

# Subgame Perfect Equilibrium

Felix Munoz-Garcia

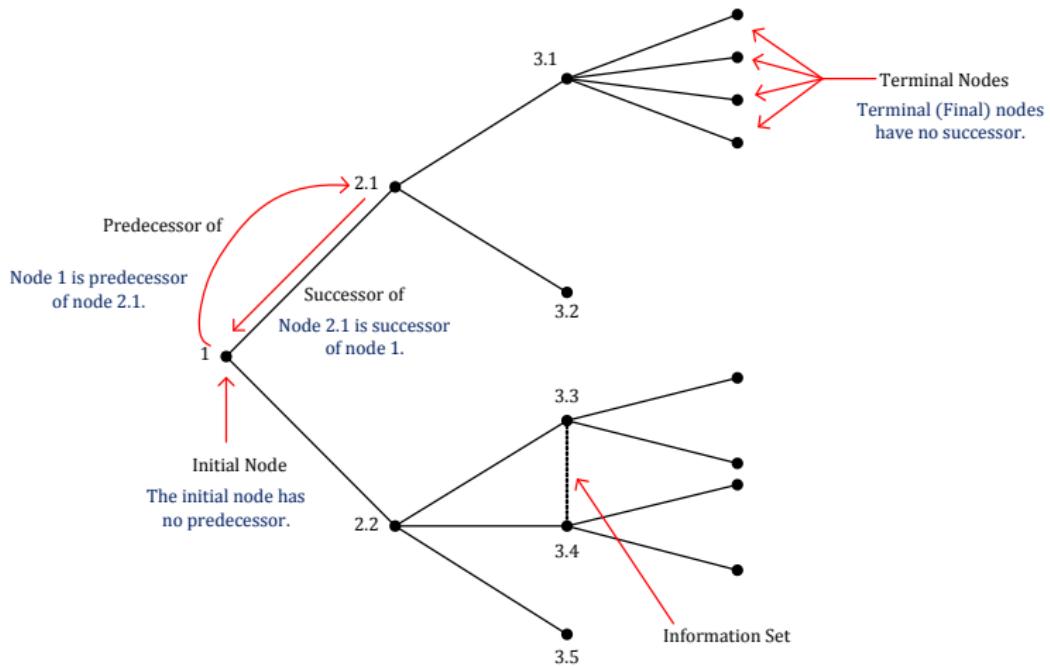
*Strategy and Game Theory* - Washington State University

# Sequential Move Games

- Road Map:
  - Rules that game trees must satisfy.
  - How to incorporate sequential rationality in our solution concepts in order to discard strategy profiles that are not credible.
  - Backward induction and Subgame Perfect Equilibrium.
  - Applications.
- References:
  - Watson, Ch. 14-16.
  - Harrington, Ch. 8-9.
  - Osborne, Ch. 5-6.

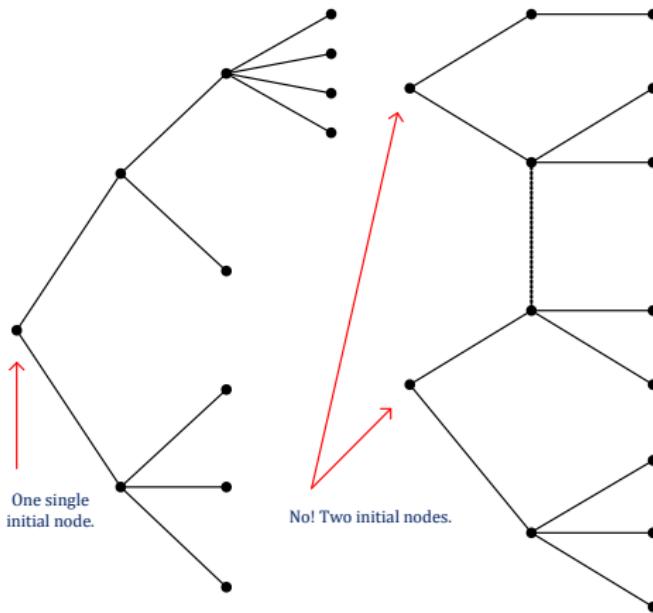
# Sequential Move Games

- Trees

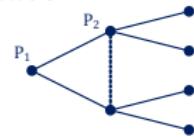


# Tree Rules

- 1) Every node is the successor of the initial node.



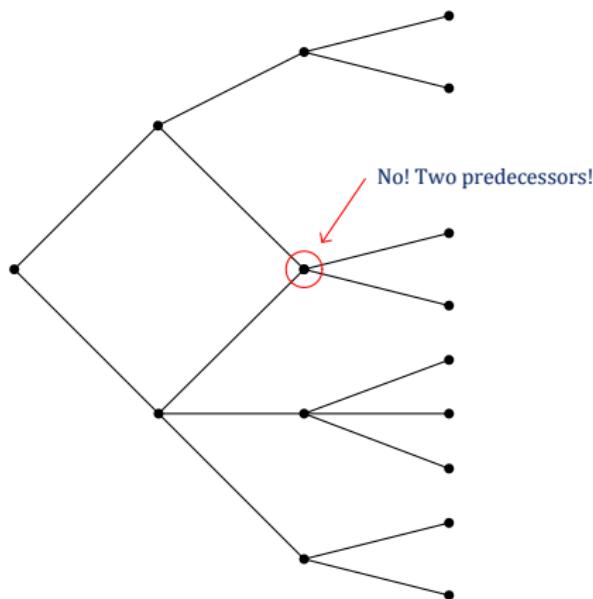
If they refer to the same player: two selves?  
If they refer to two different players acting simultaneously: We have a way to represent such situations!



# Tree Rules

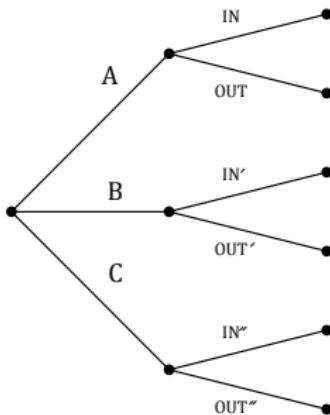
2) Every node, except the initial node, has exactly one immediate predecessor.

- The initial node has no predecessor.

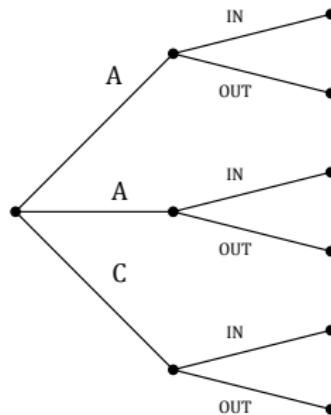


# Tree Rules

3) Multiple branches, extending from the same node, have different action labels.



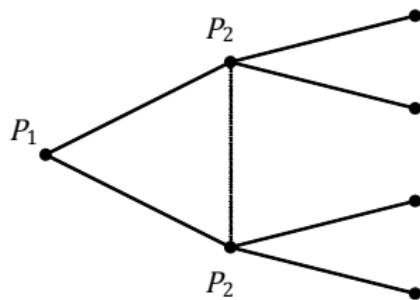
Correct



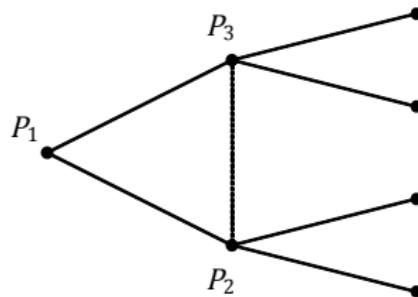
No, you must be referring to a different action.  
Otherwise collapse everything under the same name.

# Tree Rules

4) Each information set contains decision nodes for only one of the players.



