# Dynamic Games of Complete Information

Lorenzo Ferrari

University of Rome Tor Vergata

Academic Year 2021/2022

# Outline

# A Very Famous Dynamic Game

# 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

# 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.*

Dynamic games are represented graphically using the **game tree**.

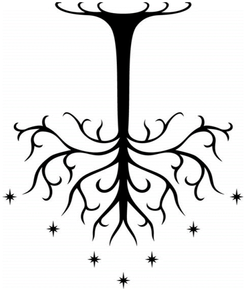
# Building Elements

1. **Players**: $N = \{\}$
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 $T \rightarrow u_i$.

## The Game Tree

- Trees consist of **nodes** connected by **branches**.
- Each branch is an **arrow**, pointing from one node to another.
- (Immediate) **successors** of a node are reached by moving forward.
- (Immediate) **predecessors** of a node are reached by moving backward.



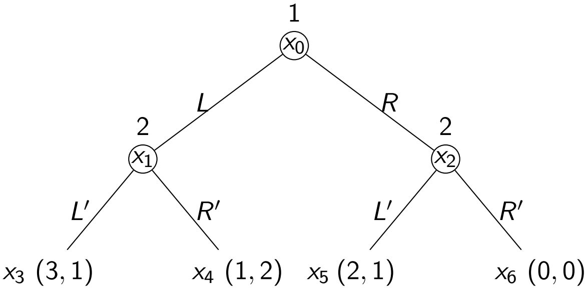
# Subgames

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

a. *begins at a decision node n that is a* **singleton** *information set;*
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

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

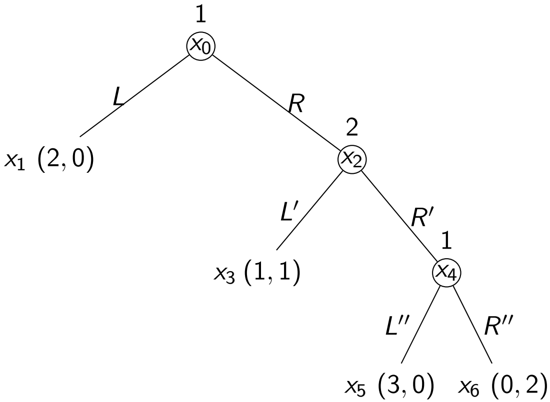
- Notice that it is possible to find the *NE* of this game.
- **Subgames**: 3
  1. The one starting at $x_1$.
  2. The one starting at $x_2$.
  3. The whole game.

# 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 (if the game reaches $x_2$, she would play $L'$).
- We need a **stronger solution concept** than *NE*.

# Another Example




- $N = \{1, 2\}$
- Decision nodes: $X_1 = \{x_0, x_4\}$ and $X_2 = \{x_2\}$
- 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 | $L, L''$ | $(2, 0)$ | $(2, 0)$ |
| | $L, R''$ | $(2, 0)$ | $(2, 0)$ |
| | $R, L''$ | $(1, 1)$ | $(3, 0)$ |
| | $R, R''$ | $(1, 1)$ | $(0, 2)$ |

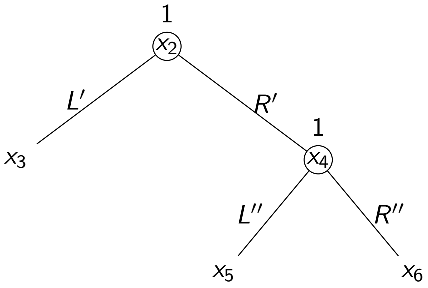
# Another Example (3)

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

- Notice that we have **2** *NE* in this game. However, $((L, R''), L')$ includes 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.

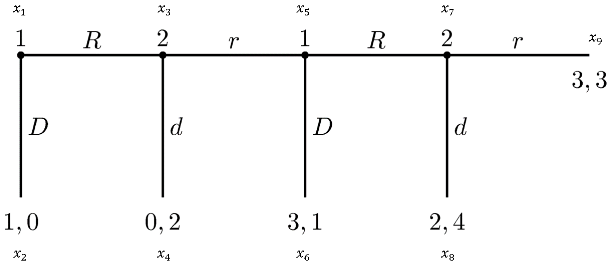


## The Centipede Game (1)

The game is as follows:

- In stage 1, Player 1 has **two piles of coins** in front of her: one contains 1 coin and the other 0 coins.
- Each player has two moves available: either **take** $(D, d)$ the larger pile of coins and give the smaller pile to the other or **push** $(R, r)$ both piles across the table to the other player.
- Each time the piles of coins pass across the table, the quantity of coins in the two piles increases by a certain amount.
- The game is composed of 4 stages and if nobody takes the big pile both players get 3. If either Player takes the big pile the game ends.

