

# Dynamic Games of Complete Information

Lorenzo Ferrari

University of Rome Tor Vergata

# Outline

- Introduction
- Extensive Form
- Backwards Induction and SPNE
- Stackelberg
- Introducing Imperfect Info
- Bank Runs

# Assumptions

- **Dynamic**: the moves occur in *sequence*
- **Complete**: the players' payoffs from each feasible combination of moves are *common knowledge*

Information can be either **perfect** or **imperfect**:

1. **Perfect**: all previous moves observed before the next is chosen
2. **Imperfect**: may not be observed before the next is chosen

We start with **Dynamic Games** of **complete** and **perfect** info

# Outline

- Introduction
- Extensive Form
- Backwards Induction and SPNE
- Stackelberg
- Introducing Imperfect Info
- Bank Runs

# Definition of Extensive form

**Definition:** *the extensive-form representation of a game specifies: (1) the **players** of the game, (2a) **when** each player has the move, (2b) **what** each player **can do** at each of his or her opportunities to move, (2c) **what** each player **knows** at each of his or her opportunities to move, and (3) the **payoff** received by each player for each combination of moves that could be chosen by the players.*

# Building Elements

1. **Players:**  $N = \{\}$  (1)
2. **Nodes**  $X$ 
  - **Root** or starting point of the game  $r \in X \rightarrow r = \{\}$
  - **Decision Nodes** for each player  $X_i \in X \rightarrow X_i = \{\}$
  - **Terminal Nodes**  $T \in X \rightarrow T = \{\}$
3. **Information Sets** Set of decision nodes that share the same information about the story of the game  $I_i \in X \rightarrow I = \{\}$ 

Notice that if the game is of *perfect info*,  $I_i = X_i$  (singletons)
4. **Strategies**  $S_i$ : A *complete plan of action* that specifies a feasible action for the player in **every contingency in which the player might be called to act** (at each info set)
5. **Payoffs** at each terminal node  $T \rightarrow u_i$

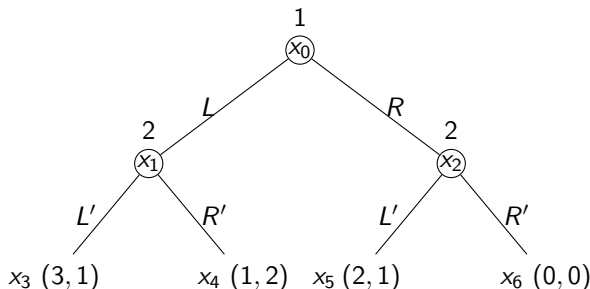
# Subgames

**Definition:** *A subgame in an extensive-form game*

- a. *begins at a decision node  $n$  that is a **singleton** information set (excluding the root)*
- b. *includes all the decision and terminal nodes following  $n$  in the game tree (but not the nodes that do not follow  $n$ ), and*
- c. *does not cut any information sets*

Notice that the whole game is a subgame of itself (improper)

# Example (1)



- $N = \{1, 2\}$
- Decision nodes:  $X_1 = \{x_0\}$  and  $X_2 = \{x_1, x_2\}$
- Info sets:  $I_1 = \{x_0\}$  and  $I_2 = \{\{x_1\}, \{x_2\}\}$
- Root:  $r = \{x_0\}$
- Terminal nodes:  $T = \{x_3, x_4, x_5, x_6\}$
- Strategies:  $S_1 = \{L, R\}$  and  $S_2 = \{L'L', L'R', R'L', R'R'\}$



## Example (2)

- A dynamic game of complete information can be represented using the corresponding **normal form**

		Player 2			
		$L'L'$	$L'R'$	$R'L'$	$R'R'$
Player 1	$L$	(3, 1)	(3, 1)	(1, 2)	(1, 2)
	$R$	(2, 1)	(0, 0)	(2, 1)	(0, 0)

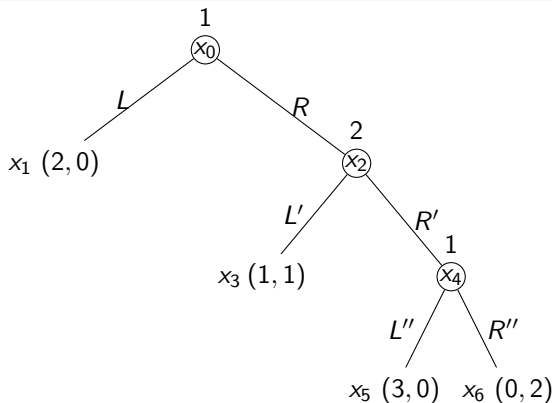
- Notice that it is possible to find the *NE* of this game
- Subgames:** 3
  - The one starting at  $x_1$
  - The one starting at  $x_2$
  - The whole game

