



भारतीय प्रौद्योगिकी संस्थान मुंबई
Indian Institute of Technology Bombay

CS 6001: Game Theory and Algorithmic Mechanism Design

Week 1 – What is Game Theory and Mechanism Design?

Swaprava Nath

July, 2026

Week 1 Roadmap



Lecture	Chunks (20–25 min each)	Quizzes
Lecture 1	1. Why Game Theory & Mechanism Design 2. Common Knowledge (Islander Puzzle) 3. Mechanism Design Warm-up 4. The Game of Chess, Formally	Quiz 1 after Chunk 2 Quiz 2 after Chunk 4
Lecture 2	1. Proof of the Chess Theorem 2. The Halving Game & Strategic Form 3. Classic Games in Strategic Form 4. Coordination & the AI Safety Race	Quiz 1 after Chunk 1 Quiz 2 after Chunk 3



► Lecture 1: Why Game Theory, and the Game of Chess

► Lecture 2: Proving the Chess Theorem, and Strategic Form



Chunk 1

Why Game Theory and Mechanism Design?

($\approx$ 20–25 minutes)

Typical Engineering Courses



- Circuit **analysis**



Typical Engineering Courses



- Circuit **analysis**



analysis: given the circuit, find the behavior

Typical Engineering Courses



- Circuit **analysis** and **synthesis**



analysis: given the circuit, find the behavior

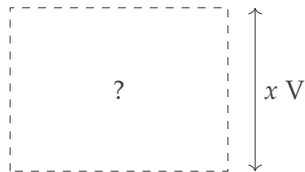
Typical Engineering Courses



- Circuit **analysis** and **synthesis**



analysis: given the circuit, find the behavior



synthesis: given the behavior we want,
design the circuit

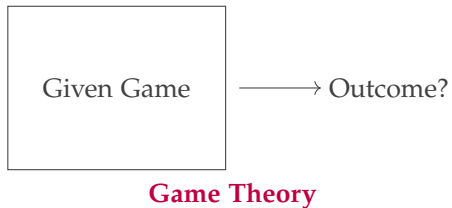
Now Replace the Circuit with People



Now Replace the Circuit with People



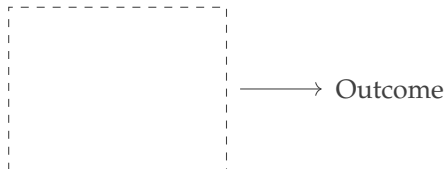
Now Replace the Circuit with People



Now Replace the Circuit with People



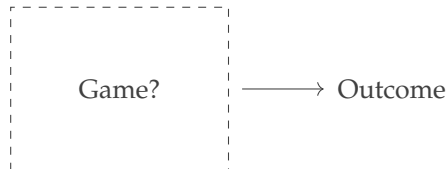
Game Theory



Now Replace the Circuit with People



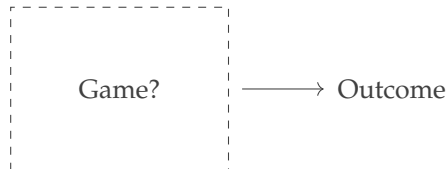
Game Theory



Now Replace the Circuit with People



Game Theory

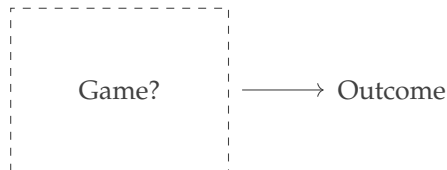


Mechanism Design

Now Replace the Circuit with People



Game Theory



Mechanism Design

- Social **analysis** and **synthesis** – same intellectual move, applied to people instead of resistors

A Motivating Game: The AI Lab Dilemma



Two AI labs, **Alpha** and **Beta**, are racing to release the next big model. Each can either **Test Thoroughly** (slow, safe) or **Ship Fast** (skip long safety evaluations).

		Beta	
		Test Thoroughly	Ship Fast
Alpha	Test Thoroughly	5, 5	0, 6
	Ship Fast	6, 0	1, 1

A Motivating Game: The AI Lab Dilemma



Two AI labs, **Alpha** and **Beta**, are racing to release the next big model. Each can either **Test Thoroughly** (slow, safe) or **Ship Fast** (skip long safety evaluations).

		Beta	
		Test Thoroughly	Ship Fast
Alpha	Test Thoroughly	5, 5	0, 6
	Ship Fast	6, 0	1, 1

Question

If you were advising Alpha's leadership, what would you recommend, and why?

What Just Happened There?



- A **Game** is a formal representation of the **strategic** interaction between **players**

What Just Happened There?



- A **Game** is a formal representation of the **strategic** interaction between **players**
- The choices available to players are **actions** (Test Thoroughly, Ship Fast)

What Just Happened There?



- A **Game** is a formal representation of the **strategic** interaction between **players**
- The choices available to players are **actions** (Test Thoroughly, Ship Fast)
- A **strategy** maps the state of the game to an action

What Just Happened There?



- A **Game** is a formal representation of the **strategic** interaction between **players**
- The choices available to players are **actions** (Test Thoroughly, Ship Fast)
- A **strategy** maps the state of the game to an action
 - In single-shot games like this one, strategy and action coincide

What Just Happened There?



- A **Game** is a formal representation of the **strategic** interaction between **players**
- The choices available to players are **actions** (Test Thoroughly, Ship Fast)
- A **strategy** maps the state of the game to an action
 - In single-shot games like this one, strategy and action coincide
 - Not true once the game has multiple stages – more on this soon, with chess

What Just Happened There?



- A **Game** is a formal representation of the **strategic** interaction between **players**
- The choices available to players are **actions** (Test Thoroughly, Ship Fast)
- A **strategy** maps the state of the game to an action
 - In single-shot games like this one, strategy and action coincide
 - Not true once the game has multiple stages – more on this soon, with chess
- Games come in many *representations*: **Normal form**, **Extensive form**, **Static**, **Dynamic**, **Repeated**, **Stochastic**, ...



Definition

Game theory is the formal study of strategic interaction between players who are **rational** and **intelligent**.



Definition

Game theory is the formal study of strategic interaction between players who are **rational** and **intelligent**.

- A player is **rational** if she picks an action to achieve her most preferred outcome, i.e., maximize her payoff



Definition

Game theory is the formal study of strategic interaction between players who are **rational** and **intelligent**.

- A player is **rational** if she picks an action to achieve her most preferred outcome, i.e., maximize her payoff
- A player is **intelligent** if she knows the rules of the game perfectly and can compute the *best* action, knowing that the other players are also rational and intelligent



Definition

Game theory is the formal study of strategic interaction between players who are **rational** and **intelligent**.

- A player is **rational** if she picks an action to achieve her most preferred outcome, i.e., maximize her payoff
- A player is **intelligent** if she knows the rules of the game perfectly and can compute the *best* action, knowing that the other players are also rational and intelligent
- **Goal of game theory:** **predict** the outcome of a game (revisit the AI Lab Dilemma with this lens)

Assumptions of Game Theory



This course is an axiomatic analysis of multi-agent behavior, built on these axioms:

- **Rationality**: a player picks actions to *maximize* her utility

Assumptions of Game Theory



This course is an axiomatic analysis of multi-agent behavior, built on these axioms:

- **Rationality**: a player picks actions to *maximize* her utility
- **Intelligence**: a player knows the rules perfectly and reasons about other rational, intelligent players

Assumptions of Game Theory



This course is an axiomatic analysis of multi-agent behavior, built on these axioms:

- **Rationality**: a player picks actions to *maximize* her utility
- **Intelligence**: a player knows the rules perfectly and reasons about other rational, intelligent players
- **Common Knowledge** (CK): a **fact** is common knowledge if

Assumptions of Game Theory



This course is an axiomatic analysis of multi-agent behavior, built on these axioms:

- **Rationality**: a player picks actions to *maximize* her utility
- **Intelligence**: a player knows the rules perfectly and reasons about other rational, intelligent players
- **Common Knowledge** (CK): a **fact** is common knowledge if
 - all players know the fact,

