



भारतीय प्रौद्योगिकी संस्थान मुंबई
Indian Institute of Technology Bombay

CS 6001: Game Theory and Algorithmic Mechanism Design

Week 4 – Correlated Equilibrium, Extensive Form Games, and Subgame Perfection

Swaprava Nath

August, 2026

Week 4 Roadmap



Lecture	Chunks (20–25 min each)	Quizzes
Lecture 7	1. Why a Mediator? Correlated Strategy & Equilibrium 2. Worked Examples of CE 3. Computing CE 4. CE vs. Other Equilibria	Quiz 1 after Chunk 2 Quiz 2 after Chunk 4
Lecture 8	1. PIEFG: Formal Model 2. Strategies, NFG Translation, Non-Credible Threats 3. Subgame Perfection & Backward Induction 4. Limitations of SPNE & the Centipede Game	Quiz 1 after Chunk 2 Quiz 2 after Chunk 4



► Lecture 7: Correlated Equilibrium

► Lecture 8: Extensive Form Games and Subgame Perfection

Recap



- MSNE $\rightarrow$ weakest notion of equilibrium so far

Recap



- MSNE $\rightarrow$ weakest notion of equilibrium so far
- Existence is guaranteed for finite games

Recap



- MSNE $\rightarrow$ weakest notion of equilibrium so far
- Existence is guaranteed for finite games
- Finding MSNE is computationally expensive



Chunk 1

Why a Mediator? Correlated Strategy and Equilibrium

(≈ 20 – 25 minutes)

Delivery Robots at a Blind Junction



Two **autonomous** delivery robots, Chandra and Aryabhata, meet at a blind junction with no signal. Each must choose to **Wait** or **Go**. If both wait, nothing moves. If one goes while the other waits, the mover delivers faster. If both go at once, they collide.

		Aryabhata	
		Wait	Go
Chandra	Wait	0, 0	1, 2
	Go	2, 1	-10, -10

Delivery Robots at a Blind Junction



Two **autonomous** delivery robots, Chandra and Aryabhata, meet at a blind junction with no signal. Each must choose to **Wait** or **Go**. If both wait, nothing moves. If one goes while the other waits, the mover delivers faster. If both go at once, they collide.

		Aryabhata	
		Wait	Go
Chandra	Wait	0,0	1,2
	Go	2,1	-10,-10

Nash solutions here are: one robot waits while the other goes, or a large probability of waiting for both. In practice, something simpler happens. (similar to a traffic intersection)

Enter the Junction Controller



- A **junction controller** tells each robot when to wait and when to go, and both robots agree in advance to follow it (**Why?**)

Enter the Junction Controller



- A **junction controller** tells each robot when to wait and when to go, and both robots agree in advance to follow it (**Why?**)
- This **trusted third party** is called the **mediator**

Enter the Junction Controller



- A **junction controller** tells each robot when to wait and when to go, and both robots agree in advance to follow it (**Why?**)
- This **trusted third party** is called the **mediator**
- **Role:**

Enter the Junction Controller



- A **junction controller** tells each robot when to wait and when to go, and both robots agree in advance to follow it (**Why?**)
- This **trusted third party** is called the **mediator**
- **Role:**
 - randomize over **strategy profiles** (not individual strategies)

Enter the Junction Controller



- A **junction controller** tells each robot when to wait and when to go, and both robots agree in advance to follow it (**Why?**)
- This **trusted third party** is called the **mediator**
- **Role:**
 - randomize over **strategy profiles** (not individual strategies)
 - suggest the corresponding action to each robot

Enter the Junction Controller



- A **junction controller** tells each robot when to wait and when to go, and both robots agree in advance to follow it (**Why?**)
- This **trusted third party** is called the **mediator**
- **Role:**
 - randomize over **strategy profiles** (not individual strategies)
 - suggest the corresponding action to each robot
- If the suggested strategies are **enforceable**, this is an equilibrium (**correlated**)

Correlated Strategy and Equilibrium



Definition (Correlated Strategy)

A **correlated strategy** is a mapping $\pi : S_1 \times S_2 \times \cdots \times S_n \rightarrow [0, 1]$ s.t. $\sum_{s \in S} \pi(s) = 1$.

Correlated Strategy and Equilibrium



Definition (Correlated Strategy)

A **correlated strategy** is a mapping $\pi : S_1 \times S_2 \times \cdots \times S_n \rightarrow [0, 1]$ s.t. $\sum_{s \in S} \pi(s) = 1$.

Example: $\pi(\text{Wait}, \text{Wait}) = 0$, $\pi(\text{Wait}, \text{Go}) = \pi(\text{Go}, \text{Wait}) = \frac{1}{2}$, and $\pi(\text{Go}, \text{Go}) = 0$

Correlated Strategy and Equilibrium



Definition (Correlated Strategy)

A **correlated strategy** is a mapping $\pi : S_1 \times S_2 \times \cdots \times S_n \rightarrow [0, 1]$ s.t. $\sum_{s \in S} \pi(s) = 1$.

Example: $\pi(\text{Wait}, \text{Wait}) = 0$, $\pi(\text{Wait}, \text{Go}) = \pi(\text{Go}, \text{Wait}) = \frac{1}{2}$, and $\pi(\text{Go}, \text{Go}) = 0$

Question

What is a correlated equilibrium?



Correlated Strategy and Equilibrium

Definition (Correlated Strategy)

A **correlated strategy** is a mapping $\pi : S_1 \times S_2 \times \cdots \times S_n \rightarrow [0, 1]$ s.t. $\sum_{s \in S} \pi(s) = 1$.

Example: $\pi(\text{Wait}, \text{Wait}) = 0$, $\pi(\text{Wait}, \text{Go}) = \pi(\text{Go}, \text{Wait}) = \frac{1}{2}$, and $\pi(\text{Go}, \text{Go}) = 0$

Question

What is a correlated equilibrium?

Answer

A *correlated strategy* is a **correlated equilibrium** when no player *gains* by deviating from the suggested strategy while others follow the suggested strategies



Correlated Strategy and Equilibrium

Definition (Correlated Strategy)

A **correlated strategy** is a mapping $\pi : S_1 \times S_2 \times \cdots \times S_n \rightarrow [0, 1]$ s.t. $\sum_{s \in S} \pi(s) = 1$.

Example: $\pi(\text{Wait}, \text{Wait}) = 0$, $\pi(\text{Wait}, \text{Go}) = \pi(\text{Go}, \text{Wait}) = \frac{1}{2}$, and $\pi(\text{Go}, \text{Go}) = 0$

Question

What is a correlated equilibrium?