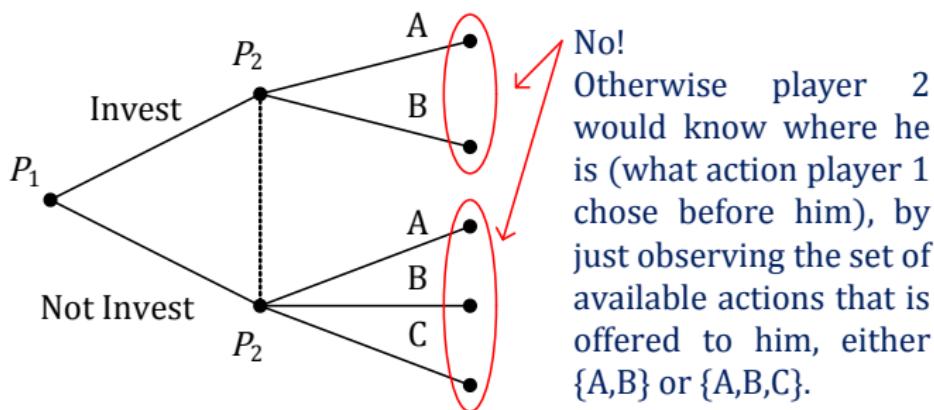
Correct



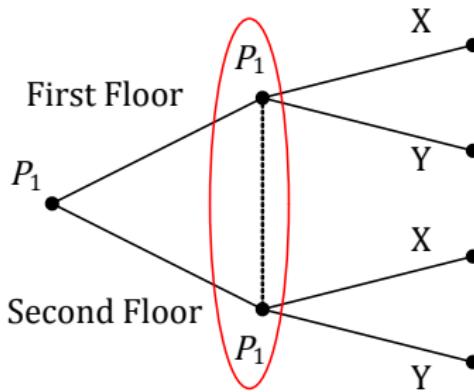
Incorrect otherwise  $P_2$  knows he is called on to move after B.  
(no uncertainty)

## Tree Rules

5) All nodes in a given information set have the same number of immediate successors, and they must have the same number of action labels leading to these successors.



# Perfect vs. Imperfect Recall

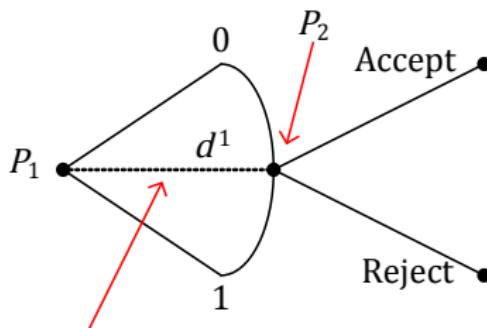


This is imperfect recall: Where did I park my car, in the first or the second floor?

- Is it realistic to assume **perfect recall**? Yes, if stakes are high.

# Continuum of Actions

- What if we want to represent that one player can choose among a continuum of actions?
  - Draw infinitely many branches? No!

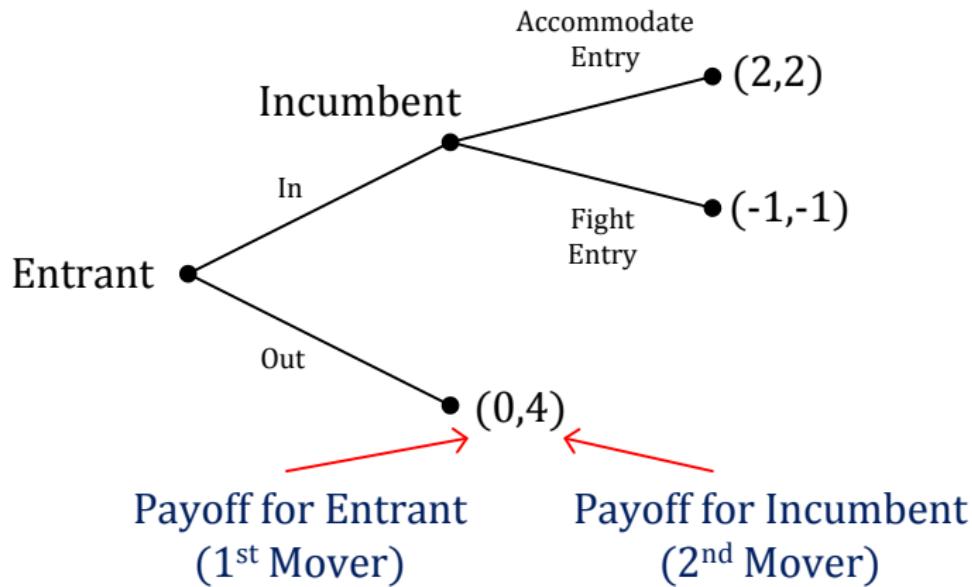


Sometimes we add this line, where  $d_1$  is the division of the pie that player 1 chooses between 0 (0%) and 1 (100%).

# Introducing a new solution concept

- Why do we need a new solution concept?
  - Because when we apply NE to sequential-move games, some NE predictions do not seem sensible (or credible).
- Let us see one example of this: "Entry and Predation"

# Entry and Predation



## Entry and Predation

- Normal form representation of the game:

		Incumbent	
		Accom.	Fight
In		<u>2</u> , <u>2</u>	-1, -1
Entrant	Out	0, 4	0, 4

- There are two psNE for this game: (In, Accommodate) and (Out, Fight).

# Entry and Predation

- ➊ The strategy profile (Out, Fight) is indeed a NE of the game since:
  - ➊  $BR_{Entrant}(Fight) = Out$
  - ➋  $BR_{Incumbent}(Out) = \{Fight, Accom\}$
- ➋ But is this equilibrium credible?
  - ➊ No! The entrant's beliefs about the incumbent's decision to Fight after he enters are not rational (in a sequential way): once the entrant is in, the best thing that the incumbent can do is to Accommodate.
  - ➋ Then, only Accommodate is sequentially rational, and (Out,Fight) is not sequentially rational.
  - ➌ The NE (In, Accom) satisfies sequential rationality.
- ➌ But, how can we define Sequential rationality more formally?

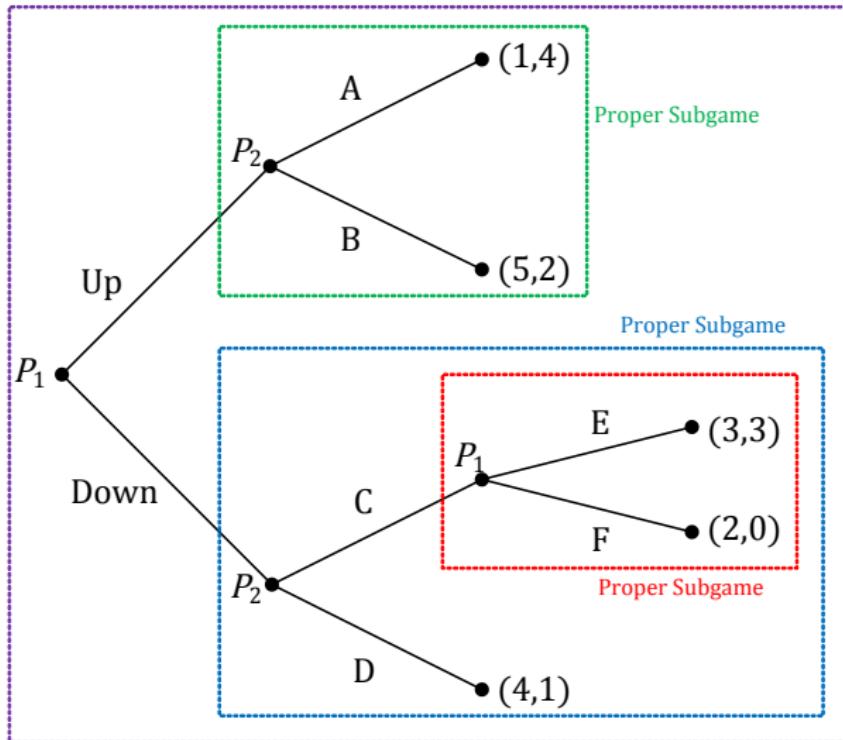
# Sequential Rationality

- Player  $i$ 's strategy is *sequentially rational* if it specifies an optimal action for player  $i$  at any node (or information set) of the game where he is called on to move, even those information sets that player  $i$  does not believe (ex-ante) that will be reached in the game.
- How to satisfy this long definition when solving games?
  - Using Backward Induction:
    - starting from every terminal node, every player uses optimal actions at every subgame of the game tree.
    - Before we describe Backward Induction we must define what we mean by subgames.