# Example (3)

		Player 2			
		$L'L'$	$L'R'$	$R'L'$	$R'R'$
Player 1	$L$	(3, 1)	(3, 1)	(1, 2)	(1, 2)
	$R$	(2, 1)	(0, 0)	(2, 1)	(0, 0)

- Notice that we have two *NE* in this game. However,  $(L, R'R')$  is not **sequentially rational** for player 2
- We need a **stronger solution concept** than *NE*

# Another Example



- $N = \{1, 2\}$
- Decision nodes:  $X_1 = I_1 = \{x_0, x_4\}$  and  $X_2 = I_2 = \{x_2\}$
- Info sets:  $I_1 = \{\{x_0\}, \{x_4\}\}$  and  $I_2 = \{x_2\}$
- Root:  $r = \{x_0\}$
- Terminal nodes:  $T = \{x_3, x_5, x_6\}$
- Strategies:  $S_1 = \{(L, L''), (L, R''), (R, L''), (R, R'')\}$  and  $S_2 = \{L', R'\}$

## Another Example (2)

- Corresponding normal-form

		Player 2	
		$L'$	$R'$
Player 1	$LL''$	(2, 0)	(2, 0)
	$LR''$	(2, 0)	(2, 0)
	$RL''$	(1, 1)	(3, 0)
	$RR''$	(1, 1)	(0, 2)

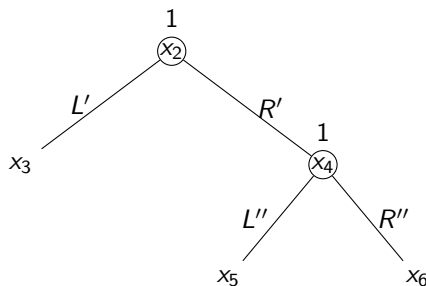
## Another Example (3)

		Player 2	
		$L'$	$R'$
Player 1	$LL''$	(2, 0)	(2, 0)
	$LR''$	(2, 0)	(2, 0)
	$RL''$	(1, 1)	(3, 0)
	$RR''$	(1, 1)	(0, 2)

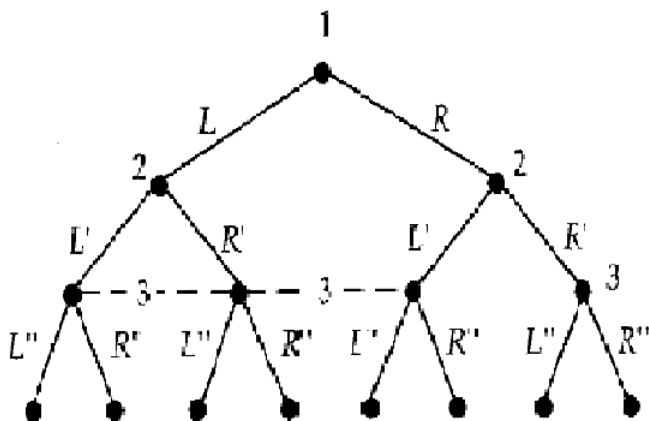
- Notice that we have **4 NE** in this game. However,  $(LL'', R')$ ,  $(LR'', L')$ , and  $(LR'', R')$  include actions that are not **sequentially rational**
- Again, we need a **stronger solution concept** than **NE**

## Another Example (4) -Subgames

- 3 subgames
  1. The one starting at  $x_2$
  2. The one starting at  $x_4$
  3. The whole game

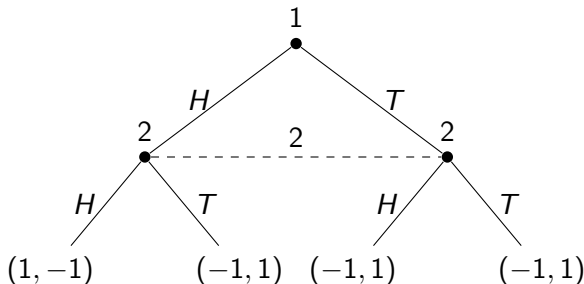


# How Many Subgames?



# Static Games of Complete Info and Extensive Form

- It is possible to represent **static games of complete information** using the extensive form



- Player 2 does not know if she is at the left or right node. We denote this with a *dashed* line between the two decisions node of Player 2
- Subgames: 1, the whole game



# Outline

- Introduction
- Extensive Form
- **Backwards Induction and SPNE**
- Stackelberg
- Introducing Imperfect Info
- Bank Runs

# How to Solve Dynamic Games: Theory (1)

- $N = \{\text{Player 1, Player 2}\}$
- $a_i \in A_i$  are actions for Player  $i$  (not strategies)
- Game is *sequential*:
  1. *First stage*: Player 1 chooses  $a_1$
  2. *Second stage*: Player 2 observes  $a_1$  and then chooses  $a_2$
- Payoffs are  $u_1(a_1, a_2)$  and  $u_2(a_1, a_2)$  chooses