Answer

A *correlated strategy* is a **correlated equilibrium** when no player *gains* by deviating from the suggested strategy while others follow the suggested strategies

The correlated strategy π is common knowledge

Correlated Equilibrium: Formal Definition



Definition (Correlated Equilibrium)

A **correlated equilibrium** is a correlated strategy π s.t.

Correlated Equilibrium: Formal Definition



Definition (Correlated Equilibrium)

A **correlated equilibrium** is a correlated strategy π s.t.

$$\pi(s_i, s_{-i})$$

$$\pi(s_i, s_{-i})$$



Correlated Equilibrium: Formal Definition

Definition (Correlated Equilibrium)

A **correlated equilibrium** is a correlated strategy π s.t.

$$\pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i})$$



Correlated Equilibrium: Formal Definition

Definition (Correlated Equilibrium)

A **correlated equilibrium** is a correlated strategy π s.t.

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i})$$



Correlated Equilibrium: Formal Definition

Definition (Correlated Equilibrium)

A **correlated equilibrium** is a correlated strategy π s.t.

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i})$$

Correlated Equilibrium: Formal Definition



Definition (Correlated Equilibrium)

A **correlated equilibrium** is a correlated strategy π s.t.

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i}), \quad \forall s_i, s'_i \in S_i, \forall i \in N.$$



Correlated Equilibrium: Formal Definition

Definition (Correlated Equilibrium)

A **correlated equilibrium** is a correlated strategy π s.t.

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i}), \quad \forall s_i, s'_i \in S_i, \forall i \in N.$$

Discussion:

- The mediator suggests actions after running its randomization device π
- Every agent's best response is to follow it, if others follow it too



Chunk 2

Worked Examples of Correlated Equilibrium

($\approx$ 20–25 minutes)

Chess or Football Game



		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

Chess or Football Game



		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

MSNE: $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

Chess or Football Game



		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

MSNE: $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

Question

Is $\pi(C,C) = \frac{1}{2} = \pi(F,F)$ a CE?

Chess or Football Game



		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

MSNE: $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

Question

Is $\pi(C,C) = \frac{1}{2} = \pi(F,F)$ a CE?

Yes!

Chess or Football Game



		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

MSNE: $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

Question

Is $\pi(C,C) = \frac{1}{2} = \pi(F,F)$ a CE?

Yes!

Expected utility: MSNE = $\frac{2}{3}$, CE = $\frac{3}{2}$



Chess or Football Game

		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

		Aryabhata	
		Wait	Go
Chandra	Wait	0,0	1,2
	Go	2,1	-10,-10

Blind Junction Once More

MSNE: $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

Question

Is $\pi(C,C) = \frac{1}{2} = \pi(F,F)$ a CE?

Yes!

Expected utility: MSNE = $\frac{2}{3}$, CE = $\frac{3}{2}$



Chess or Football Game

		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

		Aryabhata	
		Wait	Go
Chandra	Wait	0,0	1,2
	Go	2,1	-10,-10

Blind Junction Once More

MSNE: $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

What are the MSNEs?

Question

Is $\pi(C, C) = \frac{1}{2} = \pi(F, F)$ a CE?

Yes!

Expected utility: MSNE = $\frac{2}{3}$, CE = $\frac{3}{2}$



Chess or Football Game

		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

MSNE: $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

Question

Is $\pi(C,C) = \frac{1}{2} = \pi(F,F)$ a CE?

Yes!

Expected utility: MSNE = $\frac{2}{3}$, CE = $\frac{3}{2}$

		Aryabhata	
		Wait	Go
Chandra	Wait	0,0	1,2
	Go	2,1	-10,-10

Blind Junction Once More

What are the MSNEs?

Question

Is $\pi(W,G) = \pi(W,W) = \pi(G,W) = \frac{1}{3}$ a CE?



Chess or Football Game

		Kabir	
		C	F
Rohan	C	2,1	0,0
	F	0,0	1,2

Chess or Football Game

MSNE: $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

Question

Is $\pi(C,C) = \frac{1}{2} = \pi(F,F)$ a CE?

Yes!

Expected utility: MSNE = $\frac{2}{3}$, CE = $\frac{3}{2}$

		Aryabhata	
		Wait	Go
Chandra	Wait	0,0	1,2
	Go	2,1	-10,-10

Blind Junction Once More

What are the MSNEs?

Question

Is $\pi(W,G) = \pi(W,W) = \pi(G,W) = \frac{1}{3}$ a CE?

Question

Are there other CEs of this game?



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

Computing Correlated Equilibrium

(≈ 20 – 25 minutes)

Computing Correlated Equilibrium



CE finding reduces to solving a set of linear inequalities.

Computing Correlated Equilibrium



CE finding reduces to solving a set of linear inequalities.

Two sets of constraints

Computing Correlated Equilibrium



CE finding reduces to solving a set of linear inequalities.

Two sets of constraints

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i}), \forall s_i, s'_i \in S_i, \forall i \in N$$

Computing Correlated Equilibrium



CE finding reduces to solving a set of linear inequalities.

Two sets of constraints

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i}), \forall s_i, s'_i \in S_i, \forall i \in N$$

Total number of inequalities = $O(n \cdot m^2)$, assuming $|S_i| = m, \forall i \in N$

Computing Correlated Equilibrium



CE finding reduces to solving a set of linear inequalities.

Two sets of constraints

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i}), \forall s_i, s'_i \in S_i, \forall i \in N$$

Total number of inequalities = $O(n \cdot m^2)$, assuming $|S_i| = m, \forall i \in N$

$$\pi(s) \geq 0, \forall s \in S, \quad \sum_{s \in S} \pi(s) = 1$$

Computing Correlated Equilibrium



CE finding reduces to solving a set of linear inequalities.

Two sets of constraints

$$\sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi(s_i, s_{-i}) \cdot u_i(s'_i, s_{-i}), \forall s_i, s'_i \in S_i, \forall i \in N$$

Total number of inequalities = $O(n \cdot m^2)$, assuming $|S_i| = m, \forall i \in N$

$$\pi(s) \geq 0, \forall s \in S, \quad \sum_{s \in S} \pi(s) = 1$$

m^n inequalities

Computing Correlated Equilibrium (contd.)



- These inequalities together form a **feasibility linear program**, easier to solve than the MSNE problem

¹take log of both quantities to see this

Computing Correlated Equilibrium (contd.)



- These inequalities together form a **feasibility linear program**, easier to solve than the MSNE problem
- **MSNE**: total number of support profiles to check = $O(2^{mn})$

¹take log of both quantities to see this

Computing Correlated Equilibrium (contd.)



