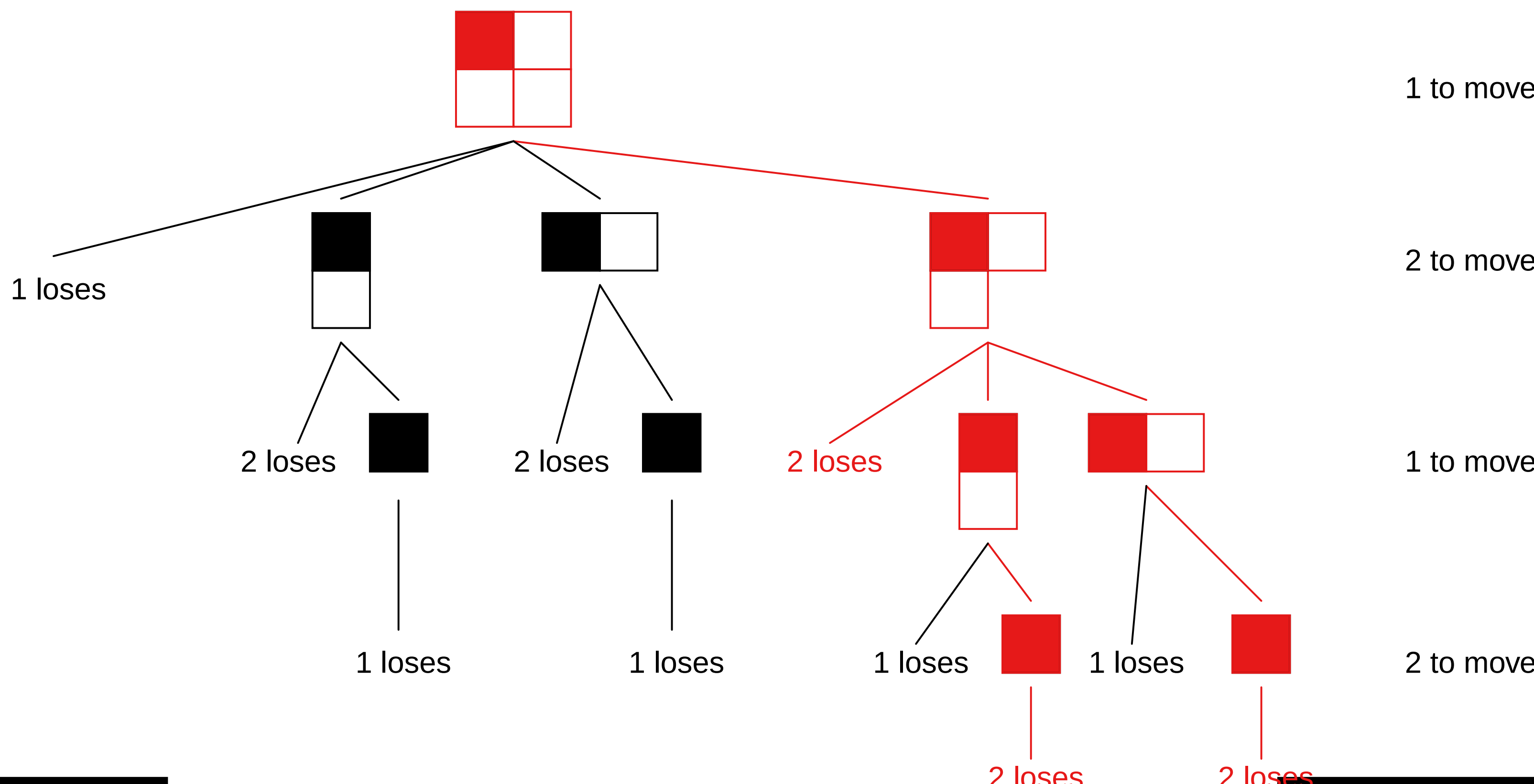
If there are chance moves involved, then we get several paths ending in several leaves, together with a probability for each leaf that it will occur. The probability for a leaf to occur is calculated by multiplying the probabilities occuring along the unique path leading to it. The probabilites for all the possible leaves will add up to one.

Playing strategies against each other

We have a look at an example to show how playing one strategy against another works.

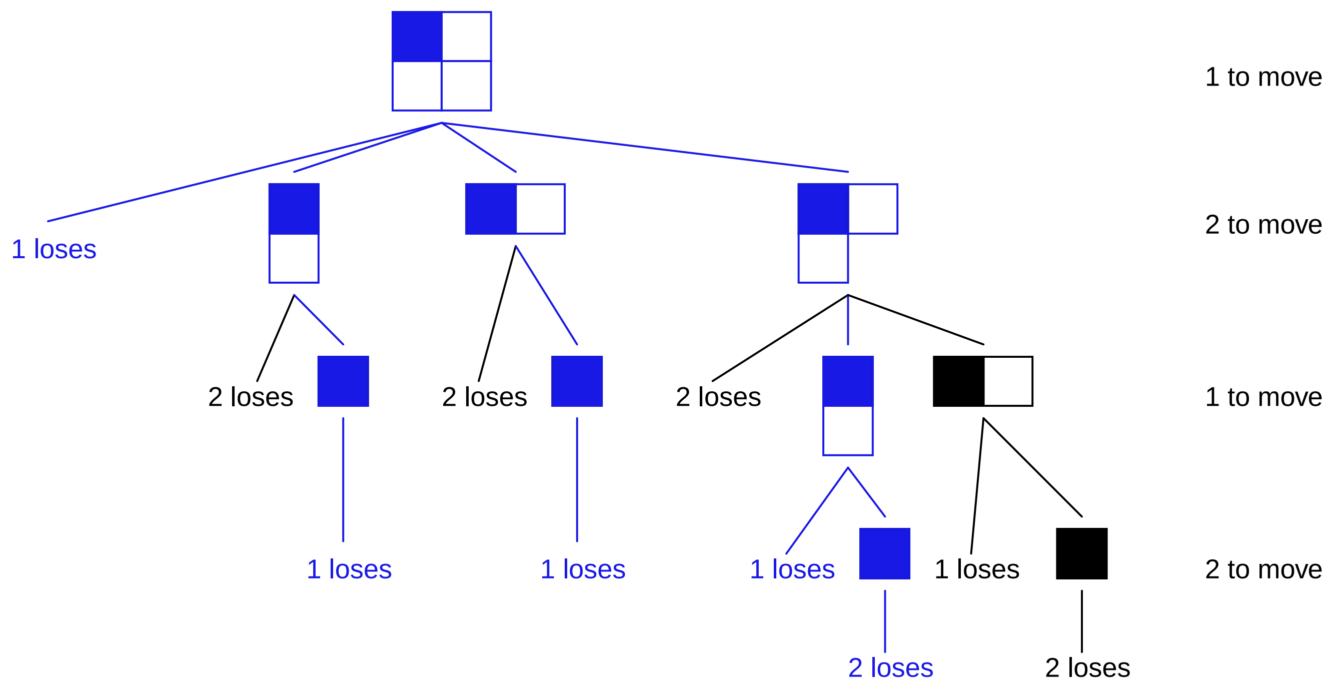
Playing strategies against each other

Here is a strategy for (2×2)-Chomp for Player 1.