We solve these games using **backwards induction** (looking forward, solving backwards)

## How to Solve Dynamic Games: Theory (2)

- Start from **Stage 2**, where Player 2 chooses  $a_2$  as to maximise

$$\max_{a_2 \in A_2} u_2(a_1, a_2).$$

Solution is Player 2's **reaction function** to Player 1's action

$$R_2(a_1)$$

- Player 1 can solve 2's problem as well (complete info) and **anticipate**  $R_2(a_1)$ . Player 1 maximises

$$\max_{a_1 \in A_1} u_1(a_1, R_2(a_1)).$$

- The backwards-induction **outcome** of the game is  $(a_1^*, R_2(a_1^*))$ . Since Player 2 responds **optimally** to  $a_1^*$ , no **non-credible threats**

# Subgame-Perfect Nash Equilibrium

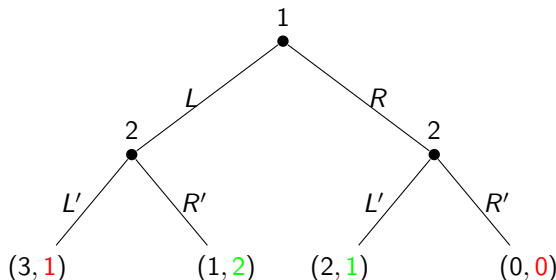
**Definition:** A Nash equilibrium is **subgame-perfect** if the players' strategies constitute a Nash equilibrium in every subgame

- Related to the concept of backwards induction
- but not exactly the same thing

**Definition** In the two-stage game of complete and perfect information the backwards-induction outcome is  $(a_1^*, R_2(a_1^*))$  but the subgame-perfect Nash equilibrium is  $(a_1^*, R_2(a_1))$

- The difference is that the **SPNE** includes the strategies played by player 2 in all the contingencies in which she may be called to act

## Example - Second Stage

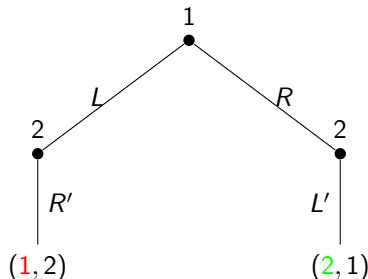


Start with Player 2:

- If player 1 chose  $L$  player 2 chooses  $R'$  as  $2 > 1$
- If player 1 chose  $R$  player 2 chooses  $L'$  as  $1 > 0$

We can **delete** the branches that will not be chosen

## Example - First Stage



Player 1 knows the reactions of Player 2 to  $L$  and  $R$ :

- If player 1 chooses  $L$  she gets 1
- If player 1 chooses  $R$  she gets 2

Player 1 will choose  $R$  as  $2 > 1$

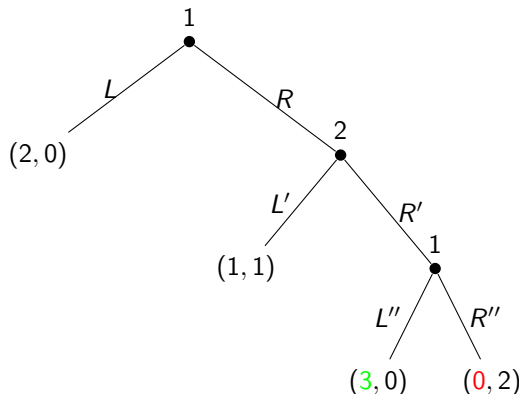
- The *backwards-induction* outcome of this game is  $(R, L')$
- The *SPNE* is  $(R, R', L')$

# Example: Corresponding Normal Form and NE

		Player 2			
		$L'L'$	$L'R'$	$R'L'$	$R'R'$
Player 1	$L$	(3, 1)	(3, 1)	(1, 2)	(1, 2)
	$R$	(2, 1)	(0, 0)	(2, 1)	(0, 0)

- Notice that we have two *NE* in this game. However,  $(L, R'R')$  is not **sequentially rational** for player 2

## Another Example - Third Stage

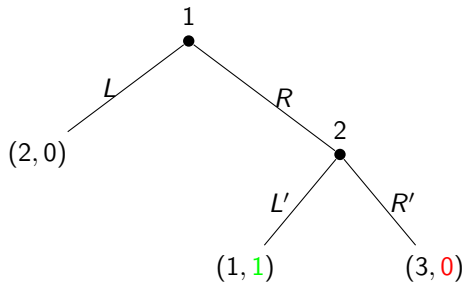


- If player 2 chose  $R'$  player 1 chooses  $L''$  as  $3 > 0$

We can **delete** the branches that will not be chosen



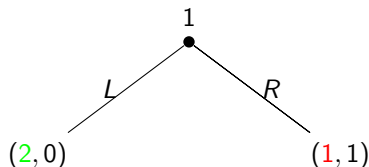
## Another Example - Second Stage



- If player 1 chose  $R$  player 2 chooses  $L'$  as  $1 > 0$

We can **delete** the branches that will not be chosen

## Another Example - First Stage



Player 1 knows the reactions of Player 2 to  $L$  and  $R$ :

- If player 1 chooses  $L$  she gets 2
- If player 1 chooses  $R$  she gets 1

Player 1 will choose  $R$  as  $2 > 1$ .