- These inequalities together form a **feasibility linear program**, easier to solve than the MSNE problem
- **MSNE**: total number of support profiles to check = $O(2^{mn})$
- **CE**: number of inequalities = $O(m^n)$, exponentially smaller than the above ¹

¹take log of both quantities to see this

Computing Correlated Equilibrium (contd.)



- These inequalities together form a **feasibility linear program**, easier to solve than the MSNE problem
- **MSNE**: total number of support profiles to check = $O(2^{mn})$
- **CE**: number of inequalities = $O(m^n)$, exponentially smaller than the above ¹
- This LP can also optimize an objective, e.g., maximize the sum of the players' utilities

¹take log of both quantities to see this



Chunk 4

Correlated Equilibrium vs. Other Notions

($\approx$ 20–25 minutes)

Comparison with Previous Equilibrium Notions



Theorem

*For every **MSNE** σ^* , there exists a **CE** π^**

Comparison with Previous Equilibrium Notions



Theorem

For every **MSNE** σ^* , there exists a **CE** π^*

Proof: Use $\pi^*(s_1, \dots, s_n) = \prod_{i=1}^n \sigma_i^*(s_i)$ and the MSNE characterization theorem

Comparison with Previous Equilibrium Notions



Theorem

For every **MSNE** σ^* , there exists a **CE** π^*

Proof: Use $\pi^*(s_1, \dots, s_n) = \prod_{i=1}^n \sigma_i^*(s_i)$ and the MSNE characterization theorem

- $u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*)$, for all $s'_i \in S_i, i \in N$, an equivalent definition of MSNE

Comparison with Previous Equilibrium Notions



Theorem

For every **MSNE** σ^* , there exists a **CE** π^*

Proof: Use $\pi^*(s_1, \dots, s_n) = \prod_{i=1}^n \sigma_i^*(s_i)$ and the MSNE characterization theorem

- 1 $u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*)$, for all $s'_i \in S_i, i \in N$, an equivalent definition of MSNE
- 2 $\forall s_i \in \delta(\sigma_i^*), u_i(\sigma_i^*, \sigma_{-i}^*) = \sum_{s_i \in \delta(\sigma_i^*)} \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) = u_i(s_i, \sigma_{-i}^*)$



Comparison with Previous Equilibrium Notions

Theorem

For every **MSNE** σ^* , there exists a **CE** π^*

Proof: Use $\pi^*(s_1, \dots, s_n) = \prod_{i=1}^n \sigma_i^*(s_i)$ and the MSNE characterization theorem

- 1 $u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*)$, for all $s'_i \in S_i, i \in N$, an equivalent definition of MSNE
- 2 $\forall s_i \in \delta(\sigma_i^*), u_i(\sigma_i^*, \sigma_{-i}^*) = \sum_{s_i \in \delta(\sigma_i^*)} \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) = u_i(s_i, \sigma_{-i}^*)$
- 3 for all $s_i \in \delta(\sigma_i^*)$ and $s'_i \in S_i$ (using the characterization theorem)



Comparison with Previous Equilibrium Notions

Theorem

For every **MSNE** σ^* , there exists a **CE** π^*

Proof: Use $\pi^*(s_1, \dots, s_n) = \prod_{i=1}^n \sigma_i^*(s_i)$ and the MSNE characterization theorem

- 1 $u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*)$, for all $s'_i \in S_i, i \in N$, an equivalent definition of MSNE
- 2 $\forall s_i \in \delta(\sigma_i^*), u_i(\sigma_i^*, \sigma_{-i}^*) = \sum_{s_i \in \delta(\sigma_i^*)} \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) = u_i(s_i, \sigma_{-i}^*)$
- 3 for all $s_i \in \delta(\sigma_i^*)$ and $s'_i \in S_i$ (using the characterization theorem)

$$u_i(s_i, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*)$$



Comparison with Previous Equilibrium Notions

Theorem

For every **MSNE** σ^* , there exists a **CE** π^*

Proof: Use $\pi^*(s_1, \dots, s_n) = \prod_{i=1}^n \sigma_i^*(s_i)$ and the MSNE characterization theorem

- 1 $u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*)$, for all $s'_i \in S_i, i \in N$, an equivalent definition of MSNE
- 2 $\forall s_i \in \delta(\sigma_i^*), u_i(\sigma_i^*, \sigma_{-i}^*) = \sum_{s_i \in \delta(\sigma_i^*)} \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) = u_i(s_i, \sigma_{-i}^*)$
- 3 for all $s_i \in \delta(\sigma_i^*)$ and $s'_i \in S_i$ (using the characterization theorem)

$$\begin{aligned} u_i(s_i, \sigma_{-i}^*) &\geq u_i(s'_i, \sigma_{-i}^*) \\ \implies \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) &\geq \sigma_i^*(s_i) u_i(s'_i, \sigma_{-i}^*) \end{aligned}$$



Comparison with Previous Equilibrium Notions

Theorem

For every **MSNE** σ^* , there exists a **CE** π^*

Proof: Use $\pi^*(s_1, \dots, s_n) = \prod_{i=1}^n \sigma_i^*(s_i)$ and the MSNE characterization theorem

- ① $u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*)$, for all $s'_i \in S_i, i \in N$, an equivalent definition of MSNE
- ② $\forall s_i \in \delta(\sigma_i^*), u_i(\sigma_i^*, \sigma_{-i}^*) = \sum_{s_i \in \delta(\sigma_i^*)} \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) = u_i(s_i, \sigma_{-i}^*)$
- ③ for all $s_i \in \delta(\sigma_i^*)$ and $s'_i \in S_i$ (using the characterization theorem)

$$\begin{aligned}
 & u_i(s_i, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*) \\
 \implies & \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) \geq \sigma_i^*(s_i) u_i(s'_i, \sigma_{-i}^*) \\
 \implies & \sum_{s_{-i} \in S_{-i}} \sigma_i^*(s_i) \sigma_{-i}^*(s_{-i}) u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \sigma_i^*(s_i) \sigma_{-i}^*(s_{-i}) u_i(s'_i, s_{-i})
 \end{aligned}$$