Assumptions of Game Theory



This course is an axiomatic analysis of multi-agent behavior, built on these axioms:

- **Rationality**: a player picks actions to *maximize* her utility
- **Intelligence**: a player knows the rules perfectly and reasons about other rational, intelligent players
- **Common Knowledge** (CK): a **fact** is common knowledge if
 - all players know the fact,
 - all players know that all players know the fact,

Assumptions of Game Theory



This course is an axiomatic analysis of multi-agent behavior, built on these axioms:

- **Rationality**: a player picks actions to *maximize* her utility
- **Intelligence**: a player knows the rules perfectly and reasons about other rational, intelligent players
- **Common Knowledge** (CK): a **fact** is common knowledge if
 - all players know the fact,
 - all players know that all players know the fact,
 - all players know that all players know that all players know the fact,

Assumptions of Game Theory



This course is an axiomatic analysis of multi-agent behavior, built on these axioms:

- **Rationality**: a player picks actions to *maximize* her utility
- **Intelligence**: a player knows the rules perfectly and reasons about other rational, intelligent players
- **Common Knowledge** (CK): a **fact** is common knowledge if
 - all players know the fact,
 - all players know that all players know the fact,
 - all players know that all players know that all players know the fact,
 - ...ad infinitum



Chunk 2

Common Knowledge – The Blue-Eyed Islanders

($\approx$ 20–25 minutes)

Why Common Knowledge Is Not Just Philosophy



- Think of a large group chat where someone posts “I know that you know that I saw the leaked exam paper.”

Why Common Knowledge Is Not Just Philosophy



- Think of a large group chat where someone posts “I know that you know that I saw the leaked exam paper.”
- Or a blockchain protocol upgrade: it only works if *every* node knows that every other node has upgraded, and knows that they know it, and so on

Why Common Knowledge Is Not Just Philosophy



- Think of a large group chat where someone posts “I know that you know that I saw the leaked exam paper.”
- Or a blockchain protocol upgrade: it only works if *every* node knows that every other node has upgraded, and knows that they know it, and so on
- Common knowledge is what turns private information into *coordinated public action*

Why Common Knowledge Is Not Just Philosophy



- Think of a large group chat where someone posts “I know that you know that I saw the leaked exam paper.”
- Or a blockchain protocol upgrade: it only works if *every* node knows that every other node has upgraded, and knows that they know it, and so on
- Common knowledge is what turns private information into *coordinated public action*
- We will see this with a classic puzzle, then come back to why it matters for strategic reasoning

Implication of CK: the Blue-Eyed Islander Problem



- **Location:** an isolated island (no mirrors, no reflecting surfaces)

Implication of CK: the Blue-Eyed Islander Problem



- **Location:** an isolated island (no mirrors, no reflecting surfaces)
- Three residents live there; each person's eyes are either **blue** or black, but eye color is never discussed

Implication of CK: the Blue-Eyed Islander Problem



- **Location:** an isolated island (no mirrors, no reflecting surfaces)
- Three residents live there; each person's eyes are either **blue** or black, but eye color is never discussed
- One day, a visiting *truth-telling* outsider announces: "At least one of you has blue eyes."

Implication of CK: the Blue-Eyed Islander Problem



- **Location:** an isolated island (no mirrors, no reflecting surfaces)
- Three residents live there; each person's eyes are either **blue** or black, but eye color is never discussed
- One day, a visiting *truth-telling* outsider announces: "At least one of you has blue eyes."
- **Consequence:** anyone who works out that his own eyes are blue must leave by the end of that day

Implication of CK: the Blue-Eyed Islander Problem



- **Location:** an isolated island (no mirrors, no reflecting surfaces)
- Three residents live there; each person's eyes are either **blue** or black, but eye color is never discussed
- One day, a visiting *truth-telling* outsider announces: "At least one of you has blue eyes."
- **Consequence:** anyone who works out that his own eyes are blue must leave by the end of that day

Implication of CK: the Blue-Eyed Islander Problem



- **Location:** an isolated island (no mirrors, no reflecting surfaces)
- Three residents live there; each person's eyes are either **blue** or black, but eye color is never discussed
- One day, a visiting *truth-telling* outsider announces: "At least one of you has blue eyes."
- **Consequence:** anyone who works out that his own eyes are blue must leave by the end of that day

Question

How does this one sentence change anything? Everyone can already see who has blue eyes.

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one
 - He leaves at the end of day 1

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one
 - He leaves at the end of day 1
- **If there were two** blue-eyed people:

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one
 - He leaves at the end of day 1
- **If there were two** blue-eyed people:
 - Each sees exactly one blue-eyed person

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one
 - He leaves at the end of day 1
- **If there were two** blue-eyed people:
 - Each sees exactly one blue-eyed person
 - If there were only one, that person would have left on day 1 – when that does not happen, each of the two realizes he himself must also be blue-eyed

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one
 - He leaves at the end of day 1
- **If there were two** blue-eyed people:
 - Each sees exactly one blue-eyed person
 - If there were only one, that person would have left on day 1 – when that does not happen, each of the two realizes he himself must also be blue-eyed
 - Both leave at the end of day 2

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one
 - He leaves at the end of day 1
- **If there were two** blue-eyed people:
 - Each sees exactly one blue-eyed person
 - If there were only one, that person would have left on day 1 – when that does not happen, each of the two realizes he himself must also be blue-eyed
 - Both leave at the end of day 2
- **With three**, the same argument pushes the departure to day 3

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one
 - He leaves at the end of day 1
- **If there were two** blue-eyed people:
 - Each sees exactly one blue-eyed person
 - If there were only one, that person would have left on day 1 – when that does not happen, each of the two realizes he himself must also be blue-eyed
 - Both leave at the end of day 2
- **With three**, the same argument pushes the departure to day 3

Percolation of Common Knowledge



Let us reason in stages.

- **If there were one** blue-eyed person:
 - He sees only black eyes around him
 - Since the outsider is always truthful, he must be the blue-eyed one
 - He leaves at the end of day 1
- **If there were two** blue-eyed people:
 - Each sees exactly one blue-eyed person
 - If there were only one, that person would have left on day 1 – when that does not happen, each of the two realizes he himself must also be blue-eyed
 - Both leave at the end of day 2
- **With three**, the same argument pushes the departure to day 3

Key Assumption of Game Theory

The fact that all players are rational and intelligent is common knowledge among them.



In-Class Quiz 1

Open the Google Form now.

Questions refer only to what was just discussed –
you cannot answer them without having been in class.



Chunk 3

Mechanism Design, Warmed Up

($\approx 20\text{--}25$ minutes)

From Analysis to Synthesis: A Fair-Division Problem



Two labmates, Riya and Karan, share a cloud GPU budget of \$1000 for the month. Their advisor wants it split *fairly*, but has no idea whose experiments need more compute, and does not want to referee any disputes.

From Analysis to Synthesis: A Fair-Division Problem



Two labmates, Riya and Karan, share a cloud GPU budget of \$1000 for the month. Their advisor wants it split *fairly*, but has no idea whose experiments need more compute, and does not want to referee any disputes.

Question

Can the advisor design a simple procedure so that whatever split results, neither labmate can complain that the other got more *by their own judgment* of what a fair share looks like?

The “Divide and Choose” Mechanism



- Riya splits the \$1000 pool into two shares, however she likes

The “Divide and Choose” Mechanism



- Riya splits the \$1000 pool into two shares, however she likes
- Karan picks whichever of the two shares he prefers; Riya gets the other

The “Divide and Choose” Mechanism



- Riya splits the \$1000 pool into two shares, however she likes
- Karan picks whichever of the two shares he prefers; Riya gets the other



The “Divide and Choose” Mechanism

- Riya splits the \$1000 pool into two shares, however she likes
- Karan picks whichever of the two shares he prefers; Riya gets the other

Why this works

- Karan cannot complain: he chose the share he valued more
- Riya cannot complain either: since she does not know which share Karan will pick, her best move is to split it into two shares *she* considers equal – so whichever one she ends up with, she is indifferent



The “Divide and Choose” Mechanism

- Riya splits the \$1000 pool into two shares, however she likes
- Karan picks whichever of the two shares he prefers; Riya gets the other

Why this works

- Karan cannot complain: he chose the share he valued more
 - Riya cannot complain either: since she does not know which share Karan will pick, her best move is to split it into two shares *she* considers equal – so whichever one she ends up with, she is indifferent
-
- The advisor never needs to know either student’s private valuation of “useful compute” – the **mechanism** does the work
 - This is **mechanism design**: design the rules of the game so that self-interested play leads to a good outcome

Why Design a Game at All?