- The *backwards-induction* outcome of this game is  $(L)$
- The *SPNE* is  $(LL'', L')$

# Another Example: Corresponding Normal Form and NE

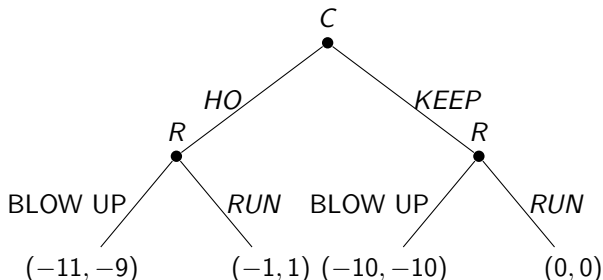
		Player 2	
		$L'$	$R'$
Player 1	$LL''$	(2, 0)	(2, 0)
	$LR''$	(2, 0)	(2, 0)
	$RL''$	(1, 1)	(3, 0)
	$RR''$	(1, 1)	(0, 2)

- Notice that we have **4 NE** in this game. However,  $(LL'', R')$ ,  $(LR'', L')$ , and  $(LR'', R')$  include actions that are not **sequentially rational**

# Bank Robbery - Extensive Form (1)

- A clerk works in a bank. A robber enters shouting: if you do not give me the money, I will blow both of us up.
- The clerk move first and can *hand over* or *keep* the money. The robber observes the clerk's choice and then decides to *blow up* or *run*

Game tree



## Bank Robbery - Extensive Form (2)

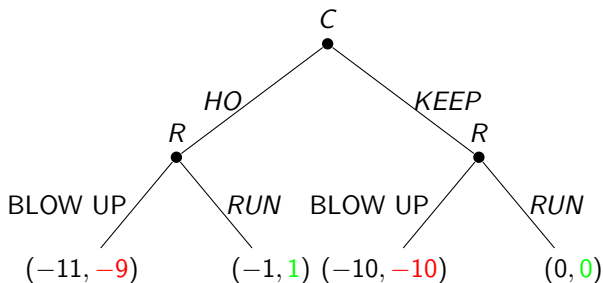
- $N = \{Clerk, Robber\}$
- $X_C = I_C = \{x_0\}$  and  $X_R = I_R = \{x_1, x_2\}$
- $I_C = \{x_0\}$  and  $I_R = \{\{x_1\}, \{x_2\}\}$
- $r = \{x_0\}$
- $T = \{x_3, x_4, x_5, x_6\}$
- $S_C = \{HO, KEEP\}$  and  
 $S_R = \{B B, B R, R B, R R\}$

# Bank Robbery - Corresponding Normal Form

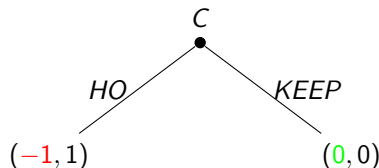
		Player 2			
		$B\ B$	$B\ R$	$R\ B$	$R\ R$
Player 1	$HO$	$(-11, -9)$	$(-11, -9)$	$(-1, 1)$	$(-1, 1)$
	$KEEP$	$(-10, -10)$	$(0, 0)$	$(-10, -10)$	$(0, 0)$

- There are **three** *NE* in this game, but  $(KEEP, B\ R)$  and  $(HO, R\ B)$  are **off the equilibrium pattern**
- These will never be played, as  $B$  is a **non-credible threat**

# Bank Robbery - Second Stage



# Bank Robbery - First Stage



- If C chooses *HO* he gets -1
- If C chooses *K* he gets 0

Player 1 will choose *K* as  $0 > -1$ .