Comparison with Previous Equilibrium Notions

Theorem

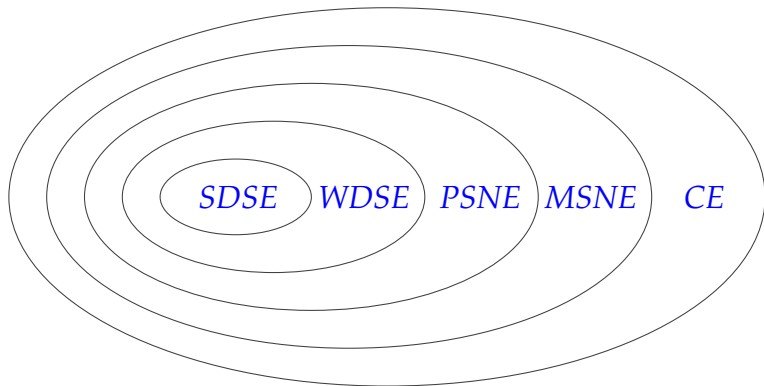
For every **MSNE** σ^* , there exists a **CE** π^*

Proof: Use $\pi^*(s_1, \dots, s_n) = \prod_{i=1}^n \sigma_i^*(s_i)$ and the MSNE characterization theorem

- ① $u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*)$, for all $s'_i \in S_i, i \in N$, an equivalent definition of MSNE
- ② $\forall s_i \in \delta(\sigma_i^*), u_i(\sigma_i^*, \sigma_{-i}^*) = \sum_{s_i \in \delta(\sigma_i^*)} \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) = u_i(s_i, \sigma_{-i}^*)$
- ③ for all $s_i \in \delta(\sigma_i^*)$ and $s'_i \in S_i$ (using the characterization theorem)

$$\begin{aligned}
 & u_i(s_i, \sigma_{-i}^*) \geq u_i(s'_i, \sigma_{-i}^*) \\
 \implies & \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) \geq \sigma_i^*(s_i) u_i(s'_i, \sigma_{-i}^*) \\
 \implies & \sum_{s_{-i} \in S_{-i}} \pi^*(s_i, s_{-i}) u_i(s_i, s_{-i}) \geq \sum_{s_{-i} \in S_{-i}} \pi^*(s_i, s_{-i}) u_i(s'_i, s_{-i})
 \end{aligned}$$

Venn Diagram of Games Having an Equilibrium





Two AI Labs: Test Thoroughly or Ship Fast

Two AI labs, Saarthi AI and Drishti AI, are racing to release a new model. Each lab can Test Thoroughly (slower, safer) or Ship Fast (quick, risky). If both test, both build trust and profit well. If one ships fast while the other tests, the fast shipper grabs the market. If both ship fast, a public failure hurts both.

		Drishti AI	
		Test	Ship
Saarthi AI	Test	5, 5	0, 6
	Ship	6, 0	1, 1

Two AI Labs

Question

What is the CE of this game?

Summary so Far



- Normal form games

Summary so Far



- Normal form games
- Rationality, intelligence, common knowledge

Summary so Far



- Normal form games
- Rationality, intelligence, common knowledge
- Strategy and action

Summary so Far



- Normal form games
- Rationality, intelligence, common knowledge
- Strategy and action
- Dominance, strict and weak, equilibrium: SDSE, WDSE

Summary so Far



- Normal form games
- Rationality, intelligence, common knowledge
- Strategy and action
- Dominance, strict and weak, equilibrium: SDSE, WDSE
- Unilateral deviation, PSNE, generalization: MSNE, existence (Nash)

Summary so Far



- Normal form games
- Rationality, intelligence, common knowledge
- Strategy and action
- Dominance, strict and weak, equilibrium: SDSE, WDSE
- Unilateral deviation, PSNE, generalization: MSNE, existence (Nash)
- Characterization of MSNE, computing, hardness

Summary so Far



- Normal form games
- Rationality, intelligence, common knowledge
- Strategy and action
- Dominance, strict and weak, equilibrium: SDSE, WDSE
- Unilateral deviation, PSNE, generalization: MSNE, existence (Nash)
- Characterization of MSNE, computing, hardness
- Trusted mediator, correlated strategies, equilibrium



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 7

See you next class.



► Lecture 7: Correlated Equilibrium

► Lecture 8: Extensive Form Games and Subgame Perfection



Chunk 1

Perfect Information Extensive Form Games: Formal Model

($\approx$ 20–25 minutes)

Why a Richer Representation?



- Normal form is fine for simultaneous moves, but many games unfold in stages, e.g., **chess**

Why a Richer Representation?

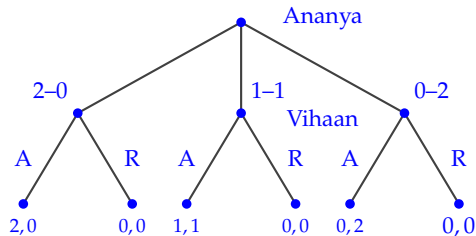


- Normal form is fine for simultaneous moves, but many games unfold in stages, e.g., **chess**
- Players act one after another, and the sequence of actions is the **history** of the game



The Hackathon Prize Split

- Ananya and Vihaan just won a hackathon. The prize is 2 credits.
- Ananya proposes a split: keep both (2-0), split evenly (1-1), or give both to Vihaan (0-2)
- Vihaan then **Accepts** or **Rejects**
- **Rejects** → organizers withhold the whole prize, both get 0



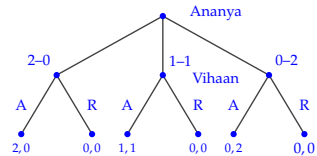


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players



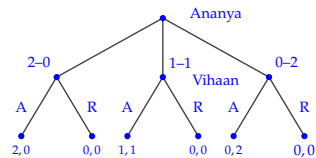


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players
- A : a set of all possible actions (of all players)



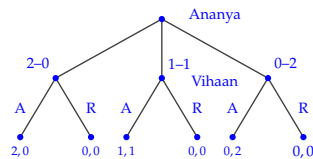


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players
- A : a set of all possible actions (of all players)
- H : a set of all **sequences of actions** satisfying



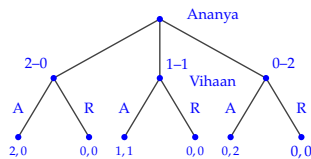


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players
- A : a set of all possible actions (of all players)
- H : a set of all **sequences of actions** satisfying
 - empty history $\emptyset \in H$



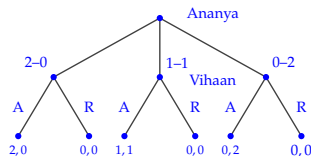


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players
- A : a set of all possible actions (of all players)
- H : a set of all **sequences of actions** satisfying
 - empty history $\emptyset \in H$
 - if $h \in H$, every sub-sequence h' of h starting at the root $\in H$