# The Centipede Game (2)




- $N = \{1, 2\}$
- Decision nodes: $X_1 = \{x_1, x_5\}$ and $X_2 = \{x_3, x_7\}$
- Info sets: $I_1 = \{\{x_1\}, \{x_5\}\}$ and $I_2 = \{\{x_3\}, \{x_7\}\}$
- Root: $r = \{x_1\}$
- Terminal nodes: $T = \{x_2, x_4, x_6, x_8, x_9\}$
- Strategies: $S_1 = \{(D, D), (D, R), (R, D)(R, R)\}$ and $S_2 = \{(d, d), (d, r), (r, d)(r, r)\}$

# The Centipede Game (3)

- Corresponding normal form:

Player 2

| Player 1 | | $d, d$ | $d, r$ | $r, d$ | $r, r$ |
|---|---|---|---|---|---|
| | $D, D$ | $(1, 0)$ | $(1, 0)$ | $(1, 0)$ | $(1, 0)$ |
| | $D, R$ | $(1, 0)$ | $(1, 0)$ | $(1, 0)$ | $(1, 0)$ |
| | $R, D$ | $(0, 2)$ | $(0, 2)$ | $(3, 1)$ | $(3, 1)$ |
| | $R, R$ | $(0, 2)$ | $(0, 2)$ | $(2, 4)$ | $(3, 3)$ |

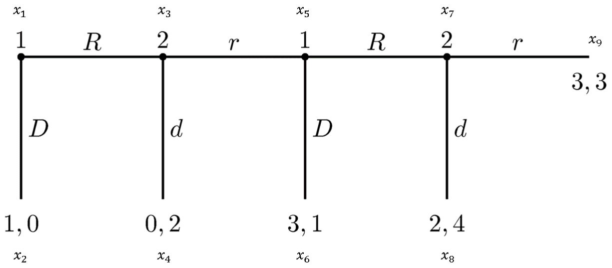
# The Centipede Game (4)

| | | Player 2 | | | |
|---|---|---|---|---|---|
| | | $d, d$ | $d, r$ | $r, d$ | $r, r$ |
| Player 1 | $D, D$ | $(1, 0)$ | $(1, 0)$ | $(1, 0)$ | $(1, 0)$ |
| | $D, R$ | $(1, 0)$ | $(1, 0)$ | $(1, 0)$ | $(1, 0)$ |
| | $R, D$ | $(0, 2)$ | $(0, 2)$ | $(3, 1)$ | $(3, 1)$ |
| | $R, R$ | $(0, 2)$ | $(0, 2)$ | $(2, 4)$ | $(3, 3)$ |

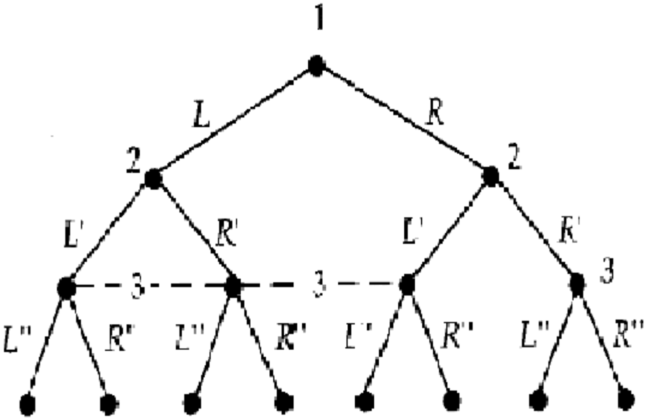
- Notice that we have **4** *NE* in this game. However, $((D, D), (d, r))$, $((D, R), (d, d))$, and $((D, R), (d, r))$ include actions that are not **sequentially rational**.
- Again, we need a **stronger solution concept** than *NE*.

# The Centipede Game (5)

- 4 subgames
  1. The one starting at $x_3$.
  2. The one starting at $x_5$.
  3. The one starting at $x_7$.
  4. The whole game.

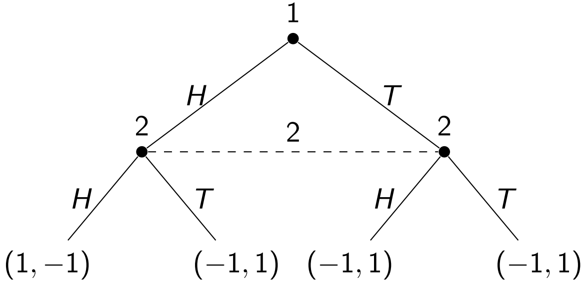


# How Many Subgames?



# 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

# How to Solve Dynamic Games: Theory (1)

- $N = \{\text{Player } 1, \text{Player } 2\}$.
- $a_i \in A_i$ 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 than chooses $a_2$.

- Payoffs are $u_1(a_1, a_2)$ and $u_2(a_1, a_2)$.

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

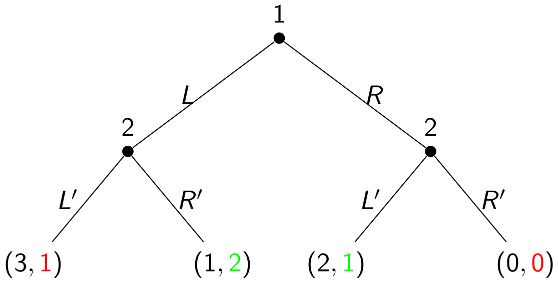
**Definition**: *A Nash equilibrium is* **subgame-perfect** *if the players' strategies constitute a Nash equilibrium in every subgame*.