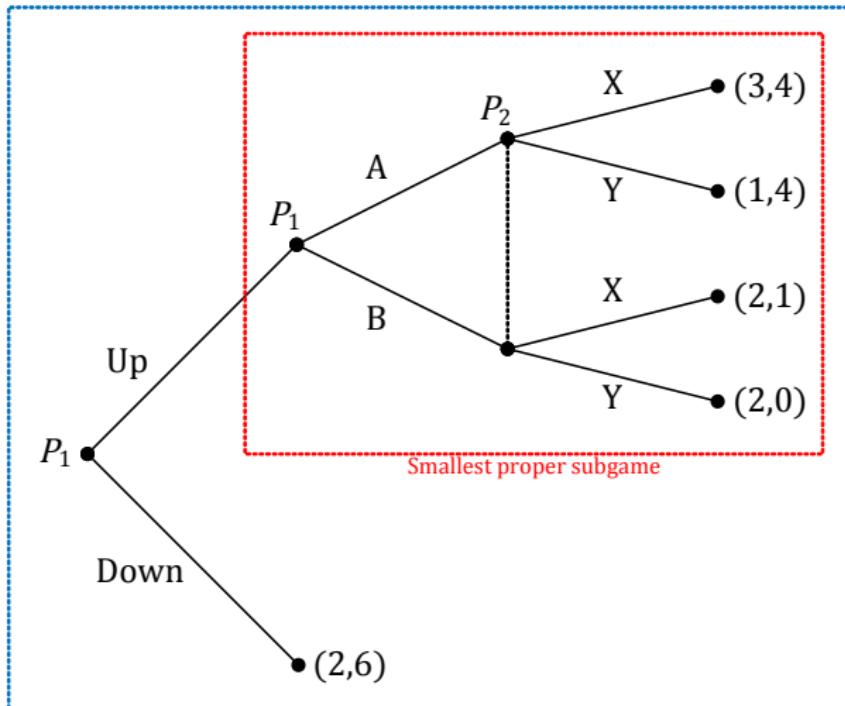
# Sequential Rationality

- **Subgame:** Given an extensive form game, a node  $x$  is said to *initiate a subgame* if neither  $x$  nor any of its successors are in an information set that contains nodes that are not successors of  $x$ .
- Hence, a subgame is a tree structure defined by such a node  $x$  and its successors. Graphically, a subgame can be identified by drawing a circle around a section of the game tree without "breaking" any information set.
  - Graphical representation. →

# Sequential Rationality - Examples

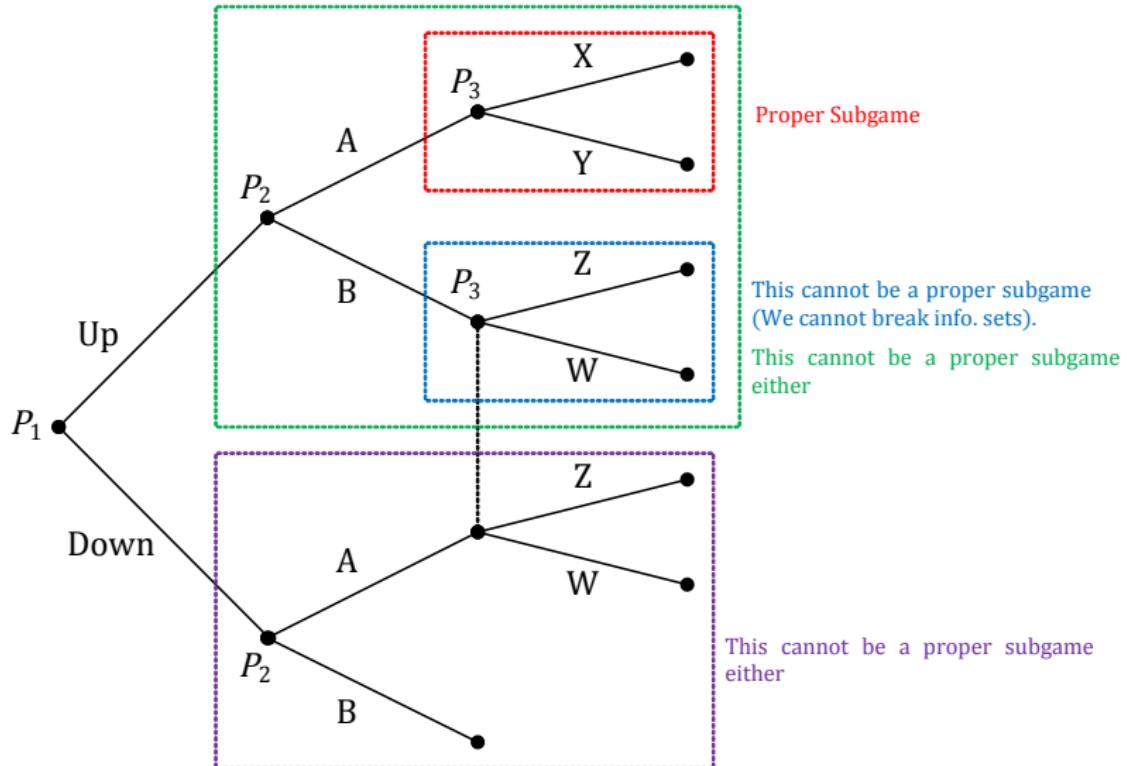


# Sequential Rationality - Examples



The game as a whole is the second smallest subgame.

# Sequential Rationality - Examples



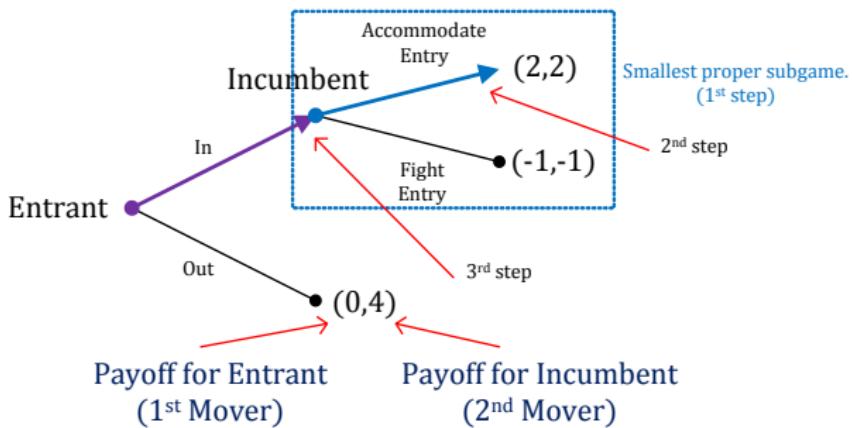
# Sequential Rationality

- After describing what is a proper subgame and what is not, we are ready to solve sequential-move games.
- How can we guarantee that our solution for these games embodies the notion of "sequential rationality"?
  - By using the so-called "backward induction."
  - In particular, we find the strategy that every player  $i$  finds optimal when he is called to move **at every proper subgame** along the game tree.



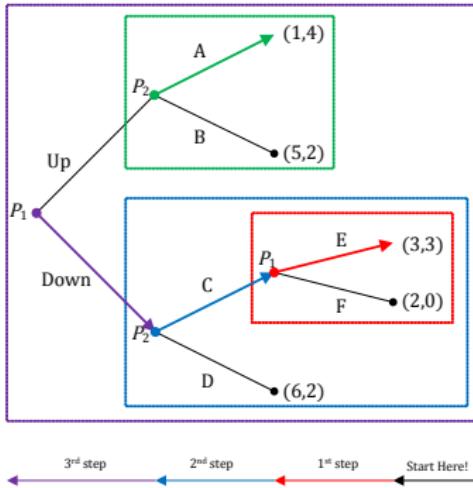
- Once we are done applying backwards induction, we can claim that:
- Strategy profile  $(s_1^*, s_2^*, \dots, s_N^*)$  is a Subgame Perfect Nash Equilibrium (SPNE) of the game since it specifies a NE for each proper subgames of the game.
- Let's do a few examples together. →

# Using Backward Induction - Entry and Predation Game



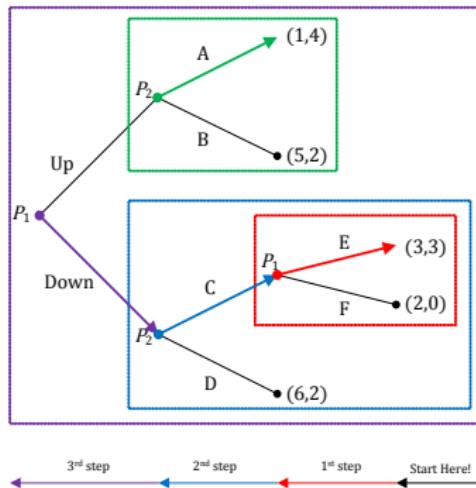
- Hence, there is only one Subgame Perfect Equilibrium in this game: (In, Accommodate)
- Among the two psNE we found, i.e., (In, Accommodate) and (Out, Fight), only the first equilibrium is sequentially rational.

# Backward Induction



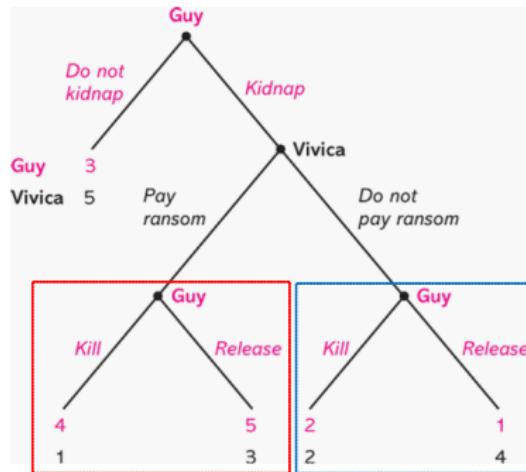
- **1st step:** What is optimal for player 1 in the last subgame?
- **2nd step:** Given the outcome of the 1st step, what is optimal for player 2?
- **3rd step:** Given the outcome of the 2nd step, what is optimal for player 1?