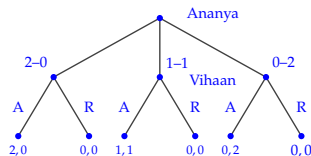


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players
- A : a set of all possible actions (of all players)
- H : a set of all **sequences of actions** satisfying
 - empty history $\emptyset \in H$
 - if $h \in H$, every sub-sequence h' of h starting at the root $\in H$
 - a history $h = (a^{(0)}, a^{(1)}, \dots, a^{(T-1)})$ is **terminal** if $\nexists a^{(T)} \in A$ s.t. $(a^{(0)}, \dots, a^{(T)}) \in H$



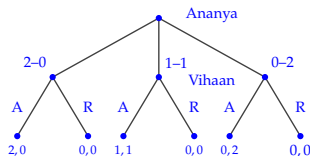


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players
- A : a set of all possible actions (of all players)
- H : a set of all **sequences of actions** satisfying
 - empty history $\emptyset \in H$
 - if $h \in H$, every sub-sequence h' of h starting at the root $\in H$
 - a history $h = (a^{(0)}, a^{(1)}, \dots, a^{(T-1)})$ is **terminal** if $\nexists a^{(T)} \in A$ s.t. $(a^{(0)}, \dots, a^{(T)}) \in H$
 - $Z \subseteq H$: set of all terminal histories



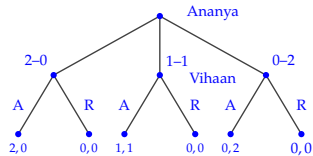


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players
- A : a set of all possible actions (of all players)
- H : a set of all **sequences of actions** satisfying
 - empty history $\emptyset \in H$
 - if $h \in H$, every sub-sequence h' of h starting at the root $\in H$
 - a history $h = (a^{(0)}, a^{(1)}, \dots, a^{(T-1)})$ is **terminal** if $\nexists a^{(T)} \in A$ s.t. $(a^{(0)}, \dots, a^{(T)}) \in H$
 - $Z \subseteq H$: set of all terminal histories
- $X : H \setminus Z \rightarrow 2^A$: action set selection function



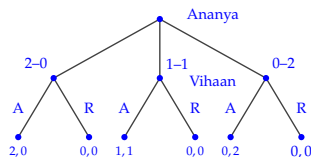


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

- N : a set of players
- A : a set of all possible actions (of all players)
- H : a set of all **sequences of actions** satisfying
 - empty history $\emptyset \in H$
 - if $h \in H$, every sub-sequence h' of h starting at the root $\in H$
 - a history $h = (a^{(0)}, a^{(1)}, \dots, a^{(T-1)})$ is **terminal** if $\nexists a^{(T)} \in A$ s.t. $(a^{(0)}, \dots, a^{(T)}) \in H$
 - $Z \subseteq H$: set of all terminal histories
- $X : H \setminus Z \rightarrow 2^A$: action set selection function
- $P : H \setminus Z \rightarrow N$: player function



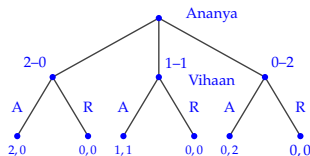


Formal Capture of a PIEFG

Formal capture

PIEFG $\langle N, A, H, X, P, (u_i)_{i \in N} \rangle$

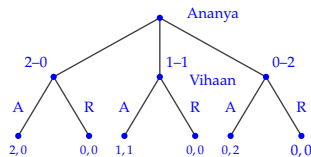
- N : a set of players
- A : a set of all possible actions (of all players)
- H : a set of all **sequences of actions** satisfying
 - empty history $\emptyset \in H$
 - if $h \in H$, every sub-sequence h' of h starting at the root $\in H$
 - a history $h = (a^{(0)}, a^{(1)}, \dots, a^{(T-1)})$ is **terminal** if $\nexists a^{(T)} \in A$ s.t. $(a^{(0)}, \dots, a^{(T)}) \in H$
 - $Z \subseteq H$: set of all terminal histories
- $X : H \setminus Z \rightarrow 2^A$: action set selection function
- $P : H \setminus Z \rightarrow N$: player function
- $u_i : Z \rightarrow \mathbb{R}$: utility of i



Working Out the Tuple on the Hackathon Example



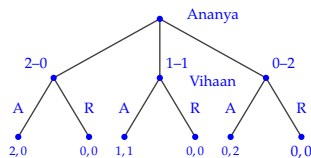
- $N = \{1, 2\}$ – Ananya and Vihaan respectively





Working Out the Tuple on the Hackathon Example

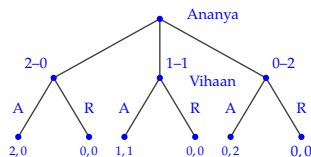
- $N = \{1, 2\}$ – Ananya and Vihaan respectively
- $A = \{2-0, 1-1, 0-2, A, R\}$





Working Out the Tuple on the Hackathon Example

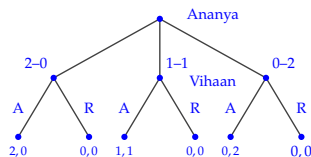
- $N = \{1, 2\}$ – Ananya and Vihaan respectively
- $A = \{2-0, 1-1, 0-2, A, R\}$
- $H = \{\emptyset, (2-0), (1-1), (0-2), (2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$





Working Out the Tuple on the Hackathon Example

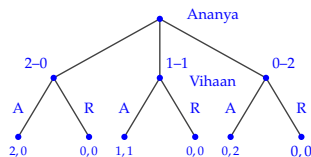
- $N = \{1, 2\}$ – Ananya and Vihaan respectively
- $A = \{2-0, 1-1, 0-2, A, R\}$
- $H = \{\emptyset, (2-0), (1-1), (0-2), (2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $Z = \{(2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$





Working Out the Tuple on the Hackathon Example

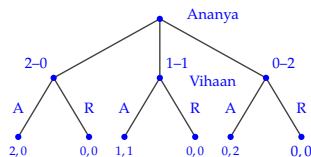
- $N = \{1, 2\}$ – Ananya and Vihaan respectively
- $A = \{2-0, 1-1, 0-2, A, R\}$
- $H = \{\emptyset, (2-0), (1-1), (0-2), (2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $Z = \{(2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $X(\emptyset) = \{(2-0), (1-1), (0-2)\}$