- 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

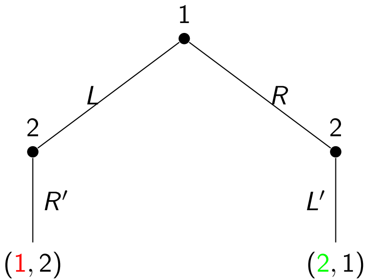



Start with Player 2:

- 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



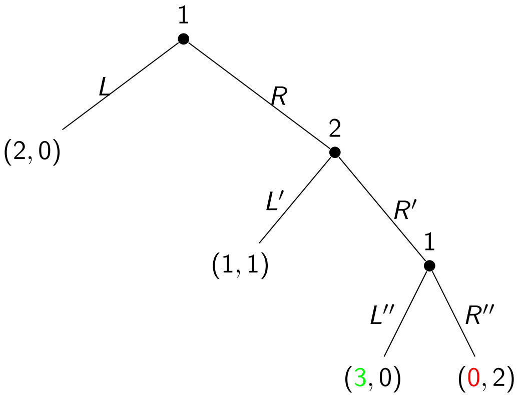

Player 1 knows the reactions of Player 2 to $L$ and $R$:

- 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 $(R, R^{'}L^{'})$.

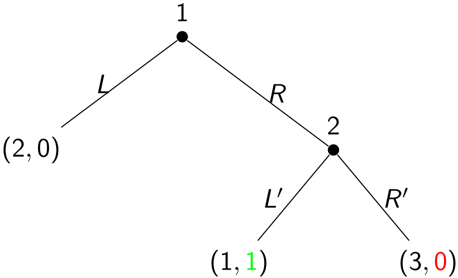
# Another Example - Third Stage




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

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

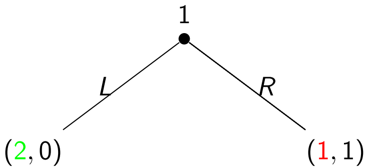
# Another Example - Second Stage




- 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



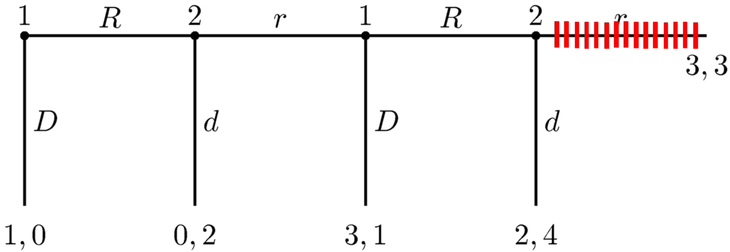

Player 1 knows the reactions of Player 2 to $L$ and $R$:

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

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

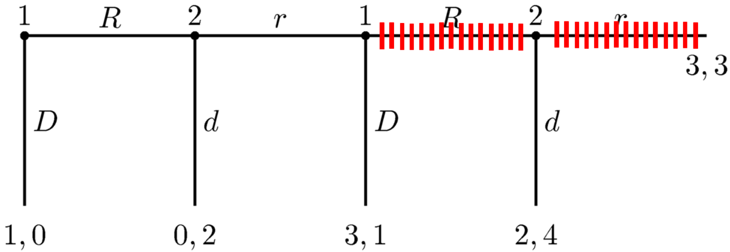
- The *backwards-induction* outcome of this game is $(L)$.
- The *SPNE* is $((L, L''), L')$.

# The Centipede Game: Fourth Stage



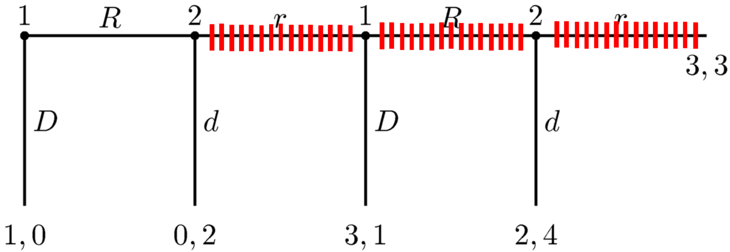

In stage four Player 2 chooses $d$ as $4 > 3$. We can **delete** the branches that will not be chosen.

# The Centipede Game: Third Stage



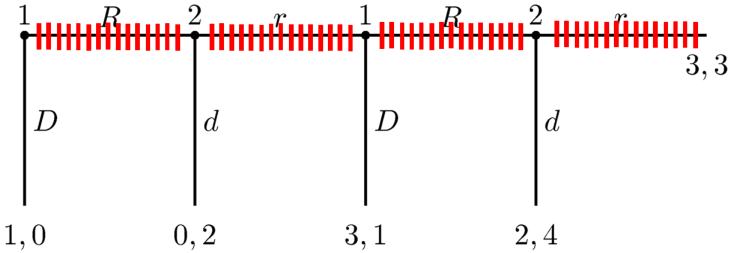

In stage three Player 1 chooses $D$ as $3 > 2$. We can **delete** the branches that will not be chosen.

# The Centipede Game: Second Stage




In stage two Player 2 chooses $d$ as $2 > 0$. We can **delete** the branches that will not be chosen.

# The Centipede Game: First Stage



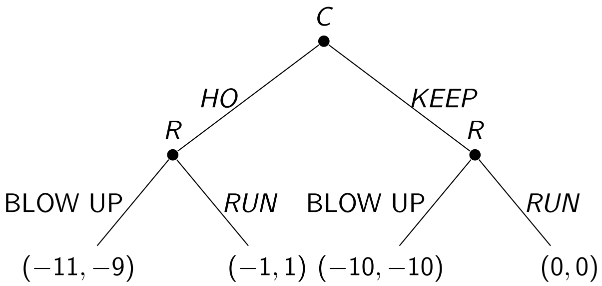

In stage one Player 1 chooses $D$ as $1 > 0$. We can **delete** the branches that will not be chosen.

- The *backwards-induction* outcome of this game is $(D)$.
- The *SPNE* is $((D, D), (d, d))$.

