Algorithmic Game Theory

Javad Salimi
Salimi.sartakhti@gmail.com
1396

Some Formal Definitions

DYNAMIC GAMES II

Dynamic Games II

1. First Mover or Second Mover?
2. Zermelo Theorem
3. Perfect Information/Pure Strategy
4. Imperfect Information/Information Set
5. Information vs Time

First mover advantage

- Is being the first mover always good?
  - ***Yes, sometimes***: as in the Cournot Stackelberg model
  - ***Not always***, as in the Rock, Paper, Scissors game
  - Sometimes neither being the first nor the second is good

The NIM game

- We have two players
- There are two piles of stones, A and B
- Each player, in turn, decides to delete some stones from whatever pile
- The player that remains with the last stone wins

Let’s play the game

The NIM game (2)

- If piles are equal → second mover advantage
- If piles are unequal → first mover advantage

- You'll know who will win the game from the initial setup
- You can solve through backward induction

Dynamic Games II

1. First Mover or Second Mover?
2. Zermelo Theorem
3. Perfect Information/Pure Strategy
4. Imperfect Information/Information Set
5. Information vs Time

The Zermelo Theorem

- Consider a general 2-Player game
- We assume ***<u>perfect information</u>***
  - Players know where they are in the game tree and how they got there
- We assume a finite game, i.e. a game-tree with a finite number of nodes
- There can be three or fewer outcomes: W_1 (player 1 wins), L_1 (player 2 wins), T (tie)

The Zermelo Theorem

The result (or solution) of this game is:

1. Either player 1 can force a **win (over player 2)**
2. Or player 1 can force a **tie**
3. Or player 2 can force a **loss (on player 1)**

The Zermelo Theorem

- This theorem appears to be trivial:
  - Three possible outcomes
  - Games are subdivided in three categories:
    - Those in which, ***<u>whatever player 2 does</u>***, player 1 can win (provided he/she plays well)
    - Those in which player 1 can always force a draw/tie
    - Those in which, player 1 is toast, and can only loose

Examples of games

- **NIM**
  - Depends on number of stones in the first stage
- **Tic-tac-toe**:
  - If players play correctly, you can always force a tie
  - If players make wrong moves, they can loose
- **Chess** → has a solution!
- In fact, the theorem doesn't tell you how to play, it just tells you there is a solution!

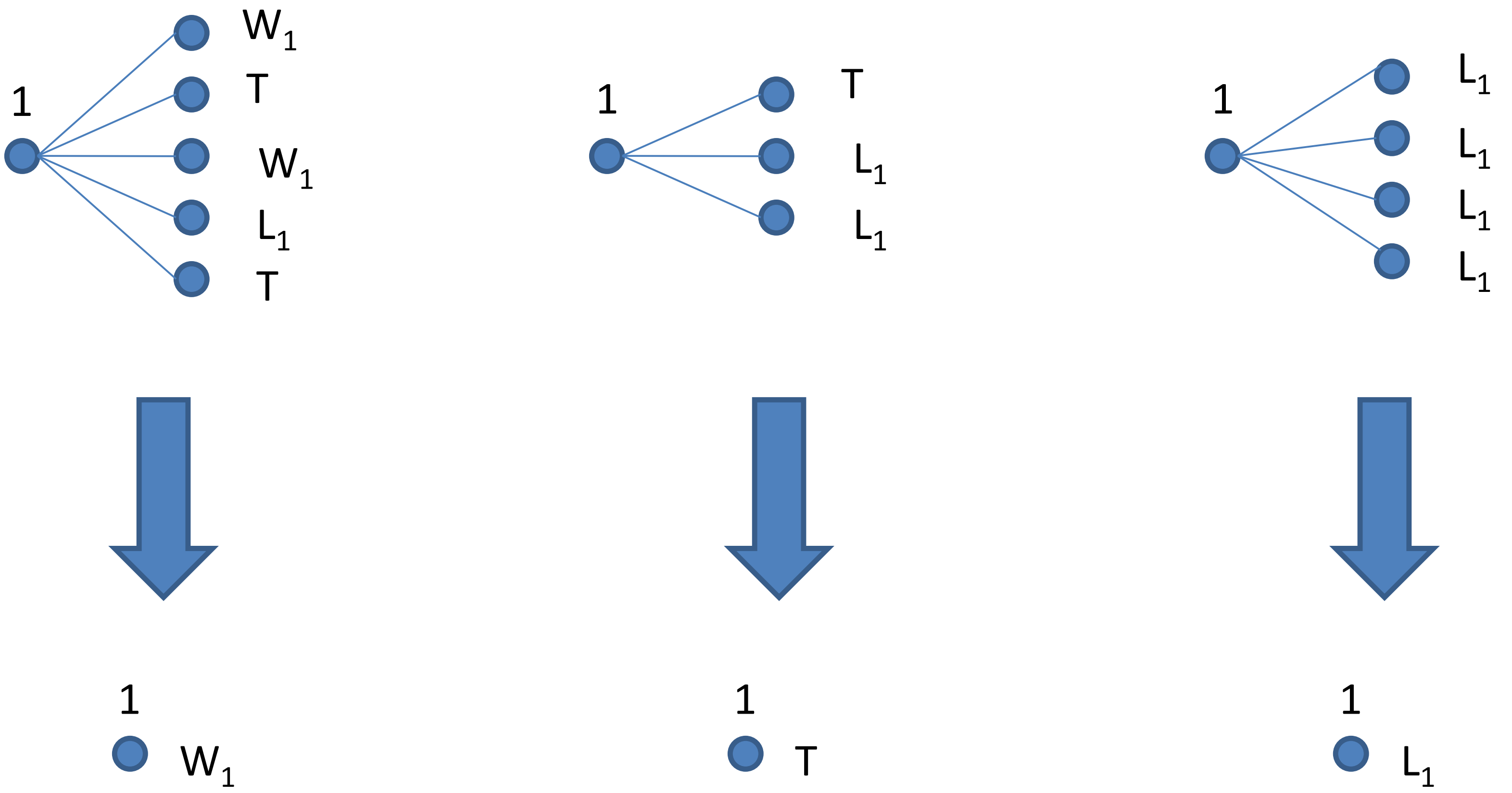
The Zermelo Theorem proof (1)