# Backward Induction



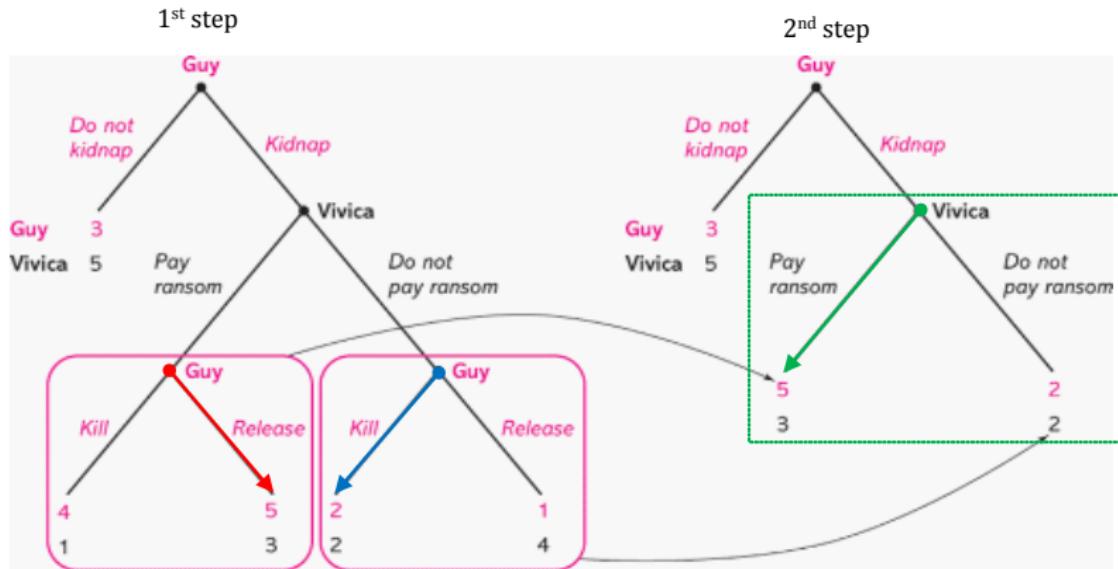
- Hence the SPNE of this game is  $\{(Down, E), (A, C)\}$  where the first parenthesis applies to P1 and the second to P2.

# Kidnapping Game (Harrington)

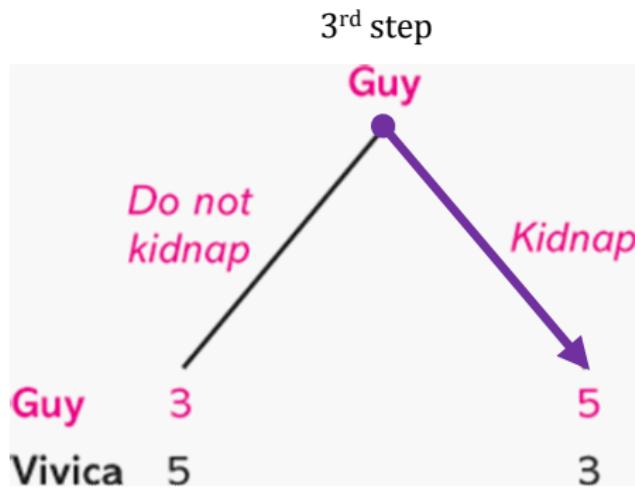


- After identifying the smallest proper subgames, let's find optimal strategies for player "Guy" in these subgames. →

# Kidnapping Game

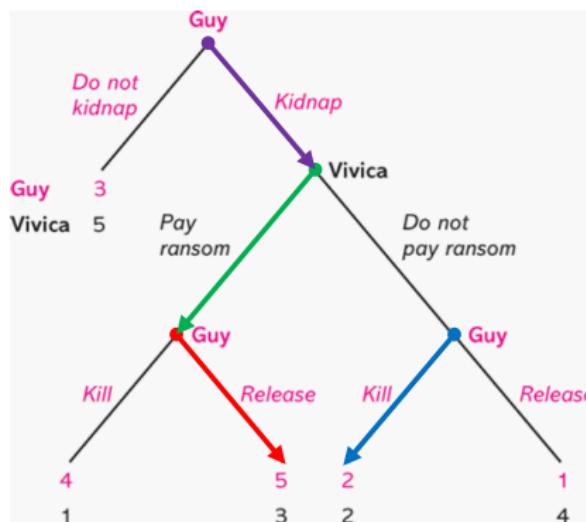


# Kidnapping Game



# Kidnapping Game

- Alternatively, you can find spNE without having to redraw the reduced versions of the game tree, as we do below for the same example:



- One spNE:  $\{(Kidnap, \text{Release after Pay, Kill after no Pay}), \text{Pay Ransom}\}$

# Kidnapping Game

- We found a unique SPNE by applying backward induction.
- But, how many NEs are in this game?
- In order to find that, we need to first represent this game in its normal form.
  - For that, we first need to know how many strategies player 1 has (rows in the matrix) and how many strategies player 2 has (columns in the matrix).
  - $S_2 = \{Pay, Don'tPay\} \longrightarrow 2$  columns in the following matrix
  - $S_1$  must take into account all combinations of player 1's actions  $2 \times 2 \times 2 \longrightarrow 8$  rows in the following matrix.

# Kidnapping Game

- 5 psNE!

		Vivica (kin of victim)	
		Pay ransom	Do not pay ransom
Guy (kidnapper)	Do not kidnap/Kill/Kill	3, 5	3, 5
	Do not kidnap/Kill/Release	3, 5	3, 5
	Do not kidnap/Release/Kill	3, 5	3, 5
	Do not kidnap/Release/Release	3, 5	3, 5
	Kidnap/Kill/Kill	4, 1	2, 2
	Kidnap/Kill/Release	4, 1	1, 4
	Kidnap/Release/Kill	5, 3	2, 2
	Kidnap/Release/Release	5, 3	1, 4

- However, all of the NE that involve Do not kidnap (Highlighted yellow) are not sequentially rational.
- Only the SPNE is sequentially rational (Highlighted green).
  - We found it by applying backward induction in the game tree a few slides ago.

# Kidnapping Game

- Just for curiosity, which strategy profiles survive the application of IDSDS?
  - For Guy (row player), Kidnap/Kill/Kill and Kidnap/Kill/Release are strictly dominated by a mixed strategy.
  - In particular, we can construct a mixed strategy between Do not Kidnap/Kill/Kill (with probability  $\frac{2}{3}$ ) and Kidnap/Release/Kill (with probability  $\frac{1}{3}$ ) that yields an expected utility of 4.3 for Guy when Vivica pays the ransom (left column) and 2.6 when Vivica does not pay the ransom (right column).
  - See next slide.

# Kidnapping Game

		Vivica (kin of victim)	
		Pay ransom	Do not pay ransom
Guy (kidnapper)	Do not kidnap/Kill/Kill	(3) 5	(3) 5
	Do not kidnap/Kill/Release	3, 5	3, 5
	Do not kidnap/Release/Kill	3, 5	3, 5
	Do not kidnap/Release/Release	3, 5	3, 5
	Kidnap/Kill/Kill	4, 1	2, 2
	Kidnap/Kill/Release	4, 1	1, 4
	Kidnap/Release/Kill	(5) 3	(2) 2
	Kidnap/Release/Release	5, 3	1, 4

Prob.  $\frac{2}{3} \rightarrow$

Guy (kidnapper)

Prob.  $\frac{1}{3} \rightarrow$

$$EU = \frac{2}{3} * 3 + \frac{1}{3} * 5 = 4.3 \quad EU = \frac{2}{3} * 3 + \frac{1}{3} * 2 = 2.6$$

# Kidnapping Game

- Once we have deleted the rows corresponding to Kidnap/Kill/Kill and Kidnap/Kill/Release...
- We move to Vivica, and we cannot find any strictly dominated strategy for her.