- In **game theory**, the game is given; we predict what rational, intelligent players will do

Why Design a Game at All?



- In **game theory**, the game is given; we predict what rational, intelligent players will do
- In **mechanism design**, we work backward: we want a particular outcome (fairness, efficiency, revenue, truthfulness) and we design the rules so that this outcome emerges from self-interested play

Why Design a Game at All?



- In **game theory**, the game is given; we predict what rational, intelligent players will do
- In **mechanism design**, we work backward: we want a particular outcome (fairness, efficiency, revenue, truthfulness) and we design the rules so that this outcome emerges from self-interested play
- This is sometimes called *inverse game theory*

Why Design a Game at All?



- In **game theory**, the game is given; we predict what rational, intelligent players will do
- In **mechanism design**, we work backward: we want a particular outcome (fairness, efficiency, revenue, truthfulness) and we design the rules so that this outcome emerges from self-interested play
- This is sometimes called *inverse game theory*
- Recurring instances of this in modern computing: matching riders to drivers, allocating ad slots, splitting compute clusters, running elections, matching residents to hospitals

Why Design a Game at All?



- In **game theory**, the game is given; we predict what rational, intelligent players will do
- In **mechanism design**, we work backward: we want a particular outcome (fairness, efficiency, revenue, truthfulness) and we design the rules so that this outcome emerges from self-interested play
- This is sometimes called *inverse game theory*
- Recurring instances of this in modern computing: matching riders to drivers, allocating ad slots, splitting compute clusters, running elections, matching residents to hospitals
- The rest of this course develops the tools to do this in a principled manner



Chunk 4

The Game of Chess, Formally

($\approx$ 20–25 minutes)

From Handshake Games to Chess



- The AI Lab Dilemma was a *one-shot* game: one action each, then done

From Handshake Games to Chess



- The AI Lab Dilemma was a *one-shot* game: one action each, then done
- Most interesting games unfold over many moves, where each player reacts to what has happened so far

From Handshake Games to Chess



- The AI Lab Dilemma was a *one-shot* game: one action each, then done
- Most interesting games unfold over many moves, where each player reacts to what has happened so far
- Chess is the cleanest classical example, and it is how modern game-playing AI (AlphaZero, MuZero, and self-play agents in general) is still analyzed today

From Handshake Games to Chess



- The AI Lab Dilemma was a *one-shot* game: one action each, then done
- Most interesting games unfold over many moves, where each player reacts to what has happened so far
- Chess is the cleanest classical example, and it is how modern game-playing AI (AlphaZero, MuZero, and self-play agents in general) is still analyzed today
- Let us build up the formal machinery using chess as our running example



Description

- Two-player game: White (**W**) and Black (**B**) – 16 pieces each



Description

- Two-player game: White (**W**) and Black (**B**) – 16 pieces each
- Every piece has some legal moves – these are the **actions**



Description

- Two-player game: White (**W**) and Black (**B**) – 16 pieces each
- Every piece has some legal moves – these are the **actions**
- White moves first, players alternate



Description

- Two-player game: White (**W**) and Black (**B**) – 16 pieces each
- Every piece has some legal moves – these are the **actions**
- White moves first, players alternate
- Ends in



Description

- Two-player game: White (**W**) and Black (**B**) – 16 pieces each
- Every piece has some legal moves – these are the **actions**
- White moves first, players alternate
- Ends in
 - ❶ **Win for W:** W captures B's king



Description

- Two-player game: White (**W**) and Black (**B**) – 16 pieces each
- Every piece has some legal moves – these are the **actions**
- White moves first, players alternate
- Ends in
 - 1 Win for W: W captures B's king
 - 2 Win for B: B captures W's king



Description

- Two-player game: White (**W**) and Black (**B**) – 16 pieces each
- Every piece has some legal moves – these are the **actions**
- White moves first, players alternate
- Ends in
 - 1 **Win for W**: W captures B's king
 - 2 **Win for B**: B captures W's king
 - 3 **Draw**: everything else (stalemate, agreement, repetition, insufficient material, and so on)

Natural Questions from a Game Theorist's Perspective



Question

Does **W** have a *winning strategy*: a complete plan of moves that wins no matter what B does?

Natural Questions from a Game Theorist's Perspective



Question

Does **W** have a *winning strategy*: a complete plan of moves that wins no matter what B does?

Question

Does **B** have a winning strategy?

Natural Questions from a Game Theorist's Perspective



Question

Does **W** have a *winning strategy*: a complete plan of moves that wins no matter what B does?

Question

Does **B** have a winning strategy?

Question

If neither, does either player have at least a strategy that *guarantees a draw*?

Natural Questions from a Game Theorist's Perspective



Question

Does **W** have a *winning strategy*: a complete plan of moves that wins no matter what B does?

Question

Does **B** have a winning strategy?

Question

If neither, does either player have at least a strategy that *guarantees a draw*?

- None of these need hold in general – and “the game can end” does not by itself answer any of them

What Is a Strategy? Board Position vs. Game Situation



- Denote a **board position** at time k by x_k

What Is a Strategy? Board Position vs. Game Situation



- Denote a **board position** at time k by x_k
- A board position alone does not tell us the sequence of moves that produced it – several move sequences can land on the same board

What Is a Strategy? Board Position vs. Game Situation



- Denote a **board position** at time k by x_k
- A board position alone does not tell us the sequence of moves that produced it – several move sequences can land on the same board
- To reason about strategies (and to model an opponent), we need the full history: the **game situation**

What Is a Strategy? Board Position vs. Game Situation



- Denote a **board position** at time k by x_k
- A board position alone does not tell us the sequence of moves that produced it – several move sequences can land on the same board
- To reason about strategies (and to model an opponent), we need the full history: the **game situation**
- A **game situation** is a finite sequence $(x_0, x_1, \dots, x_k)$ of board positions such that:

What Is a Strategy? Board Position vs. Game Situation



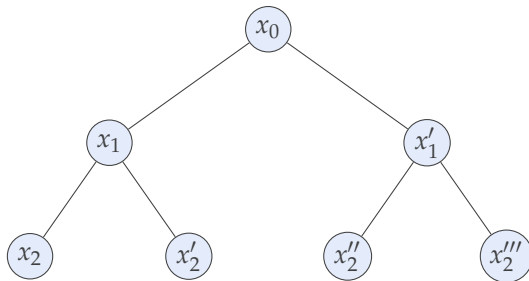
- Denote a **board position** at time k by x_k
- A board position alone does not tell us the sequence of moves that produced it – several move sequences can land on the same board
- To reason about strategies (and to model an opponent), we need the full history: the **game situation**
- A **game situation** is a finite sequence $(x_0, x_1, \dots, x_k)$ of board positions such that:
 - x_0 is the opening position

What Is a Strategy? Board Position vs. Game Situation

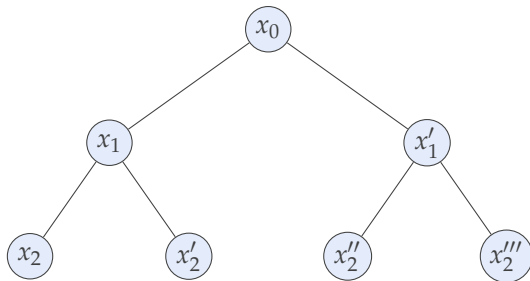


- Denote a **board position** at time k by x_k
- A board position alone does not tell us the sequence of moves that produced it – several move sequences can land on the same board
- To reason about strategies (and to model an opponent), we need the full history: the **game situation**
- A **game situation** is a finite sequence $(x_0, x_1, \dots, x_k)$ of board positions such that:
 - x_0 is the opening position
 - $x_k \rightarrow x_{k+1}$: for k even, this is a single legal move of W; for k odd, a single legal move of B

The Game Tree



The Game Tree



Strategy: a mapping from every game situation to an action – **a complete plan for every situation that could ever arise**, not just the moves along one particular play.