- We're going to prove the theorem, in a sketchy way, as this is relates to backward induction

- Proof methodology:
  Induction on maximum length of a game N
  – We'll start with an induction hypothesis
  – And we'll prove this is true for longer games

The Zermelo Theorem proof (2)

- If N = 1

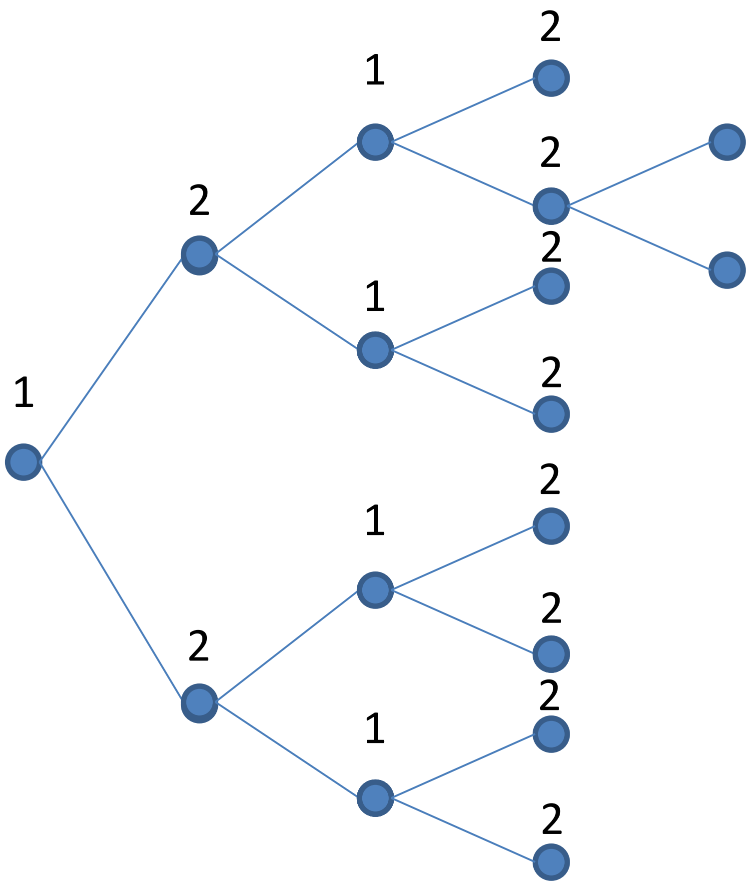


The Zermelo Theorem proof (3)

- Induction hypothesis:
  Suppose the claim is true for all games of length $\leq N$
- We claim, therefore it will be true for games of length $N+1$