Working Out the Tuple on the Hackathon Example

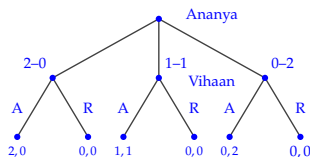
- $N = \{1, 2\}$ – Ananya and Vihaan respectively
- $A = \{2-0, 1-1, 0-2, A, R\}$
- $H = \{\emptyset, (2-0), (1-1), (0-2), (2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $Z = \{(2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $X(\emptyset) = \{(2-0), (1-1), (0-2)\}$
- $X(2-0) = X(1-1) = X(0-2) = \{A, R\}$





Working Out the Tuple on the Hackathon Example

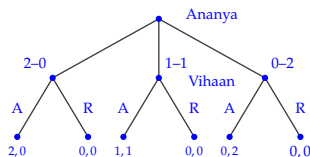
- $N = \{1, 2\}$ – Ananya and Vihaan respectively
- $A = \{2-0, 1-1, 0-2, A, R\}$
- $H = \{\emptyset, (2-0), (1-1), (0-2), (2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $Z = \{(2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $X(\emptyset) = \{(2-0), (1-1), (0-2)\}$
- $X(2-0) = X(1-1) = X(0-2) = \{A, R\}$
- $P(\emptyset) = 1, P(2-0) = P(1-1) = P(0-2) = 2$





Working Out the Tuple on the Hackathon Example

- $N = \{1, 2\}$ – Ananya and Vihaan respectively
- $A = \{2-0, 1-1, 0-2, A, R\}$
- $H = \{\emptyset, (2-0), (1-1), (0-2), (2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $Z = \{(2-0, A), (2-0, R), (1-1, A), (1-1, R), (0-2, A), (0-2, R)\}$
- $X(\emptyset) = \{(2-0), (1-1), (0-2)\}$
- $X(2-0) = X(1-1) = X(0-2) = \{A, R\}$
- $P(\emptyset) = 1, P(2-0) = P(1-1) = P(0-2) = 2$
- $u_1(2-0, A) = 2, u_1(1-1, A) = 1, u_2(1-1, A) = 1, u_2(0-2, A) = 2$ [zero at other terminal histories]





Chunk 2

Strategies, NFG Translation, and Non-Credible Threats

($\approx$ 20–25 minutes)

Strategy in an Extensive Form Game



The **strategy** of a player in an EFG is a tuple of actions at every history where the player moves:

$$S_i = \prod_{\{h \in H: P(h)=i\}} X(h)$$

Remember:

- A strategy is a **complete contingency plan**

Strategy in an Extensive Form Game



The **strategy** of a player in an EFG is a tuple of actions at every history where the player moves:

$$S_i = \prod_{\{h \in H: P(h)=i\}} X(h)$$

Remember:

- A strategy is a **complete contingency plan**
- It specifies an action at every node where the player could move, even nodes that are never actually reached together



Strategy in an Extensive Form Game

The **strategy** of a player in an EFG is a tuple of actions at every history where the player moves:

$$S_i = \prod_{\{h \in H: P(h)=i\}} X(h)$$

Remember:

- A strategy is a **complete contingency plan**
- It specifies an action at every node where the player could move, even nodes that are never actually reached together
- Here $S_1 = \{2-0, 1-1, 0-2\}$, and $S_2 = \{A, R\}^3 = \{AAA, AAR, ARA, ARR, RAA, RAR, RRA, RRR\}$

Transforming the PIEFG into an NFG



Once S_1 and S_2 are known, the game can be written as an NFG.

		Vihaan							
		AAA	AAR	ARA	ARR	RAA	RAR	RRA	RRR
Ananya	2-0	2,0	2,0	2,0	2,0	0,0	0,0	0,0	0,0
	1-1	1,1	1,1	0,0	0,0	1,1	1,1	0,0	0,0
	0-2	0,2	0,0	0,2	0,0	0,2	0,0	0,2	0,0



- A Nash equilibrium like $(2-0, RRA)$ is not quite reasonable: why would Vihaan commit to R at 1-1?



- A Nash equilibrium like $(2-0, RRA)$ is not quite reasonable: why would Vihaan commit to R at 1-1?
- Similarly, $(2-0, RRR)$ relies on a **non-credible threat**: if the history 1-1 is ever reached, Vihaan's rational choice there is not R



- A Nash equilibrium like $(2-0, RRA)$ is not quite reasonable: why would Vihaan commit to R at 1-1?
- Similarly, $(2-0, RRR)$ relies on a **non-credible threat**: if the history 1-1 is ever reached, Vihaan's rational choice there is not R
- So PSNE alone is not a good enough predictor of outcomes in PIEFGs



- A Nash equilibrium like $(2-0, RRA)$ is not quite reasonable: why would Vihaan commit to R at $1-1$?
- Similarly, $(2-0, RRR)$ relies on a **non-credible threat**: if the history $1-1$ is ever reached, Vihaan's rational choice there is not R
- So PSNE alone is not a good enough predictor of outcomes in PIEFGs
- Also, the NFG representation of a sequential game is hugely redundant, the EFG is far more succinct



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

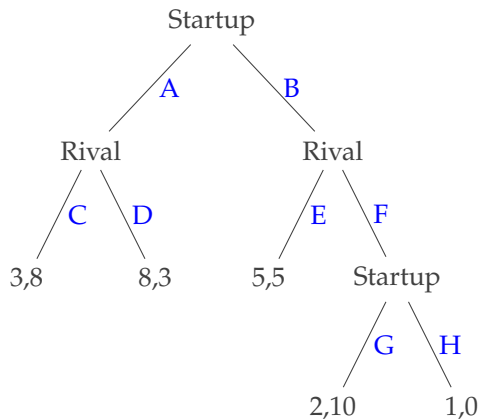
Subgame Perfection and Backward Induction

($\approx$ 20–25 minutes)



Another Game: A Feature Launch Race

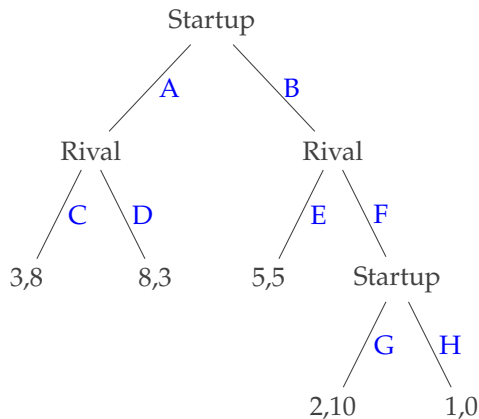
A startup decides to Launch Now (A) or Later (B) – a feature or product. If it launches, a rival can Copy (C) or Ignore (D). If it chooses to launch later, the rival can Launch First (E) or Hold Back (F); if the rival holds back, the startup then chooses Give Up (G) or Grand Release (H).