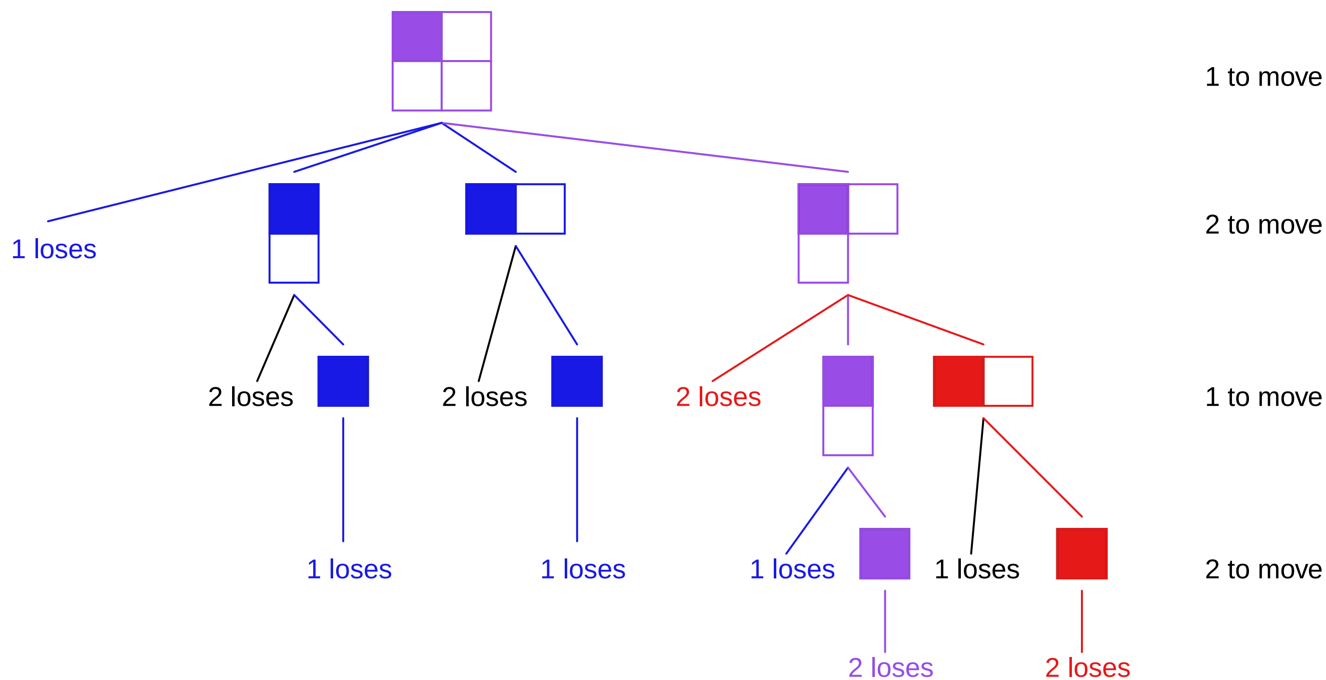
Playing strategies against each other

Here is a strategy for (2×2)-Chomp for Player 1. And here is a strategy for the same game for Player 2.



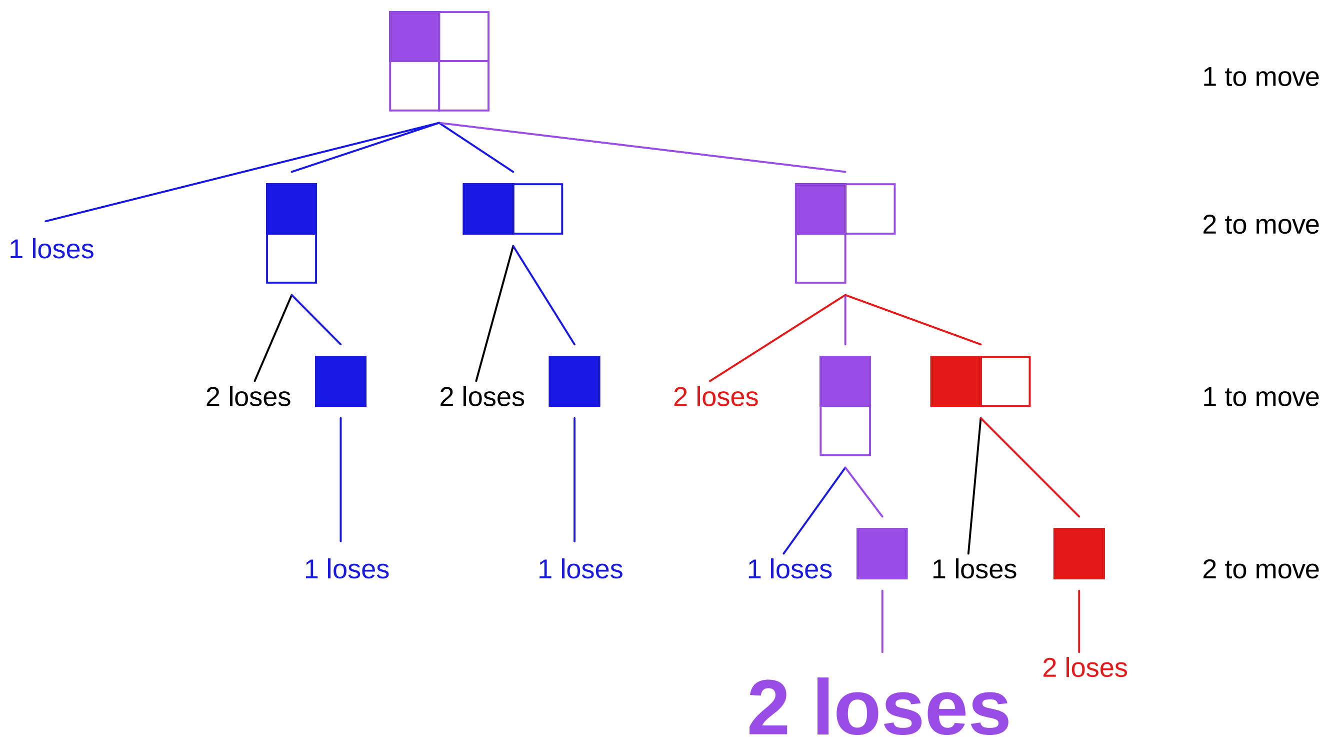
Playing strategies against each other

Here is a strategy for (2×2)-Chomp for Player 1. And here is a strategy for the same game for Player 2. And here they are together.



Playing strategies against each other

Here is a strategy for (2×2)-Chomp for Player 1. And here is a strategy for the same game for Player 2. And here they are together. The outcome of playing the one against the other is immediately obvious.



Why bother?

You might rightly argue that playing games in this way is a very boring thing to do! Everybody picks a strategy and then we arrive at the outcome. Where's all the fun gone?

Why bother?

You might rightly argue that playing games in this way is a very boring thing to do! Everybody picks a strategy and then we arrive at the outcome. Where's all the fun gone?

Playing games like this indeed isn't any fun. However, it is a very useful point of view to take if one is trying to analyse a game.

Why bother?

You might rightly argue that playing games in this way is a very boring thing to do! Everybody picks a strategy and then we arrive at the outcome. Where's all the fun gone?

Playing games like this indeed isn't any fun. However, it is a very useful point of view to take if one is trying to analyse a game.

It does have, however, one great disadvantage: It can only be applied if the game is small enough so that one can indeed determine all strategies for all the players.

Why bother?

You might rightly argue that playing games in this way is a very boring thing to do! Everybody picks a strategy and then we arrive at the outcome. Where's all the fun gone?

Playing games like this indeed isn't any fun. However, it is a very useful point of view to take if one is trying to analyse a game.

It does have, however, one great disadvantage: It can only be applied if the game is small enough so that one can indeed determine all strategies for all the players.

This is quite hopeless for games like Chess and Go, or even Checkers, simply because the game tree is far too big to hold in a computer—and we've just got some idea of how the number of strategies grows with the size of the game tree!

Why bother?

You might rightly argue that playing games in this way is a very boring thing to do! Everybody picks a strategy and then we arrive at the outcome. Where's all the fun gone?

Playing games like this indeed isn't any fun. However, it is a very useful point of view to take if one is trying to analyse a game.

It does have, however, one great disadvantage: It can only be applied if the game is small enough so that one can indeed determine all strategies for all the players.

This is quite hopeless for games like Chess and Go, or even Checkers, simply because the game tree is far too big to hold in a computer.

Hence when we talk about small games we mean games which are amenable to an analysis *via* strategies. They will keep us busy for quite a while, maybe longer than you'd thought. This is the most mathematical part of the course.

Games *via* strategies

We take the idea of presenting games *via* their strategies to its logical conclusion.

Games *via* strategies

If there are just two players, and no elements of chance involved, then we can present a game in the form of a matrix. We list the strategies for Player 1 along the rows and those for Player 2 along the columns, and record in the centre the outcome of playing the one against the other.

Games *via* strategies

In Paper-Stone-Scissors, Player 1 and 2 have three choices each which we can encode as follows.

Games *via* strategies

In Paper-Stone-Scissors, Player 1 and 2 have three choices each which we can encode as follows.

		2		
		P	R	S
	P			
1	R			
	S			

Games *via* strategies

In Paper-Stone-Scissors, Player 1 and 2 have three choices each which we can encode as follows. We record the result as

- 1: win for Player 1
- 2: win for Player 2
- D: draw.

		2		
		P	R	S
	P			
1	R			
	S			

Games *via* strategies

In Paper-Stone-Scissors, Player 1 and 2 have three choices each which we can encode as follows. We record the result as

- 1: win for Player 1
- 2: win for Player 2
- D: draw.

		2		
		P	R	S
	P	D	1	2
1	R	2	D	1
	S	1	2	D

Games *via* strategies

In Paper-Stone-Scissors, Player 1 and 2 have three choices each which we can encode as follows. We record the result as

- 1: win for Player 1
- 2: win for Player 2
- D: draw.

		2		
		P	R	S
	P	D	1	2
1	R	2	D	1
	S	1	2	D

Games given likes this are known as matrix games.

Question

Question. How would you give a matrix version of our miniature Risk game?

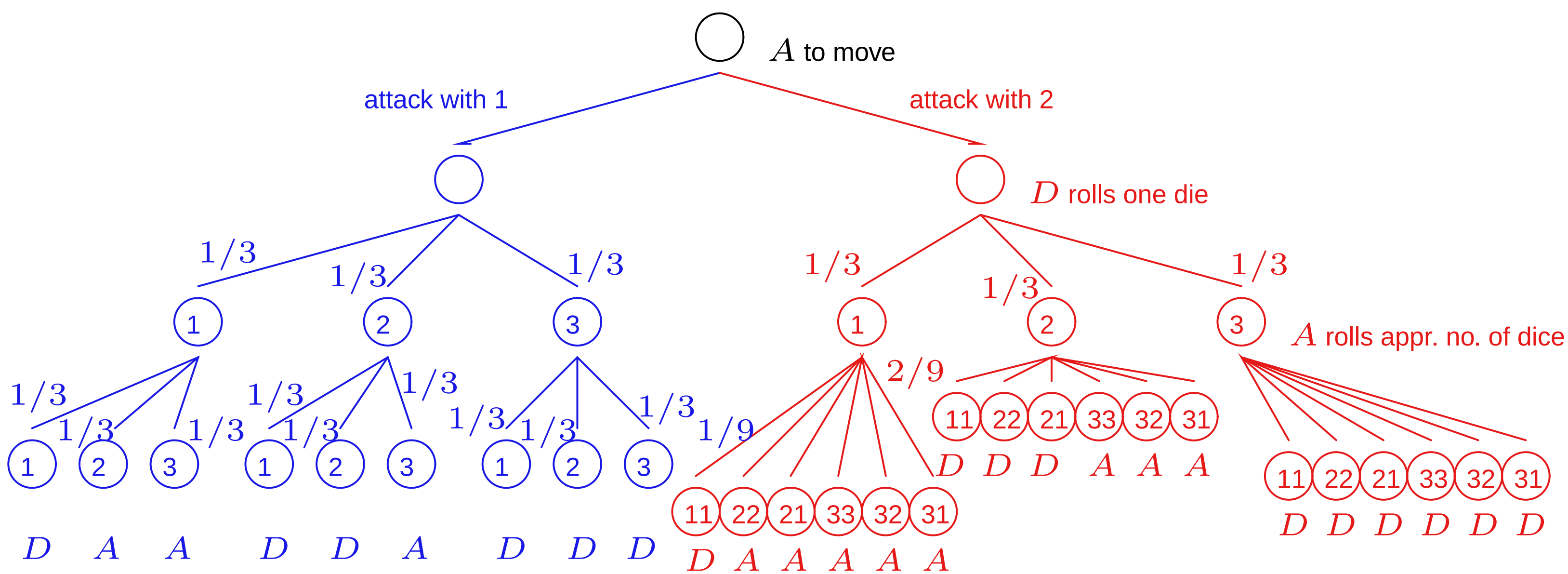
Games *via* strategies–chance

The problem with matrix games as we've described them so far is that as soon as elements as chance are involved, we don't get a unique result from playing strategies against each other.

Games *via* strategies–chance

The problem with matrix games as we've described them so far is that as soon as elements as chance are involved, we don't get a unique result from playing strategies against each other.

Recall the miniature Risk game we looked at earlier: Player 1 can only choose between the strategies 'attack with 1' and 'attack with 2', while Player 2 has only one strategy. But there are plenty of possible outcomes for either strategy.



Games *via* strategies–chance

The problem with matrix games as we've described them so far is that as soon as elements as chance are involved, we don't get a unique result from playing strategies against each other.

However, if all our outcomes are given as numbers then we can calculate an expected number as the outcome of playing one strategy against another.

Games *via* strategies–chance

The problem with matrix games as we've described them so far is that as soon as elements as chance are involved, we don't get a unique result from playing strategies against each other.

However, if all our outcomes are given as numbers then we can calculate an expected number as the outcome of playing one strategy against another.

If, for each player and each outcome of the game, we have a number to indicate what the outcome means for this player, then we can turn every game into a matrix.

Pay-off functions

It is for this reason that our definition of ‘game’ talks about something called a ‘pay-off function’.

Pay-off functions

It is for this reason that our definition of ‘game’ talks about something called a ‘pay-off function’.

Definition 1 *A game is given by*

- *a finite set of players,*
- *a finite game tree,*
- *for each node of the tree, a player whose turn it is in that position and*
- *for each final node and each player a pay-off function.*

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows.

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows.

		2		
		P	R	S
	P	0	1	-1
1	R	-1	0	1
	S	1	-1	0

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows. The pay-offs for Player 2 can be given similarly.

		2		
		P	R	S
	P	0	1	-1
1	R	-1	0	1
	S	1	-1	0

		2		
		P	R	S
	P	0	-1	1
1	R	1	0	-1
	S	-1	1	0

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows. The pay-offs for Player 2 can be given similarly. Or we can combine the two matrices into one.

		2		
		P	R	S
	P	$(0,0)$	$(1,-1)$	$(-1,1)$
1	R	$(-1,1)$	$(0,0)$	$(1,-1)$
	S	$(1,-1)$	$(-1,1)$	$(0,0)$

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows. The pay-offs for Player 2 can be given similarly. Or we can combine the two matrices into one.

		2		
		P	R	S
	P	$(0,0)$	$(1,-1)$	$(-1,1)$
1	R	$(-1,1)$	$(0,0)$	$(1,-1)$
	S	$(1,-1)$	$(-1,1)$	$(0,0)$

We say that a game for two players is in normal form when it is given *via* a matrix with real numbers as entries.

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows. The pay-offs for Player 2 can be given similarly. Or we can combine the two matrices into one.

		2		
		P	R	S
	P	$(0,0)$	$(1,-1)$	$(-1,1)$
1	R	$(-1,1)$	$(0,0)$	$(1,-1)$
	S	$(1,-1)$	$(-1,1)$	$(0,0)$

We say that a game for two players is in normal form when it is given *via* a matrix with real numbers as entries.

Question. What would happen if there were more than 2 players?

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows. The pay-offs for Player 2 can be given similarly.

		2		
		P	R	S
	P	0	1	-1
1	R	-1	0	1
	S	1	-1	0

		2		
		P	R	S
	P	0	-1	1
1	R	1	0	-1
	S	-1	1	0

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows. The pay-offs for Player 2 can be given similarly.

		2		
		P	R	S
	P	0	1	-1
1	R	-1	0	1
	S	1	-1	0

		2		
		P	R	S
	P	0	-1	1
1	R	1	0	-1
	S	-1	1	0

Note that if we add the entries in the left hand table to the corresponding ones in the right hand table we get a table consisting entirely of 0s.

Example

Assume that for Paper-Stone-Scissors the loser pays the winner one (pound, point, chocolate bar), and the pay-off is 0 for both if the game ends in a draw. Then the pay-offs for this game for Player 1 are as follows. The pay-offs for Player 2 can be given similarly.

		2		
		P	R	S
	P	0	1	-1
1	R	-1	0	1
	S	1	-1	0

		2		
		P	R	S
	P	0	-1	1
1	R	1	0	-1
	S	-1	1	0

Note that if we add the entries in the left hand table to the corresponding ones in the right hand table we get a table consisting entirely of 0s. This is because all payments to Player 1 are pay-outs by Player 2 and *vice versa*.

Zero-sum games

It turns out that games with this property are important enough to deserve their own name.

Zero-sum games

It turns out that games with this property are important enough to deserve their own name.

Definition 3 *A game is a zero-sum game if for each final position the pay-offs for all players add up to* 0*.*

Zero-sum games

It turns out that games with this property are important enough to deserve their own name.

Definition 3 *A game is a zero-sum game if for each final position the pay-offs for all players add up to* 0.

We can view such games as closed systems: Whatever the numbers stand for, none are created or lost to the system. The units just move from one player to another.

Zero-sum games

It turns out that games with this property are important enough to deserve their own name.

Definition 3 *A game is a zero-sum game if for each final position the pay-offs for all players add up to* 0*.*

We can view such games as closed systems.

Zero-sum games with two players are particularly easy to describe: If we give the pay-off function for one player, we just have to multiply it with -1 to get the pay-off function for the other player.

		2		
		P	R	S
	P	0	1	-1
1	R	-1	0	1
	S	1	-1	0

		2		
		P	R	S
	P	0	-1	1
1	R	1	0	-1
	S	-1	1	0

Zero-sum games

It turns out that games with this property are important enough to deserve their own name.

Definition 3 *A game is a zero-sum game if for each final position the pay-offs for all players add up to 0.*

We can view such games as closed systems.

Zero-sum games with two players are particularly easy to describe: If we give the pay-off function for one player, we just have to multiply it with -1 to get the pay-off function for the other player.

Hence in such a case it is sufficient to give just the one matrix, which normally is that for Player 1. In fact, whenever we give a game by just such a matrix, the implicit assumption is that we are describing a 2-person zero-sum game.

Non zero-sum games

There are plenty of games which are not zero-sum.

Non zero-sum games

There are plenty of games which are not zero-sum.

Question. Do you know any interesting zero-sum or non zero-sum games?

Non zero-sum games

There are plenty of games which are not zero-sum.

Question. Do you know any interesting zero-sum or non zero-sum games?

Card games played for points usually are not zero-sum games (and if you want to turn a series of such as something where you play for money you typically have to do something like turning differences in score into money).

Non zero-sum games

There are plenty of games which are not zero-sum.

Question. Do you know any interesting zero-sum or non zero-sum games?

Card games played for points usually are not zero-sum games (and if you want to turn a series of such as something where you play for money you typically have to do something like turning differences in score into money).

Battle games are another example: Defeating somebody else's forces does not mean that one gets these forces as assets somehow.

Non zero-sum games

There are plenty of games which are not zero-sum.

Question. Do you know any interesting zero-sum or non zero-sum games?

Card games played for points usually are not zero-sum games (and if you want to turn a series of such as something where you play for money you typically have to do something like turning differences in score into money).

Battle games are another example: Defeating somebody else's forces does not mean that one gets these forces as assets somehow.

Games played in a casino typically aren't zero-sum either—the casino takes a cut which the players never see again.

Non zero-sum games

There are plenty of games which are not zero-sum.

Question. Do you know any interesting zero-sum or non zero-sum games?

Card games played for points usually are not zero-sum games (and if you want to turn a series of such as something where you play for money you typically have to do something like turning differences in score into money).

Battle games are another example: Defeating somebody else's forces does not mean that one gets these forces as assets somehow.

Games played in a casino typically aren't zero-sum either—the casino takes a cut which the players never see again.

Some of the most interesting games are non zero-sum. We will see plenty of examples in this course, in particular since most of the known techniques do not work as well for these kinds of games.

Example

We now consider how to incorporate chance into turning a game into its matrix form.

Example

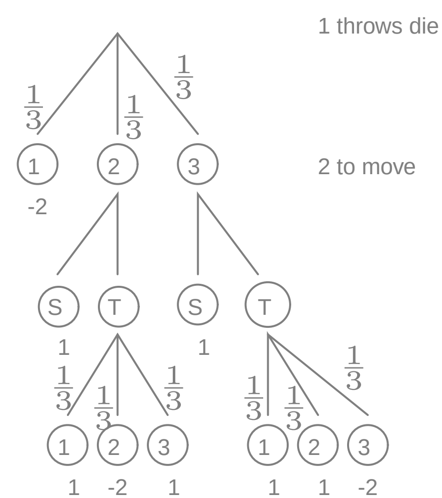
We now consider how to incorporate chance into turning a game into its matrix form.

Consider the following game between two players. Player 1 rolls a a three-faced die. If he throws 1 he pays two units to Player 2. If he throws 2 or 3, Player 2 has a choice. She can either choose to pay one unit to Player 1 (she stops the game) or she can throw the die. If she repeats Player 1's throw, he has to pay her two units. Otherwise she pays him one unit. The game tree is given below, with the pay-off being given for Player 1.

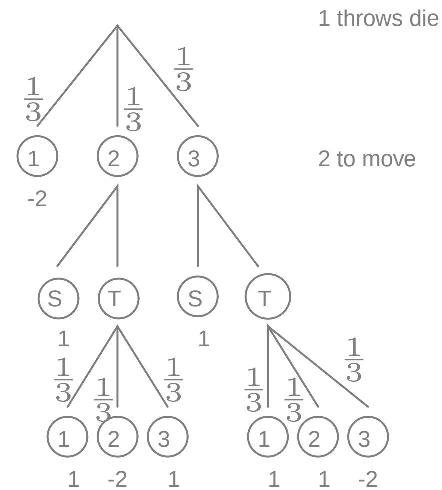
Example

We now consider how to incorporate chance into turning a game into its matrix form.

Consider the following game between two players. Player 1 rolls a a three-faced die. If he throws 1 he pays two units to Player 2. If he throws 2 or 3, Player 2 has a choice. She can either choose to pay one unit to Player 1 (she stops the game) or she can throw the die. If she repeats Player 1's throw, he has to pay her two units. Otherwise she pays him one unit. The game tree is given below, with the pay-off being given for Player 1.

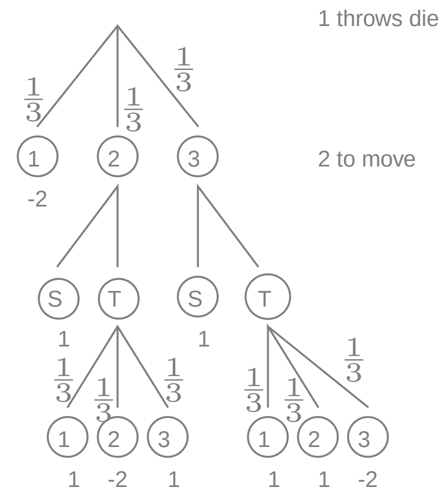


Example



Player 1 has only one strategy (he never gets a choice). Player 2 has four strategies (she can choose to throw the die or not, and is allowed to make that dependent on Player 1's throw). We can encode her strategies by saying what she will do when Player 1 throws a 2, and what she will do when Player 1 throws a 3, stop (S) or throw the die (T). So $S|T$ means that she will stop if he throws a 2, but throw if he throws a 3. The matrix will look something like this:

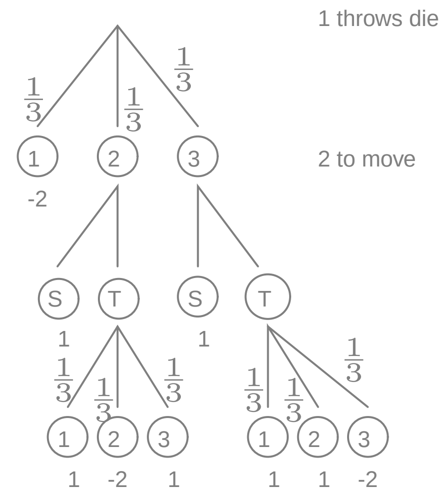
Example



Player 1 has only one strategy. Player 2 has four strategies. We can encode her strategies by saying what she will do when Player 1 throws a 2, and what she will do when Player 1 throws a 3, stop (S) or throw the die (T). So $S|T$ means that she will stop if he throws a 2, but throw if he throws a 3. The matrix will look something like this:

	2			
	$S\|S$	$S\|T$	$T\|S$	$T\|T$
1				

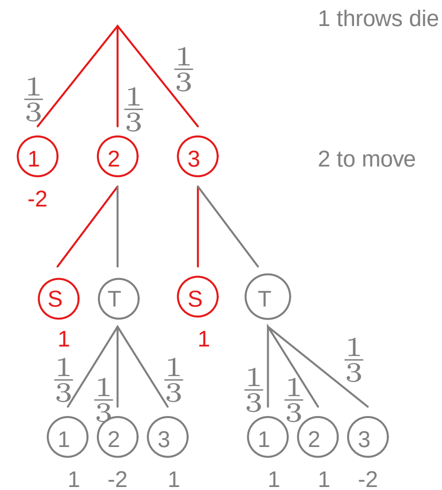
Example



	2			
	$S\|S$	$S\|T$	$T\|S$	$T\|T$
1				

What are the expected pay-offs for the outcome of these strategies?

Example

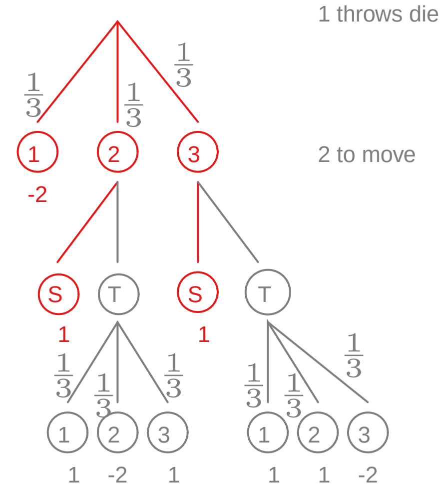


	2			
	$S\|S$	$S\|T$	$T\|S$	$T\|T$
1				

What are the expected pay-offs for the outcome of these strategies?

Case $S|S$. For each of the possible outcomes of playing this strategy, take the probability that it will occur and multiply it with the pay-off, then add all these up.

Example

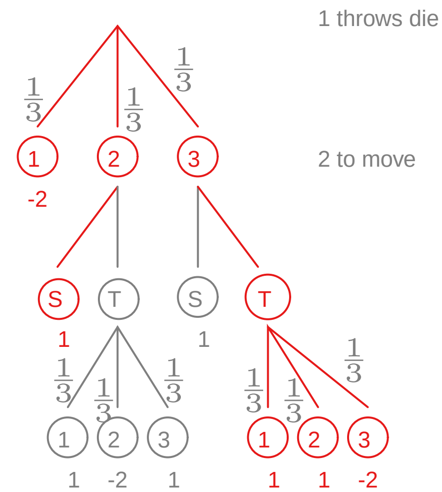


	2			
	$S\|S$	$S\|T$	$T\|S$	$T\|T$
1	0			

What are the expected pay-offs for the outcome of these strategies?
Case $S|S$. For each of the possible outcomes of playing this strategy, take the probability that it will occur and multiply it with the pay-off, then add all these up. Hence we get

$$(1/3 \times (-2)) + (1/3 \times 1) + (1/3 \times 1) = 0.$$

Example

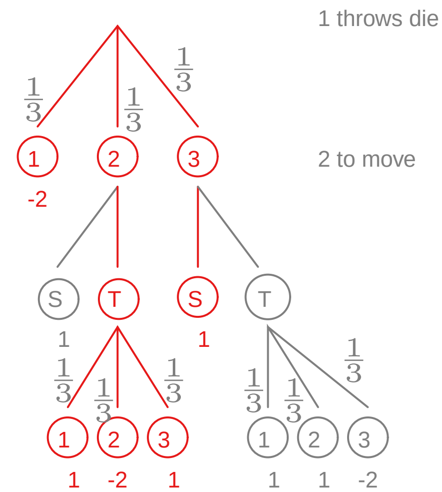


	2			
	$S\|S$	$S\|T$	$T\|S$	$T\|T$
1	0	$-1/3$		

What are the expected pay-offs for the outcome of these strategies?
Case $S|T$.

$$(1/3\times(-2))+(1/3\times 1)+(1/9\times 1)+(1/9\times 1)+(1/9\times(-2)) = -3/9 = -1/3.$$

Example

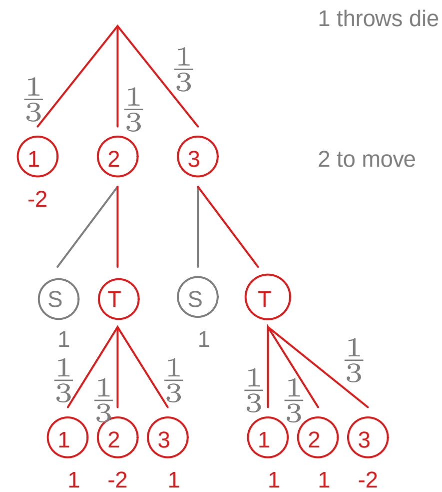


	2			
	$S\|S$	$S\|T$	$T\|S$	$T\|T$
1	0	$-1/3$	$-1/3$	

What are the expected pay-offs for the outcome of these strategies?

Case $T|S$. This is a symmetric variation of case $S|T$, and the pay-off is the same.

Example



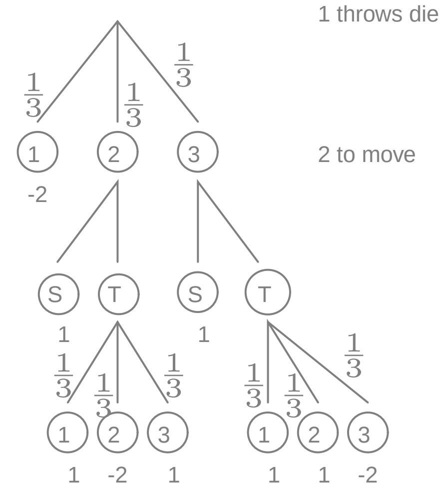
	2			
	$S\|S$	$S\|T$	$T\|S$	$T\|T$
1	0	$-1/3$	$-1/3$	$-2/3$

What are the expected pay-offs for the outcome of these strategies?

Case $T|T$.

$$(1/3 \times (-2)) + (1/9 \times 1) + (1/9 \times 1) + (1/9 \times (-2)) + \\ (1/9 \times 1) + (1/9 \times 1) + (1/9 \times (-2)) \\ = -2/3.$$

Example



	2			
	$S\|S$	$S\|T$	$T\|S$	$T\|T$
1	0	$-1/3$	$-1/3$	$-2/3$

Question. Which player would you rather be in this game?

A result

We are now ready to state and prove the only result of this introductory section.

A result

We are now ready to state and prove the only result of this introductory section.

Theorem 1.1 *Consider a game with two players, 1 and 2, of perfect information without chance, which can only have three different outcomes: Player 1 wins, Player 2 wins, or they draw. Then one of the following must be true.*

(i) *Player 1 has a strategy which allows him always to win;*

(ii) *Player 2 has a strategy which allows him always to win;*

(iii) *Player 1 and Player 2 both have strategies which ensure that they will not lose (which means that either side can enforce a draw).*

A result

We are now ready to state and prove the only result of this introductory section.

Theorem 1.1 *Consider a game with two players, 1 and 2, of perfect information without chance, which can only have three different outcomes: Player 1 wins, Player 2 wins, or they draw. Then one of the following must be true.*

(i) Player 1 has a strategy which allows him always to win;

(ii) Player 2 has a strategy which allows him always to win;

(iii) Player 1 and Player 2 both have strategies which ensure that they will not lose (which means that either side can enforce a draw).

If a player has a strategy which allows him to always win, no matter what the other player does, we call that strategy a winning strategy.

A result

We are now ready to state and prove the only result of this introductory section.

Theorem 1.1 *Consider a game with two players, 1 and 2, of perfect information without chance, which can only have three different outcomes: Player 1 wins, Player 2 wins, or they draw. Then one of the following must be true.*

(i) Player 1 has a strategy which allows him always to win;

(ii) Player 2 has a strategy which allows him always to win;

(iii) Player 1 and Player 2 both have strategies which ensure that they will not lose (which means that either side can enforce a draw).

If a player has a strategy which allows him to always win, no matter what the other player does, we call that strategy a winning strategy. Then we can restate the theorem by saying that in such a game, it is the case that either one of the players has a winning strategy or that they can both enforce a draw.

Proof

Induction over the height of the game tree.

Proof

Induction over the height of the game tree.

Base case: The height is 0. Then the result must be stated with the only node, win for Player 1, win for Player 2 or draw, and the Theorem is obviously true.

Proof

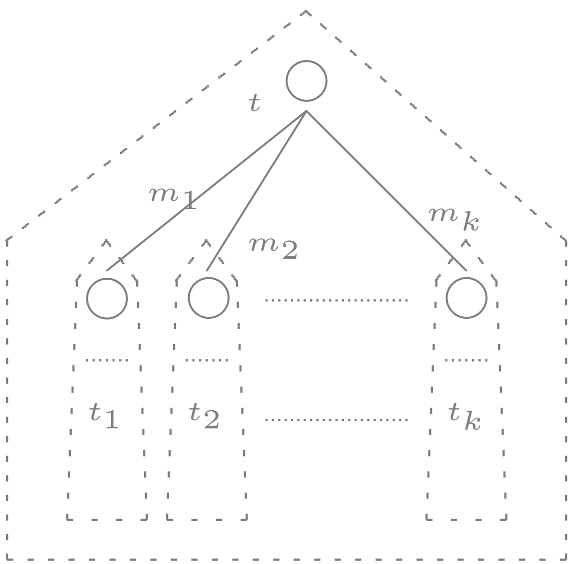
Induction over the height of the game tree.

Induction hypothesis: We can label the root of a game tree of height at most n with a number which indicates which case occurs. We use 1 if Player 1 can force a win, -1 if Player 2 can force a win and 0 if both sides can enforce a draw.

Proof

Induction over the height of the game tree.

Induction step.



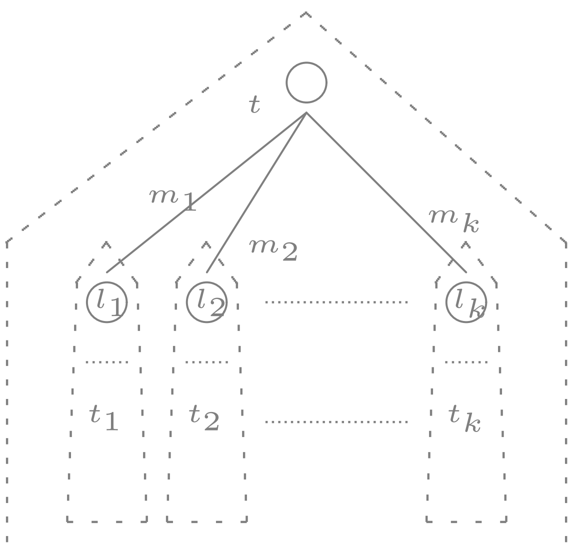
Let t be a tree of height $n + 1$.

Proof

Induction over the height of the game tree.
Induction step.

Let t be a tree of height $n+1$. By the induction hypothesis we can label the roots of the game trees reached by making a first move of t with a number (say l_i for tree t_i) as follows:

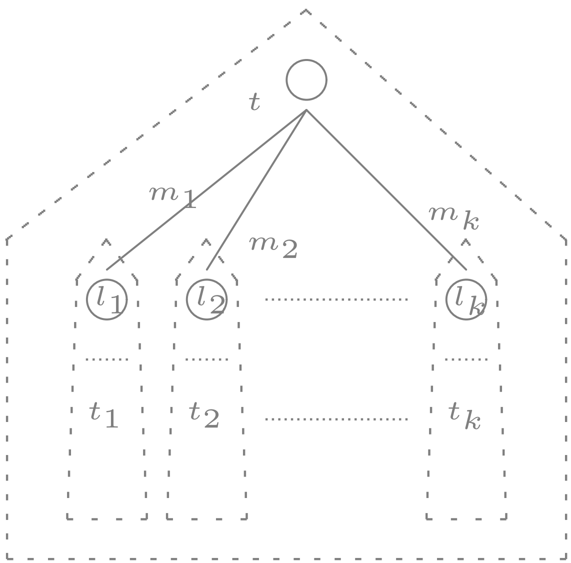
- it bears label $l_i = 1$ if Player 1 wins the game rooted there;
- it bears label $l_i = -1$ if Player 2 wins the game rooted there;
- it bears label $l_i = 0$ if both sides can enforce a draw in the game rooted there.



Proof

Induction over the height of the game tree.

Induction step.

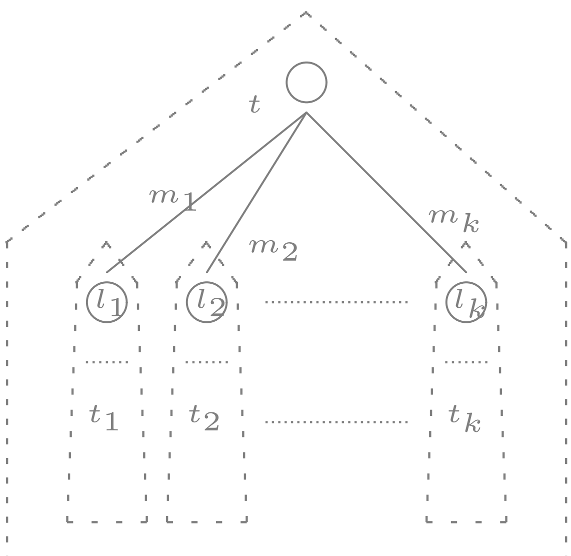


Let t be a tree of height $n+1$. Assume the first move is made by Player 1. The following cases can arise.

Proof

Induction over the height of the game tree.

Induction step.



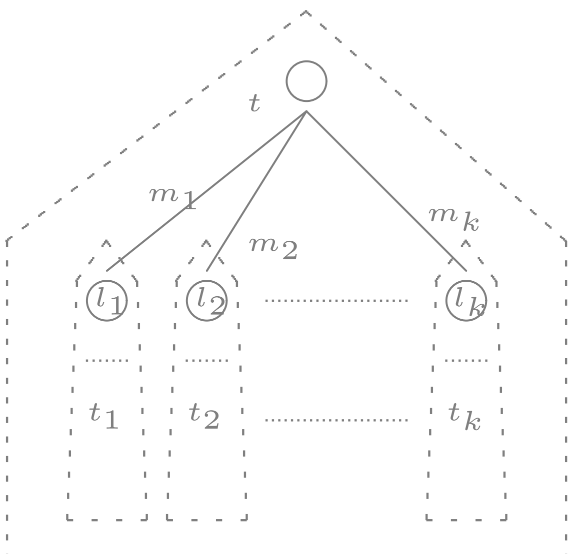
Let t be a tree of height $n+1$. Assume the first move is made by Player 1. The following cases can arise.

There is a tree labelled 1. So there is a $1 \leq i \leq k$ with $l_i = 1$. Then by making move m_i Player 1 can ensure that he will win the subsequent game t_i (because he has a winning strategy there), and thus the overall game t. Hence t gets label 1.

Proof

Induction over the height of the game tree.

Induction step.



Let t be a tree of height $n+1$. Assume the first move is made by Player 1. The following cases can arise.

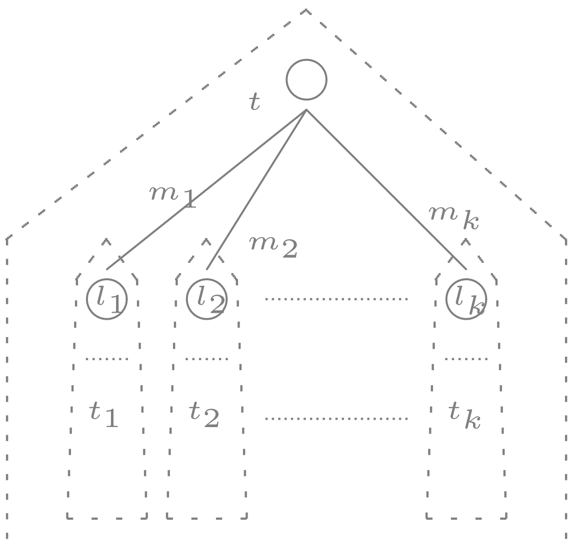
There is a tree labelled 1.

There is no tree labelled 1, but one labelled 0. So $l_i \leq 0$ is true for all $1 \leq i \leq n$ and $l_i = 0$ for one particular i. Then by making the first move m_i Player 1 transforms the game into t_i, where he can ensure a draw.

Proof

Induction over the height of the game tree.

Induction step.



Let t be a tree of height $n+1$. Assume the first move is made by Player 1. The following cases can arise.

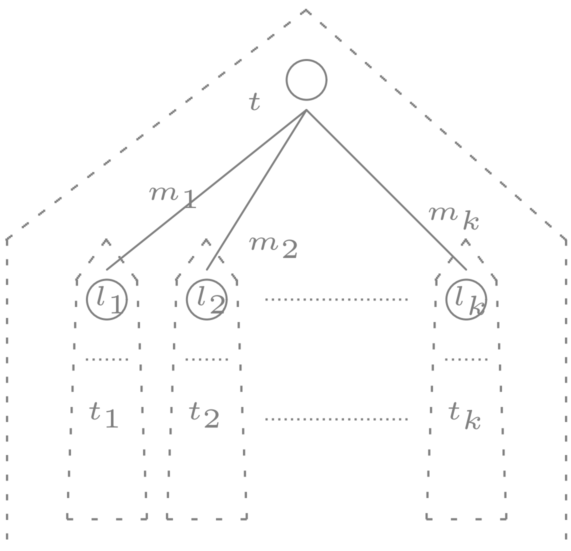
There is a tree labelled 1.

There is no tree labelled 1, but one labelled 0. So $l_i \leq 0$ is true for all $1 \leq i \leq n$ and $l_i = 0$ for one particular i. Then by making the first move m_i Player 1 transforms the game into t_i, where he can ensure a draw. But the fact that all the trees have a label of at most 0 means that Player 2 can enforce at least a draw in all games $t_1, \dots, t_k$, and thus in the game t no matter what Player 1's first move is. So t gets label 0.

Proof

Induction over the height of the game tree.

Induction step.



Let t be a tree of height $n+1$. Assume the first move is made by Player 1. The following cases can arise.

There is a tree labelled 1.

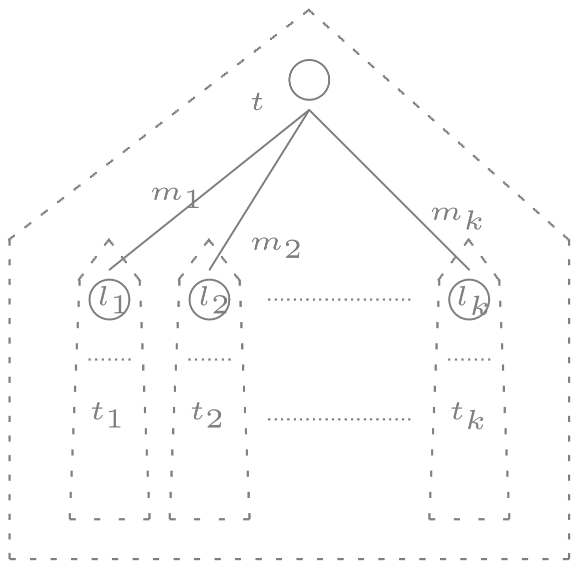
There is no tree labelled 1, but one labelled 0.

All trees are labelled -1. Then no matter which first move m_i Player 1 makes, Player 2 can enforce a win in the subsequent game t_i. Hence she can enforce a win in t and it gets label -1.

Proof

Induction over the height of the game tree.

Induction step.



Let t be a tree of height $n+1$. Assume the first move is made by Player 1. The following cases can arise.

There is a tree labelled 1.

There is no tree labelled 1, but one labelled 0.

All trees are labelled -1.

The case where Player 2 makes the first move is analogous.

Summary of Section 1

- Games can be represented using a game tree.

Summary of Section 1

- Games can be represented using a game tree.
- Elements of chance are then modelled by having a player (sometimes called Nature), with probabilities labelling such moves. Incomplete information is modelled by including in the game tree information about any nodes which cannot be distinguished by the player about to move.

Summary of Section 1

- Games can be represented using a game tree.
- Elements of chance are then modelled by having a player (sometimes called Nature), with probabilities labelling such moves. Incomplete information is modelled by including in the game tree information about any nodes which cannot be distinguished by the player about to move.
- The pay-off function for a player assigns a value to each of the possible outcomes (final positions) possible in the game.

Summary of Section 1

- Games can be represented using a game tree.
- Elements of chance are then modelled by having a player (sometimes called Nature), with probabilities labelling such moves. Incomplete information is modelled by including in the game tree information about any nodes which cannot be distinguished by the player about to move.
- The pay-off function for a player assigns a value to each of the possible outcomes (final positions) possible in the game.
- A strategy for a player is a complete game plan for that player.

Summary of Section 1

- Games can be represented using a game tree.
- Elements of chance are then modelled by having a player (sometimes called Nature), with probabilities labelling such moves. Incomplete information is modelled by including in the game tree information about any nodes which cannot be distinguished by the player about to move.
- The pay-off function for a player assigns a value to each of the possible outcomes (final positions) possible in the game.
- A strategy for a player is a complete game plan for that player.
- Small games have an alternative description *via* a matrices which show the pay-off for each player depending on the strategies chosen by all the players. Larger games have too many strategies for all of them to be listed. A game given in this way is known to be in normal form.

Summary of Section 1

- Games can be represented using a game tree.
- Elements of chance are then modelled by having a player (sometimes called Nature), with probabilities labelling such moves. Incomplete information is modelled by including in the game tree information about any nodes which cannot be distinguished by the player about to move.
- The pay-off function for a player assigns a value to each of the possible outcomes (final positions) possible in the game.
- A strategy for a player is a complete game plan for that player.
- Small games have an alternative description *via* a matrices which show the pay-off for each player depending on the strategies chosen by all the players. A game given in this way is known to be in normal form.
- In 2-player games of perfect information without chance either one of the players can force a win, or they can both force a draw.