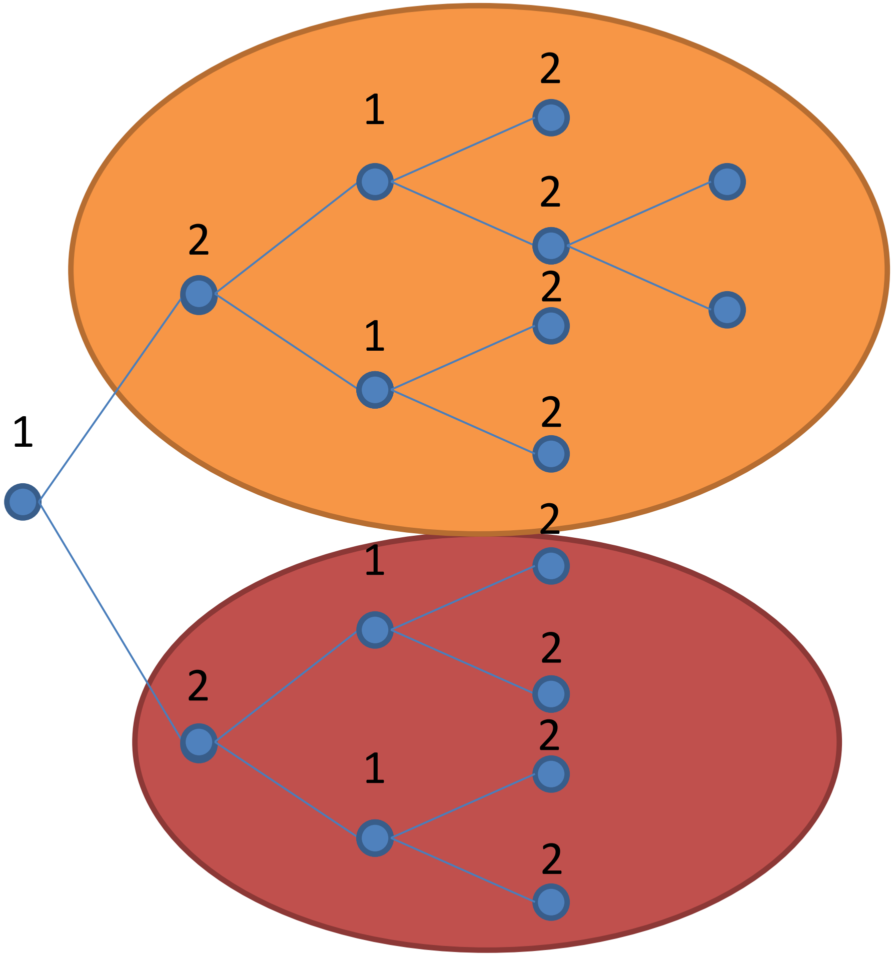
- Let's take an example

The Zermelo Theorem proof (4)



- What is the maximum length of the game?

The Zermelo Theorem proof (5)

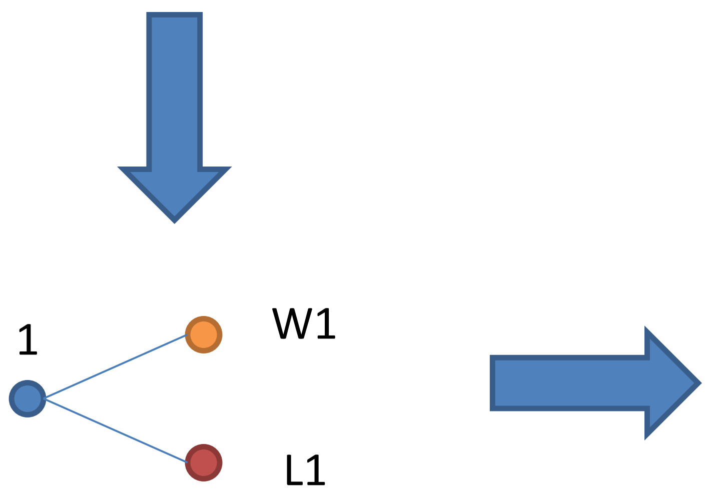


We have two **sub-games**

- The upper **sub-game**: follows “1” and it has length 3
- The lower **sub-game**: follows “1” and has length 2

The Zermelo Theorem proof (6)

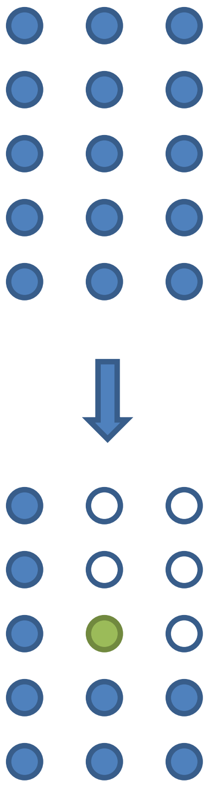
- By induction hypothesis (for N=3), upper sub-game has a solution, say "W_1"
- Again, by induction hypothesis (N=2), lower sub-game has a solution, say "L_1"



- This game has a solution, it is a game of length 1 we know already!

A more Complex Example

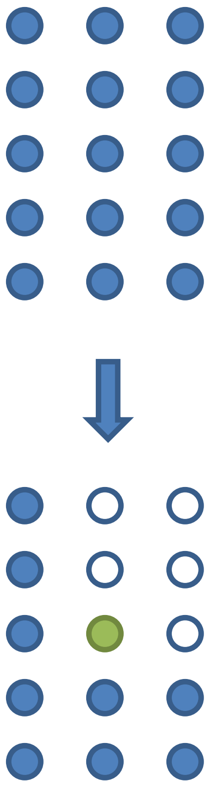
- Suppose we have an array of stones, and two players
- Sequential moves, each player can delete some stones
  - Select one, delete all stones that lie above and right
- **The looser is the person who ends up removing the last rock**



A more Complex Example

- According to Zermelo's Theorem, this game has a solution and the advantage depends on NxM, the size of the array

- Think about it!



Dynamic Games II

1. First Mover or Second Mover?
2. Zermelo Theorem
3. Perfect Information/Pure Strategy
4. Imperfect Information/Information Set
5. Information vs Time

FORMAL DEFINITIONS

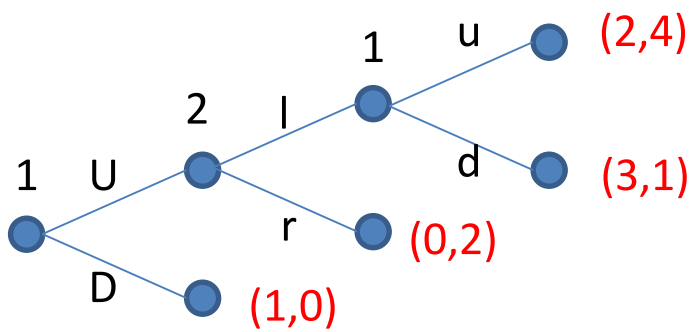
Perfect Information Game

A game of ***perfect information*** is one in which at each node of the game tree, the player whose turn is to move **knows** which node she is at and **how** she got there