- The *backwards-induction* outcome of this game is  $(K, R)$
- The *SPNE* is  $(K, R, R)$

Blow up is not a credible threat



# Outline

- Introduction
- Extensive Form
- Backwards Induction and SPNE
- **Stackelberg**
- Introducing Imperfect Info
- Bank Runs

# Assumptions (1)

- Players: two firms

$$N = \{Firm\ i, Firm\ j\}.$$

- Strategies: quantity of a *homogeneous* good to be produced (infinite is not included in the production interval)

$$S_i = S_j = [0, \infty) \text{ or}$$

$$q_i = q_j = [0, \infty).$$

- Inverse market demand is  $p(q) = a - Q$
- Payoffs: firm's profits

$$\pi_i = [a - Q - c]q_i,$$

where  $Q = q_i + q_j$ ,  $c$  is constant *marginal cost* of production, and  $a$  is a parameter

## Assumptions (2)

- General motors is the US during the 60s-70s (Ford, Chrysler were **followers**)

Dynamic game of complete information:

- Firm  $i$  **chooses** a quantity  $q_i \geq 0$
- Firm  $j$  **observes**  $q_i$  and picks  $q_j \geq 0$
- Payoffs are given by **profit functions**

Solved using **backwards-induction**

## Stage 2: Firm $j$ 's Maximisation Problem

- We proceed **backwards**. Firm  $j$  picks  $q_j$  for *any arbitrary quantity* (no star) produced by Firm  $i$  by maximising

$$\max_{q_j} \pi_j(q_i, q_j) = [a - (q_i + q_j) - c]q_j.$$

- Take the first derivative and equate to zero

$$\frac{\delta \pi_j(q_i, q_j)}{\delta q_j} = a - 2q_j - q_i - c = 0.$$

$$q_j = R_j(q_i) = \frac{a - q_i - c}{2}$$

- As the game is dynamic this is **truly** Firm 2's reaction function, not only the best response

## Stage 1: Firm $i$ 's Maximisation Problem

- Firm  $i$  can also solve Firm  $j$ 's maximisation problem, anticipating the latter's response to any  $q_i$
- The latter replaces  $R_j$  in its maximisation problem

$$\max_{q_i} \pi_i(q_i, R_j(q_i)) = [a - (q_i + R_j(q_i)) - c]q_i.$$

$$\max_{q_i} [a - q_i - \left(\frac{a - q_i - c}{2}\right) - c]q_i = \left[\frac{a - q_i - c}{2}\right]q_i.$$

- Take the first derivative and equate to zero

$$\frac{\delta \pi_i(q_i, R_j(q_i))}{\delta q_j} = \frac{a - 2q_i - c}{2} = 0.$$

$$q_i^* = \frac{a - c}{2} \text{ and } q_j^* = \frac{a - \left(\frac{a - c}{2}\right) - c}{2} = \frac{a - c}{4}.$$

# Equilibrium Profits and First-Mover Advantage

- To find equilibrium payoffs (profits), we plug equilibrium quantities in the firms' profit function

$$\begin{aligned}\pi_i^s(q_i^*, q_j^*) &= \left[ a - \left( \frac{a-c}{2} + \frac{a-c}{4} \right) - c \right] \frac{a-c}{2} = \\ &= \left[ \frac{4a - a + c - 2a + 2c - 4c}{4} \right] \frac{a-c}{2} = \left[ \frac{a-c}{4} \right] \frac{a-c}{2} = \frac{(a-c)^2}{8}.\end{aligned}$$

$$\begin{aligned}\pi_j^s(q_i^*, q_j^*) &= \left[ a - \left( \frac{a-c}{2} + \frac{a-c}{4} \right) - c \right] \frac{a-c}{4} = \\ &= \left[ \frac{4a - a + c - 2a + 2c - 4c}{4} \right] \frac{a-c}{4} = \left[ \frac{a-c}{4} \right] \frac{a-c}{4} = \frac{(a-c)^2}{16}.\end{aligned}$$

- Firm  $i$  has the so-called **first-mover advantage**, i.e.  $\pi_i^s > \pi_j^s$

# Stackelberg vs Cournot (1)

- Overall *industry profits* are given by

$$\pi^s = \pi_i^s + \pi_j^s = \frac{(a-c)^2}{8} + \frac{(a-c)^2}{16} = \frac{3(a-c)^2}{16}$$

- In Cournot industry profits are

$$\pi^c = \frac{(a-c)^2}{9} + \frac{(a-c)^2}{9} = \frac{2(a-c)^2}{9}$$

- Industry profits are larger in Cournot

$$\pi^c > \pi^s$$

- Why is this the case? In Stackelberg overall quantity is **larger** and **prices are lower**

## Stackelberg vs Cournot (2)

- Overall quantity in Stackelberg is

$$Q^s = \frac{(a-c)}{2} + \frac{(a-c)}{4} = \frac{3(a-c)}{4}$$

- Overall quantity in Cournot is

$$Q^c = \frac{(a-c)}{3} + \frac{(a-c)}{3} = \frac{2(a-c)}{3}$$

- Industry quantity is larger in Stackelberg

$$Q^s > Q^c$$



# Outline

- Introduction
- Extensive Form
- Backwards Induction and SPNE
- Stackelberg
- **Introducing Imperfect Info**
- Bank Runs