Formal Definition of a Strategy



Definition (Strategy, MSZ Ch. 1)

A **strategy** for W is a function s_W that maps every game situation $(x_0, x_1, \dots, x_k) \in H$ (the set of all game situations), k even, to a board position x_{k+1} such that $x_k \rightarrow x_{k+1}$ is a single legal move for W .

Formal Definition of a Strategy



Definition (Strategy, MSZ Ch. 1)

A **strategy** for W is a function s_W that maps every game situation $(x_0, x_1, \dots, x_k) \in H$ (the set of all game situations), k even, to a board position x_{k+1} such that $x_k \rightarrow x_{k+1}$ is a single legal move for W .

- Strategy s_B for B is defined analogously (for k odd)
- A pair (s_W, s_B) pins down a unique **play** of the game: start at x_0 , then $x_1 = s_W(x_0)$, $x_2 = s_B(x_0, x_1)$, and so on

Winning and Drawing Strategies



- A **winning strategy** for W is s_W^* such that for *every* s_B , the play (s_W^*, s_B) ends in a win for W

Winning and Drawing Strategies



- A **winning strategy** for W is s_W^* such that for *every* s_B , the play (s_W^*, s_B) ends in a win for W
- A **strategy guaranteeing at least a draw** for W is s_W' such that for every s_B , (s_W', s_B) ends in a draw or a win for W

Winning and Drawing Strategies



- A **winning strategy** for W is s_W^* such that for *every* s_B , the play (s_W^*, s_B) ends in a win for W
- A **strategy guaranteeing at least a draw** for W is s_W' such that for every s_B , (s_W', s_B) ends in a draw or a win for W
- Analogous definitions for B

Winning and Drawing Strategies



- A **winning strategy** for W is s_W^* such that for *every* s_B , the play (s_W^*, s_B) ends in a win for W
- A **strategy guaranteeing at least a draw** for W is s_W' such that for every s_B , (s_W', s_B) ends in a draw or a win for W
- Analogous definitions for B
- It is far from obvious that such strategies even exist – this is exactly the question von Neumann answered

An Early Result (von Neumann, 1928)



Theorem

In chess, exactly one of the following is true:

- 1 *W has a winning strategy*

An Early Result (von Neumann, 1928)



Theorem

In chess, exactly one of the following is true:

- ❶ *W has a winning strategy*
- ❷ *B has a winning strategy*

An Early Result (von Neumann, 1928)



Theorem

In chess, exactly one of the following is true:

- ❶ *W has a winning strategy*
- ❷ *B has a winning strategy*
- ❸ *Each player has a strategy guaranteeing at least a draw*

An Early Result (von Neumann, 1928)



Theorem

In chess, exactly one of the following is true:

- ❶ *W has a winning strategy*
- ❷ *B has a winning strategy*
- ❸ *Each player has a strategy guaranteeing at least a draw*

Remarkable – but read it carefully.



An Early Result (von Neumann, 1928)

Theorem

In chess, exactly one of the following is true:

- ❶ *W has a winning strategy*
- ❷ *B has a winning strategy*
- ❸ *Each player has a strategy guaranteeing at least a draw*

Remarkable – but read it carefully.

- It rules out the fourth possibility: that *nobody* can guarantee anything
- It does **not** say *which* alternative holds, or *what* the winning/drawing strategy actually is



An Early Result (von Neumann, 1928)

Theorem

In chess, exactly one of the following is true:

- ❶ *W has a winning strategy*
- ❷ *B has a winning strategy*
- ❸ *Each player has a strategy guaranteeing at least a draw*

Remarkable – but read it carefully.

- It rules out the fourth possibility: that *nobody* can guarantee anything
- It does **not** say *which* alternative holds, or *what* the winning/drawing strategy actually is

If we ever discovered which case holds, chess would stop being interesting to play.

Why an AI Researcher Should Care



- AlphaZero and similar self-play systems learn to play chess, Go, and shogi *without* being told which alternative of the theorem holds

Why an AI Researcher Should Care



- AlphaZero and similar self-play systems learn to play chess, Go, and shogi *without* being told which alternative of the theorem holds
- What they are really doing is searching for strategies close to a winning or drawing strategy, using self-play instead of exhaustive proof

Why an AI Researcher Should Care



- AlphaZero and similar self-play systems learn to play chess, Go, and shogi *without* being told which alternative of the theorem holds
- What they are really doing is searching for strategies close to a winning or drawing strategy, using self-play instead of exhaustive proof
- The theorem tells us such a strategy *exists* for one of the players; it says nothing about how hard it is to *find*

Why an AI Researcher Should Care



- AlphaZero and similar self-play systems learn to play chess, Go, and shogi *without* being told which alternative of the theorem holds
- What they are really doing is searching for strategies close to a winning or drawing strategy, using self-play instead of exhaustive proof
- The theorem tells us such a strategy *exists* for one of the players; it says nothing about how hard it is to *find*
- This gap between existence and tractability will reappear throughout the course, especially once we get to computing equilibria



In-Class Quiz 2

Open the Google Form now.

Questions refer only to what was just discussed –
you cannot answer them without having been in class.



End of Lecture 1

See you next class.



► Lecture 1: Why Game Theory, and the Game of Chess

► Lecture 2: Proving the Chess Theorem, and Strategic Form



Chunk 1

Proof of the Chess Theorem

($\approx$ 20–25 minutes)



An Early Result (von Neumann, 1928)

Theorem

In chess, exactly one of the following is true:

- ❶ *W has a winning strategy*
- ❷ *B has a winning strategy*
- ❸ *Each player has a strategy guaranteeing at least a draw*

Remarkable – but read it carefully.

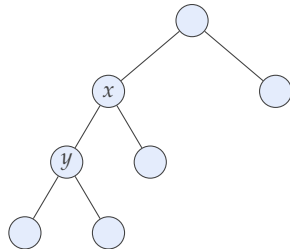
- It rules out the fourth possibility: that *nobody* can guarantee anything
- It does **not** say *which* alternative holds, or *what* the winning/drawing strategy actually is

If we ever discovered which case holds, chess would stop being interesting to play.

Setup of the Proof



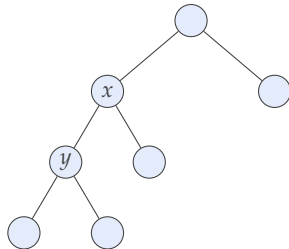
- Each vertex of the game tree is a game situation; edges are moves



Setup of the Proof



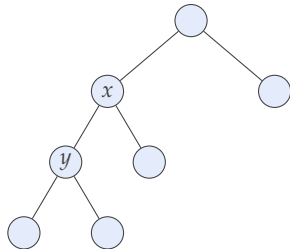
- Each vertex of the game tree is a game situation; edges are moves
- $\Gamma(x)$: the subtree rooted at x (including x)



Setup of the Proof



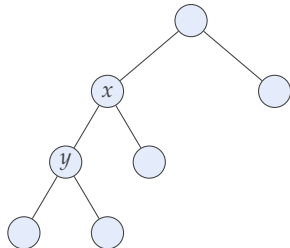
- Each vertex of the game tree is a game situation; edges are moves
- $\Gamma(x)$: the subtree rooted at x (including x)
- n_x : number of vertices in $\Gamma(x)$



Setup of the Proof



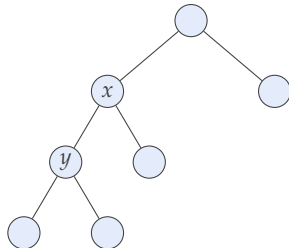
- Each vertex of the game tree is a game situation; edges are moves
- $\Gamma(x)$: the subtree rooted at x (including x)
- n_x : number of vertices in $\Gamma(x)$
- For y a descendant of x (with $y \neq x$), $\Gamma(y)$ is a subtree of $\Gamma(x)$, so $n_y < n_x$



Setup of the Proof



- Each vertex of the game tree is a game situation; edges are moves
- $\Gamma(x)$: the subtree rooted at x (including x)
- n_x : number of vertices in $\Gamma(x)$
- For y a descendant of x (with $y \neq x$), $\Gamma(y)$ is a subtree of $\Gamma(x)$, so $n_y < n_x$
- Idea: induct on n_x , from the leaves upward



Proof of the Chess Theorem: Base Case



The proof proceeds by induction on n_x .