Pure Strategy

A ***pure strategy*** for player i in a game of perfect information is a ***complete plan*** of actions: it specifies which action i will take at each of its decision nodes

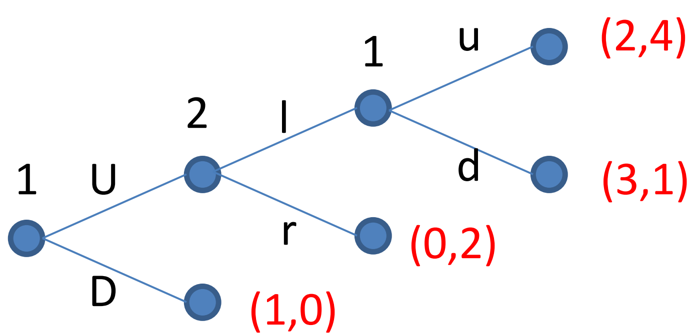
Example 1



- Strategies
  - Player 2:
    [l], [r]
  - Player 1:
    [U,u], [U,d]
    [D, u], [D,d]

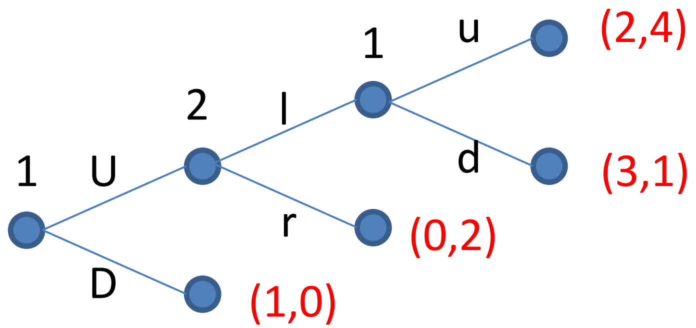
Hey, they look redundant, but we need them!

Example I



- Note:
  - In this game it appears that player 2 may never have the possibility to play her strategies
  - This is also true for player 1!

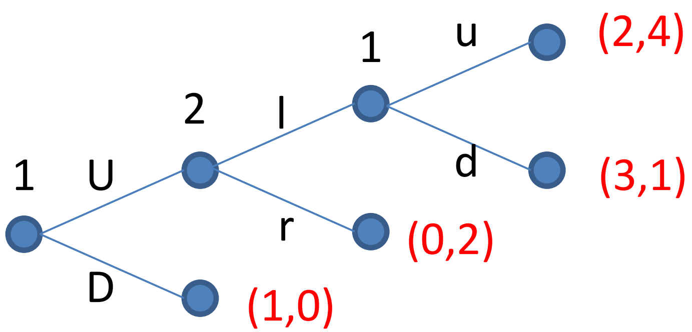
Example I



- Backward Induction
  - Start from the end
    - "d" → higher payoff
  - Summarize game
    - "r" → higher payoff
  - Summarize game
    - "D" → higher payoff

➢ BI :: {[D,d],r}

Example I

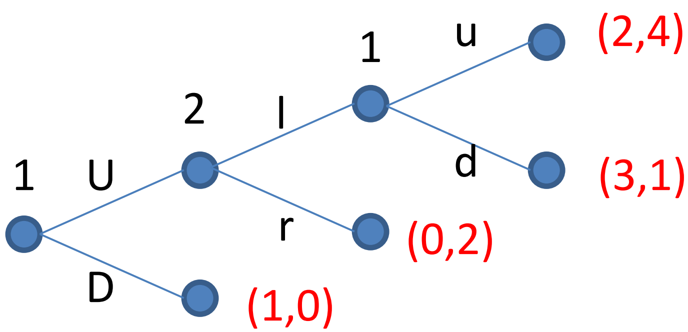


	l	r
U u	2,4	0,2
U d	3,1	0,2
D u	1,0	1,0
D d	1,0	1,0

From the ***extensive form***

To the ***normal form***

Example I



	l	r
U u	2,4	0,2
U d	3,1	0,2
D u	1,0	1,0
D d	1,0	1,0

Backward Induction

{[D, d],r}

?

Nash Equilibrium

{[D, d],r}
{[D, u],r}

Wait! We will find an answer to this later.

Dynamic Games II

1. First Mover or Second Mover?
2. Zermelo Theorem
3. Perfect Information/Pure Strategy
4. Imperfect Information/Information Set
5. Information vs Time

Let's be in the real world!