# 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)

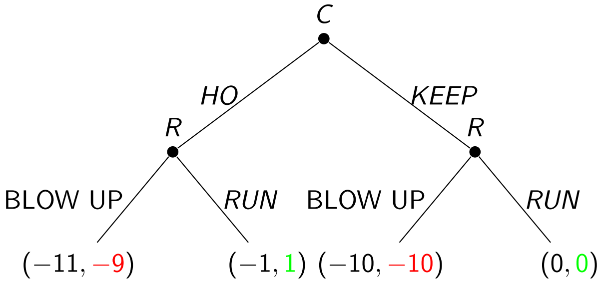
- $N = \{Clerk, Robber\}$
- $X_C = \{x_0\}$ and $X_R = \{x_1, x_2\}$
- $I_C = \{x_0\}$ and $I_R = \{\{x_1\}, \{x_2\}\}$
- $r = \{x_0\}$
- $T = \{x_3, x_4, x_5, x_6\}$
- $S_C = \{HO, KEEP\}$ and
  $S_R = \{B\ B, B\ R, R\ B,\ R\ R\}$

# Bank Robbery - Corresponding Normal Form

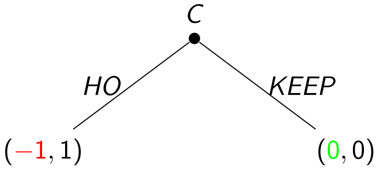
| | | Player 2 | | | |
|---|---|---|---|---|---|
| | | $B\ B$ | $B\ R$ | $R\ B$ | $R\ R$ |
| Player 1 | $HO$ | $(-11,-9)$ | $(-11,-9)$ | $(-1,1)$ | $(-1,1)$ |
| | $KEEP$ | $(-10,-10)$ | $(0,0)$ | $(-10,-10)$ | $(0,0)$ |

- 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

## 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)

$$S_i = S_j = [0, \infty) or$$
$$q_i = q_j = [0, \infty).$$

- 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 in the US during the 60s-70s (Ford and 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

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

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

# Equilibrium Price

- To find equilibrium payoffs (profits), we plug equilibrium quantities in demand:

$$p^s = a - \frac{a-c}{2} - \frac{a-c}{4} = \frac{4a - 2a + 2c - a + c}{4} = \frac{a+3c}{4}.$$

# Equilibrium Profits and First-Mover Advantage

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

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

$$\left[\frac{4a - a + c - 2a + 2c - 4c}{4}\right]\frac{a-c}{2} = \left[\frac{a-c}{4}\right]\frac{a-c}{2} = \frac{(a-c)^2}{8}.$$

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

$$\left[\frac{4a - a + c - 2a + 2c - 4c}{4}\right]\frac{a-c}{4} == \left[\frac{a-c}{4}\right]\frac{a-c}{4} = \frac{(a-c)^2}{16}.$$

- 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

$$Q^s > Q^c.$$

# Stackelberg vs Cournot (3)

- Price in Stackelberg is

$$p^s = \frac{(a+3c)}{4}.$$

- Price in Cournot is

$$p^c = \frac{(a+2c)}{3}.$$

- A higher equilibrium quantity in Stackelberg implies that price is lower in Stackelberg than in Cournot:

$$p^s = \frac{(a+3c)}{4} < \frac{(a+2c)}{3} = p^c.$$

# Extensions

- Effect of including **fixed costs** $F$ (i.e. costs independent from quantity) on $NE$ equilibrium prices, quantities and profits?
- Effect of the two firms facing **different marginal costs** on $NE$ equilibrium prices, quantities and profits?
- Effect of changing the **slope** $b$ of the demand function?

# Outline

# Setting

Players 1 and 2 bargain over **how to share** a dollar. Player 1 and 2's shares are respectively $s_i$ and $1 - s_i$ ($i$ means that Player $i$ makes the offer).

Period 1:

- Stage 1: Player 1 makes a proposal $0 < s_1 < 1$.
- Stage 2: Player 2 accepts or rejects the offer. If 2 accepts the game ends and the split is implemented. Otherwise, it proceeds to stage 3.

Period 2:

- Stage 3, Player 2 makes a proposal $0 < s_2 < 1$.
- Stage 4, Player 1 accepts or rejects the offer. If 1 accepts the game ends and the split is implemented.

Period 3:

- In case of rejection, the game ends and the **exogenous** split $s, 1 - s$ with $0 < s < 1$ is implemented.

Assume players are **impatient**, i.e. they discount payoffs received in later periods by the **factor** $0 < \delta < 1$. Suppose players accept when indifferent.

# Backwards Induction

We solve the game using **backwards induction**.

Period 2: Player 2 knows that Player 1 will accept the offer in period 2 only if $s_2 \geq \delta s$, i.e. what she gets in the next period (discounted). Optimal offer is $s_2^* = \delta s$.

Period 1: Player 1 knows that 2 can obtain $1 - s_2^*$ by rejecting her offer in period 1 (since Player 1 will accept in period 2). To make her indifferent, she must offer $1 - s_1^* = \delta(1 - s_2^*)$, i.e. what Player 2 gets by rejecting in period 1, and would get $s_1^* = 1 - \delta(1 - s_2^*)$.