		Vivica (kin of victim)	
		Pay ransom	Do not pay ransom
Guy (kidnapper)	Do not kidnap/Kill/Kill	3, 5	3, 5
	Do not kidnap/Kill/Release	3, 5	3, 5
	Do not kidnap/Release/Kill	3, 5	3, 5
	Do not kidnap/Release/Release	3, 5	3, 5
	Kidnap/Release/Kill	5, 3	2, 2
	Kidnap/Release/Release	5, 3	1, 4

- Hence, the 12 remaining cells are the 12 strategy profiles that survive IDSDS.

# Kidnapping Game

- We are getting more precise in our predictions!

IDSDS, e.g., 12 in the previous example

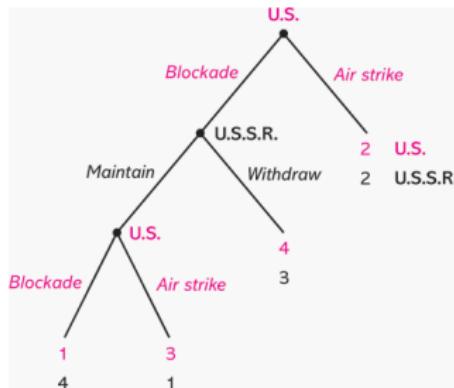
NE, e.g., 5 in the previous example

SPNE, e.g., 1 in the previous example

## Another Example: The Cuban Missile Crisis



## Another Example: The Cuban Missile Crisis

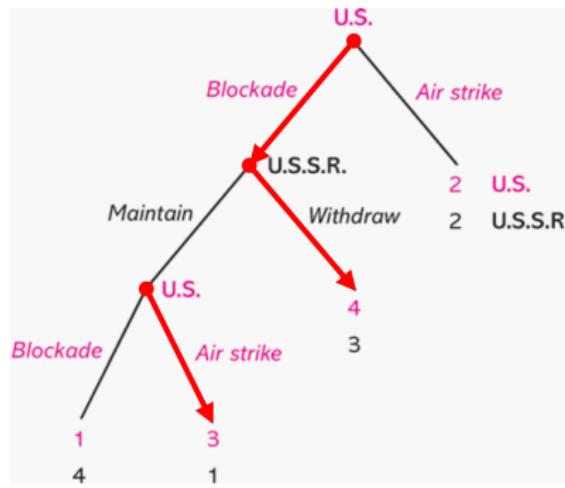


### Assumptions :

- The US prefers that the USSR withdraw the missiles without an air strike (i.e.,  $4 > 2$ ).
- The USSR prefers to maintain the missiles if no air strike ensues (i.e.,  $4 > 3$ ), but prefers to withdraw them if maintaining the missiles triggers an air strike (i.e.,  $3 > 1$ )
- If the missiles are maintained, however, the US prefers to launch an air strike (i.e.,  $3 > 1$ ).

## Another Example: The Cuban Missile Crisis

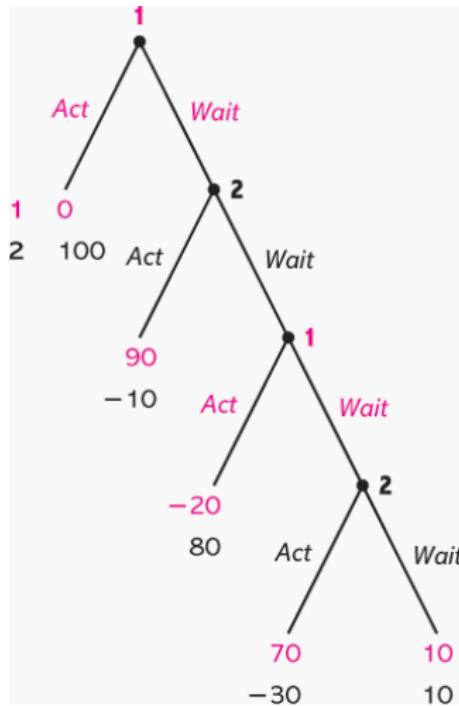
- Let's apply backward induction to find the Subgame Perfect Nash Equilibrium (SPNE) of this game.



- Hence, SPNE is...

$\{(Blockade, Air\ strike\ if\ USSR\ maintains), Withdraw\}$

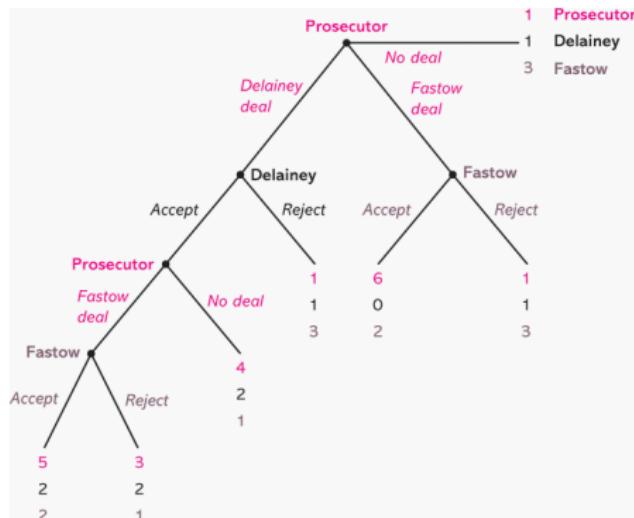
## Practice - I: War of Attrition



- Answer in Harrington, page 238.

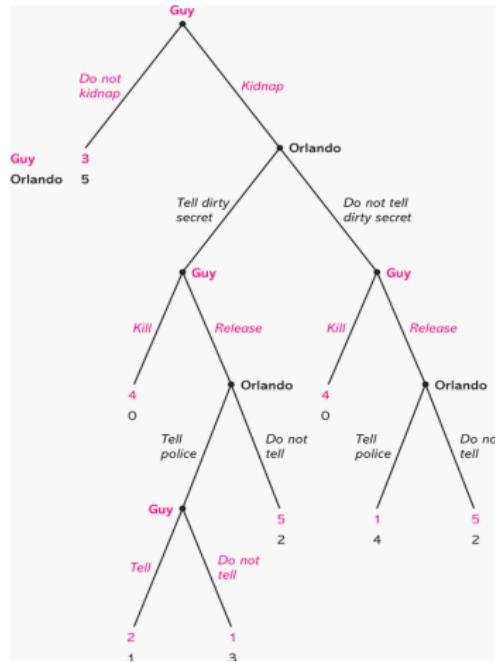
## Practice - II: Enron and Prosecutorial Perogative

- Delaney → Midlevel executive.
- Fastow → CFO.



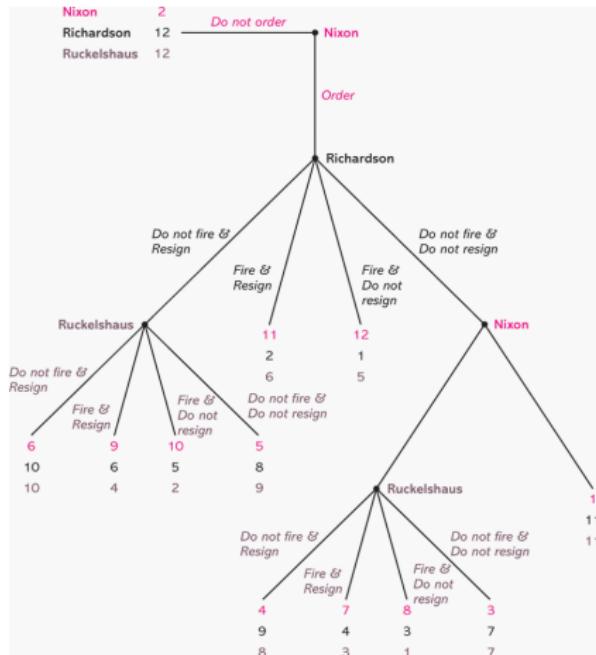
- Hence, SPNE is... in Harrington, pp. 227-229.