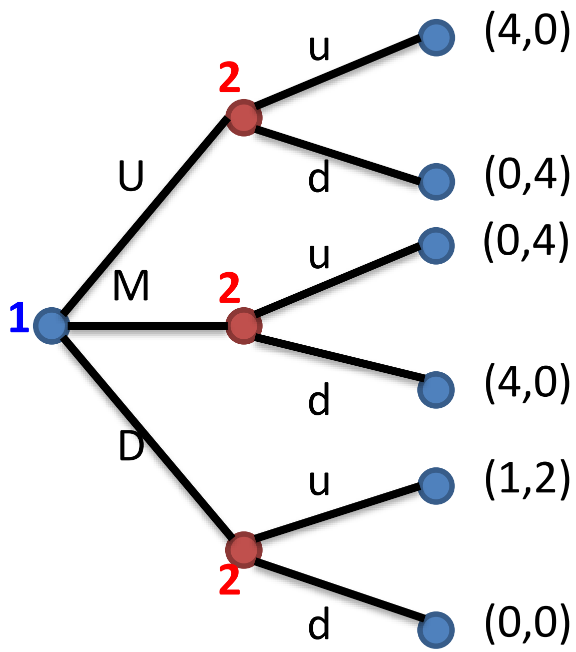
IMPERFECT INFORMATION

Brief Review

- We have seen **simultaneous** move games, in which players **cannot** observe strategies and have to reason based on the idea of **best response**
- We have seen **sequential** move games, in which **observation** is allowed, and players reason using **backward induction**

- Now, let's study a class of games in which these two approaches **are blended**

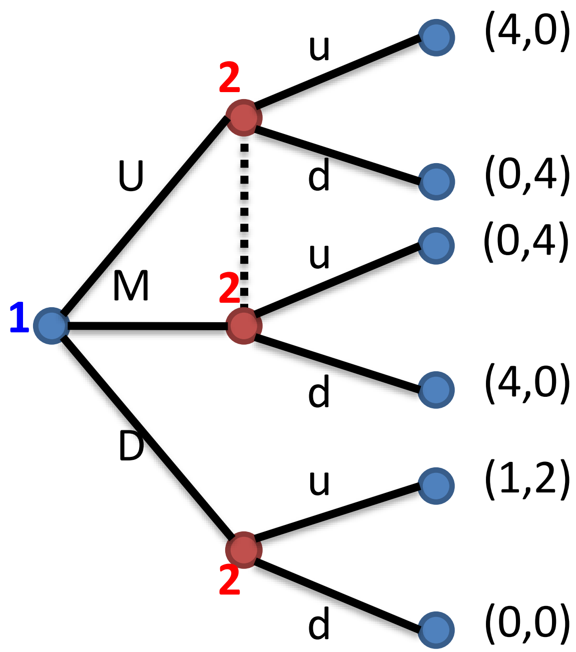
A Simple Dynamic Game



- Sequential move game
- Assume for a moment perfect information

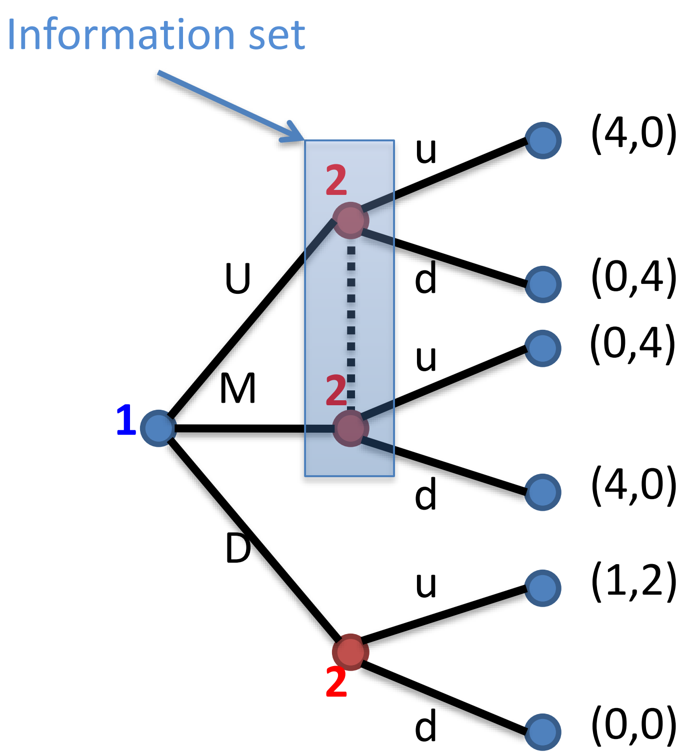
- We know how to solve it using **backward induction**
  - Player 1 knows that if he chooses U or M, player 2 can crush him
  - Player 2 has a huge second mover advantage in the first branches of the tree

Imperfect Information



- Sequential move game
- **Imperfect information**
  - Player 2 cannot distinguish where she is on (some parts of) the tree
- If player 1 chooses D, player 2 can observe it
- If player 1 chooses U or M, player 2 doesn't know which choice was made