Player 1 compares $s_1^* = 1 - \delta(1 - s_2^*)$ and what she would get in Period 2, i.e. $\delta s_2^* = \delta^2 s$. Notice that

$$s_1^* = 1 - \delta(1 - s_2^*) = 1 - \delta(1 - \delta s) = \delta^2 s + 1 - \delta > \delta^2 s \text{ since } 1 > \delta.$$

The game thus ends in period 1 with the following share (2 accepts):

$$s_1^* = 1 - \delta(1 - s_2^*) = 1 - \delta(1 - \delta s).$$

# Outline

## 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.

**Note**: A player may also not remember her previous moves. This is called **imperfect recall**.

# Example of Imperfect Recall: Forgetful Driver

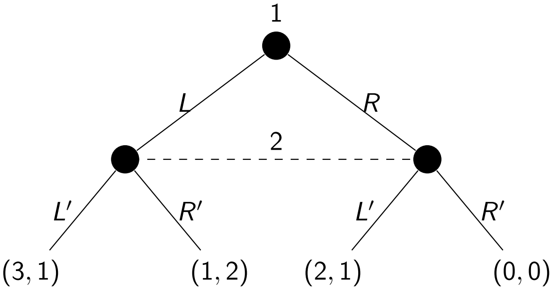
This one-player game demonstrates imperfect recall.

- The player is driving to a friend's house at night.
- The player reaches an **intersection**, where he must turn **left** or **right**. If he turns right, he will find a **police checkpoint**, where he will be delayed for the entire evening. If he turns left, he will eventually reach **another intersection** requiring another right/left decision.
- At this one, a right turn will bring him to his **friend's house**, while a **left turn** will take him to the **police checkpoint**.
- When he has to make the second decision, the player **does not recall** how many intersections he passed through or what decisions he made previously.

# 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)



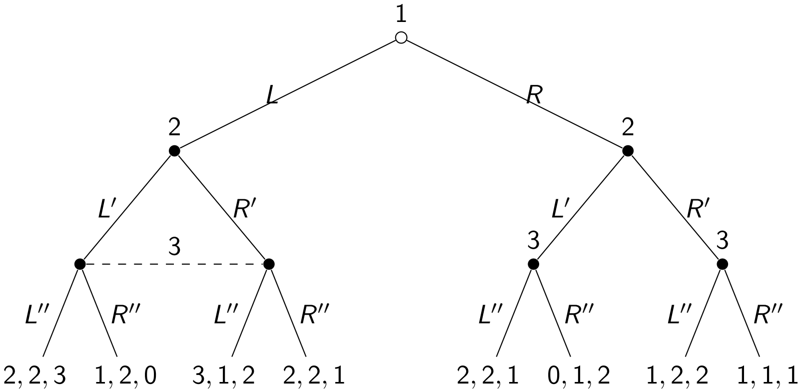

- $N = \{1, 2\}$
- Decision nodes: $X_1 = \{x_0\}$ and $X_2 = \{x_1, x_2\}$
- Info sets: $I_1 = \{x_0\}$ and $I_2 = \{x_1, x_2\}$
- Root: $r = \{x_0\}$
- Terminal nodes: $T = \{x_3, x_4, x_5, x_6\}$
- Strategies: $S_1 = \{L, R\}$ and $S_2 = \{L', R'\}$

# Example (2)

| | | Player 2 | |
|---|---|---|---|
| | | $L^{'}$ | $R^{'}$ |
| Player 1 | $L$ | $(\underline{3}, 1)$ | $(\underline{1}, \underline{2})$ |
| | $R$ | $(2, \underline{1})$ | $(0, 0)$ |

- 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 $x_6$.

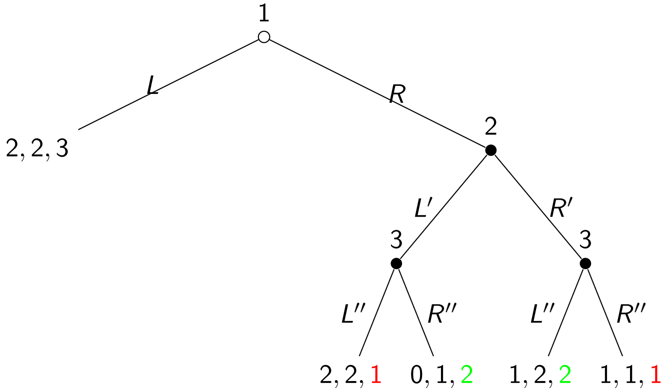
# Example 2 - Third Stage

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

| | | Player 3 | |
|---|---|---|---|
| | | $L''$ | $R''$ |
| Player 2 | $L'$ | $(\underline{2}, \underline{3})$ | $(\underline{2}, 0)$ |
| | $R'$ | $(1, \underline{2})$ | $(\underline{2}, 1)$ |

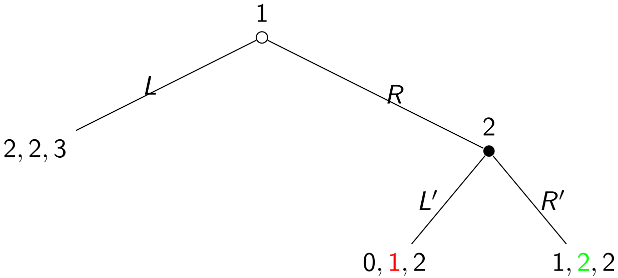
- The only *NE* in this subgame is $(L', L'')$. We can delete the branches that will not be chosen.
- Notice that $L''$ is **strictly dominant** for Player 3.
- $L'$ is **weakly dominant** for Player 2.