Question

Does the theorem hold when $n_x = 1$?

Proof of the Chess Theorem: Base Case



The proof proceeds by induction on n_x .

Question

Does the theorem hold when $n_x = 1$?

- If W's king has been captured, B has already won – the empty strategy is a winning strategy for B





Proof of the Chess Theorem: Base Case

The proof proceeds by induction on n_x .

Question

Does the theorem hold when $n_x = 1$?

- If W's king has been captured, B has already won – the empty strategy is a winning strategy for B
- If B's king has been captured, W has already won





Proof of the Chess Theorem: Base Case

The proof proceeds by induction on n_x .

Question

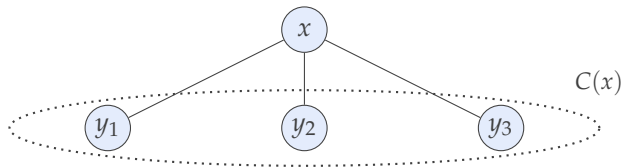
Does the theorem hold when $n_x = 1$?

- If W's king has been captured, B has already won – the empty strategy is a winning strategy for B
- If B's king has been captured, W has already won
- If both kings remain, $n_x = 1$ means the game has ended in a draw





Inductive Step: $n_x > 1$

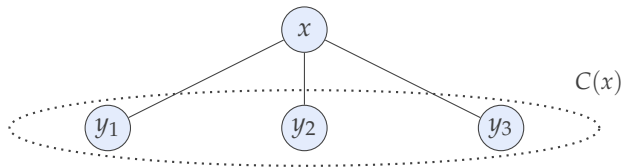


Notation

- x is a vertex with $n_x > 1$



Inductive Step: $n_x > 1$

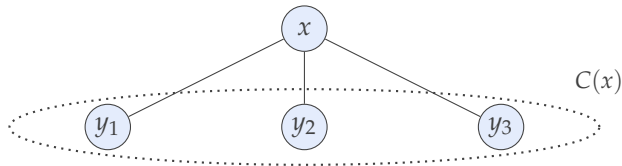


Notation

- x is a vertex with $n_x > 1$
- **Induction hypothesis:** for every $y \in C(x)$ (children of x , reachable in one move), the theorem already holds in $\Gamma(y)$, since $n_y < n_x$



Inductive Step: $n_x > 1$



Notation

- x is a vertex with $n_x > 1$
- **Induction hypothesis:** for every $y \in C(x)$ (children of x , reachable in one move), the theorem already holds in $\Gamma(y)$, since $n_y < n_x$
- We now show the theorem then also holds for x

Inductive Step: The Three Cases (WLOG W Moves at x)



- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0

Inductive Step: The Three Cases (WLOG W Moves at x)



- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0
 - Then W has a winning strategy at x too: move to y_0 , then follow that winning strategy

Inductive Step: The Three Cases (WLOG W Moves at x)



- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0
 - Then W has a winning strategy at x too: move to y_0 , then follow that winning strategy
- **Case 2:** $\forall y \in C(x)$, B has a winning strategy

Inductive Step: The Three Cases (WLOG W Moves at x)



- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0
 - Then W has a winning strategy at x too: move to y_0 , then follow that winning strategy
- **Case 2:** $\forall y \in C(x)$, B has a winning strategy
 - Then B wins at x too: whatever W plays, B just follows the winning strategy of the resulting subgame

Inductive Step: The Three Cases (WLOG W Moves at x)



- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0
 - Then W has a winning strategy at x too: move to y_0 , then follow that winning strategy
- **Case 2:** $\forall y \in C(x)$, B has a winning strategy
 - Then B wins at x too: whatever W plays, B just follows the winning strategy of the resulting subgame
- **Case 3:** neither Case 1 nor Case 2 holds

Inductive Step: The Three Cases (WLOG W Moves at x)



- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0
 - Then W has a winning strategy at x too: move to y_0 , then follow that winning strategy
- **Case 2:** $\forall y \in C(x)$, B has a winning strategy
 - Then B wins at x too: whatever W plays, B just follows the winning strategy of the resulting subgame
- **Case 3:** neither Case 1 nor Case 2 holds
 - Since Case 1 fails, W has no winning strategy in any $y \in C(x)$

Inductive Step: The Three Cases (WLOG W Moves at x)



- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0
 - Then W has a winning strategy at x too: move to y_0 , then follow that winning strategy
- **Case 2:** $\forall y \in C(x)$, B has a winning strategy
 - Then B wins at x too: whatever W plays, B just follows the winning strategy of the resulting subgame
- **Case 3:** neither Case 1 nor Case 2 holds
 - Since Case 1 fails, W has no winning strategy in any $y \in C(x)$
 - By the induction hypothesis, in every such y , either B has a winning strategy, or both players have a draw-guaranteeing strategy



Inductive Step: The Three Cases (WLOG W Moves at x)

- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0
 - Then W has a winning strategy at x too: move to y_0 , then follow that winning strategy
- **Case 2:** $\forall y \in C(x)$, B has a winning strategy
 - Then B wins at x too: whatever W plays, B just follows the winning strategy of the resulting subgame
- **Case 3:** neither Case 1 nor Case 2 holds
 - Since Case 1 fails, W has no winning strategy in any $y \in C(x)$
 - By the induction hypothesis, in every such y , either B has a winning strategy, or both players have a draw-guaranteeing strategy
 - Since Case 2 also fails, $\exists y' \in C(x)$ where B does *not* have a winning strategy – so by the previous line, both players have a draw-guaranteeing strategy at y'

Inductive Step: The Three Cases (WLOG W Moves at x)



- **Case 1:** $\exists y_0 \in C(x)$ where condition 1 is true: W has a winning strategy in y_0
 - Then W has a winning strategy at x too: move to y_0 , then follow that winning strategy
- **Case 2:** $\forall y \in C(x)$, B has a winning strategy
 - Then B wins at x too: whatever W plays, B just follows the winning strategy of the resulting subgame
- **Case 3:** neither Case 1 nor Case 2 holds
 - Since Case 1 fails, W has no winning strategy in any $y \in C(x)$
 - By the induction hypothesis, in every such y , either B has a winning strategy, or both players have a draw-guaranteeing strategy
 - Since Case 2 also fails, $\exists y' \in C(x)$ where B does *not* have a winning strategy – so by the previous line, both players have a draw-guaranteeing strategy at y'
 - W moves to y' and then plays the draw-guaranteeing strategy there; B does the same

Where Exactly Did We Use Chess Being... Chess?



The induction only used three structural properties of chess (MSZ, Ch. 1, p. 6) – worth isolating them, since the theorem is really about *any* game with these properties, not chess specifically.

- **(C1) Finite:** the game tree has finitely many vertices

Where Exactly Did We Use Chess Being... Chess?



The induction only used three structural properties of chess (MSZ, Ch. 1, p. 6) – worth isolating them, since the theorem is really about *any* game with these properties, not chess specifically.

- **(C1) Finite:** the game tree has finitely many vertices
- **(C2) No chance moves:** no dice, no card draws – the strategies alone determine the play

Where Exactly Did We Use Chess Being... Chess?