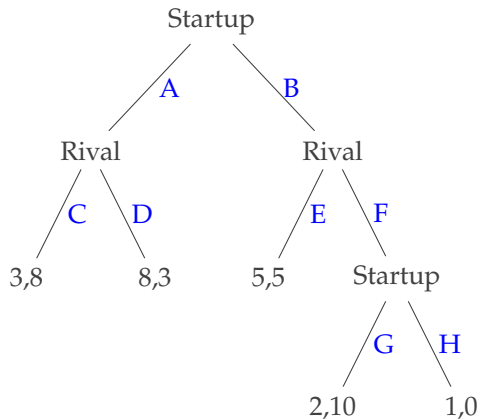
Another Game: A Feature Launch Race

A startup decides to Launch Now (A) or Later (B) – a feature or product. If it launches, a rival can Copy (C) or Ignore (D). If it chooses to launch later, the rival can Launch First (E) or Hold Back (F); if the rival holds back, the startup then chooses Give Up (G) or Grand Release (H).



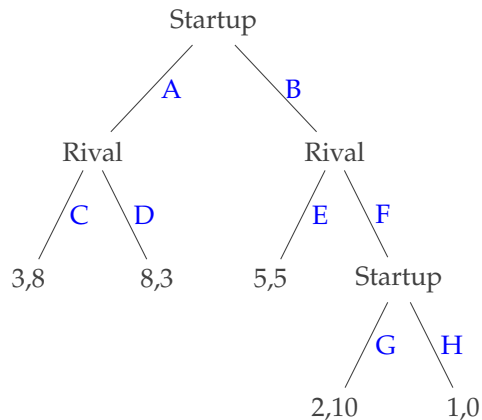
Equilibrium guarantees are weak when this PIEFG is read off its NFG representation.

PSNEs of the Feature Launch Race



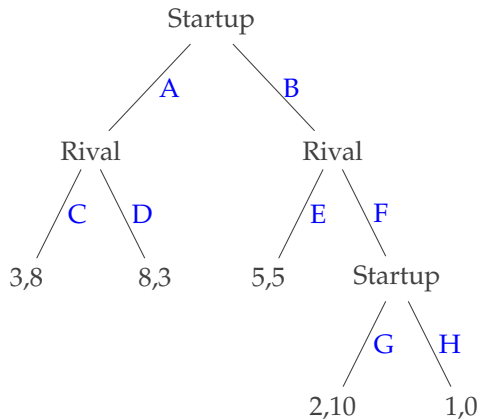
- Strategies of the Startup: AG, AH, BG, BH

PSNEs of the Feature Launch Race



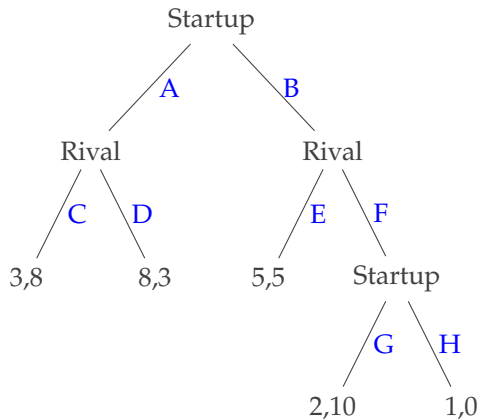
- Strategies of the Startup: AG, AH, BG, BH
- Strategies of the Rival: CE, CF, DE, DF

PSNEs of the Feature Launch Race



- Strategies of the Startup: AG, AH, BG, BH
- Strategies of the Rival: CE, CF, DE, DF
- PSNEs? $(AG, CF), (AH, CF), (BH, CE)$ – is there a non-credible threat here?

PSNEs of the Feature Launch Race



- Strategies of the Startup: AG, AH, BG, BH
- Strategies of the Rival: CE, CF, DE, DF
- PSNEs? $(AG, CF), (AH, CF), (BH, CE)$ – is there a non-credible threat here?
- A better notion of rational outcome should consider history and ensure utility maximization at every point

Subgame and Subgame Perfection



Subgame: the game rooted at an intermediate vertex

Subgame and Subgame Perfection



Subgame: the game rooted at an intermediate vertex

Definition (Subgame)

The subgame of a PIEFG G rooted at a history h is the *restriction* of G to the descendants of h .

Subgame and Subgame Perfection



Subgame: the game rooted at an intermediate vertex

Definition (Subgame)

The subgame of a PIEFG G rooted at a history h is the *restriction* of G to the descendants of h .

The set of subgames of G is the collection of all subgames at some history of G

Subgame and Subgame Perfection



Subgame: the game rooted at an intermediate vertex

Definition (Subgame)

The subgame of a PIEFG G rooted at a history h is the *restriction* of G to the descendants of h .

The set of subgames of G is the collection of all subgames at some history of G

Subgame perfection: best response at every subgame



Subgame and Subgame Perfection

Subgame: the game rooted at an intermediate vertex

Definition (Subgame)

The subgame of a PIEFG G rooted at a history h is the *restriction* of G to the descendants of h .

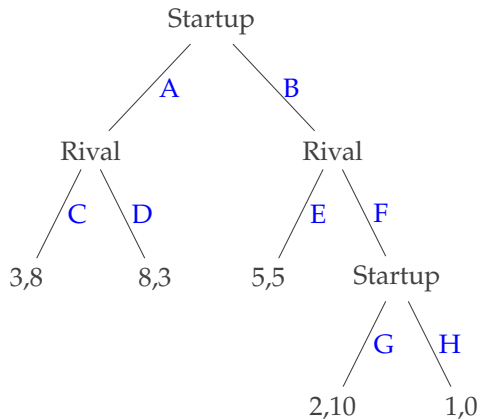
The set of subgames of G is the collection of all subgames at some history of G

Subgame perfection: best response at every subgame

Definition (Subgame Perfect Nash Equilibrium (SPNE))

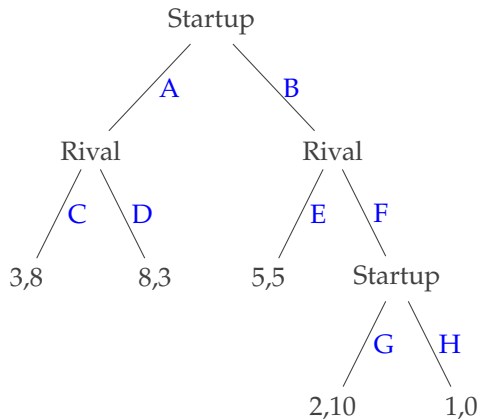
A subgame perfect Nash equilibrium (SPNE) of a PIEFG G is a strategy profile $s \in S$ s.t. for every subgame G' of G , the restriction of s to G' is a PSNE of G'

Which PSNEs Survive?