# Example 2 - Backwards Induction (1)



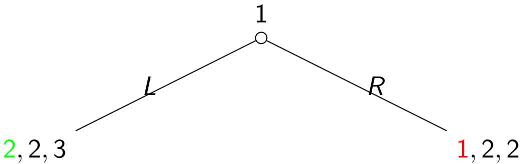

- We can delete the corresponding branches.

# Example 2 - Backwards Induction (2)



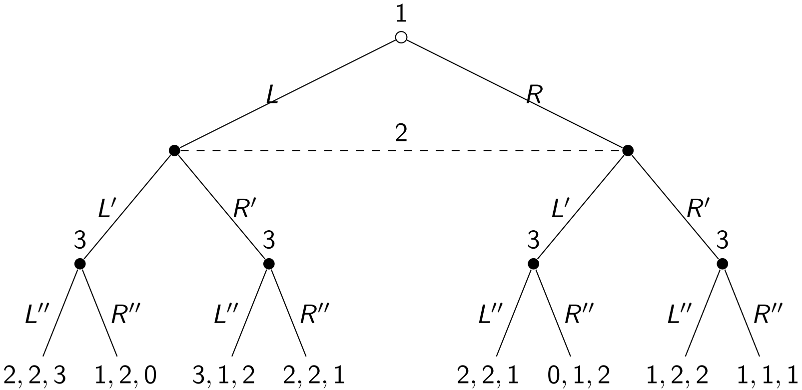

- We can delete the corresponding branches.

# 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)

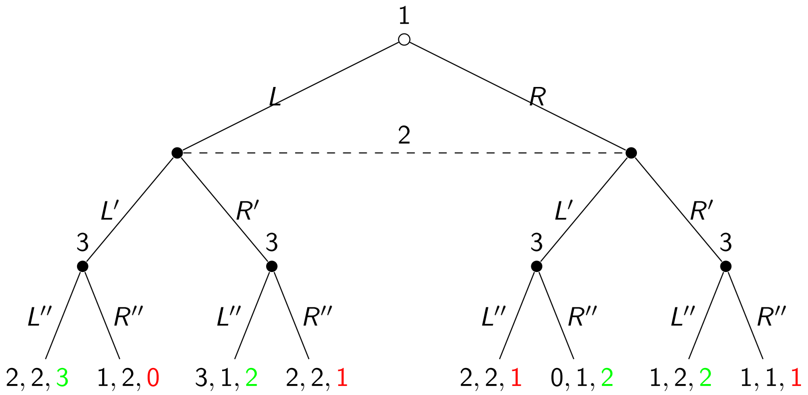



- $N = \{1, 2, 3\}$
- Decision nodes: $X_1 = \{x_0\}$ and $X_2 = \{x_1, x_2\}$ and $X_3 = \{x_3, x_4, x_5, x_6\}$
- Info sets: $I_1 = \{x_0\}$ and $I_2 = \{x_1, x_2\}$ and $I_3 = \{\{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}\}$
- Root: $r = \{x_0\}$
- Terminal nodes: $T = \{x_7, x_8, x_9, x_{10}, x_{11}, x_{12},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

# Example 3 - Strategies and Subgames

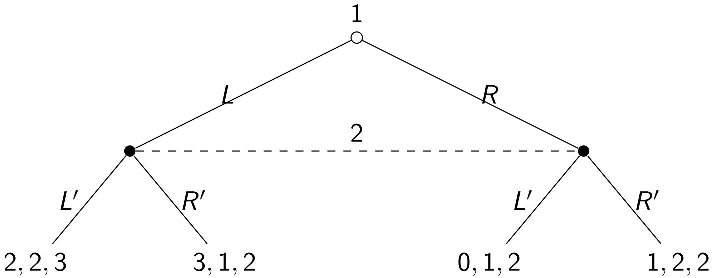
- $S_1 = \{L, R\}$
- $S_2 = \{L', R'\}$
- $S_3 = \{L''L''L''L'', R''L''L''L'', R''R''L''L'', R''R''R''L'', R''R''R''R''$
  $L''R''R''R'', L''L''R''R'', L''L''L''R'', R''R''R''L'', ...\}$
- Player 3 has many strategies.
- There are five subgames:
  1. The whole game.
  2. The game starting at $x_3$.
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

| Player 1 | | $L'$ | $R'$ |
|---|---|---|---|
| | $L$ | $(\underline{2}, \underline{2})$ | $(\underline{3}, 1)$ |
| | $R$ | $(0, 1)$ | $(1, \underline{2})$ |

- 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

# Rules of the Game (1)

- Two investors: $N = \{Investor\ 1, Investor\ 2\}$.
- 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

| Investor 1 | | $W$ | $NW$ |
|---|---|---|---|
| | $W$ | $(r, r)$ | $(D, 2r - D)$ |
| | $NW$ | $(2r - D, D)$ | *Next Stage* |

Date 1

Investor 2

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

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.

# Outline

# Rules of the Game (1)

- Two identical countries: $N = \{Country\ 1, Country\ 2\}$.

- Each country has:
  1. A government that chooses a **tariff rate** $t_1, t_2$.
  2. A **firm** producing output for both **home consumption** $(h_i)$ and **export** $(e_i)$.
  3. **Consumers** who buy either from the domestic or foreign firm.

# Rules of the Game (2)

- **Inverse demand** in country $i$ is given by home production plus imports from country $j$:

$$P_i(Q_i) = a - Q_i, \text{ with } Q_i = h_i + e_j.$$

- Firms face a **constant marginal** cost $c$, i.e.

$$C_i(h_i, e_i) = c(h_i + e_i).$$

- If country $j$ has a tariff $t_j$ in place, firm $i$ must also pay $t_j e_i$.