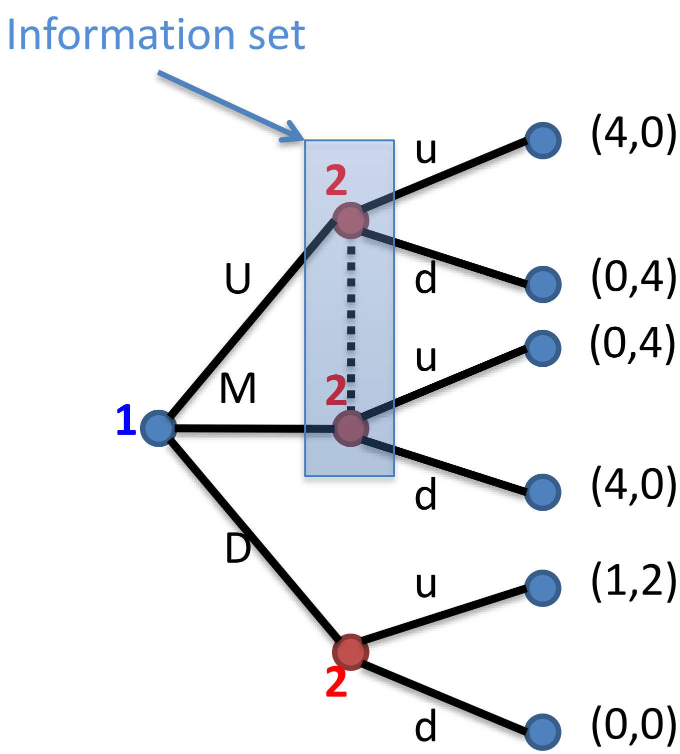
Information Set



- The idea is that the two internal nodes are in the same **information set**
  - Player 2 knows that player 1 chose whether U or M, but not which one

- How can we analyze this kind of games?

How to Solve?

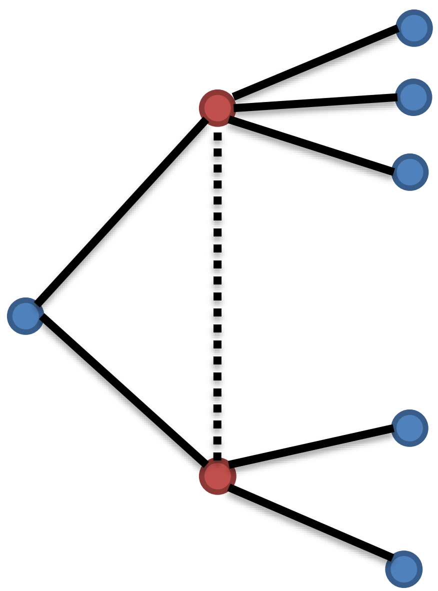


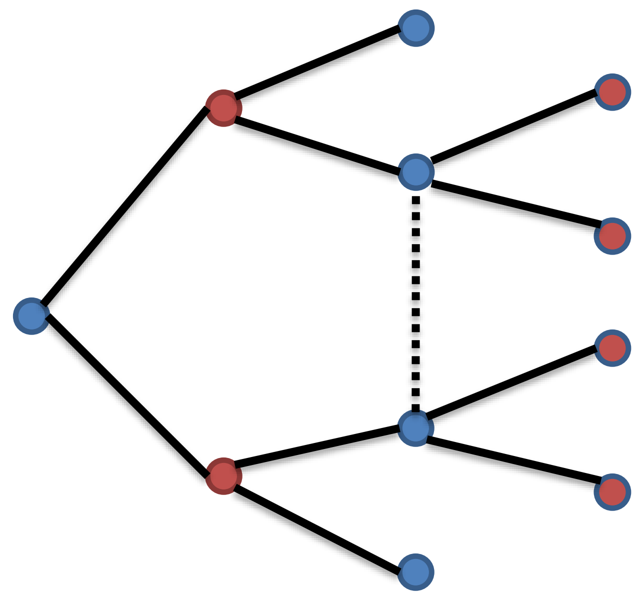
- The simple backward induction argument (player 2 could always crush player 1) does not hold anymore
- Moreover, player 1 **knows** that player 2 cannot distinguish U or M
  - Player 1 **might** decide to randomize over U and M, and hope to get an expected payoff of 2
  - A payoff of 2 is better than what player 1 could ever obtain by choosing D

Information Set

- An **information set** of player i is a collection of player i's **decision nodes** among which i **cannot** distinguish

Examples: Are these information sets?





Information Sets: Some Rules

- **Rule 1**: A player must not be able to infer in which node she is by looking at the **number of available strategies** she has

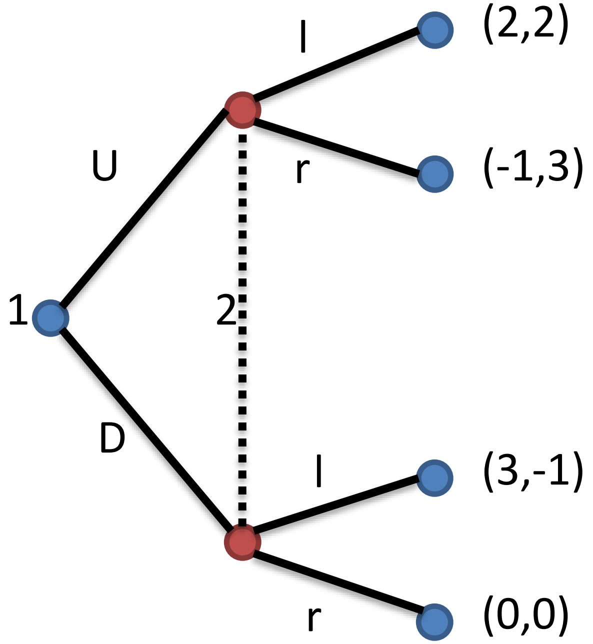
- **Rule 2**: provided a player can recall what she did earlier on in the tree, she shouldn't be able to distinguish where she is
  - This assumption is called **perfect recall**
  - ***NOTE***: perfect recall is not always realistic!