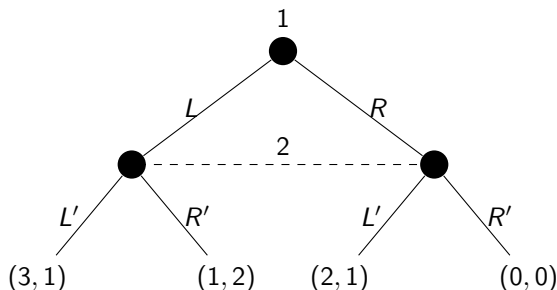
# Assumptions

- **Dynamic**: the moves occur in *sequence*
- **Complete**: the players' payoffs from each feasible combination of moves are *common knowledge*
- Information is **imperfect**: previous moves **may not be observed** before the next is chosen

# How to Solve Dynamic Games of Complete but Imperfect Info

- Solution concept is still **SPNE**
- Backwards induction is not feasible in some stages
- Some games must be solved as a **static game of complete information**

# Example (1)



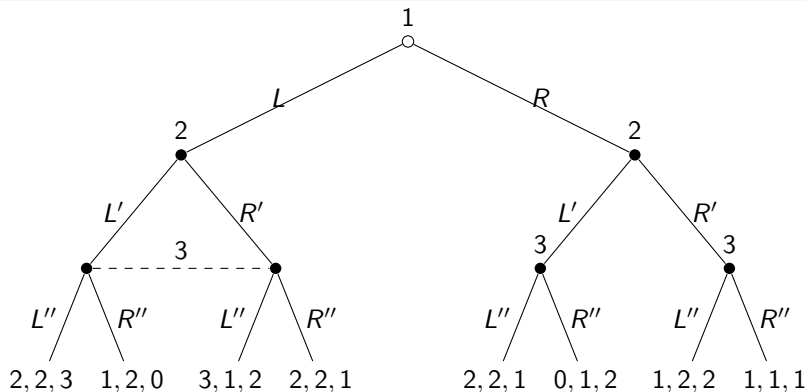
- $N = \{1, 2\}$
- Decision nodes:  $X_1 = \{x_0\}$  and  $X_2 = \{x_1, x_2\}$
- Info sets:  $I_1 = \{x_0\}$  and  $I_2 = \{x_1, x_2\}$
- Root:  $r = \{x_0\}$
- Terminal nodes:  $T = \{x_3, x_4, x_5, x_6\}$
- Strategies:  $S_1 = \{L, R\}$  and  $S_2 = \{L', R'\}$

## Example (2)

		Player 2	
		$L'$	$R'$
Player 1	$L$	( <u>3</u> , 1)	( <u>1</u> , <u>2</u> )
	$R$	(2, <u>1</u> )	(0, 0)

- The only *SPNE* of this game is  $(L, R')$
- This is the *NE* (in pure strategies) of the only subgame, the whole game

# Example 2 (1)



- $N = \{1, 2, 3\}$
- Decision nodes:  $X_1 = \{x_0\}$  and  $X_2 = \{x_1, x_2\}$  and  $X_3 = \{x_3, x_4, x_5, x_6\}$
- Info sets:  $I_1 = \{x_0\}$  and  $I_2 = \{\{x_1\}, \{x_2\}\}$  and  $I_3 = \{\{x_3, x_4\}, \{x_5\}, \{x_6\}\}$
- Root:  $r = \{x_0\}$
- Terminal nodes:  $T = \{x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}\}$
- Strategies:  $S_1 = \{L, R\}$  and  $S_2 = \{L'L', L'R', R'L', R'R'\}$  and  $S_3 = \{L''L''L'', L''R''L'', L''L''R'', L''R''R'', R''L''L'', R''R''L'', R''L''R'', R''R''R''\}$

## Example 2 (Subgames)

- Five subgames:
  1. The whole game
  2. Starting at  $x_1$
  3. Starting at  $x_2$
  4. Starting at  $x_5$
  5. Starting at  $x_6$

## Example 2 - Third Stage

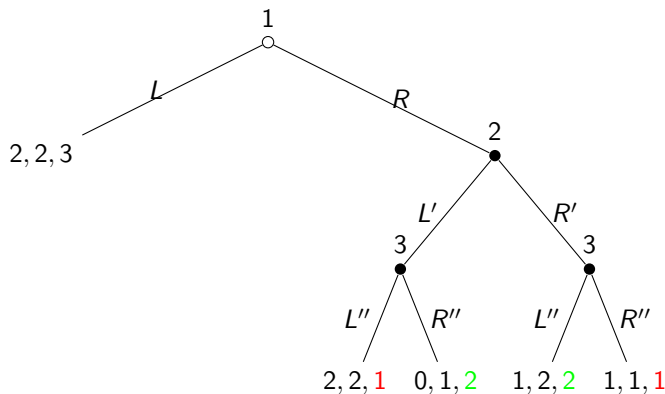
- We solve the game starting at  $x_2$  as a simultaneous game