The induction only used three structural properties of chess (MSZ, Ch. 1, p. 6) – worth isolating them, since the theorem is really about *any* game with these properties, not chess specifically.

- **(C1) Finite:** the game tree has finitely many vertices
- **(C2) No chance moves:** no dice, no card draws – the strategies alone determine the play
- **(C3) Perfect information:** at every turn, a player knows every move made so far by both sides

Where Exactly Did We Use Chess Being... Chess?



The induction only used three structural properties of chess (MSZ, Ch. 1, p. 6) – worth isolating them, since the theorem is really about *any* game with these properties, not chess specifically.

- **(C1) Finite:** the game tree has finitely many vertices
- **(C2) No chance moves:** no dice, no card draws – the strategies alone determine the play
- **(C3) Perfect information:** at every turn, a player knows every move made so far by both sides



Where Exactly Did We Use Chess Being... Chess?

The induction only used three structural properties of chess (MSZ, Ch. 1, p. 6) – worth isolating them, since the theorem is really about *any* game with these properties, not chess specifically.

- **(C1) Finite:** the game tree has finitely many vertices
- **(C2) No chance moves:** no dice, no card draws – the strategies alone determine the play
- **(C3) Perfect information:** at every turn, a player knows every move made so far by both sides

Question

Rock-Paper-Scissors, which we meet later today, is finite and has no chance moves. Does the chess theorem's conclusion still have to hold for it?



Where Exactly Did We Use Chess Being... Chess?

The induction only used three structural properties of chess (MSZ, Ch. 1, p. 6) – worth isolating them, since the theorem is really about *any* game with these properties, not chess specifically.

- **(C1) Finite:** the game tree has finitely many vertices
- **(C2) No chance moves:** no dice, no card draws – the strategies alone determine the play
- **(C3) Perfect information:** at every turn, a player knows every move made so far by both sides

Question

Rock-Paper-Scissors, which we meet later today, is finite and has no chance moves. Does the chess theorem's conclusion still have to hold for it?

- No – RPS fails (C3): both players move *simultaneously*, so neither knows the other's move when choosing. MSZ's own counterexample is Matching Pennies (Ch. 3, Example 3.20)
- This is precisely why simultaneous-move zero-sum games need a genuinely different idea – *mixed strategies* – which we return to in a few weeks

What The Proof Does NOT Give You



- The proof is *constructive in principle*: it literally builds a winning or drawing strategy bottom-up from the leaves

What The Proof Does NOT Give You



- The proof is *constructive in principle*: it literally builds a winning or drawing strategy bottom-up from the leaves
- In practice, chess has more board positions than atoms in the observable universe – the tree is finite but astronomically large

What The Proof Does NOT Give You



- The proof is *constructive in principle*: it literally builds a winning or drawing strategy bottom-up from the leaves
- In practice, chess has more board positions than atoms in the observable universe – the tree is finite but astronomically large
- So even the constructive proof gives no *feasible* algorithm – existence and tractability are very different guarantees, a gap that will resurface once we start computing equilibria



In-Class Quiz 1

Open the Google Form now.

Questions refer only to what was just discussed –
you cannot answer them without having been in class.



Chunk 2

The Halving Game, and a New Representation

($\approx$ 20–25 minutes)

Try the Argument Yourself: The Halving Game



- Two players, $+1$ and -1 , play alternately, $+1$ first

Try the Argument Yourself: The Halving Game



- Two players, $+1$ and -1 , play alternately, $+1$ first
- State of the game: a single integer, `currNumber`

Try the Argument Yourself: The Halving Game



- Two players, $+1$ and -1 , play alternately, $+1$ first
- State of the game: a single integer, `currNumber`
- On your turn, choose one of two actions:

Try the Argument Yourself: The Halving Game



- Two players, $+1$ and -1 , play alternately, $+1$ first
- State of the game: a single integer, `currNumber`
- On your turn, choose one of two actions:
 - **Decrement:** `currNumber` $\rightarrow$ `currNumber` -1

Try the Argument Yourself: The Halving Game



- Two players, $+1$ and -1 , play alternately, $+1$ first
- State of the game: a single integer, `currNumber`
- On your turn, choose one of two actions:
 - **Decrement:** `currNumber` $\rightarrow$ `currNumber` $- 1$
 - **Halve:** `currNumber` $\rightarrow$ $\lfloor \text{currNumber} / 2 \rfloor$

Try the Argument Yourself: The Halving Game



- Two players, $+1$ and -1 , play alternately, $+1$ first
- State of the game: a single integer, `currNumber`
- On your turn, choose one of two actions:
 - **Decrement:** `currNumber` $\rightarrow$ `currNumber` $- 1$
 - **Halve:** `currNumber` $\rightarrow$ $\lfloor \text{currNumber} / 2 \rfloor$
- Whoever is forced to move when `currNumber` $= 0$ wins, i.e., who sets `currNumber` $= 0$, loses



Try the Argument Yourself: The Halving Game

- Two players, $+1$ and -1 , play alternately, $+1$ first
- State of the game: a single integer, `currNumber`
- On your turn, choose one of two actions:
 - **Decrement:** $\text{currNumber} \rightarrow \text{currNumber} - 1$
 - **Halve:** $\text{currNumber} \rightarrow \lfloor \text{currNumber} / 2 \rfloor$
- Whoever is forced to move when $\text{currNumber} = 0$ wins, i.e., who sets $\text{currNumber} = 0$, loses

Question

Start at $\text{currNumber} = 6$. Does either player have a winning strategy? Try it with a partner before the quiz.

From Trees to Tables: Why We Need Strategic Form



- Chess and the Halving Game are described by *trees*: who moves when, and what they know at that point

From Trees to Tables: Why We Need Strategic Form



- Chess and the Halving Game are described by *trees*: who moves when, and what they know at that point
- But often we only care about *outcomes as a function of strategies chosen*, not the blow-by-blow dynamics

From Trees to Tables: Why We Need Strategic Form



- Chess and the Halving Game are described by *trees*: who moves when, and what they know at that point
- But often we only care about *outcomes as a function of strategies chosen*, not the blow-by-blow dynamics
- That compressed description is called the **strategic form** (or **normal form**) of a game – it will be our main tool from next week onward



Strategic Form Games: Definition

Definition (MSZ Ch. 4)

A game in **strategic form** is a triple $G = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$ where

- $N = \{1, \dots, n\}$ is the set of players
- S_i is the strategy set of player i ; write $S = S_1 \times \dots \times S_n$
- $u_i : S \rightarrow \mathbb{R}$ gives player i 's payoff for each strategy vector $s \in S$



Strategic Form Games: Definition

Definition (MSZ Ch. 4)

A game in **strategic form** is a triple $G = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$ where

- $N = \{1, \dots, n\}$ is the set of players
 - S_i is the strategy set of player i ; write $S = S_1 \times \dots \times S_n$
 - $u_i : S \rightarrow \mathbb{R}$ gives player i 's payoff for each strategy vector $s \in S$
-
- For two players, this is just a payoff matrix: rows are Player 1's strategies, columns are Player 2's, and each cell holds a pair of payoffs
 - This is exactly the table we already drew for the AI Lab Dilemma in Lecture 1

Example: Penalty Shootout Game



		Goalkeeper		
		L	C	R
Shooter	L	$-1, 1$	$1, -1$	$1, -1$
	C	$1, -1$	$-1, 1$	$1, -1$
	R	$1, -1$	$1, -1$	$-1, 1$



Example: Penalty Shootout Game

		Goalkeeper		
		L	C	R
Shooter	L	-1, 1	1, -1	1, -1
	C	1, -1	-1, 1	1, -1
	R	1, -1	1, -1	-1, 1

- $N = \{1, 2\}$, 1 = Shooter, 2 = Goalkeeper



Example: Penalty Shootout Game

		Goalkeeper		
		L	C	R
Shooter	L	-1, 1	1, -1	1, -1
	C	1, -1	-1, 1	1, -1
	R	1, -1	1, -1	-1, 1