The game proceeds as follows:

- **Stage 1**: countries **simultaneously** choose tariffs $t_i$ and $t_j$.
  **Stage 2**: Firms observe the choice by the government, and simultaneously choose $(h_1, e_1)$ and $(h_2, e_2)$.

# Rules of the Game (3)

**Payoffs** are:

- **Profits** to firm $i$ (notice that they sell in both markets):

$$\pi_i(t_i, t_j, h_i, e_i, h_j, e_j) =$$
$$= [a - (h_i + e_j)]h_i + [a - (h_j + e_i)]e_i - c(h_i + e_i) - t_j e_i.$$

- **Total welfare** to government $i$ (consumer surplus plus profit plus tariff revenue):

$$W_i(t_i, t_j, h_i, e_i, h_j, e_j) = \frac{1}{2}Q_i^2 + \pi_i(t_i, t_j, h_i, e_i, h_j, e_j) + t_i e_j.$$

- Notice that with linear demand consumer surplus is given by $\frac{1}{2}(a - p_i)(Q_i) = \frac{1}{2}Q_i^2$.

# Backwards Induction - Second Stage (1)

We start from Stage 2, i.e. governments have already chosen tariffs $t_1$ and $t_2$. Firm $i$ chooses $h_i$ and $e_i$ (given $h_j$ and $e_j$)as to **maximise profit** in the two markets:

$$\max_{h_i, e_i \geq 0} \pi_i(t_i, t_j, h_i, e_i, h_j^*, e_j^*)$$

We can solve the maximisation problem **by market**:

$$\max_{h_i \geq 0} h_i[a - (h_i + e_j^*) - c] \text{ and } \max_{e_i \geq 0} e_i[a - (h_j^* + e_i) - c] - t_j e_i.$$

# Backwards Induction - Second Stage (2)

Take firm $i$'s **first-order condition** in both markets:

$$a - 2h_i^* - e_j^* - c = 0 \implies h_i^* = \frac{a - c - e_j^*}{2} \text{ for } a - c \geq e_j^*,$$

$$a - h_j^* - 2e_i^* - c - t_j \implies e_i^* = \frac{a - c - t_j - h_j^*}{2} \text{ for } a - c - t_j \geq h_j^*.$$

The same holds for Firm $j$:

$$h_j^* = \frac{a - c - e_i^*}{2} \text{ for } a - c \geq e_i^*,$$

$$e_j^* = \frac{a - c - t_i - h_i^*}{2} \text{ for } a - c - t_i \geq h_i^*.$$

# Backwards Induction - Second Stage (3)

We have 4 equations in 4 unknowns and we can thus solve for $(h_i^*, h_j^*, e_i^*, e_j^*)$ for every value of $t_i, t_j$:

- Replace the equation of $h_j^*$ in the one for $e_i^*$ and solve for $e_i^*$:

$$e_i^* = \frac{1}{2}\left(a - c - t_j - h_j^*\right) = \frac{1}{2}\left(a - c - t_j - \frac{1}{2}\left(a - c - e_i^*\right)\right) \implies$$

$$4e_i^* = 2a - 2c - 2t_j - a + c + e_i^* \implies 3e_i^* = a - c - 2t_j \implies$$

$$e_i^* = \frac{a - c - 2t_j}{3} \text{ and } e_j^* = \frac{a - c - 2t_i}{3}.$$

- **Interpretation**: country $i$'s ($j$'s) exports are a **negative function** of country $j$'s ($i$'s) tariff, i.e. $\frac{\Delta e_i^*}{\Delta t_j} = \frac{\Delta e_j^*}{\Delta t_i} = -\frac{2}{3}$.

# Backwards Induction - Second Stage (4)

- Replace the equation of $e_j^*$ in the one for $h_i^*$ and solve for $h_i^*$:

$$h_i^* = \frac{1}{2}\left(a - c - e_j^*\right) = \frac{1}{2}\left(a - c - \left(\frac{a - c - 2t_i}{3}\right)\right) \implies$$

$$6h_i^* = 3a - 3c - a + c + 2t_i \implies 6h_i^* = 2a - 2c + 2t_i \implies$$

$$h_i^* = \frac{a - c + t_i}{3} \text{ and } h_j^* = \frac{a - c + t_j}{3}.$$

- **Interpretation**: country $i$'s ($j$'s) domestic sales are a **positive function** of country $i$'s ($j$'s) tariff, i.e. $\frac{\Delta h_i^*}{\Delta t_i} = \frac{\Delta h_j^*}{\Delta t_j} = \frac{1}{3}$.

# Backwards Induction - First Stage (1)

Governments of countries $i$ and $j$ **set tariff rates** as to maximise their **total welfare** (write them as a function of $t_i$ and $t_j^*$ only):

$$\max_{t_i \geq 0} W_i^*(t_i, t_j^*) = \frac{1}{2} Q_i^{*2} + \pi_i(t_i, t_j^*, h_i^*, e_i^*, h_j^*, e_j^*) + t_i e_j^*.$$

We need to identify all the components of welfare **as a function** of $t_i$ and $t_j^*$.