## Practice - III: Revised Kidnapping Situation



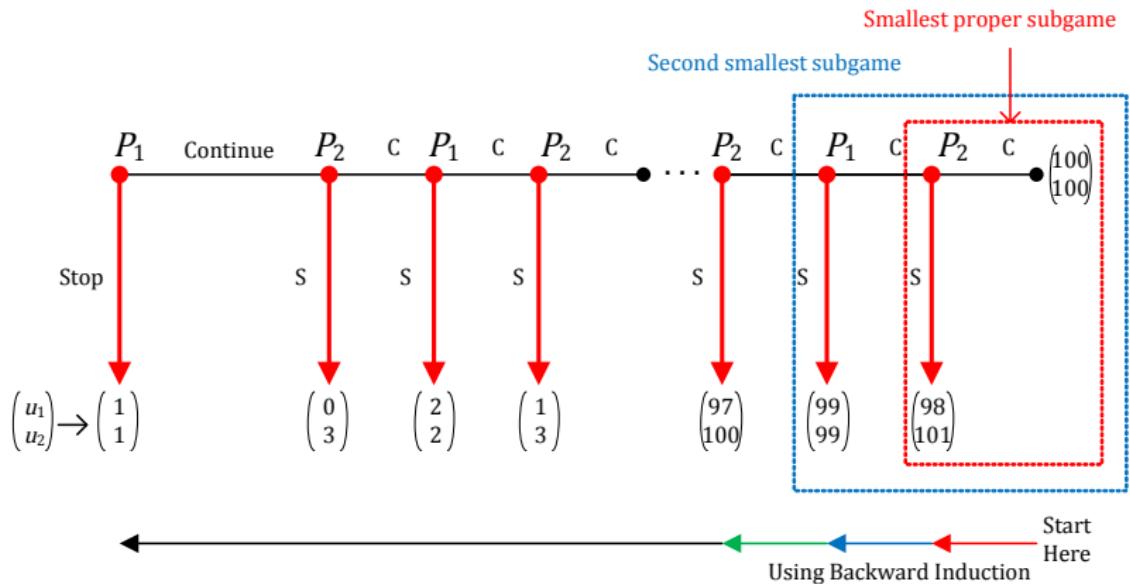
- SPNE is... Exercise 5 in Harrington, Ch. 8

# Practice - IV: Saturday Night Massacre



- SPNE is... Exercise 7 in Harrington, Ch. 8

# Centipede game



# Centipede game

- Let us use backwards induction:

1st) In the last node, P2 is called to move, so he compares

$$u_2(\text{Stop}) > u_2(\text{Continue}) \text{ since } 101 > 100$$

so he Stops.

2nd) In the previous to the last node, P1 knows that P2 will stop at the last node, then P1 compares

$$u_1(\text{Stop}) > u_1(\text{Continue}) \text{ since } 99 > 98$$

so he Stops.

.....

nth) In the first node, P1 knows that P2 will stop in the second stage, since P1 stops in the third, etc., so P1 compares

$$u_1(\text{Stop}) > u_1(\text{Continue}) \text{ since } 1 > 0$$

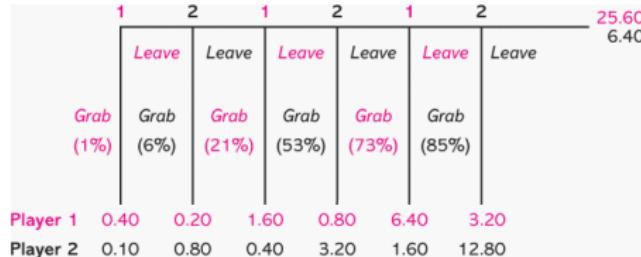
so P1 Stops.

## Centipede game

- Hence, the unique SPNE of the game is represented as  $(\text{Stop}_t, \text{Stop}_t)$  during every period  $t \in T$ , and for any finite length  $T$  of this centipede game.

- This is a rather disturbing result : because of being extremely rational and anticipating each other's actions even in 100 rounds, players forgo the opportunity to earn a lot of money.
- Why not start saying continue, and see what happens?
  - Experimentally tested.
  - (Some initial comments in Harrington. Many more in Camerer).

# Empirical test of Centipede Game



- Difference between the theoretical prediction and individuals' observed behavior in experiments.
- 1) **Bounded rationality.** People seem to use backward induction relatively well in the last 1-2 stages of the game, so they can easily anticipate what their opponent will do in just a few of posterior stages.
  - We could summarize this argument as Bounded rationality, since individuals' ability to backward induct is limited, and becomes more hindered as we move further away from the terminal nodes of the game.

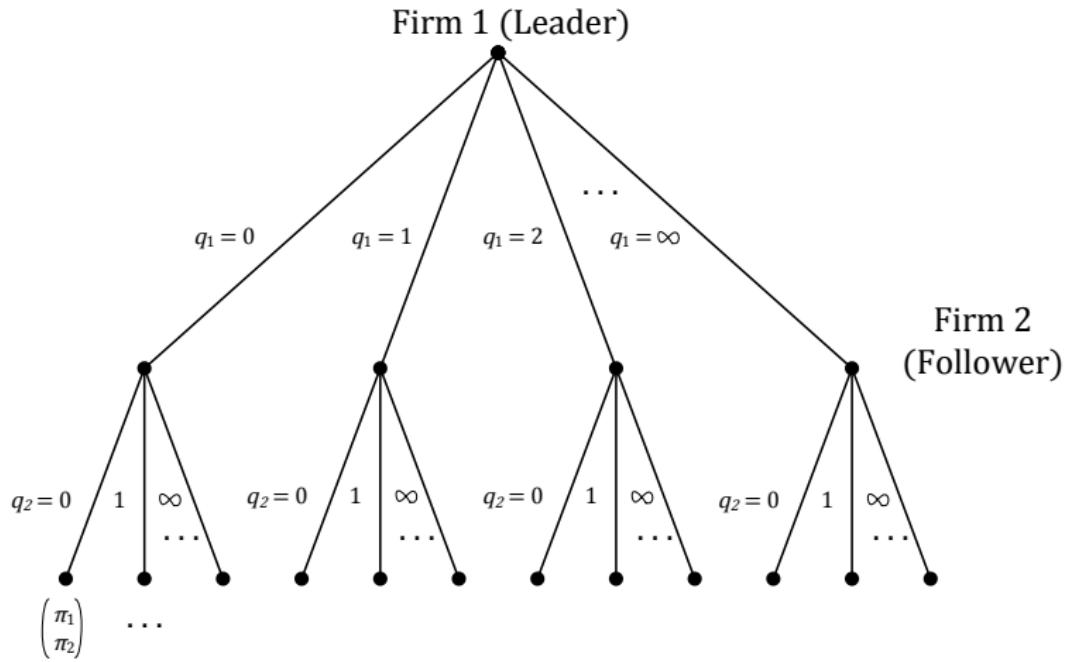
# Empirical test of Centipede Game

2) **Uncertainty about the presence of altruists in the population.** Another reason for their observed decision to leave money on the table could be their uncertainty about whether their opponent is an altruist.

- If P2 is an altruist, she values not only her own money, but also the money that P1 receives. Hence, P2 would leave money on the table rather than grab it.
- If you are in the shoes of P1 and you are uncertain about whether P2 is an altruist, you should then leave the money on the table, since P2 will respond leaving it on the table as well, and wait until the last node at which you are called on to move, where you grab the money.

- For more references, see the article "An experimental study of the centipede game" by Richard D. McKelvey and Thomas R. Palfrey, *Econometrica*, 60(4), 1992, pp. 803-836.

# Stackelberg game of sequential quantity competition



## Stackelberg game of sequential quantity competition

- Firm 1 is the leader, Firm 2 is the follower. Demand is given by

$$p(q_1, q_2) = 100 - q_1 - q_2$$

and marginal costs are \$10. Operating by backwards induction, we first solve the **follower's** profit maximization problem

$$\pi_2(q_1, q_2) = [100 - q_1 - q_2] q_2 - 10q_2$$

Taking FOCs we obtain the BRF2,

$$q_2(q_1) = 45 - \frac{q_1}{2}$$

Intuitively,  $q_2(q_1)$  represents the follower's optimal action at the smallest proper subgame (That initiated after Firm 1 chooses an output level,  $q_1$ ).

- Now, the **leader** inserts firm 2's BRF into her own profit function, since she knows how firm 2 will react to firm 1's production decision during the first stage of the game. Hence,

$$\begin{aligned}
 \underbrace{\pi_1(q_1, q_2)}_{\text{Leader's Profits}} &= \left[ 100 - q_1 - \underbrace{\left(45 - \frac{q_1}{2}\right)}_{q_2(q_1)} \right] q_1 - 10q_1 \\
 &= \frac{1}{2}(90 - q_1)q_1 = \frac{1}{2}(90q_1 - q_1^2)
 \end{aligned}$$

- Taking FOCs with respect to  $q_1$ , we obtain

$$\frac{90}{2} - \frac{2q_1}{2} = 0 \iff 90 = 2q_1 \iff q_1^* = 45$$

- Plugging this result into the follower's BRF (BRF2), we obtain

$$q_2(45) = 45 - \frac{45}{2} = 22.5$$

# Stackelberg game of sequential quantity competition

- The SPNE of Stackelberg Game is, however, more general:
  - Firm 1 chooses output  $q_1^* = 45$
  - Firm 2 responds to  $q_1$  output from Firm 1 by producing:

$$q_2(q_1) = \underbrace{45 - \frac{q_1}{2}}_{\substack{\text{More general than} \\ q_2=22.5}} \quad (BRF_2)$$

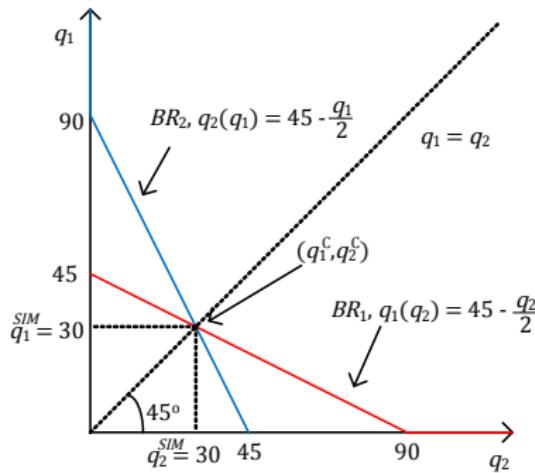
- Graphically,  $BRF_2$  represents Firm 2's best response to any production of Firm 1,  $q_1$ , that initiates any subgame (in which Firm 2 chooses output).