- $N = \{1, 2\}$, 1 = Shooter, 2 = Goalkeeper
- $S_1 = S_2 = \{L, C, R\}$



Example: Penalty Shootout Game

		Goalkeeper		
		L	C	R
Shooter	L	-1, 1	1, -1	1, -1
	C	1, -1	-1, 1	1, -1
	R	1, -1	1, -1	-1, 1

- $N = \{1, 2\}$, 1 = Shooter, 2 = Goalkeeper
- $S_1 = S_2 = \{L, C, R\}$
- $u_1(L, L) = -1, u_1(L, C) = u_1(L, R) = 1$



Example: Penalty Shootout Game

		Goalkeeper		
		L	C	R
Shooter	L	-1, 1	1, -1	1, -1
	C	1, -1	-1, 1	1, -1
	R	1, -1	1, -1	-1, 1

- $N = \{1, 2\}$, 1 = Shooter, 2 = Goalkeeper
- $S_1 = S_2 = \{L, C, R\}$
- $u_1(L, L) = -1, u_1(L, C) = u_1(L, R) = 1$
- $u_2(L, L) = 1, u_2(L, C) = u_2(L, R) = -1$



Example: Penalty Shootout Game

		Goalkeeper		
		L	C	R
Shooter	L	-1, 1	1, -1	1, -1
	C	1, -1	-1, 1	1, -1
	R	1, -1	1, -1	-1, 1

- $N = \{1, 2\}$, 1 = Shooter, 2 = Goalkeeper
- $S_1 = S_2 = \{L, C, R\}$
- $u_1(L, L) = -1, u_1(L, C) = u_1(L, R) = 1$
- $u_2(L, L) = 1, u_2(L, C) = u_2(L, R) = -1$
- (loosely) $u_1(X, X) = -1 = -u_2(X, X), u_1(X, Y) = -u_2(X, Y) = 1$

Notation You Will See Every Week From Here On



MSZ (Ch. 4, Sec. 4.4, p. 84) fixes a notation for “everyone except player i ” that we reuse for the rest of the course – worth internalizing now rather than re-deriving it every time.

Notation You Will See Every Week From Here On



MSZ (Ch. 4, Sec. 4.4, p. 84) fixes a notation for “everyone except player i ” that we reuse for the rest of the course – worth internalizing now rather than re-deriving it every time.

- For $s = (s_1, \dots, s_n) \in S$, write $s_{-i} := (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ – the strategies of everyone *but* i

Notation You Will See Every Week From Here On



MSZ (Ch. 4, Sec. 4.4, p. 84) fixes a notation for “everyone except player i ” that we reuse for the rest of the course – worth internalizing now rather than re-deriving it every time.

- For $s = (s_1, \dots, s_n) \in S$, write $s_{-i} := (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ – the strategies of everyone *but* i
- $S_{-i} := \prod_{j \neq i} S_j$ is the set all such s_{-i} can range over

Notation You Will See Every Week From Here On



MSZ (Ch. 4, Sec. 4.4, p. 84) fixes a notation for “everyone except player i ” that we reuse for the rest of the course – worth internalizing now rather than re-deriving it every time.

- For $s = (s_1, \dots, s_n) \in S$, write $s_{-i} := (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ – the strategies of everyone *but* i
- $S_{-i} := \prod_{j \neq i} S_j$ is the set all such s_{-i} can range over
- So a full strategy vector is just $s = (s_i, s_{-i})$, and we write player i 's payoff as $u_i(s_i, s_{-i})$

Notation You Will See Every Week From Here On



MSZ (Ch. 4, Sec. 4.4, p. 84) fixes a notation for “everyone except player i ” that we reuse for the rest of the course – worth internalizing now rather than re-deriving it every time.

- For $s = (s_1, \dots, s_n) \in S$, write $s_{-i} := (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ – the strategies of everyone *but* i
- $S_{-i} := \prod_{j \neq i} S_j$ is the set all such s_{-i} can range over
- So a full strategy vector is just $s = (s_i, s_{-i})$, and we write player i ’s payoff as $u_i(s_i, s_{-i})$

Why bother now

Every solution concept for the rest of this course – domination, best response, Nash equilibrium – is a statement about how $u_i(s_i, s_{-i})$ compares across choices of s_i , holding s_{-i} fixed. The notation is doing real conceptual work: it isolates exactly the one thing a single player controls.



Chunk 3

Classic Games in Strategic Form

($\approx$ 20–25 minutes)

Back to the AI Lab Dilemma: Now It Has a Name



		Beta	
		Cooperate	Defect
Alpha	Cooperate	4, 4	0, 5
	Defect	5, 0	1, 1

Back to the AI Lab Dilemma: Now It Has a Name



		Beta	
		Cooperate	Defect
Alpha	Cooperate	4, 4	0, 5
	Defect	5, 0	1, 1

- “Test Thoroughly” is relabeled **Cooperate**, “Ship Fast” is **Defect** – this is the **Prisoner’s Dilemma**, the most famous strategic-form game in existence

Back to the AI Lab Dilemma: Now It Has a Name



		Beta	
		Cooperate	Defect
Alpha	Cooperate	4, 4	0, 5
	Defect	5, 0	1, 1

- “Test Thoroughly” is relabeled **Cooperate**, “Ship Fast” is **Defect** – this is the **Prisoner’s Dilemma**, the most famous strategic-form game in existence
- Both labs are better off jointly testing thoroughly (4, 4) than jointly shipping fast (1, 1) – yet, as we will prove formally next week, each lab’s individually best move is to defect

Back to the AI Lab Dilemma: Now It Has a Name



		Beta	
		Cooperate	Defect
Alpha	Cooperate	4, 4	0, 5
	Defect	5, 0	1, 1

- “Test Thoroughly” is relabeled **Cooperate**, “Ship Fast” is **Defect** – this is the **Prisoner’s Dilemma**, the most famous strategic-form game in existence
- Both labs are better off jointly testing thoroughly (4, 4) than jointly shipping fast (1, 1) – yet, as we will prove formally next week, each lab’s individually best move is to defect
- This tension between individual and collective incentives is the central theme of half this course

Rock, Paper, Scissors – A Zero-Sum Classic



Two players simultaneously choose Rock, Paper, or Scissors. Rock breaks Scissors, Scissors cuts Paper, Paper covers Rock.

		Player II		
		Rock	Paper	Scissors
Player I	Rock	0,0	-1,1	1,-1
	Paper	1,-1	0,0	-1,1
	Scissors	-1,1	1,-1	0,0

Rock, Paper, Scissors – A Zero-Sum Classic



Two players simultaneously choose Rock, Paper, or Scissors. Rock breaks Scissors, Scissors cuts Paper, Paper covers Rock.

		Player II		
		Rock	Paper	Scissors
Player I	Rock	0, 0	-1, 1	1, -1
	Paper	1, -1	0, 0	-1, 1
	Scissors	-1, 1	1, -1	0, 0

- This is a **zero-sum** game: one player's gain is exactly the other's loss
- Matchmaking systems for competitive games (ranked ladders in esports) are built around reasoning of exactly this flavor

Chess, Compressed Into Strategic Form



- Every strategy s_W (a complete plan of moves for every game situation) is one row; every s_B is one column

Chess, Compressed Into Strategic Form



- Every strategy s_W (a complete plan of moves for every game situation) is one row; every s_B is one column
- Each cell holds the outcome of that pair of complete plans: W wins, B wins, or draw

Chess, Compressed Into Strategic Form



- Every strategy s_W (a complete plan of moves for every game situation) is one row; every s_B is one column
- Each cell holds the outcome of that pair of complete plans: W wins, B wins, or draw
- The matrix is finite – but its size is larger than we could ever write down, print, or store

Chess, Compressed Into Strategic Form



- Every strategy s_W (a complete plan of moves for every game situation) is one row; every s_B is one column
- Each cell holds the outcome of that pair of complete plans: W wins, B wins, or draw
- The matrix is finite – but its size is larger than we could ever write down, print, or store
- Von Neumann's theorem, restated in this language: some row is all W's, or some column is all B's, or there is a row and a column that are both entirely W-or-draw / B-or-draw respectively