Definition: Perfect/Imperfect Information

- A game of **perfect information** is a game in which all information sets in the game tree include just one node

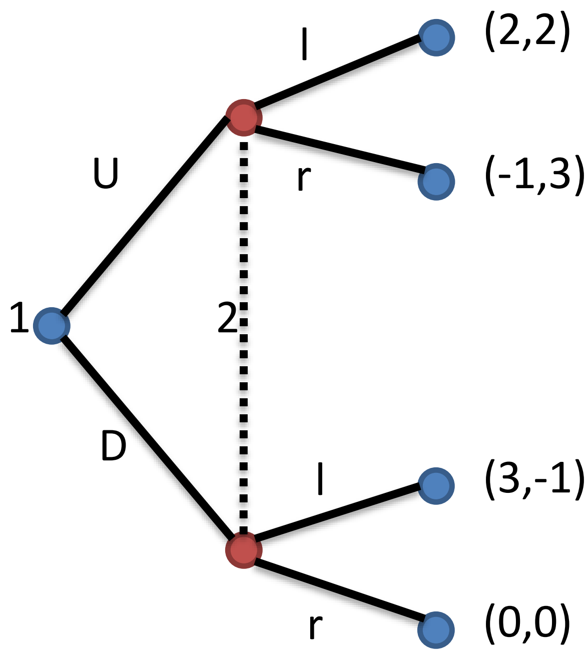
- A game of **imperfect information** is not a game of perfect information!

Simple Example



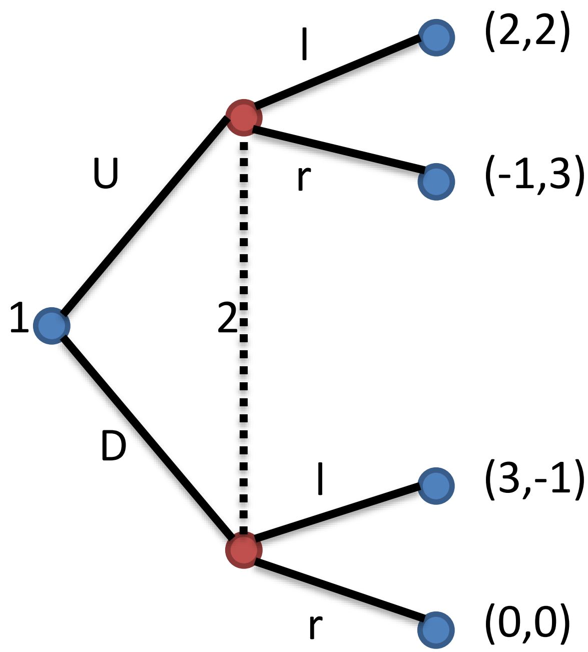
- The **information set** indicates that player 2 cannot observe whether player 1 moved *up* or *down*
  - **Perfect information**: player 2 could have chosen separately, in each node, whether to choose *left* or *right*
  - **Imperfect information**: player 2 has *only the choice of choosing left or right*, for both nodes, since she doesn't know which one she'll be at

Solution



- There's a catch here that makes the game easy:
  - Whatever is the information set, for player 2 choosing *right* is consistently better than choosing *left*
  - This game solves out rather like when using **backward induction**

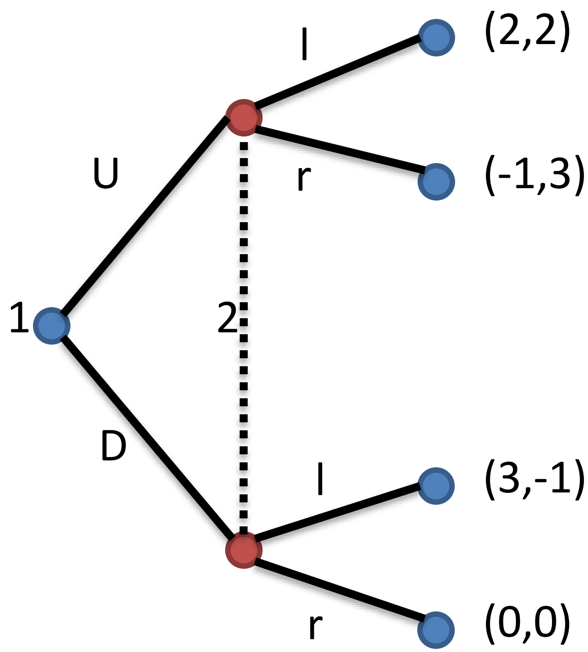
From Dynamic to Static Game



		Player 2	
		l	r
Player 1	U	2,2	-1,3
	D	3,-1	0,0

- ***Question***: What game is this?
  - Prisoners Dilemma
- Notice that by using **information sets**, we were able to represent in a tree a simultaneous move game
  - It does not really matter the **time** here, what matters is **information**

From Dynamic to Static Game



		Player 2	
		l	r
Player 1	U	2,2	-1,3
	D	3,-1	0,0

- We don't have **redundant** strategies in the matrix
- We can't have a complete action plan when we don't know where we are in the tree
  - This implies we have to revisit our definition of strategy