- For practice, you can check that this same exercise played simultaneously (a la Cournot), leads to

$$q_1^* = q_2^* = 30$$

# Stackelberg game of sequential quantity competition

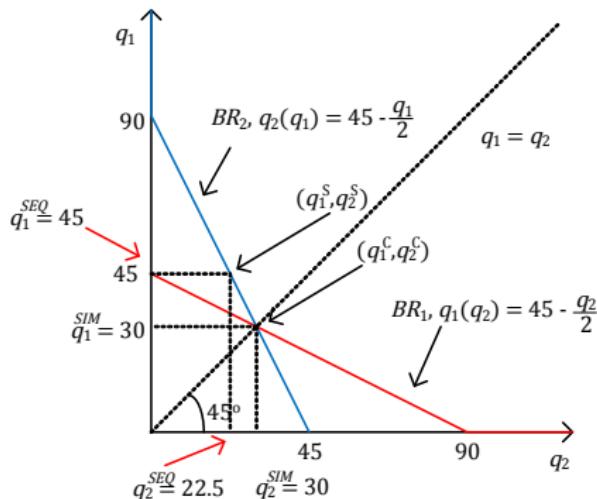
- A graphical representation of the equilibrium production levels when firms simultaneously choose their output levels (Cournot competition):



where  $(q_1^C, q_2^C)$  is the equilibrium of the simultaneous-move version of the game (Cournot).

# Stackelberg game of sequential quantity competition

- Superimposing our results about the sequential-move version of the game (Stackelberg competition) on top of the previous figure, we find:

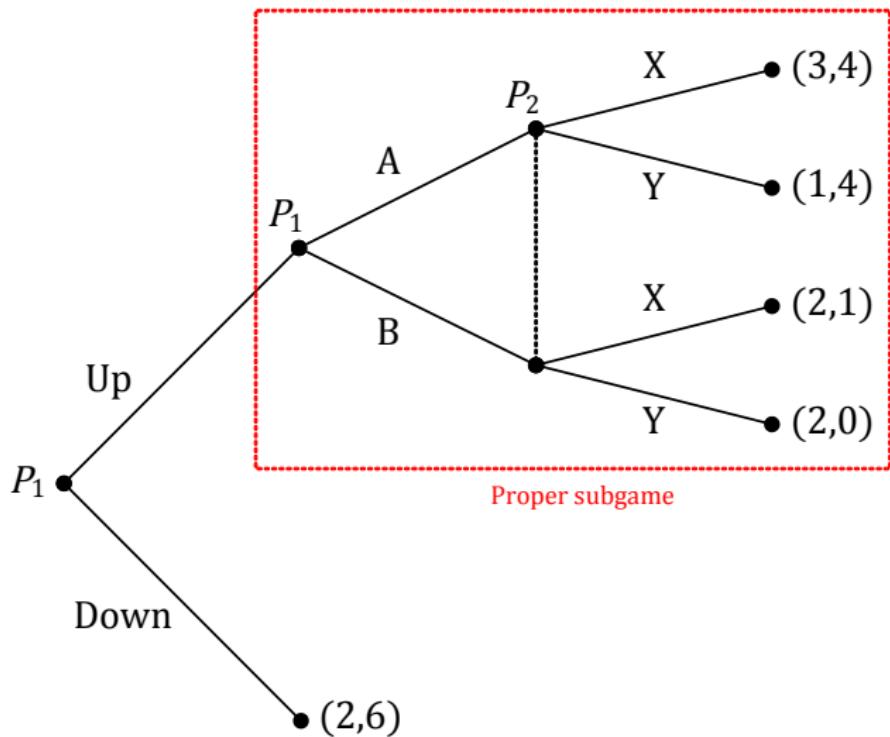


where  $(q_1^S, q_2^S)$  is the equilibrium of the sequential-move version of the game (Stackelberg).

# What if there is imperfect information?

- Harrington, Ch. 9
- What if the game includes elements of imperfect information?
  - For instance, player 2 cannot observe what player 1 does before him.
- We can still use backward induction, but...
  - Remember that backward induction requires us to always start from the **smallest proper subgame**.
- Let's do one example together.

# What if there is imperfect information?



# What if there is imperfect information?

1st) Focus on the smallest proper subgame, and find the NE of that subgame.

$P_2$

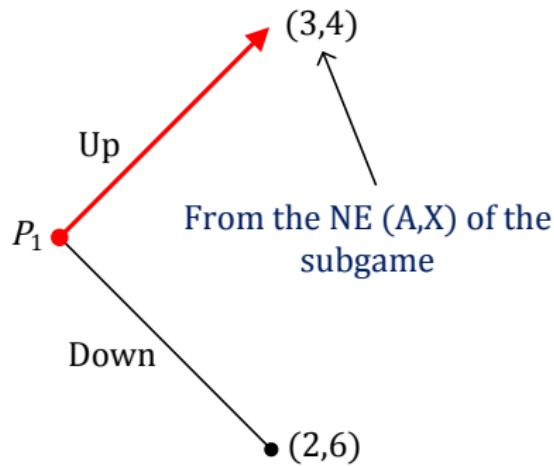
	X	Y
A	<u>3</u> , 4	1, <u>4</u>
B	2, <u>1</u>	<u>2</u> , 0

$P_1$

(A, X) is the NE of the subgame.

## What if there is imperfect information?

2nd) Given the NE you have found above, find the NE of the next subgame.



Hence, the Subgame Perfect Nash Equilibrium of this game is  $(Up/A, X)$ .

# What if there is imperfect information?

- Does this SPNE coincide with NE? No !

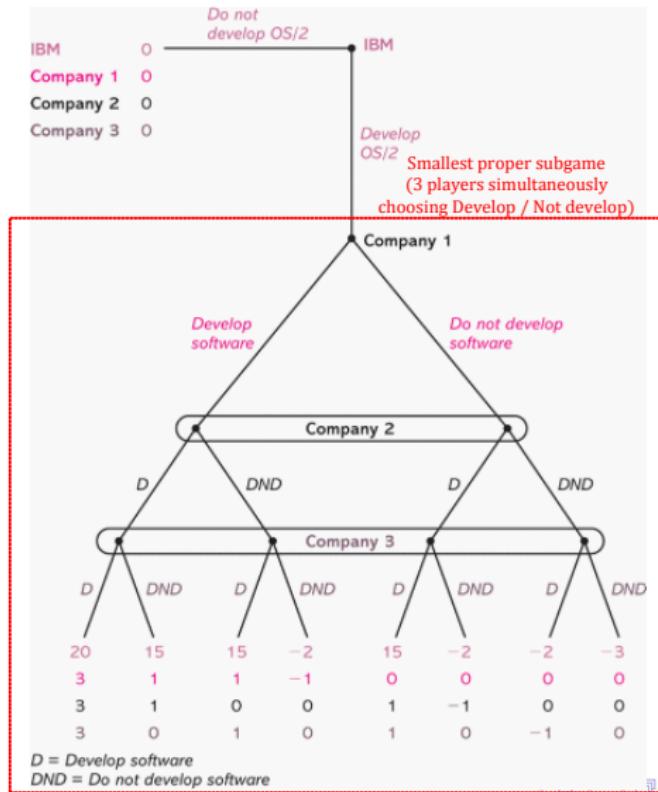
		$P_2$	
		X	Y
$P_1$	Up/A	3, 4	1, 4
	Up/B	2, 1	2, 0
	Down/A	2, 6	2, 6
	Down/B	2, 6	2, 6

- 3 psNE:  $(Up/A, X)$ ,  $(Down/A, Y)$ , and  $(Down/B, Y)$ .
- The first psNE is the unique SPNE (Highlighted green), but the latter two NE specify strategies that are not sequentially rational since they are not the NE of the proper subgame (Highlighted yellow).