		Player 3	
		$L''$	$R''$
Player 2	$L'$	( <u>2</u> , <u>3</u> )	( <u>2</u> , 0)
	$R'$	(1, <u>2</u> )	( <u>2</u> , 1)

- The only *NE* in this subgame is  $(L', L'')$ . We can delete the branches that will not be chosen
- Notice that  $L''$  is **strictly dominant** for Player 3
- $L'$  is **weakly dominant** for Player 2

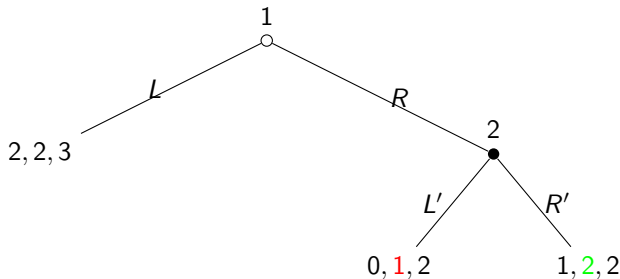


# Example 2 - Backwards Induction (1)



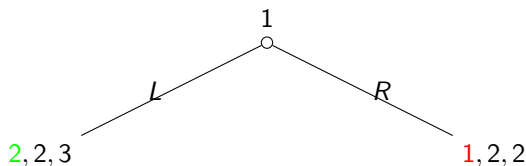
- We can delete the corresponding branches

## Example 2 - Backwards Induction (2)



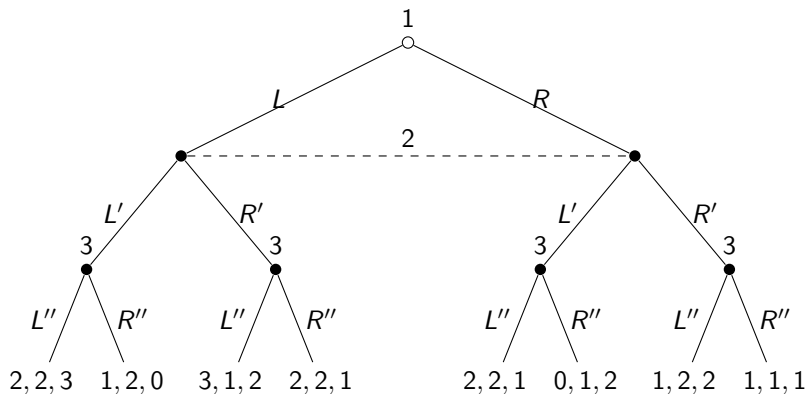
- We can delete the corresponding branches

## Example 2 - Backwards Induction (3)



- The **backwards-induction outcome** is  $(L, L', L'')$
- The **SPNE** is  $(L, L' R', L'' R'' L'')$
- Equilibrium of all the subgames that compose this game

# Example 3 (1)

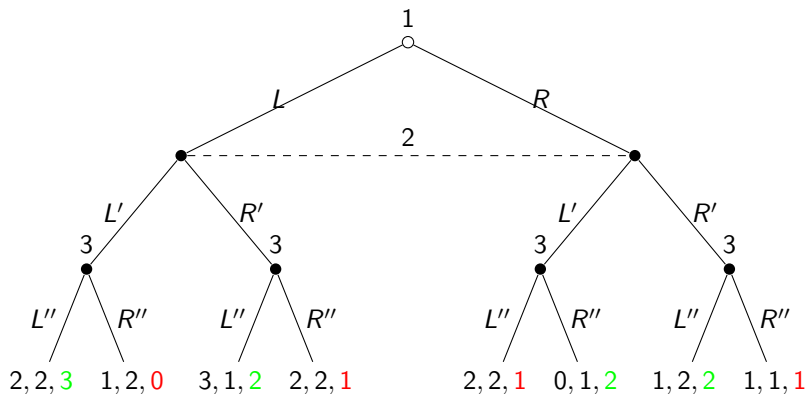


- $N = \{1, 2, 3\}$
- Decision nodes:  $X_1 = \{x_0\}$  and  $X_2 = \{x_1, x_2\}$  and  $X_3 = \{x_3, x_4, x_5, x_6\}$
- Info sets:  $I_1 = \{x_0\}$  and  $I_2 = \{x_1, x_2\}$  and  $I_3 = \{\{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}\}$
- Root:  $r = \{x_0\}$
- Terminal nodes:  $T = \{x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}\}$