Pure Strategy: A New Definition

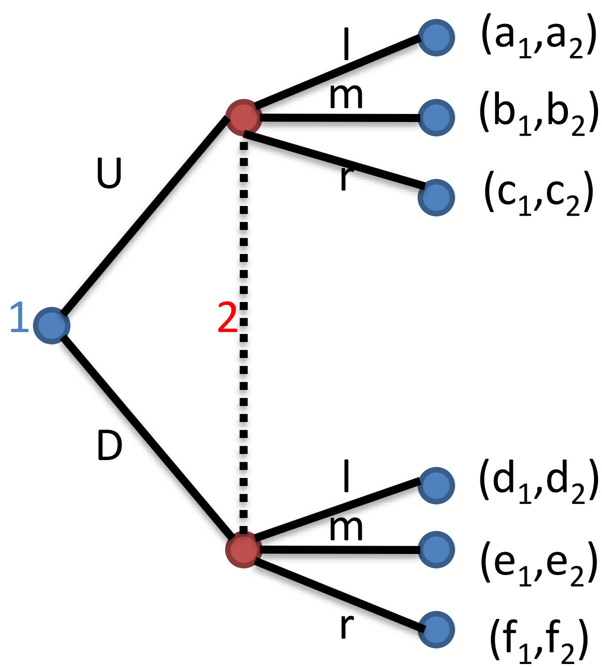
- A **pure strategy** of player *i* is a complete plan of action: it specifies what player *i* will do at each of its ***information sets***

- It looks like the same definition we saw last time, but this one involves information sets and it is more general
  - The idea remains the same: we want to transform a game tree in a matrix

Dynamic Games II

1. First Mover or Second Mover?
2. Zermelo Theorem
3. Perfect Information/Pure Strategy
4. Imperfect Information/Information Set
5. Information vs Time

Information vs Time

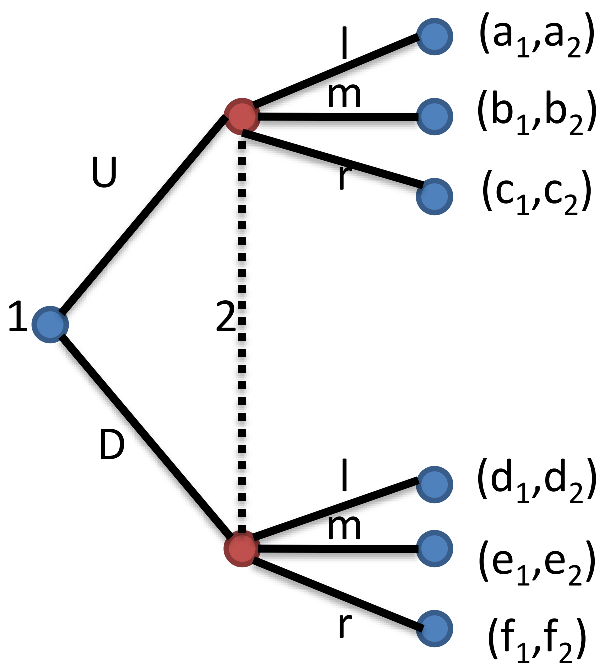


- Player 2 does not know if player 1 chooses *up* or *down*

➔ Player 2 has just three choices

- Our goal now is to transform the game into a matrix

Information vs Time



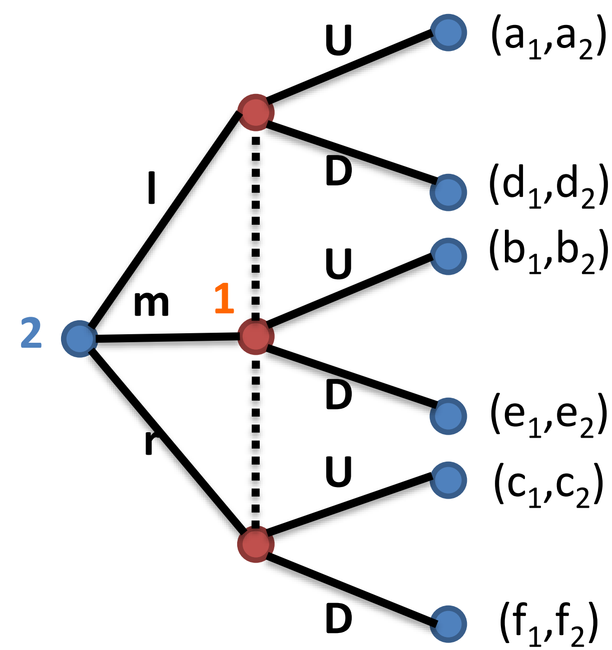
Player 2

		l	**m**	**r**
Player 1	**U**	a_1,a_2	b_1,b_2	c_1,c_2
	D	d_1,d_2	e_1,e_2	f_1,f_2

CLAIM: If we look at the matrix above **it is not obvious** that the game tree on the left is the only possible tree that could generate the matrix

Information vs Time

		Player 2		
		l	m	r
Player 1	U	a_1,a_2	b_1,b_2	c_1,c_2
	D	d_1,d_2	e_1,e_2	f_1,f_2



In the game tree to the right, player 2 moves first, then player 1 moves but she doesn't know which action player 2 chose

CLAIM: These two games trees are **equivalent**

Observations

- What matters is **not time**, but **information**
- We would like to set-up the machinery to analyze such games and predict what it is going to happen