## What if the smallest subgame is played by three players?

- Harrington, pp. 263-276.
- Then we need to find the NE of the subgame, namely, a simultaneous-move game played by three players.
- **Motivating example:** IBM developing the OS/2 operating system.
  - Microsoft developed MS-DOS for IBM in the 1980s.
  - IBM allowed Microsoft to retain the copyright of MS-DOS, which is probably one of the worst business decisions in history.
  - Afterwards, IBM started to develop an alternative operating system: OS/2.
  - However, the success of such operating system depended on the number of software companies developing compatible programs.
  - In the following game, we consider that developing OS/2 is only profitable for IBM if two or more software developers write compatible applications. →

# The OS/2 game



# The OS/2 game

- We can alternatively represent the previous subgame in which companies 1-3 simultaneously and independently select whether to develop software compatible with OS/2 , as follows:

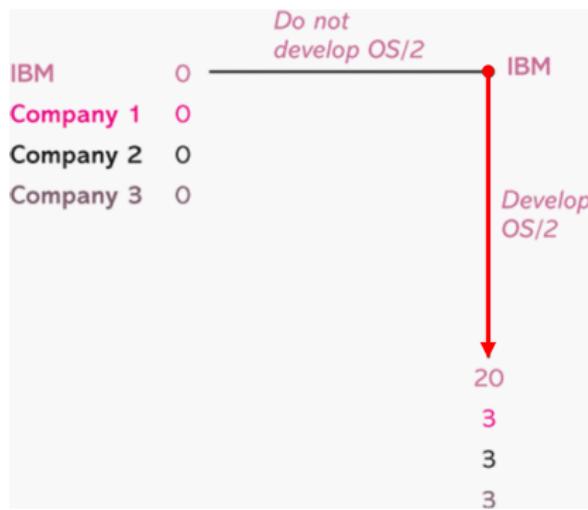
		Company 2		Company 2	
		Develop	Do not develop	Develop	Do not develop
Company 1	Develop	3, 3, 3	1, 0, 1	1, 1, 0	-1, 0, 0
	Do not develop	0, 1, 1	0, 0, -1	0, -1, 0	0, 0, 0
		Company 3, Develop		Company 3, Do not Develop	

## The OS/2 game

- Hence, we can identify two psNE in the subgame:
  - (D,D,D) with corresponding payoffs (3,3,3), and
  - (ND,ND,ND) with corresponding payoffs (0,0,0).
- Let us separately introduce each of these results at the end of the branch that has IBM developing the OS/2 system.
  - See the following two figures, one for the (D,D,D) equilibrium of the subgame and another for the (ND,ND,ND) equilibrium.

# The OS/2 game

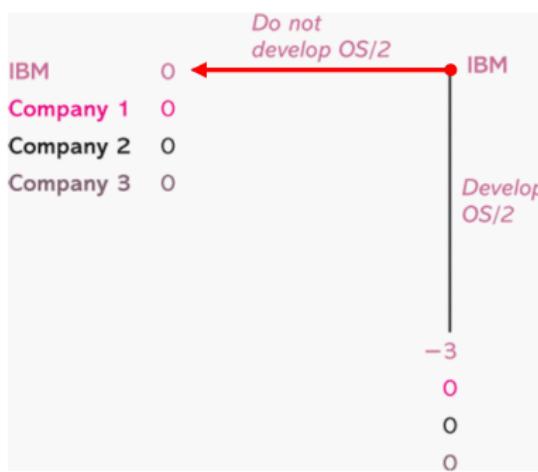
- If  $(D, D, D)$  is equilibrium of the subgame, then



Therefore,  $(\text{Develop OS/2}, D, D, D)$  is a SPNE of this game.

# The OS/2 game

- If, instead, (ND, ND, ND) is equilibrium of the subgame, then



Therefore, (Don't Develop OS/2, ND, ND, ND) is a SPNE of this game.

# The OS/2 game

- One second... did we forget something?
  - Yes! We didn't check for the possibility of a msNE in the subgame initiated by IBM's decision to develop OS/2.
  - In other words: is there a msNE in the three-player subgame?
- Since all three software developers are symmetric, if they randomize between D and ND, they must be doing so using the same probability, e.g.,  $d \in [0, 1]$ .

# The OS/2 game

- The expected payoff that company 1 obtains when developing software is

$$\begin{aligned} E\pi_1(D) &= \underbrace{d^2 3}_{\text{if firms 2 and 3 develop}} + \underbrace{d(1-d)2}_{\text{if only firm 2 develops}} + \\ &\quad \underbrace{d(1-d)2}_{\text{if only firm 3 develops}} + \underbrace{(1-d)^2(-1)}_{\text{if neither 2 nor 3 develop}} \\ &= 4d - 1 \end{aligned}$$

while that of not developing software is simply zero, i.e.,  $E\pi_1(ND) = 0$ , which is independent upon firm 2 or 3 developing software.

- Where are these payoffs coming from? →

# The OS/2 game

- Firm 1's expected profit from developing (Only look at the first row of every matrix):

If firm 2 and 3 develop

		Company 2	
		Develop	Do not develop
Company 1	Develop	3, 3, 3	1, 0, 1
	Do not develop	0, 1, 1	0, 0, -1

Company 3, Develop

If firm 3 develops but firm 2 does not.

		Company 2	
		Develop	Do not develop
Company 1	Develop	1, 1, 0	-1, 0, 0
	Do not develop	0, -1, 0	0, 0, 0

Company 3, Do not Develop

$$\begin{aligned}
 E\pi_1(Dev) &= d^2 \cdot 3 + d(1-d) \cdot 1 + (1-d)d \cdot 1 + (1-d)^2 \cdot (-1) \\
 &= 4d - 1
 \end{aligned}$$

# The OS/2 game

- Firm 1's expected profit from not developing (Second row in all matrices):

Company 2

	Develop	Do not develop
Develop	3, 3, 3	1, 0, 1
Do not develop	0, 1, 1	0, 0, -1

Company 1

If firm 2 and 3 develop

If firm 3 develops but firm 2 does not.

If firm 2 develops but firm 3 does not.

If neither firm 2 nor 3 develop

Company 2

	Develop	Do not develop
Develop	1, 1, 0	-1, 0, 0
Do not develop	0, -1, 0	0, 0, 0

Company 1

Company 3, Develop

Company 3, Do not Develop

$$\begin{aligned} E\pi_1(\text{Not dev}) &= d^2 \cdot 0 + d(1-d) \cdot 0 + (1-d)d \cdot 0 + (1-d)^2 \cdot 0 \\ &= 0 \end{aligned}$$

# The OS/2 game

- If firm 1 randomizes between Develop and Do not develop, it must be that it is indifferent between D and ND, that is

$$E\pi_1(\text{Dev}) = E\pi_1(\text{Not dev}) \implies 4d - 1 = 0$$

solving for probability  $d$ , we obtain  $d = \frac{1}{4}$ .

# The OS/2 game

- Since all three software companies are symmetric, they all develop software with probability  $d = \frac{1}{4}$ .
- Hence, IBM's expected profit from developing OS/2 is

$$E\pi_{IBM}(Dev) = \underbrace{d^3 20}_{\substack{\text{3 companies} \\ \text{develop}}} + \underbrace{3d^2(1-d)15}_{\substack{\text{Only two companies develop} \\ \text{(1 and 2, 1 and 3, or 2 and 3)}}} +$$
$$+ \underbrace{3d(1-d)^2(-2)}_{\substack{\text{Only one company develops} \\ \text{(3 possible companies)}}} + \underbrace{(1-d)^3(-3)}_{\substack{\text{No company} \\ \text{develops}}}$$

## OS/2 game

- And since  $d = \frac{1}{4}$ ,

$$\begin{aligned} E\pi_{IBM}(Dev) &= \left(\frac{1}{4}\right)^3 20 + 3 \left(\frac{1}{4}\right)^2 \left(1 - \frac{1}{4}\right) 15 + \\ &\quad + 3 \frac{1}{4} \left(1 - \frac{1}{4}\right)^2 (-2) + \left(1 - \frac{1}{4}\right)^3 (-3) \\ &= \frac{20}{64} \end{aligned}$$

# The OS/2 game

- Plugging  $E\pi_{IBM}(Dev) = \frac{20}{64}$  as the expected profit that IBM obtains from initiating the subgame...
  - We find that IBM chooses to develop OS/2.
  - Hence, we have found a third SPNE: (Develop OS/2, D with probability  $d=1/4$  for all software firms  $i = \{1, 2, 3\}$ ).