## Example 3 - Strategies and Subgames

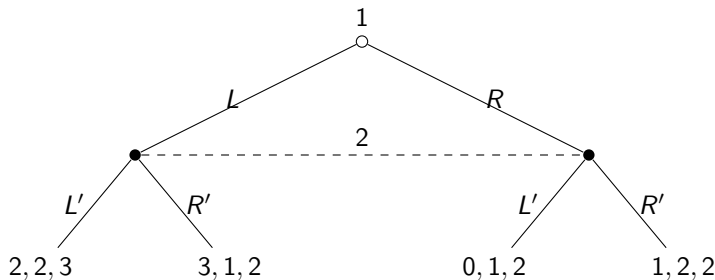
- $S_1 = \{L, R\}$
- $S_2 = \{L', R'\}$
- $S_3 = \{L''L''L''L'', R''L''L''L'', R''R''L''L'', R''R''R''L'', R''R''R''R''L''R''R''R'', L''L''R''R'', L''L''L''R'', R''R''R''L'', \dots\}$
- Player 3 has many strategies
- There are five subgames:
  1. The whole game
  2. The game starting at  $x_3$
  3. The game starting at  $x_4$
  4. The game starting at  $x_5$
  5. The game starting at  $x_6$

# Example 3 - Backwards Induction (1)



- We can delete the corresponding branches

## Example 3 - Backwards Induction (2)



- We cannot use backwards induction in the second stage!

## Example 2 - Second/First Stage

- We solve the game starting at  $x_0$  as a simultaneous game

		Player 2	
		$L'$	$R'$
Player 1	$L$	( <u>2</u> , <u>2</u> )	( <u>3</u> , 1)
	$R$	(0, 1)	(1, <u>2</u> )

- The only *NE* in this subgame is  $(L, L')$ . We can delete the branches that will not be chosen
- The backwards-induction outcome of this game is  $(L, L', L'')$
- The *SPNE* is  $(L, L', L'' L'' R'' L'')$



# Outline

- Introduction
- Extensive Form
- Backwards Induction and SPNE
- Stackelberg
- Introducing Imperfect Info
- **Bank Runs**

# Rules of the Game (1)

- Two investors:  $N = \{\text{Investor 1}, \text{Investor 2}\}$
- Each deposited a sum  $D$  with a bank
- The bank has invested  $2D$  in a **long-term project**
- If the investment is **liquidated before maturity**  $2r$  can be recovered, with

$$D > r > D/2$$

- If the investment **reaches maturity** it pays out  $2R$ , where

$$R > D$$

- Two stages: Date 1 and Date 2 (before and after maturity)
- At each Date, Investors decide to **withdraw** or **not withdraw**. If at least one withdraws, the game ends. We have two **simultaneous games**

## Rules of the Game (2)

- Payoffs at Stage 1:
  - If both withdraw, each gets  $r$  and the game ends
  - If one withdraws and the other does not, the first gets  $D$ , the second  $2r - D$ , and the game ends
  - If no investor withdraws, the game proceeds to Date 2
- Payoffs at Stage 2:
  - If both withdraw each gets  $R$  and the game ends
  - If one withdraws and the other does not, the first gets  $2R - D$ , the second  $D$ , and the game ends
  - If no investor withdraws, each gets  $R$  and the game ends

# Normal-Form Representation

		Investor 2	
		$W$	$NW$
Investor 1	$W$	$(r, r)$	$(D, 2r - D)$
	$NW$	$(2r - D, D)$	<i>Next Stage</i>

Date 1

		Investor 2	
		$W$	$NW$
Investor 1	$W$	$(R, R)$	$(2R - D, D)$
	$NW$	$(D, 2R - D)$	$(R, R)$

Date 2

# Backwards Induction - Second Stage

- We start from Date 2

		Investor 2	
		$W$	$NW$
Investor 1	$W$	$(R, R)$	$(2R - D, D)$
	$NW$	$(D, 2R - D)$	$(R, R)$

- $W$  dominates  $NW$ . To see this, notice that

$$R > D \implies 2R - D > R$$

- Only one  $NE$ ,  $(W, W)$
- We can replace this in the first stage

# Backwards Induction - First Stage

		Investor 2	
		$W$	$NW$
Investor 1	$W$	$(r, r)$	$(D, 2r - D)$
	$NW$	$(2r - D, D)$	$(R, R)$

- Notice that  $r < D \implies 2r - D < r$  and  $R > D$
- If Investor 1 (2) plays  $W$ , BR for player 2 (1) is  $W$
- If Investor 1 (2) plays  $NW$ , BR for player 2 (1) is  $NW$
- There are two *NE* in this game,  $(W, W)$  and  $(NW, NW)$
- First outcome represents a **Bank Run**, i.e. each Investor thinks that the other will play  $W$
- However, socially efficient outcome  $R$  can be achieved