# Backwards Induction - First Stage (2)

$$W_i^*(t_i, t_j^*) = \frac{1}{2} Q_i^{*2} + \pi_i(t_i, t_j^*, h_i^*, e_i^*, h_j^*, e_j^*) + t_i e_j^*.$$

Use $h_i^*$ and $e_j^*$ to find $Q_i^*$:

$$Q_i^* = h_i^* + e_j^* = \frac{a-c+t_i}{3} + \frac{a-c-2t_i}{3} = \frac{a-c+t_i+a-c-2t_i}{3} =$$

$$= \frac{2a-2c-t_i}{3} = \frac{2(a-c)-t_i}{3}.$$

This implies:

$$\frac{1}{2} Q_i^{*2} = \frac{1}{2} \frac{\left[2(a-c)-t_i\right]^2}{9} = \frac{\left[2(a-c)-t_i\right]^2}{18}.$$

# Backwards Induction - First Stage (3)

$$W_i^*(t_i, t_j^*) = \frac{1}{2}Q_i^{*2} + \pi_i(t_i, t_j^*, h_i^*, e_i^*, h_j^*, e_j^*) + t_i e_j^*.$$

Use $h_i^*$ and $e_j^*$ to find $\pi_i$:

- In domestic market:

$$h_i[a - (h_i + e_j^*) - c] = \left[\frac{a-c+t_i}{3}\right]\left[a - \left(\frac{a-c+t_i}{3}\right) - \left(\frac{a-c-2t_i}{3}\right) - c\right] =$$

$$= \left[\frac{a-c+t_i}{3}\right]\left[\frac{3a-a+c-t_i-a+c+2t_i-3c}{3}\right] = \frac{[a-c+t_i]^2}{9}.$$

- In foreign market:

$$e_i[a-(h_j^*+e_i)-c]-t_j^* e_i = \left[\frac{a-c-2t_j^*}{3}\right]\left[a-\left(\frac{a-c+t_j^*}{3}\right)-\left(\frac{a-c-2t_j^*}{3}\right)-c\right]-t_j^*\left(\frac{a-c-2t_j^*}{3}\right)$$

$$\left[\frac{a-c-2t_j^*}{3}\right]\left[\frac{3a-a+c-t_j^*-a+c+2t_j^*-3c}{3}\right] - t_j\left(\frac{a-c-2t_j^*}{3}\right) =$$

$$\left[\frac{a-c-2t_j^*}{3}\right]\left[\frac{a-c+t_j^*}{3}\right] - t_j^*\left(\frac{a-c-2t_j^*}{3}\right) =$$

$$\left[\frac{a-c-2t_j^*}{3}\right]\left[\frac{a-c+t_j^*}{3} - t_j^*\right] = \left[\frac{a-c-2t_j^*}{3}\right]\left[\frac{a-c-2t_j^*}{3}\right] = \frac{[a-c-2t_j^*]^2}{9}.$$

# Backwards Induction - First Stage (4)

Put everything together:

$$\max_{t_i \geq 0} W_i^*(t_i, t_j^*) = \frac{[2(a-c)-t_i]^2}{18} + \frac{[a-c+t_i]^2}{9} + \frac{[a-c-2t_j^*]^2}{9} + \frac{t_i(a-c-2t_i)}{3}.$$

Take the **FOC** with respect to $t_i$ (use chain rule for derivatives, i.e. $D[f(g(x))] = f^{'}(g(x)) \cdot g^{'}(x)$ and product rule $[g(x)f(x)]^{'} = f^{'}(x)g(x) + f(x) + g^{'}(x)$):

$$\frac{\delta W_i^*(t_i, t_j^*)}{\delta t_i} = 0 \implies \frac{2(-1)(2a-2c-t_i^*)}{18} + \frac{2(a-c+t_i^*)}{9} + \frac{(a-c-2t_i^*)+(-2t_i^*)}{3} =$$

$$= \frac{2t_i^* - 4a + 4c}{18} + \frac{2a - 2c + 2t_i^*}{9} + \frac{a-c-4t_i^*}{3} = 0 \implies$$

$$\implies \frac{2t_i^* - 4a + 4c + 4t_i^* + 4a - 4c + 6a - 6c - 24t_i^*}{18} = 0 \implies$$

$$\implies 6a - 6c - 18t_i^* = 0 \implies t_i^* = \frac{a-c}{3} = t_j^*.$$

Notice that $i$'s optimal tariff rate is **independent from** $j$'s (and vice-versa).

# Equilibrium Quantities and Welfare

Replace $t_i^*$ and $t_j^*$ to find equilibrium quantities:

$$h_i^* = \frac{a-c+t_i^*}{3} = \frac{a-c}{3} + \frac{a-c}{9} = \frac{3a-3c+a-c}{9} = \frac{4(a-c)}{9} = h_j^*.$$

$$e_i^* = \frac{a-c-2t_j^*}{3} = \frac{a-c}{3} - \frac{2(a-c)}{9} = \frac{3a-3c-2a+2c}{9} = \frac{a-c}{9} = e_j^*.$$

- Quantity in each market is $Q_i^*(t_i^*, t_j^*) = \dfrac{5(a-c)}{9}$.
- If tariff is set to zero, i.e. $t_i = t_j = 0$, same quantity as in Cournot $Q_i^*(0,0) = \dfrac{2(a-c)}{3}$.
- It can be shown that **aggregate total welfare**:

$$\max_{t_i, t_j \geq 0} W_i^*(t_i, t_j) + W_j^*(t_i, t_j),$$

is maximum when $t_i = t_j = 0$. The **SPNE** of this game is (Pareto) **inefficient**.