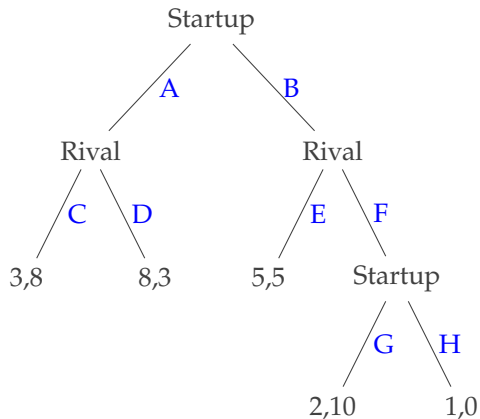
- PSNEs: (AH, CF) , (BH, CE) , (AG, CF)

Which PSNEs Survive?



- PSNEs: (AH, CF) , (BH, CE) , (AG, CF)
- Are they all SPNEs?

Which PSNEs Survive?



- PSNEs: (AH, CF) , (BH, CE) , (AG, CF)
- Are they all SPNEs?
- How do we compute them systematically?



Backward Induction

Algorithm 1: Backward Induction

```
1 Function BACK_IND(history h):  
2   if  $h \in Z$  then  
3      $\sqsubset$  return  $u(h), \emptyset$   
4    $best\_util_{P(h)} \leftarrow -\infty$   
   foreach  $a \in X(h)$  do  
5      $util\_at\_child_{P(h)}, str\_at\_child_{P(h)} \leftarrow BACK\_IND((h, a))$   
     if  $util\_at\_child_{P(h)} > best\_util_{P(h)}$  then  
6        $\sqsubset$   $best\_util_{P(h)} \leftarrow util\_at\_child_{P(h)}, best\_str_{P(h)} \leftarrow a + str\_at\_child_{P(h)}$   
7   return  $best\_util_{P(h)}, best\_str_{P(h)}$ 
```



Chunk 4

Limitations of SPNE and the Centipede Game

($\approx$ 20–25 minutes)

Limitations of SPNE



Subgame perfection is inherently built on backward induction.

Limitations of SPNE



Subgame perfection is inherently built on backward induction.

Advantages:

- SPNE is guaranteed to exist in finite PIEFGs (requires proof)



Subgame perfection is inherently built on backward induction.

Advantages:

- SPNE is guaranteed to exist in finite PIEFGs (requires proof)
- Every SPNE is a PSNE, so this is a class of games where PSNE is guaranteed to exist



Subgame perfection is inherently built on backward induction.

Advantages:

- SPNE is guaranteed to exist in finite PIEFGs (requires proof)
- Every SPNE is a PSNE, so this is a class of games where PSNE is guaranteed to exist
- The algorithm to find an SPNE is quite simple



Subgame perfection is inherently built on backward induction.

Advantages:

- SPNE is guaranteed to exist in finite PIEFGs (requires proof)
- Every SPNE is a PSNE, so this is a class of games where PSNE is guaranteed to exist
- The algorithm to find an SPNE is quite simple



Subgame perfection is inherently built on backward induction.

Advantages:

- SPNE is guaranteed to exist in finite PIEFGs (requires proof)
- Every SPNE is a PSNE, so this is a class of games where PSNE is guaranteed to exist
- The algorithm to find an SPNE is quite simple

Disadvantages and criticisms:

- The whole tree must be parsed, which can be computationally expensive, or effectively impossible: chess has $\sim 10^{150}$ vertices



Subgame perfection is inherently built on backward induction.

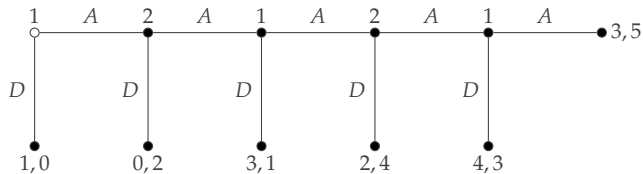
Advantages:

- SPNE is guaranteed to exist in finite PIEFGs (requires proof)
- Every SPNE is a PSNE, so this is a class of games where PSNE is guaranteed to exist
- The algorithm to find an SPNE is quite simple

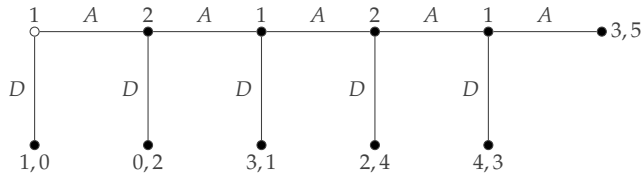
Disadvantages and criticisms:

- The whole tree must be parsed, which can be computationally expensive, or effectively impossible: chess has $\sim 10^{150}$ vertices
- The cognitive limits of real players may prevent them from actually playing an SPNE

The Centipede Game



The Centipede Game



Question

What is/are the SPNE(s) of this game?

Question

What is the problem with that prediction?



- This game has been run on many populations: random participants, university students, grandmasters, and more



- This game has been run on many populations: random participants, university students, grandmasters, and more
- Most subjects, except grandmasters, continue for a few rounds instead of quitting at the very first move



- This game has been run on many populations: random participants, university students, grandmasters, and more
- Most subjects, except grandmasters, continue for a few rounds instead of quitting at the very first move
- **Reasons claimed:** altruism, limited computational capacity, differences in incentives



- This game has been run on many populations: random participants, university students, grandmasters, and more
- Most subjects, except grandmasters, continue for a few rounds instead of quitting at the very first move
- **Reasons claimed:** altruism, limited computational capacity, differences in incentives
- **Criticism of SPNE's defining principle:** it asks “what would I do if this history were reached,” but an equilibrium reasoned from above may say that history is “never reached” at all



- This game has been run on many populations: random participants, university students, grandmasters, and more
- Most subjects, except grandmasters, continue for a few rounds instead of quitting at the very first move
- **Reasons claimed:** altruism, limited computational capacity, differences in incentives
- **Criticism of SPNE's defining principle:** it asks “what would I do if this history were reached,” but an equilibrium reasoned from above may say that history is “never reached” at all
- SPNE still explains outcomes in many games, but there is another way to extend the idea, using the players' **beliefs**



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 8

See you next class.



भारतीय प्रौद्योगिकी संस्थान मुंबई

Indian Institute of Technology Bombay