Chess, Compressed Into Strategic Form



- Every strategy s_W (a complete plan of moves for every game situation) is one row; every s_B is one column
- Each cell holds the outcome of that pair of complete plans: W wins, B wins, or draw
- The matrix is finite – but its size is larger than we could ever write down, print, or store
- Von Neumann's theorem, restated in this language: some row is all W's, or some column is all B's, or there is a row and a column that are both entirely W-or-draw / B-or-draw respectively
- This is why we almost never work with strategic form *directly* for a sequential game like chess – but the concept still matters for reasoning about what a “complete strategy” even means

The Map to Strategic Form Loses Information



- Many of chess's astronomically many "strategies" are secretly *identical*: two plans that only differ in a branch of the tree neither ever reaches produce the same row of the matrix

The Map to Strategic Form Loses Information



- Many of chess's astronomically many "strategies" are secretly *identical*: two plans that only differ in a branch of the tree neither ever reaches produce the same row of the matrix
- MSZ (Ch. 4, Sec. 4.1, p. 80) calls the version with such duplicates merged the **reduced strategic form** – still astronomically large, but smaller than the naive count

The Map to Strategic Form Loses Information



- Many of chess's astronomically many “strategies” are secretly *identical*: two plans that only differ in a branch of the tree neither ever reaches produce the same row of the matrix
- MSZ (Ch. 4, Sec. 4.1, p. 80) calls the version with such duplicates merged the **reduced strategic form** – still astronomically large, but smaller than the naive count
- Sharper fact (Thompson, 1952; MSZ Sec. 4.2, p. 82–83): *different* extensive-form games – different trees, different dynamics, different information structure – can collapse to the exact same strategic-form table



The Map to Strategic Form Loses Information

- Many of chess's astronomically many "strategies" are secretly *identical*: two plans that only differ in a branch of the tree neither ever reaches produce the same row of the matrix
- MSZ (Ch. 4, Sec. 4.1, p. 80) calls the version with such duplicates merged the **reduced strategic form** – still astronomically large, but smaller than the naive count
- Sharper fact (Thompson, 1952; MSZ Sec. 4.2, p. 82–83): *different* extensive-form games – different trees, different dynamics, different information structure – can collapse to the exact same strategic-form table

Question

If two different game trees give the identical payoff matrix, have we lost anything by working only with the matrix?



The Map to Strategic Form Loses Information

- Many of chess's astronomically many "strategies" are secretly *identical*: two plans that only differ in a branch of the tree neither ever reaches produce the same row of the matrix
- MSZ (Ch. 4, Sec. 4.1, p. 80) calls the version with such duplicates merged the **reduced strategic form** – still astronomically large, but smaller than the naive count
- Sharper fact (Thompson, 1952; MSZ Sec. 4.2, p. 82–83): *different* extensive-form games – different trees, different dynamics, different information structure – can collapse to the exact same strategic-form table

Question

If two different game trees give the identical payoff matrix, have we lost anything by working only with the matrix?

- Yes: the matrix cannot tell you *when* information arrives, or whether a threat made mid-game is credible – exactly the questions extensive form is built to answer
- This is why, once we reach sequential settings again in a few weeks (subgame perfection), we go back to trees – strategic form is the right tool for *this* week's questions, not every question



In-Class Quiz 2

Open the Google Form now.

Questions refer only to what was just discussed –
you cannot answer them without having been in class.



Chunk 4

Coordination, and the AI Safety Race

($\approx 15\text{--}20$ minutes)

A Coordination Game: Picking a Common ML Framework



Two collaborating research groups must each independently pick a deep learning framework for a joint project.

		Group B	
		PyTorch	TensorFlow
Group A	PyTorch	3, 3	0, 0
	TensorFlow	0, 0	1, 1

A Coordination Game: Picking a Common ML Framework



Two collaborating research groups must each independently pick a deep learning framework for a joint project.

		Group B	
		PyTorch	TensorFlow
Group A	PyTorch	3, 3	0, 0
	TensorFlow	0, 0	1, 1

- Both (PyTorch, PyTorch) and (TensorFlow, TensorFlow) are stable outcomes – neither group wants to unilaterally switch once matched

A Coordination Game: Picking a Common ML Framework



Two collaborating research groups must each independently pick a deep learning framework for a joint project.

		Group B	
		PyTorch	TensorFlow
Group A	PyTorch	3, 3	0, 0
	TensorFlow	0, 0	1, 1

- Both (PyTorch, PyTorch) and (TensorFlow, TensorFlow) are stable outcomes – neither group wants to unilaterally switch once matched
- But the two stable outcomes are not equally good for the groups – coordinating on the *better* equilibrium is itself a nontrivial problem, one we will return to when we study Nash equilibrium

The AI Safety Race



Two AI labs must each choose whether to **Invest in Safety Research** (costly, slows down releases) or **Skip It** (cheaper, faster releases).

		Lab B	
		Invest	Skip
Lab A	Invest	4, 4	1, 3
	Skip	3, 1	2, 2

The AI Safety Race



Two AI labs must each choose whether to **Invest in Safety Research** (costly, slows down releases) or **Skip It** (cheaper, faster releases).

		Lab B	
		Invest	Skip
Lab A	Invest	4, 4	1, 3
	Skip	3, 1	2, 2

- Best outcome for both: neither races ahead recklessly, both invest (4,4)

The AI Safety Race



Two AI labs must each choose whether to **Invest in Safety Research** (costly, slows down releases) or **Skip It** (cheaper, faster releases).

		Lab B	
		Invest	Skip
Lab A	Invest	4, 4	1, 3
	Skip	3, 1	2, 2

- Best outcome for both: neither races ahead recklessly, both invest (4,4)
- Worst joint outcome: both skip safety investment (2,2) – yet each lab, fearing the other might skip, may be tempted to skip too, to avoid being the one left behind while under-invested

The AI Safety Race



Two AI labs must each choose whether to **Invest in Safety Research** (costly, slows down releases) or **Skip It** (cheaper, faster releases).

		Lab B	
		Invest	Skip
Lab A	Invest	4, 4	1, 3
	Skip	3, 1	2, 2

- Best outcome for both: neither races ahead recklessly, both invest (4,4)
- Worst joint outcome: both skip safety investment (2,2) – yet each lab, fearing the other might skip, may be tempted to skip too, to avoid being the one left behind while under-invested
- This is the classic **Security Dilemma** structure, applied to AI governance instead of nuclear arms – same mathematics, very different headline

Week 1: What We Built, and Where We Go Next



- Games as formal objects; rationality, intelligence, and common knowledge as the axioms underlying all of game theory

Week 1: What We Built, and Where We Go Next



- Games as formal objects; rationality, intelligence, and common knowledge as the axioms underlying all of game theory
- Mechanism design as the inverse problem: design the rules, not just analyze them

Week 1: What We Built, and Where We Go Next



- Games as formal objects; rationality, intelligence, and common knowledge as the axioms underlying all of game theory
- Mechanism design as the inverse problem: design the rules, not just analyze them
- Chess as our first rigorous example of a sequential game, and von Neumann's existence theorem

Week 1: What We Built, and Where We Go Next



- Games as formal objects; rationality, intelligence, and common knowledge as the axioms underlying all of game theory
- Mechanism design as the inverse problem: design the rules, not just analyze them
- Chess as our first rigorous example of a sequential game, and von Neumann's existence theorem
- Strategic (normal) form as the compressed representation we will use for the next several weeks

Week 1: What We Built, and Where We Go Next



- Games as formal objects; rationality, intelligence, and common knowledge as the axioms underlying all of game theory
- Mechanism design as the inverse problem: design the rules, not just analyze them
- Chess as our first rigorous example of a sequential game, and von Neumann's existence theorem
- Strategic (normal) form as the compressed representation we will use for the next several weeks
- **Next week:** dominance, iterated elimination, and Nash equilibrium – the tools to actually *solve* the games we just wrote down



End of Lecture 2

See you next class.



भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay