TÉCNICO LISBOA

Multiagent decision making and Games in Extensive Form

Outline

- **Perfect-information games in extensive form**
- Strategies and equilibria
- Subgame-perfect equilibrium
- Backward induction
- Example

Extensive-form games

- Normal-form games do not incorporate any notion of **sequence or time**
  - Normal-form games assume that agents **select their actions simultaneously**
  - In many examples of normal-form games, we have considered **one-shot games**

Extensive-form games

- Extensive-form games:
  - Also known as **tree-form games**
  - An **alternative representation** that makes the **temporal structure explicit**
  - We now present the ***perfect-information*** **extensive-form games** (finite games)

Extensive-form games

- What are perfect-information games in extensive form?
  - A **tree** in the sense of **graph theory**
  - Each **node** represents the **choice** of an agent
  - Each **edge** represents an **action** of an agent
  - The **leaves** represent a **final outcome (payoffs/utility)**

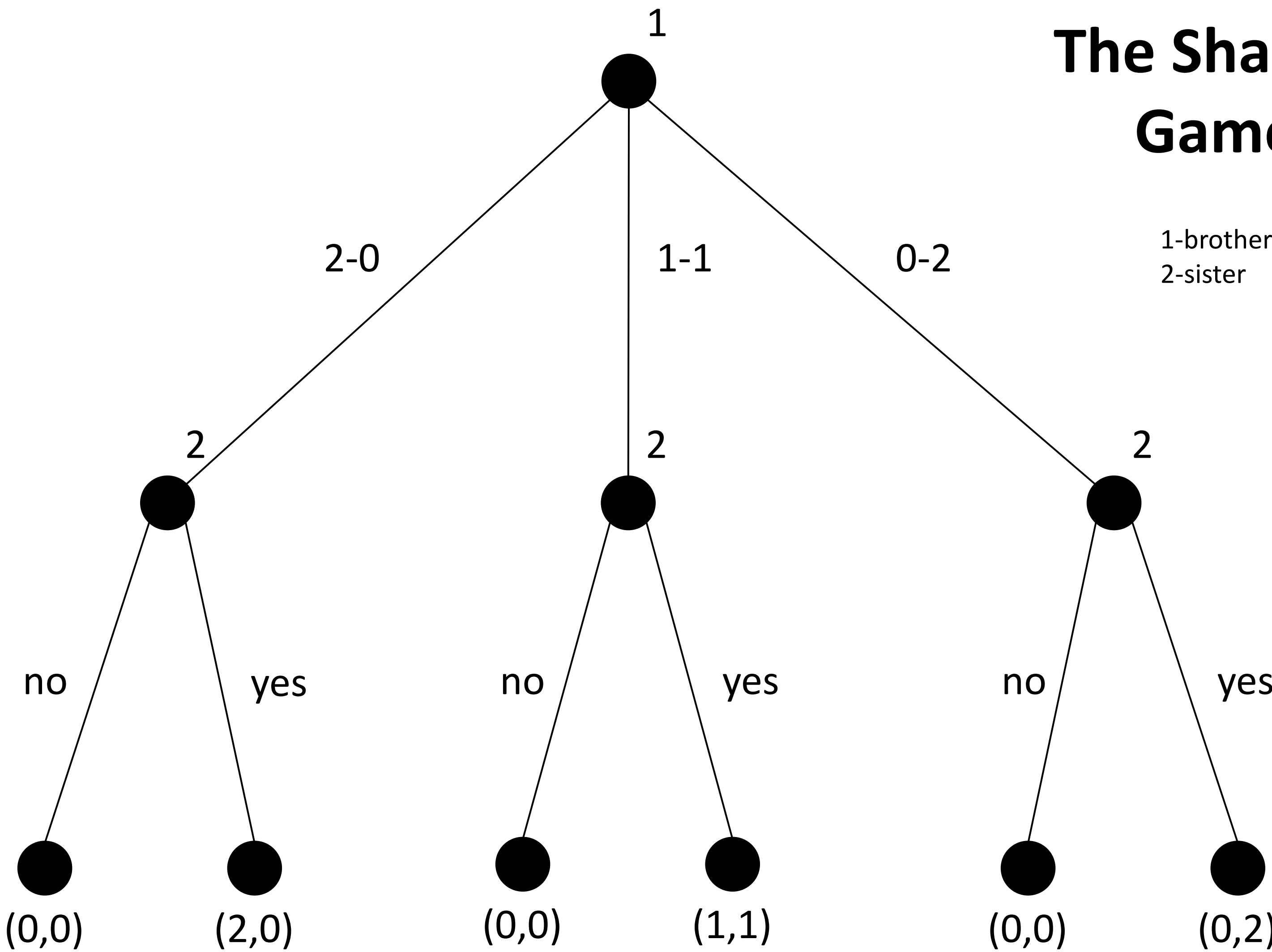
Extensive-form games

- Example: **the Sharing game**
  - Imagine a **brother and sister** have to **decide how to share two indivisible and identical presents** from their parents in the following way:
    - **First the brother suggests a split**, which can be one of three—he keeps both, she keeps both, or they each keep one.
    - Then the **sister chooses whether to accept or reject the split**.

Extensive-form games

- Example: **the Sharing game**
  - If she accepts, they each get their allocated present(s), and otherwise, neither gets any gift.
  - Assume both siblings value the two presents equally and additively

Extensive-form games



The Sharing Game

1-brother
2-sister

Extensive-form games

- **Definition (Perfect-information game):** *A (finite) perfect-information game (in extensive form) is a tuple* $G = (N, A, H, Z, \chi, \rho, \sigma, u)$*, where:*
  - N *is a* ***set of*** $\boldsymbol{n}$ ***agents****;*
  - A *is a (single)* ***set of actions****;*
  - H *is a set of* ***nonterminal choice nodes****;*
  - Z *is a set of* ***terminal nodes****, disjoint from* H*;*

Extensive-form games

- **Definition (Perfect-information game):** *A (finite) perfect-information game (in extensive form) is a tuple* $G = (N, A, H, Z, \chi, \rho, \sigma, u)$*, where:*

  - $\chi: H \longmapsto 2^A$ *is the* ***action function****, which assigns to each choice node a set of possible actions;*

  - $\rho: H \longmapsto N$ *is the* ***player function****, which assigns to each nonterminal node a player (agent)* $i \in N$ *who chooses an action at that node;*

Extensive-form games

- **Definition (Perfect-information game):** *A (finite) perfect-information game (in extensive form) is a tuple* $G = (N, A, H, Z, \chi, \rho, \sigma, u)$*, where:*

  - $\sigma: H \times A \longmapsto H \cup Z$ *is the* ***successor function****, which maps a choice node and an action to a new choice node or terminal node, such that for all* $h_1, h_2 \in H$ *and* $a_1, a_2 \in A$*, if* $\sigma(h_1, a_1) = \sigma(h_2, a_2)$ *then* $h_1 = h_2$ *and* $a_1 = a_2$*; and*

  - $u = (u_1, \dots, u_n)$*, where* $u_i: Z \longmapsto \mathbb{R}$ *is a* ***real-valued utility function*** *for player (agent)* i *on the terminal nodes* Z.

Outline

- Perfect-information games in extensive form
- **Strategies and equilibria**
- Subgame-perfect equilibrium
- Backward induction
- Example

Strategies and Equilibria

- A **pure strategy** for an agent in a perfect-information game is:
  - A **complete specification of which deterministic action to take at every node** belonging to that agent

Strategies and Equilibria

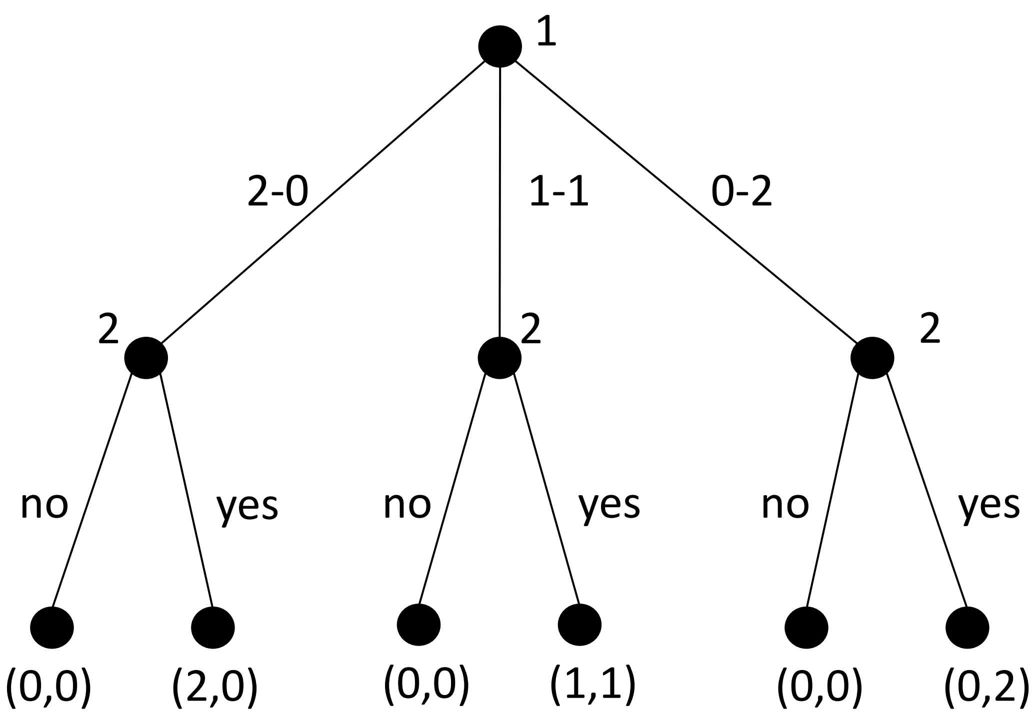
- **Definition (Pure strategies):** *Let* $G = (N, A, H, Z, \chi, \rho, \sigma, u)$ *be a perfect-information extensive-form game. Then the pure strategies of agent* i *consist of the Cartesian product*

$$\Pi_{h \in H, \rho(h)=i}\ \chi(h).$$

Action function

Strategies and Equilibria

The Sharing game



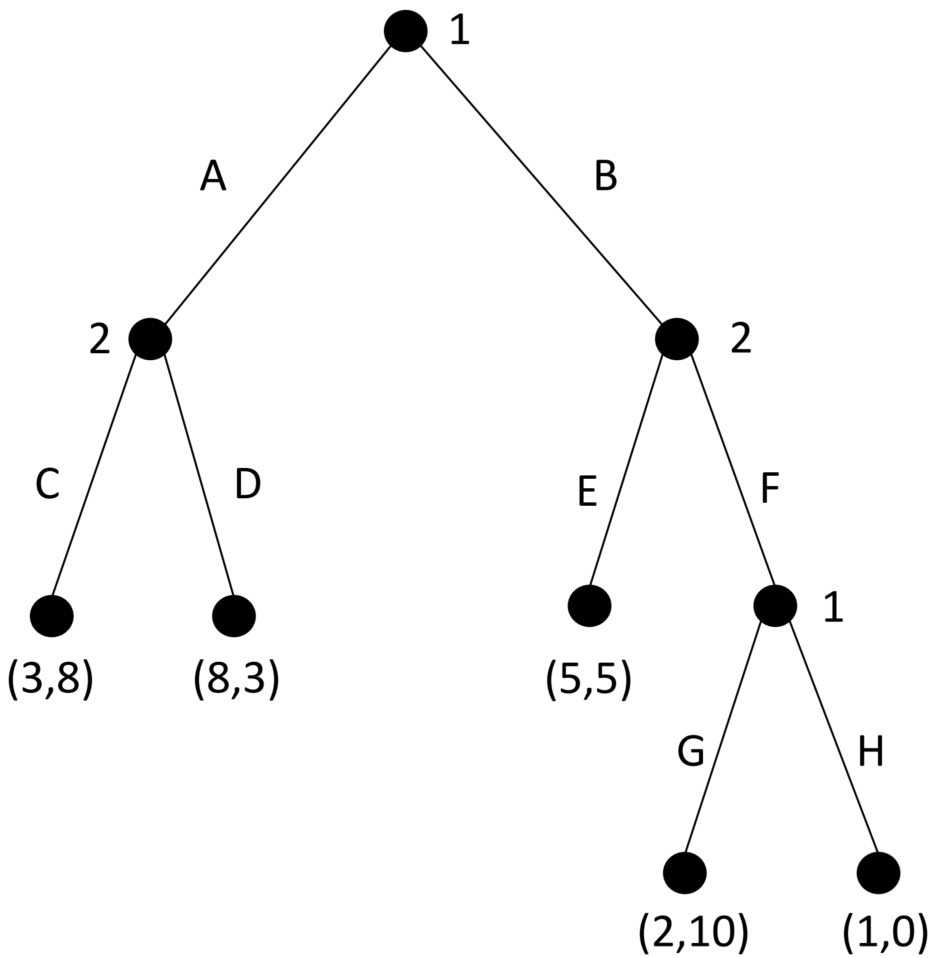
Pure Strategies:

S_1 = {2-0, 1-1, 0-2}

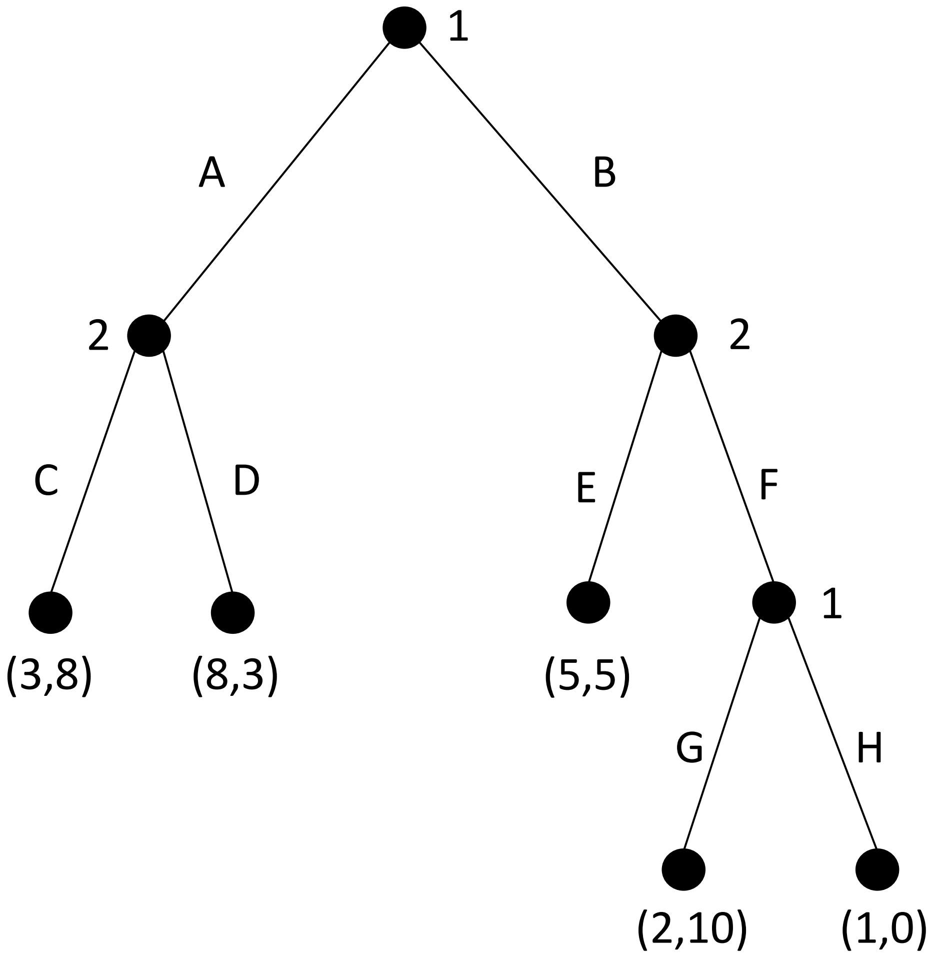
S_2 = {(*yes*,*yes*,*yes*), (*yes*,*yes*,*no*), (*yes*,*no*,*yes*), (*yes*,*no*,*no*), (*no*,*yes*,*yes*), (*no*, *yes*, *no*), (*no*, *no*, *yes*), (*no*, *no*, *no*)}

Strategies and Equilibria

Let us now consider another game



Strategies and Equilibria



Pure Strategies:

S_1 = {(A,G), (A,H), (B,G), (B,H)}

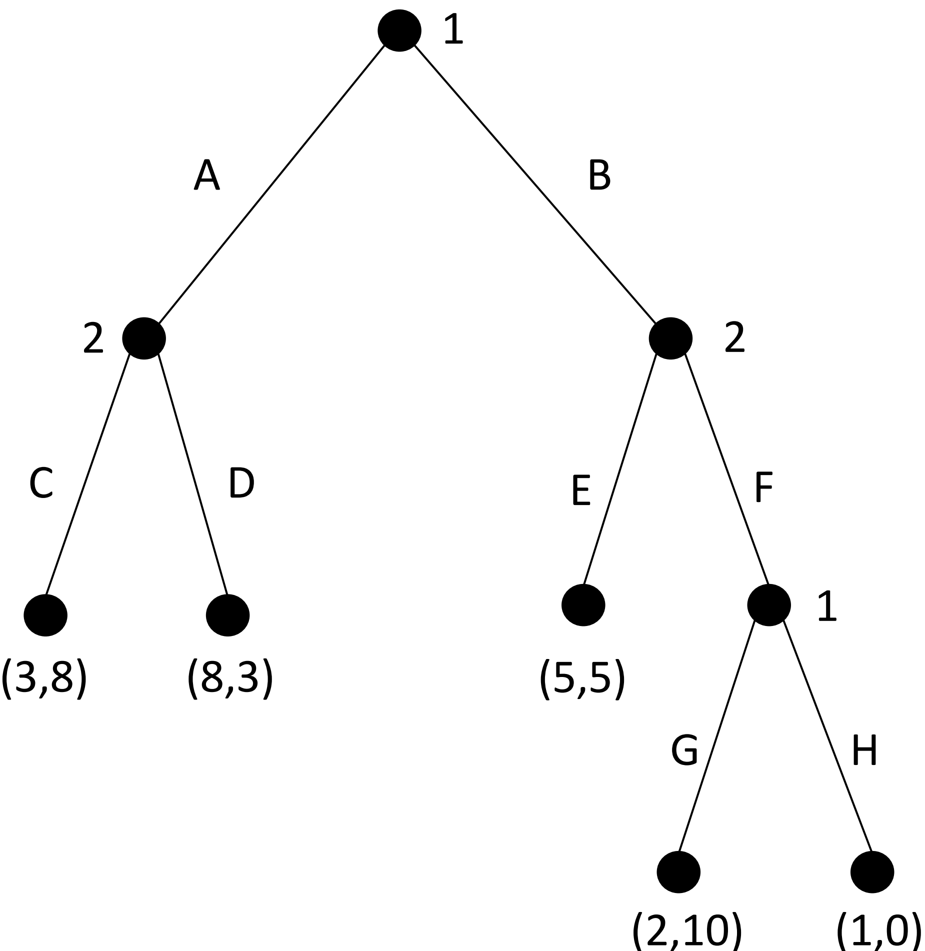
S_2 = {(C,E), (C,F), (D,E), (D,F)}

Strategies and Equilibria

- **So how do we compute the Nash equilibria?**
  - The definition of best response is the same as we’ve seen so far!
  - The definition of Nash equilibria is the same too!

Strategies and Equilibria

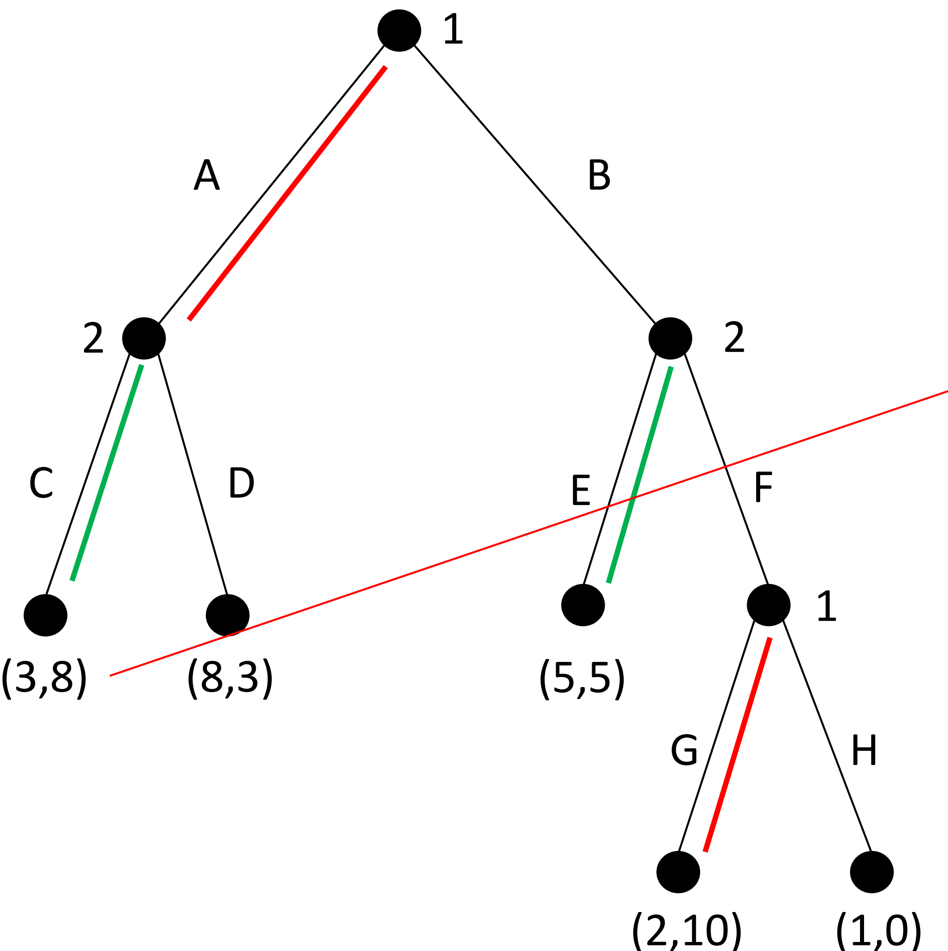
- We convert a perfect-information game to an equivalent normal-form game



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)				
(A,H)				
(B,G)				
(B,H)				

Strategies and Equilibria

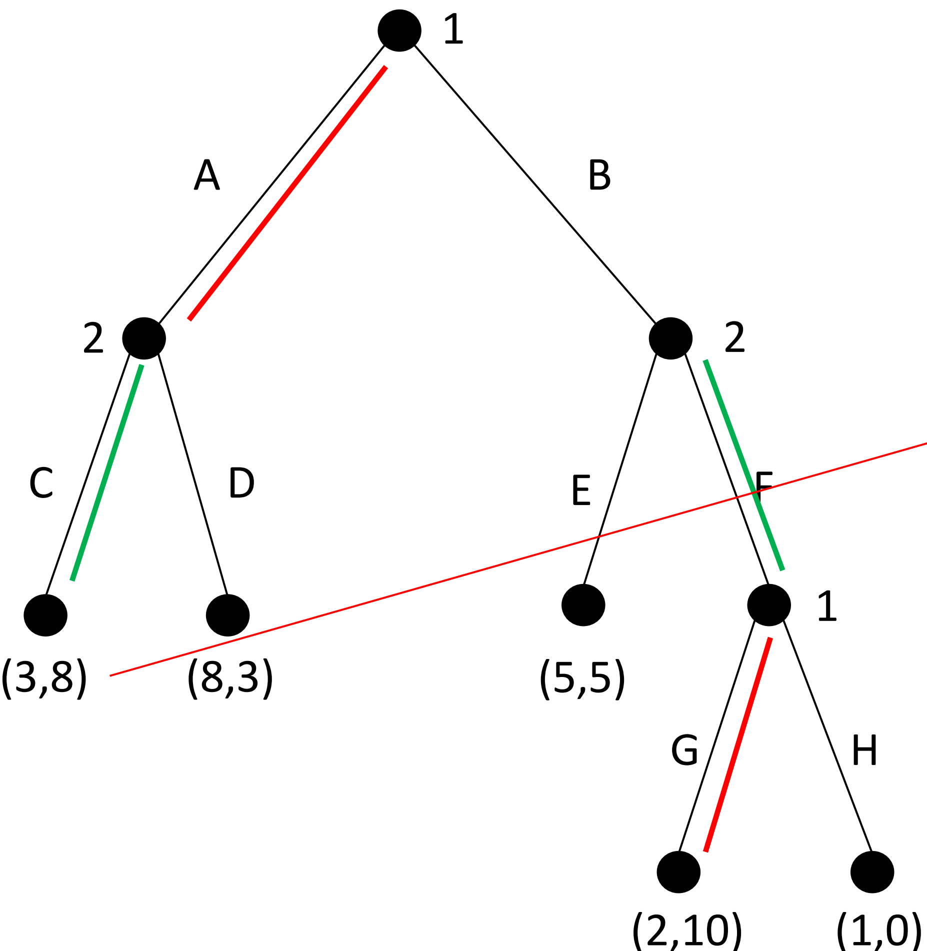
- We convert a perfect-information game to an equivalent normal-form game



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8			
(A,H)				
(B,G)				
(B,H)				

Strategies and Equilibria

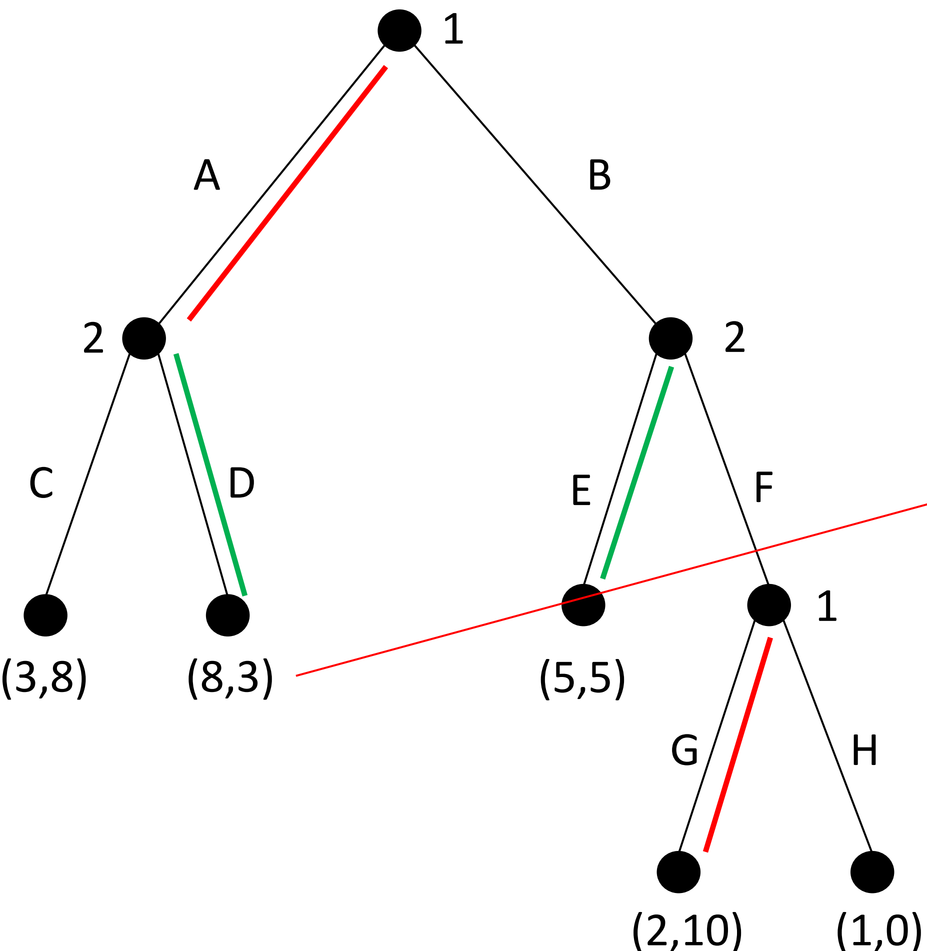
- We convert a perfect-information game to an equivalent normal-form game



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8		
(A,H)				
(B,G)				
(B,H)				

Strategies and Equilibria

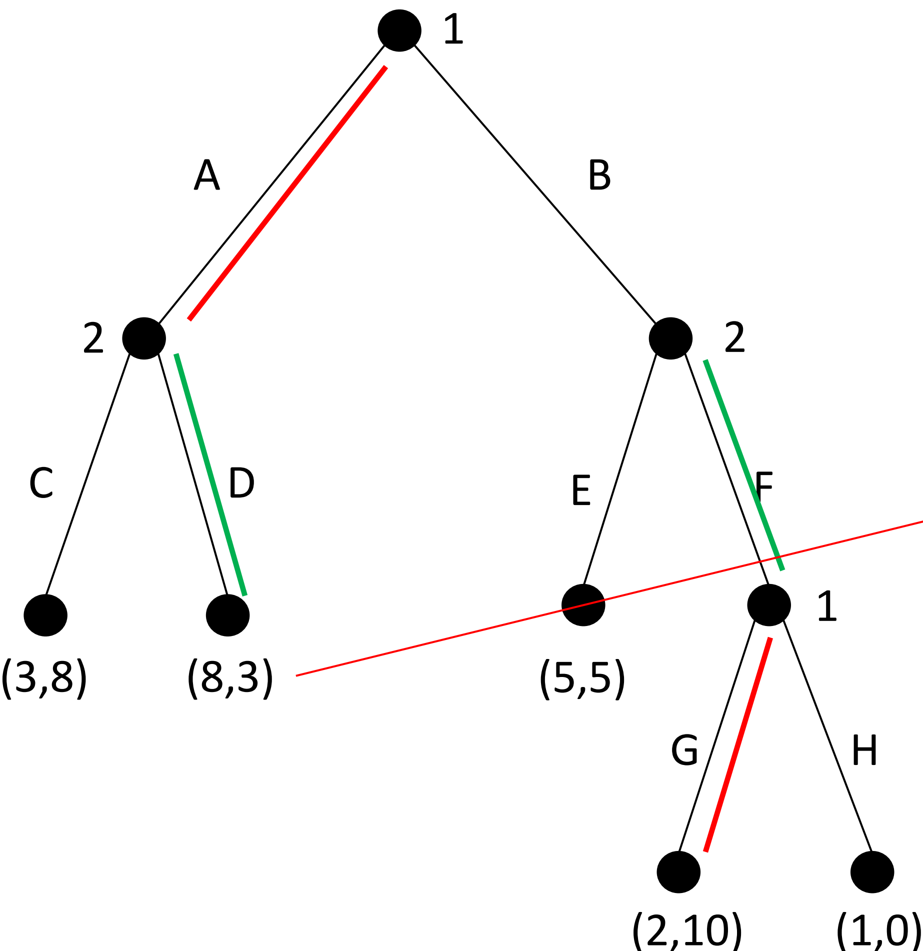
- We convert a perfect-information game to an equivalent normal-form game



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8	8, 3	
(A,H)				
(B,G)				
(B,H)				

Strategies and Equilibria

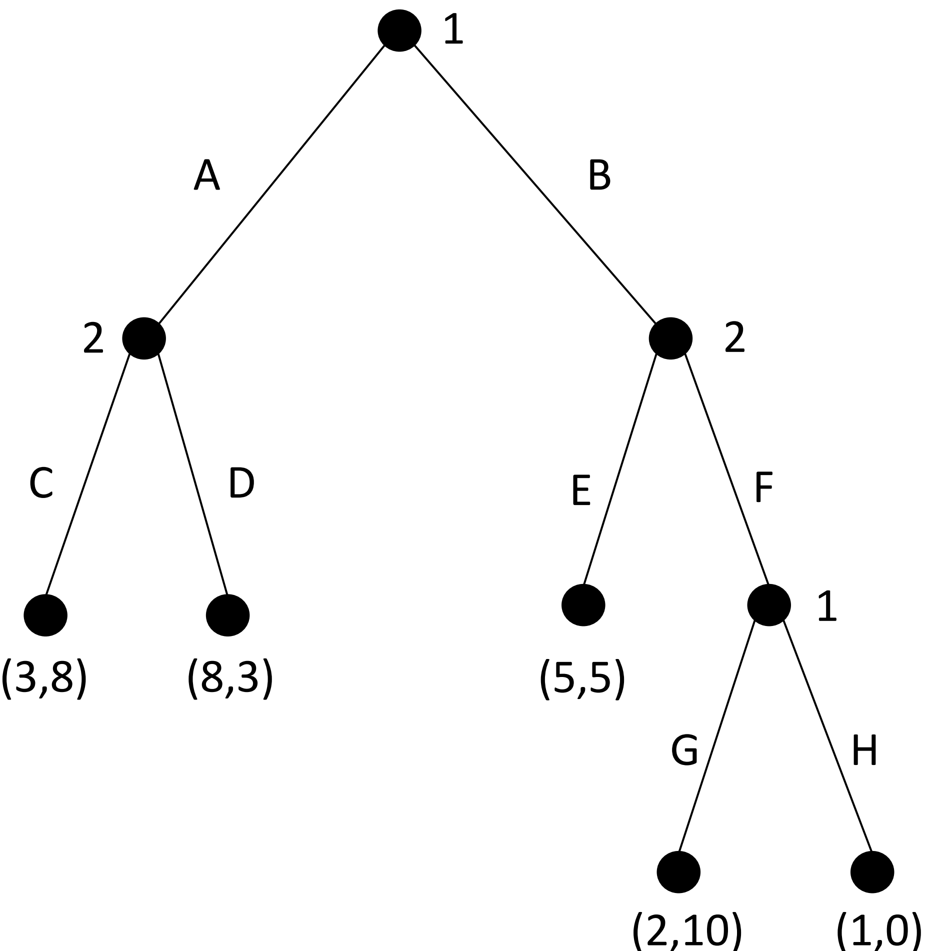
- We convert a perfect-information game to an equivalent normal-form game



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8	8, 3	8, 3
(A,H)				
(B,G)				
(B,H)				

Strategies and Equilibria

- We convert a perfect-information game to an equivalent normal-form game



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8	8, 3	8, 3
(A,H)	3, 8	3, 8	8, 3	8, 3
(B,G)	5, 5	2, 10	5, 5	2, 10
(B,H)	5, 5	1, 0	5, 5	1, 0

Strategies and Equilibria

- The **temporal structure** of the extensive-form game can generate a certain **redundancy** within the normal-form game

- In the previous example:

  - Normal form – 16 outcomes

  - Extensive form – 5 outcomes

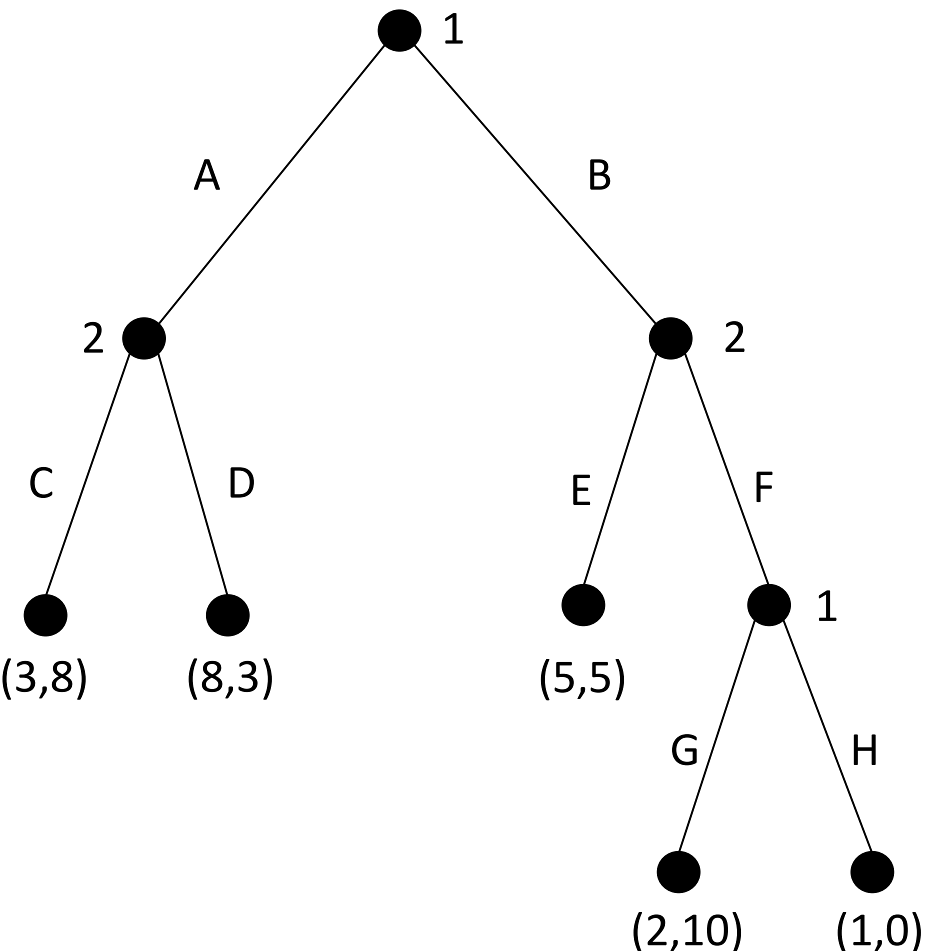
Strategies and Equilibria

- **Theorem**: *Every (finite) perfect-information game in extensive form has a pure-strategy Nash equilibrium.*

- Intuition:
  - Since agents take turns, and everyone gets to see everything that happened thus far before making a move, it is never necessary to introduce randomness into action selection in order to find an equilibrium.

Strategies and Equilibria

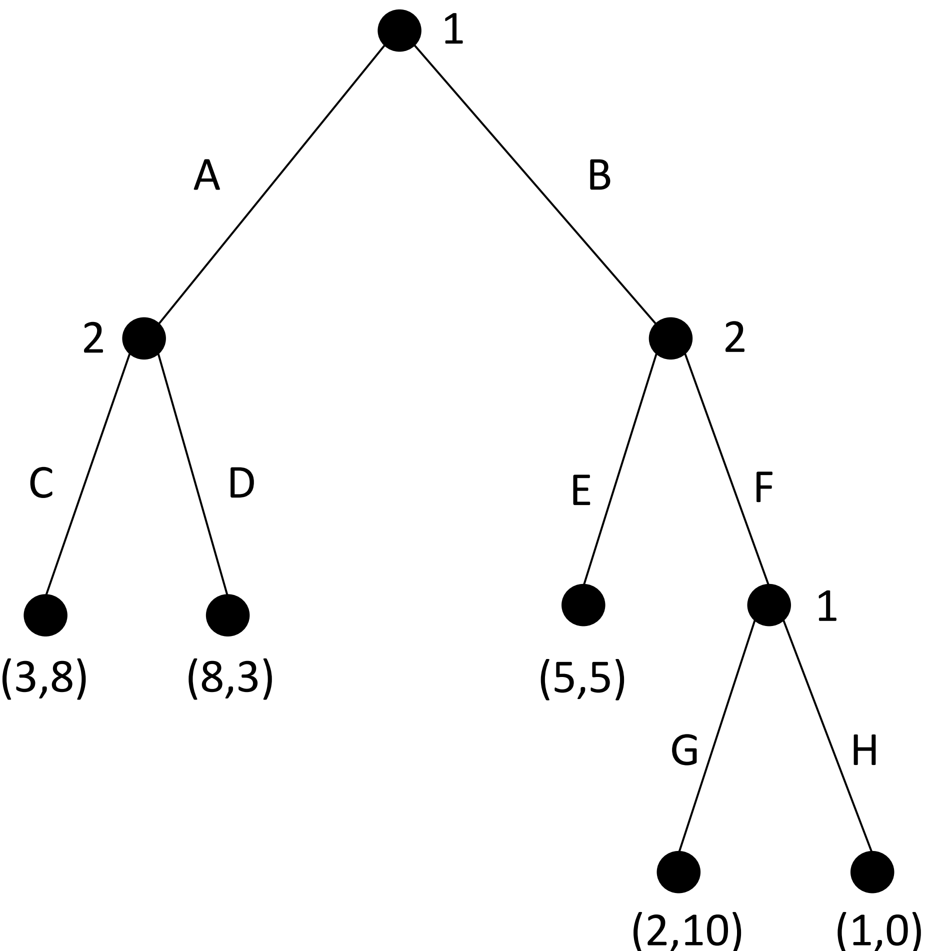
- Find the Nash equilibria?



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8	8, 3	8, 3
(A,H)	3, 8	3, 8	8, 3	8, 3
(B,G)	5, 5	2, 10	5, 5	2, 10
(B,H)	5, 5	1, 0	5, 5	1, 0

Strategies and Equilibria

- We convert a perfect-information game to an equivalent normal-form game



	(C,E)	**(C,F)**	**(D,E)**	**(D,F)**
(A,G)	3, 8	3, 8	8, 3	8, 3
(A,H)	3, 8	3, 8	8, 3	8, 3
(B,G)	5, 5	2, 10	5, 5	2, 10
(B,H)	5, 5	1, 0	5, 5	1, 0

Nash equilibria in this game:
{(*A*, *G*), (*C*, *F*)}, {(*A*, *H*), (*C*, *F*)},
and {(*B*, *H*), (*C*, *E*)}

Exercise

- A market has a single firm (monopoly) with \$2M in cash

- A new firm is deciding whether to enter (or not) the market
  - The new firm has \$2M to start the operation

- If the new firm enters, the monopolist may accept the decision or declare a price war

  - If the monopolist accepts, the new firm makes a \$1M profit and the monopolist loses \$1M

  - A price war is unprofitable for both firms (i.e., they lose all they have)

Exercise

- Model this game as a perfect-information extensive-form game? Present the game tree.
- Convert a perfect-information game to an equivalent normal-form game
- Find the Nash equilibria
- Analyze the equilibria and see if there is anything strange

Outline

- Perfect-information games in extensive form
- Strategies and equilibria
- **Subgame-perfect equilibrium**
- Backward induction
- Example

Subgame-Perfect Equilibrium

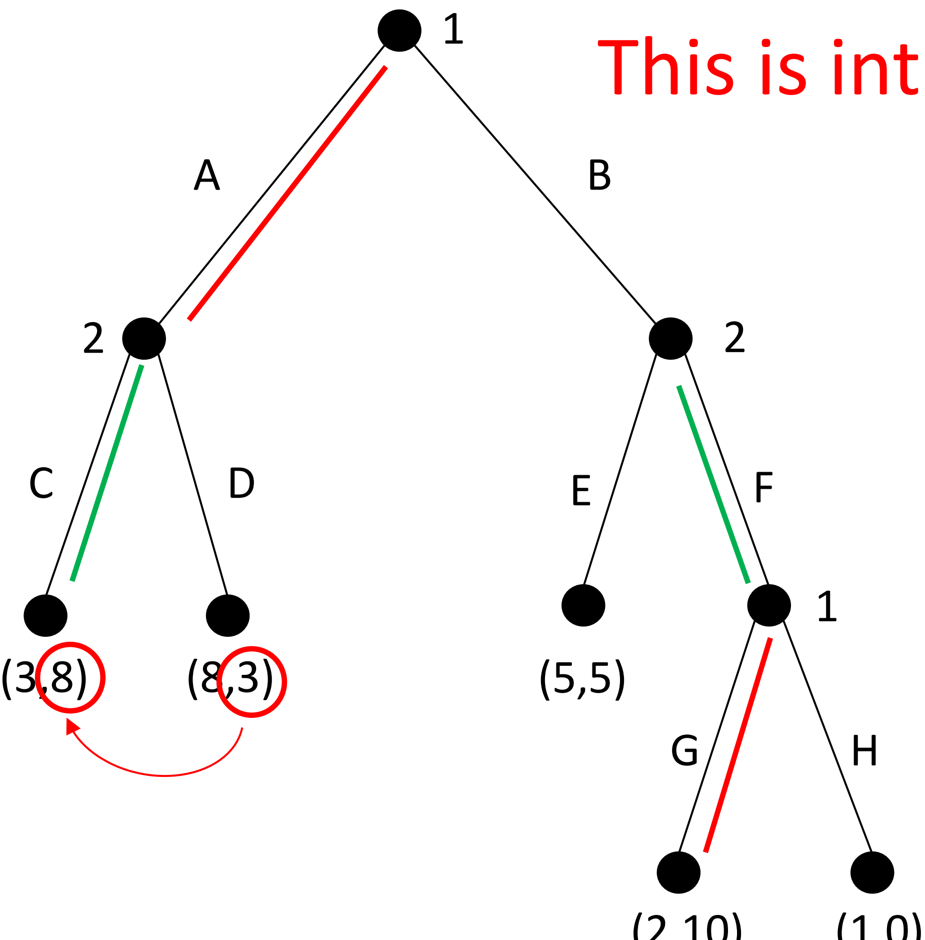
- Examining the converted normal-form game of an extensive-form game **obscures the game's temporal nature**

- Let us see this with our previous example

  - Let us focus on the equilibria $\{(A, G), (C, F)\}$ and $\{(B, H), (C, E)\}$

Subgame-Perfect Equilibrium

- Agent 2 has **no incentive to deviate** by **changing *C* to *D***!



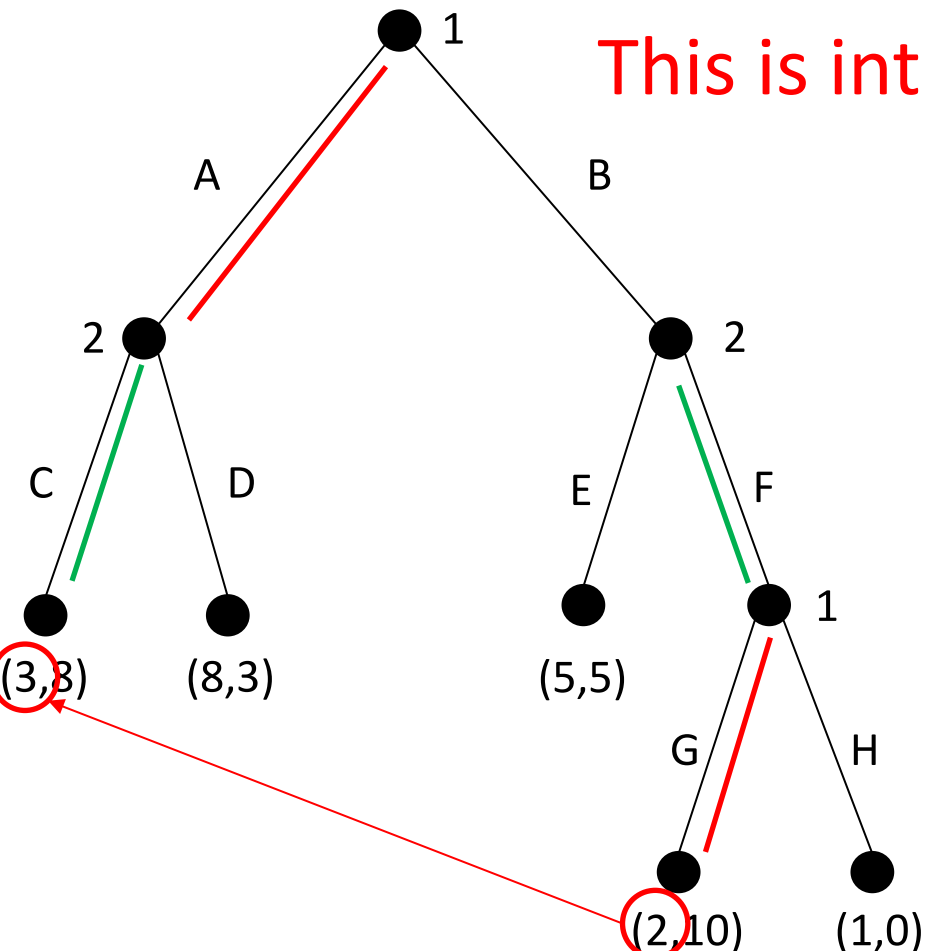
This is intuitive!

	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8	8, 3	8, 3
(A,H)	3, 8	3, 8	8, 3	8, 3
(B,G)	5, 5	2, 10	5, 5	2, 10
(B,H)	5, 5	1, 0	5, 5	1, 0

Subgame-Perfect Equilibrium

- Agent 1 has **no incentive to deviate** by **changing *A* to *B***!

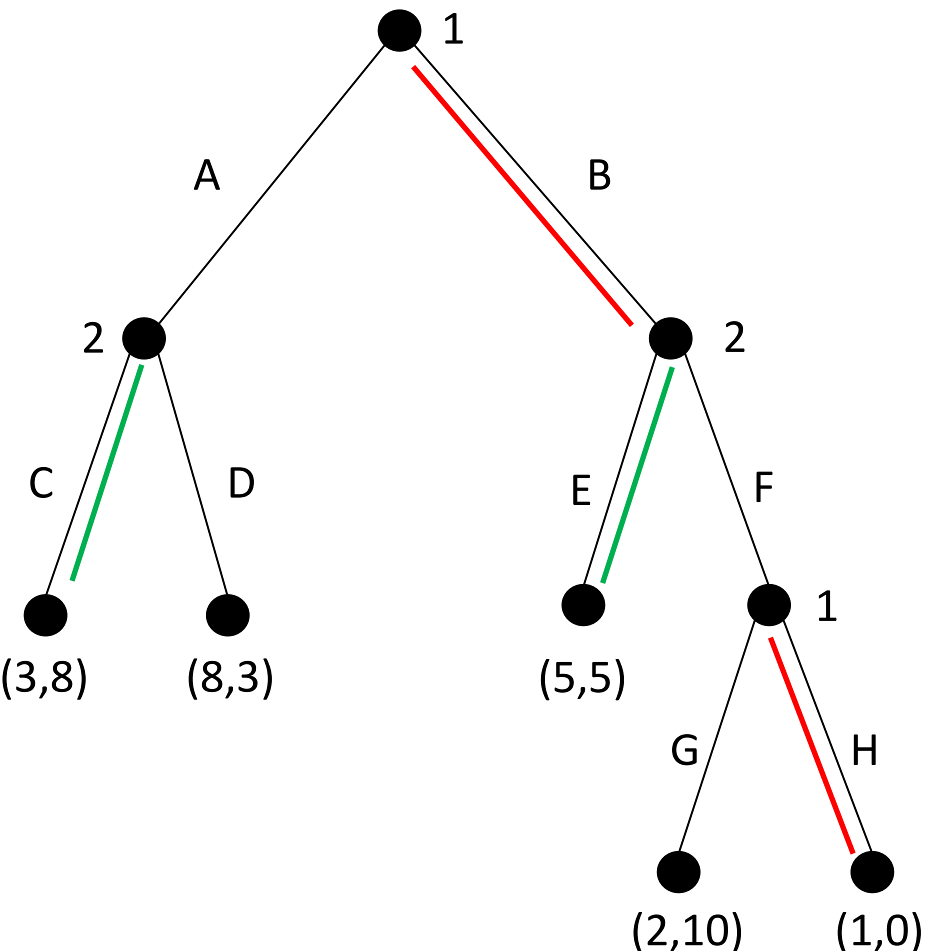
This is intuitive!



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8	8, 3	8, 3
(A,H)	3, 8	3, 8	8, 3	8, 3
(B,G)	5, 5	2, 10	5, 5	2, 10
(B,H)	5, 5	1, 0	5, 5	1, 0

Subgame-Perfect Equilibrium

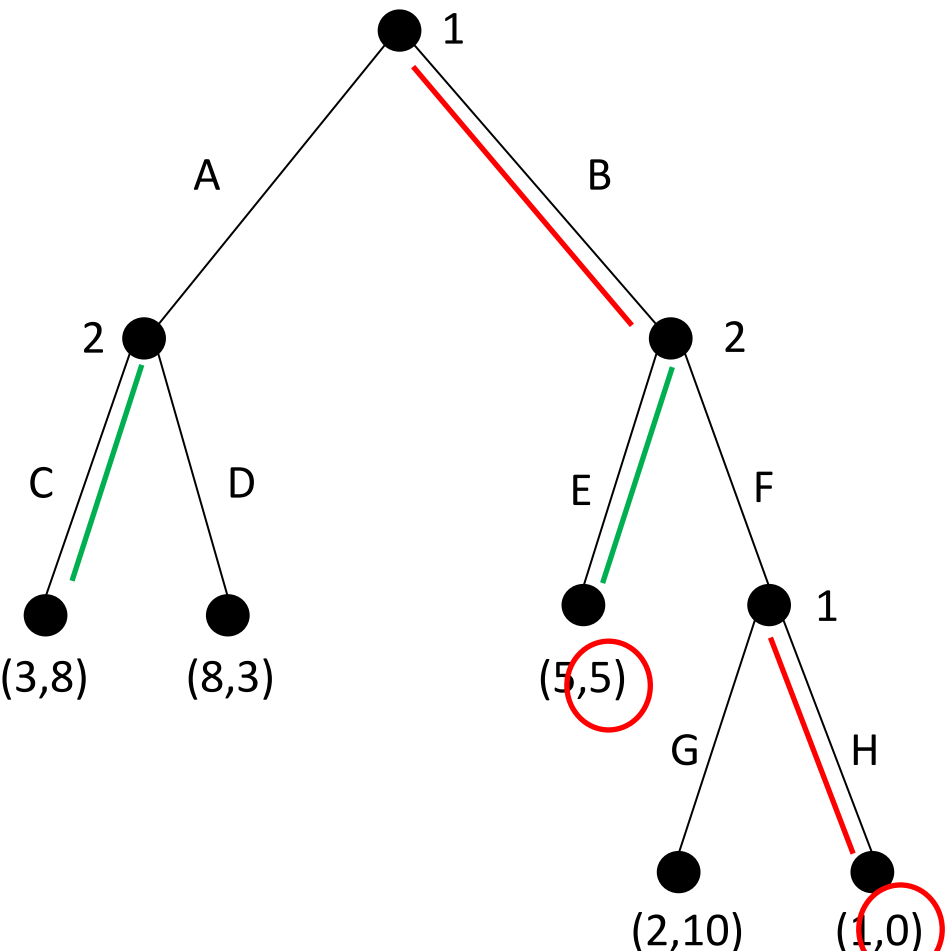
- This is also a Nash equilibrium. However…



	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8	8, 3	8, 3
(A,H)	3, 8	3, 8	8, 3	8, 3
(B,G)	5, 5	2, 10	5, 5	2, 10
(B,H)	5, 5	1, 0	5, 5	1, 0

Subgame-Perfect Equilibrium

- This is also a Nash equilibrium. However…

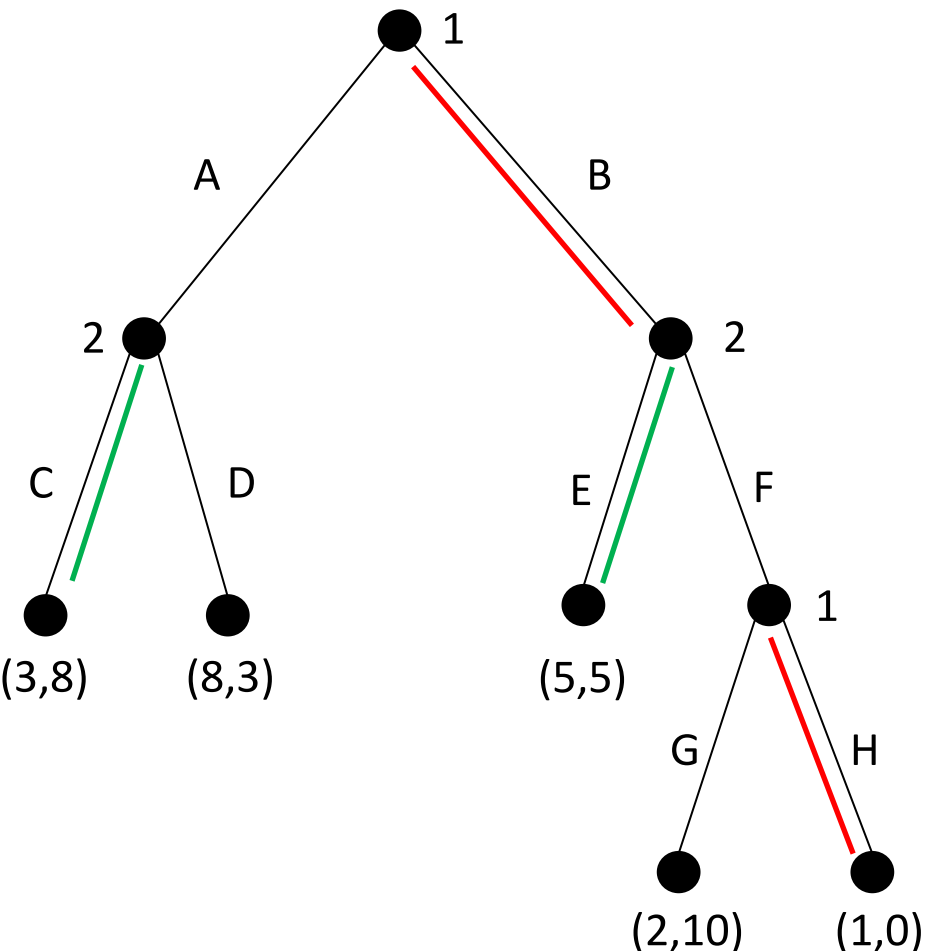


Agent 2 chooses action *E* because he knows that agent 1 would choose *H* afterwards

The behavior of agent 1 is called a *threat*

Subgame-Perfect Equilibrium

- This is also a Nash equilibrium. However…



What if agent 2 does not consider agent 1's *threat* to be credible?

If agent 2 played *F*, would agent 1 really follow through on his threat? NO

Subgame-Perfect Equilibrium

- In other words, there is something *intuitively wrong* with the equilibrium (B,H),(C,E)

- To formally capture the reason, let us define the notion of subgame.

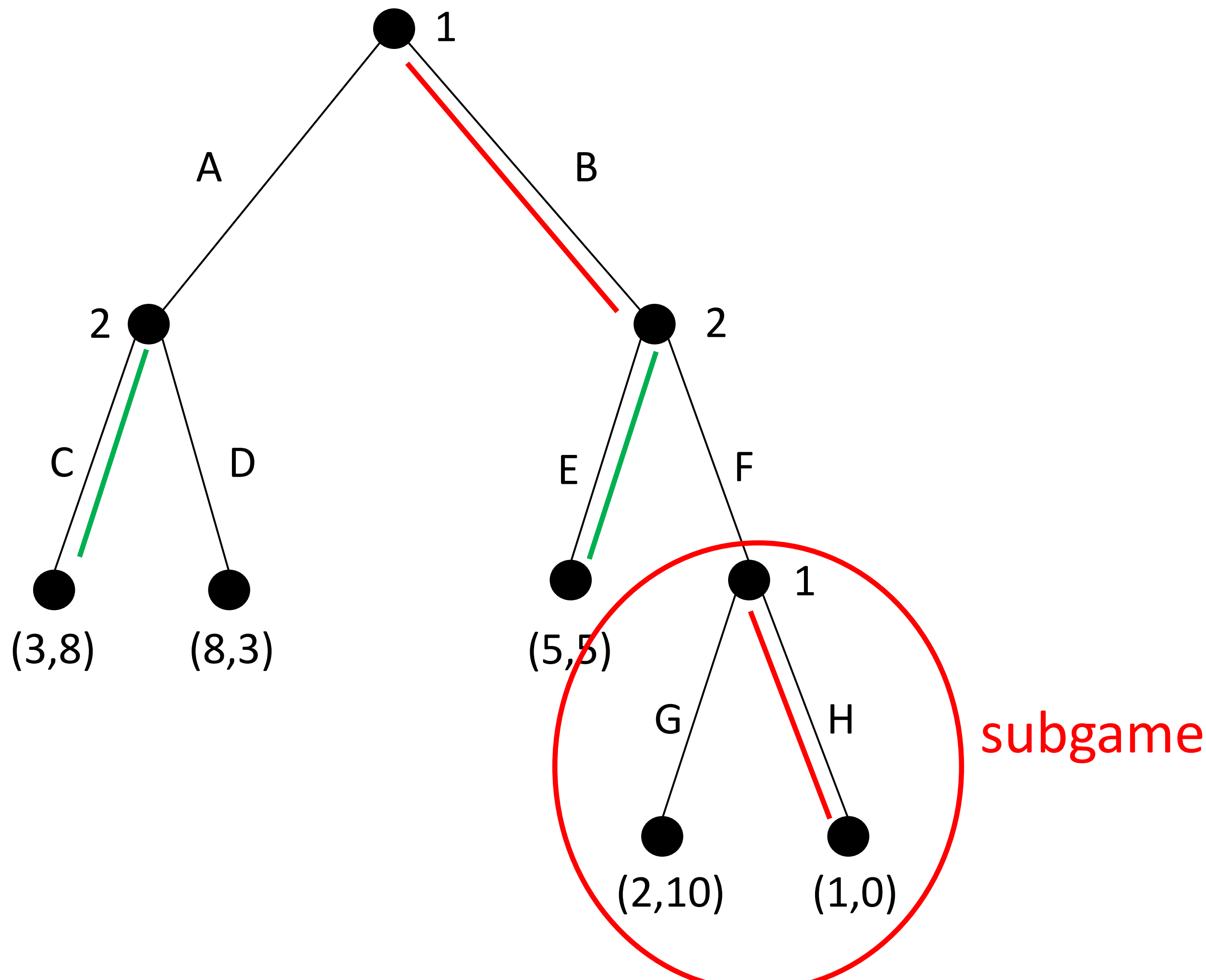
Subgame-Perfect Equilibrium

- **Definition (Subgame of *G* rooted at *h*)**: *Given a perfect-information extensive-form game G, the subgame of G rooted at node h is the restriction of G to the descendants of h.*

- **Definition (Subgames of *G*):** *The set of subgames of G consists of all of subgames of G rooted at some node in G.*

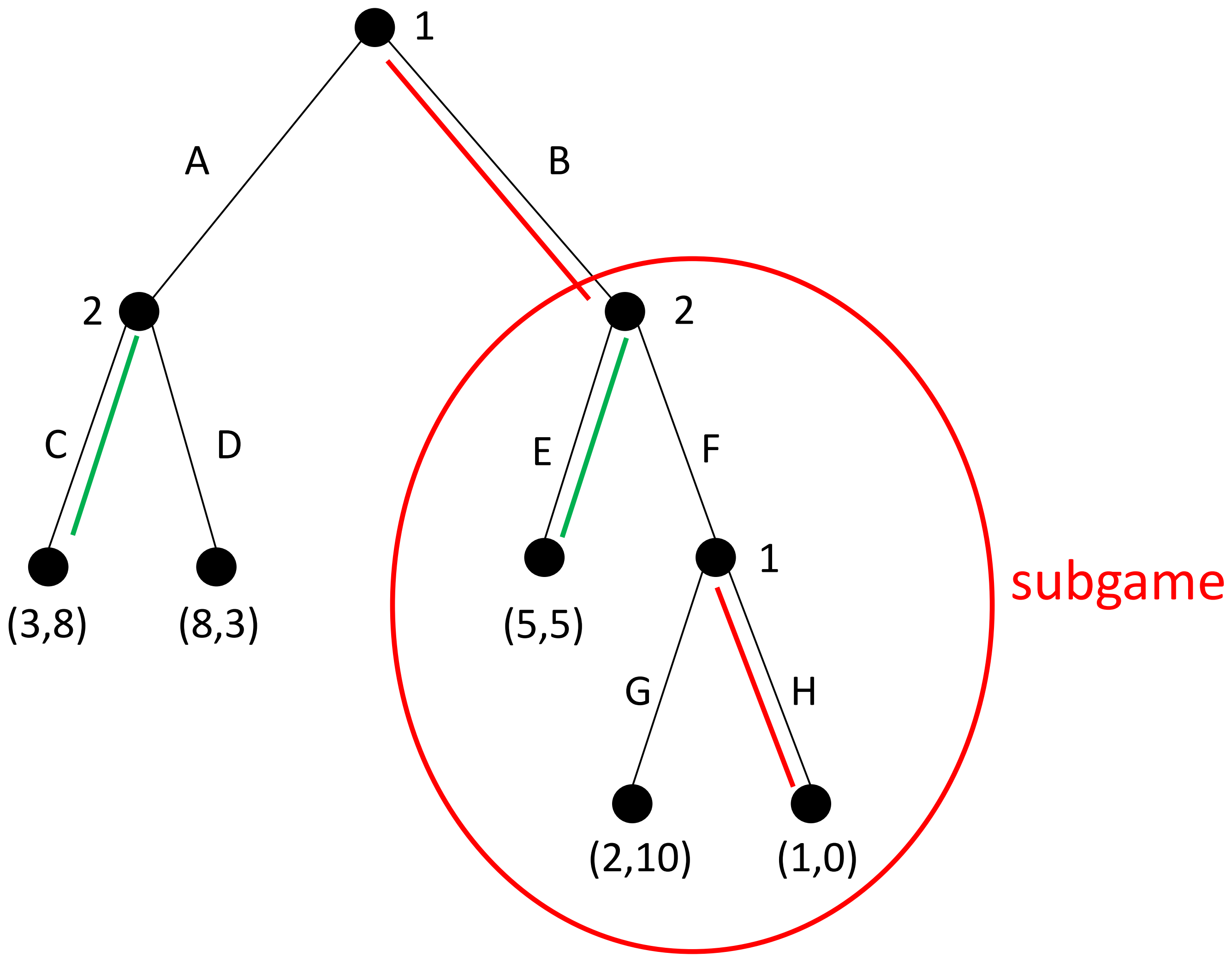
Subgame-Perfect Equilibrium

- For example



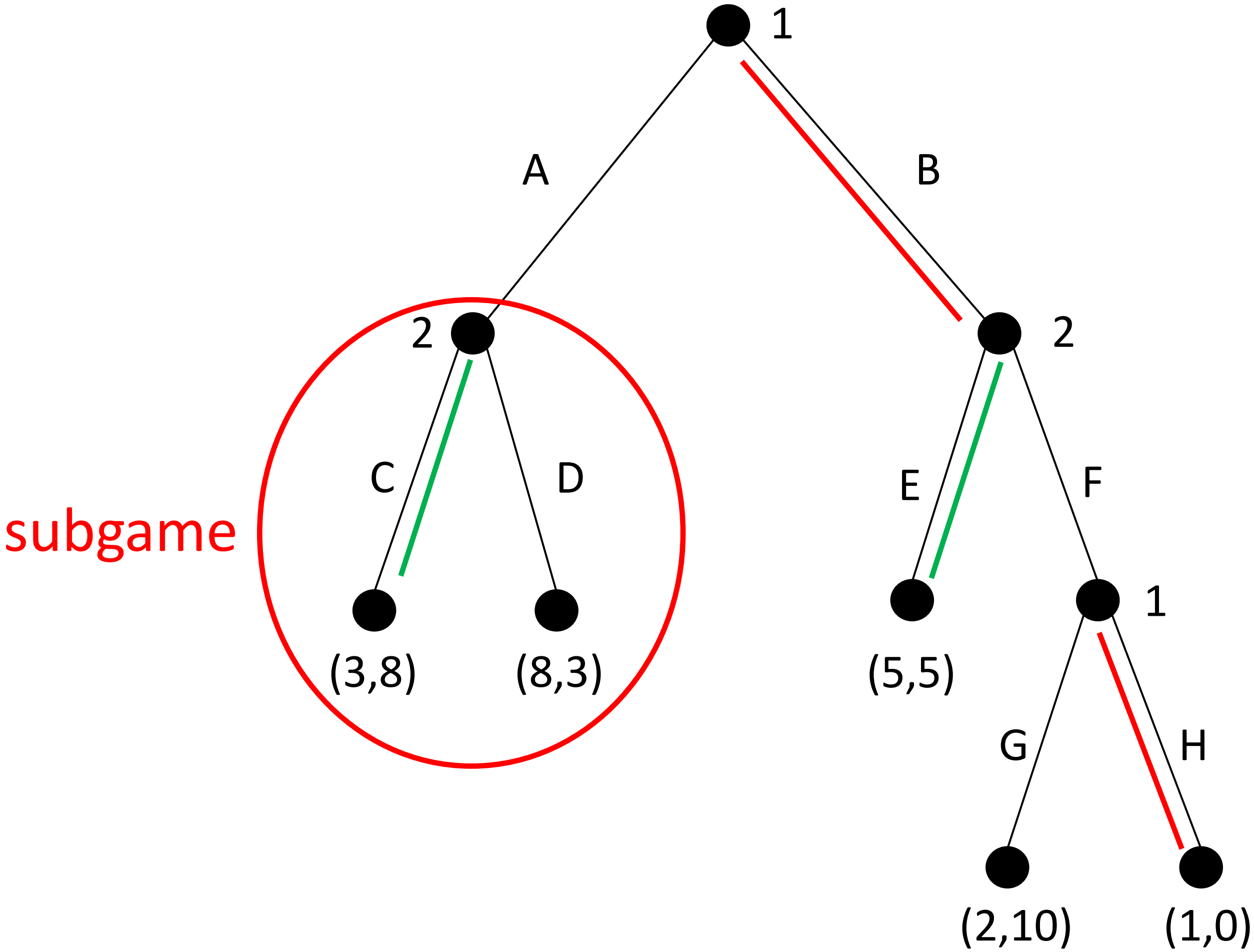
Subgame-Perfect Equilibrium

- For example



Subgame-Perfect Equilibrium

- For example

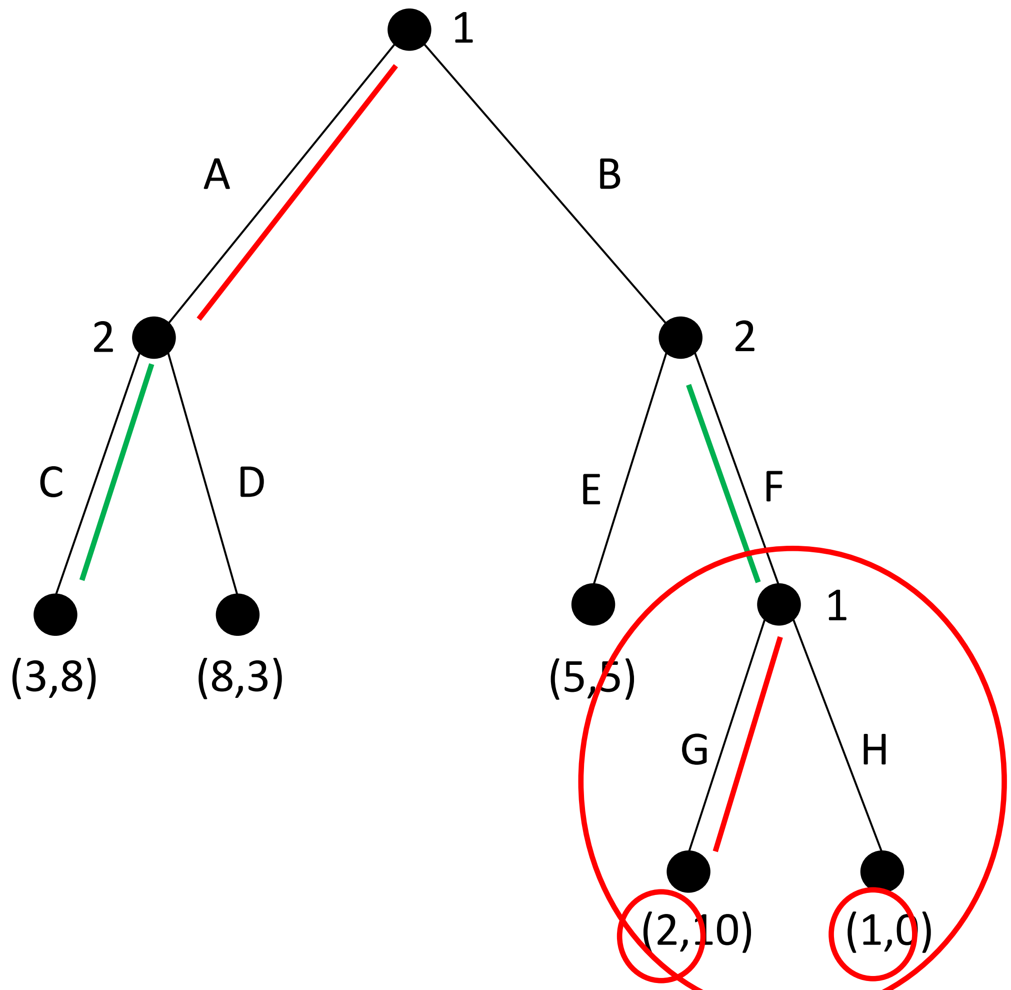


Subgame-Perfect Equilibrium

- **Definition (Subgame-perfect equilibrium):** *The subgame-perfect equilibria (SPE) of a game* G *are all strategy profiles* s *such that for any subgame* G' *of* G*, the restriction of* s *to* G' *is a Nash equilibrium of* G'*.*

Subgame-Perfect Equilibrium

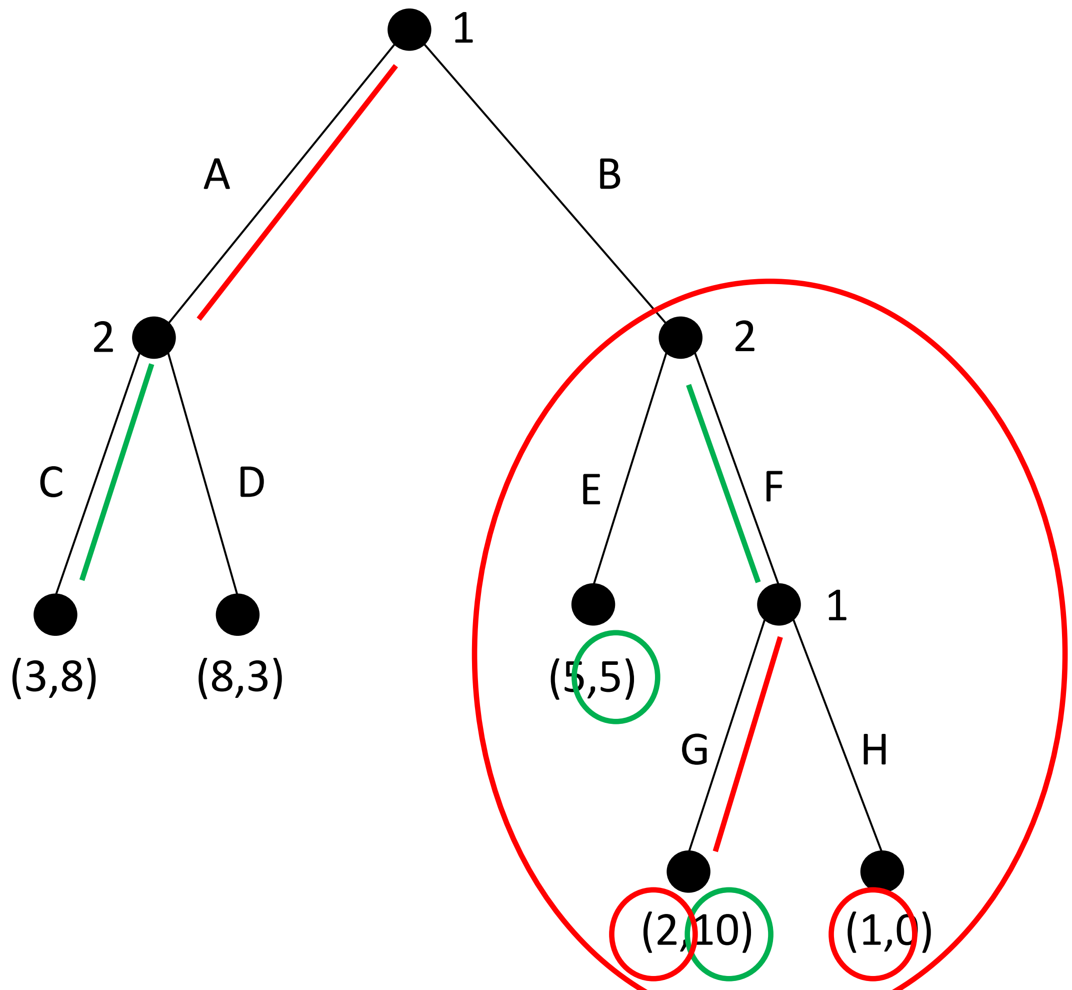
- Consider the NE {(*A*,*G*),(*C*,*F*)}, is it a **subgame-perfect equilibrium**?



Is this subgame a NE? YES

Subgame-Perfect Equilibrium

- Consider the NE {(*A*,*G*),(*C*,*F*)}, is it a **subgame-perfect equilibrium**?

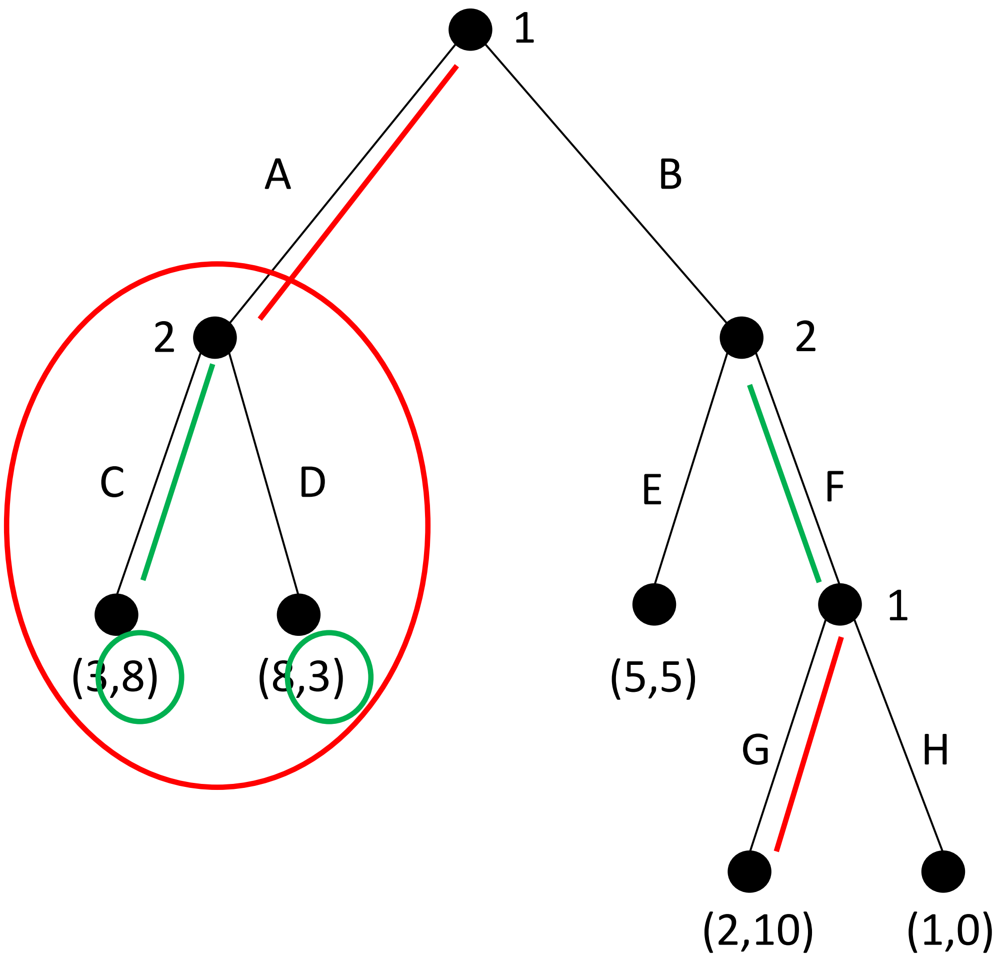


Is this subgame a NE? YES

Subgame-Perfect Equilibrium

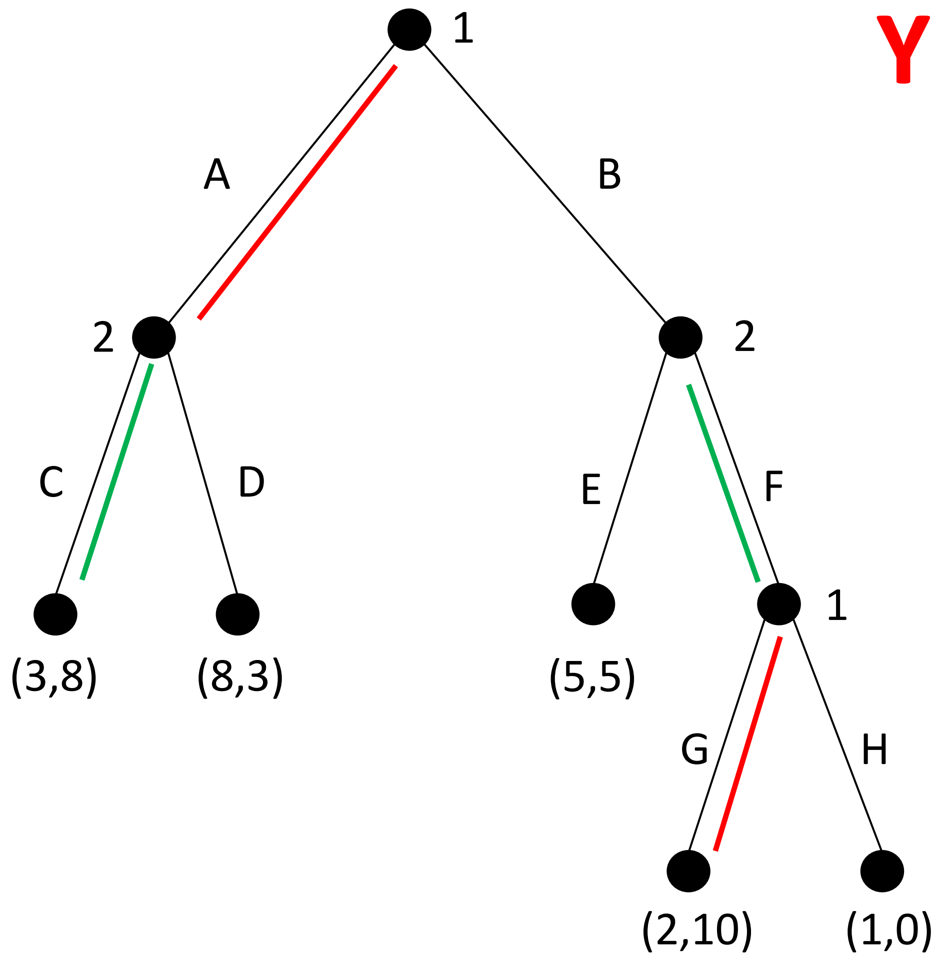
- Consider the NE {(*A*,*G*),(*C*,*F*)}, is it a **subgame-perfect equilibrium**?

Is this subgame a NE? YES



Subgame-Perfect Equilibrium

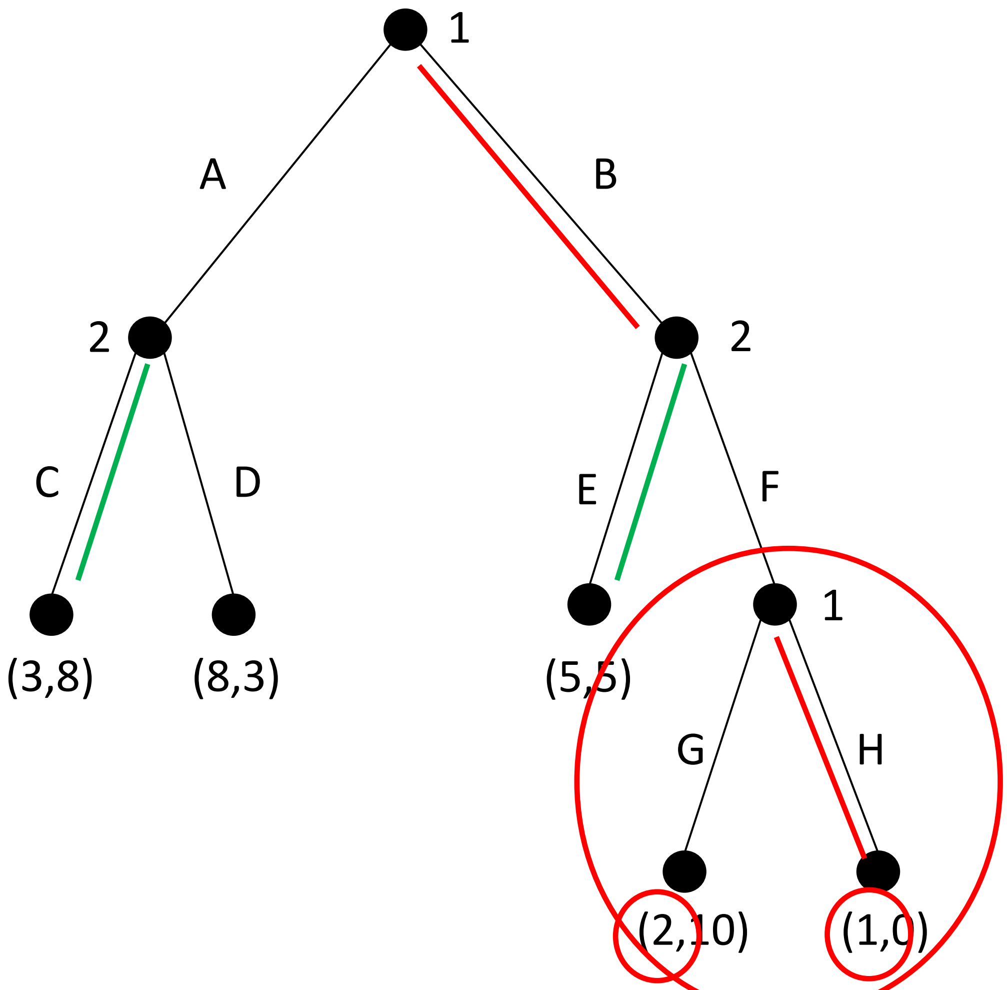
- Consider the NE {(*A*,*G*),(*C*,*F*)}, is it a **subgame-perfect equilibrium**?



YES

Subgame-Perfect Equilibrium

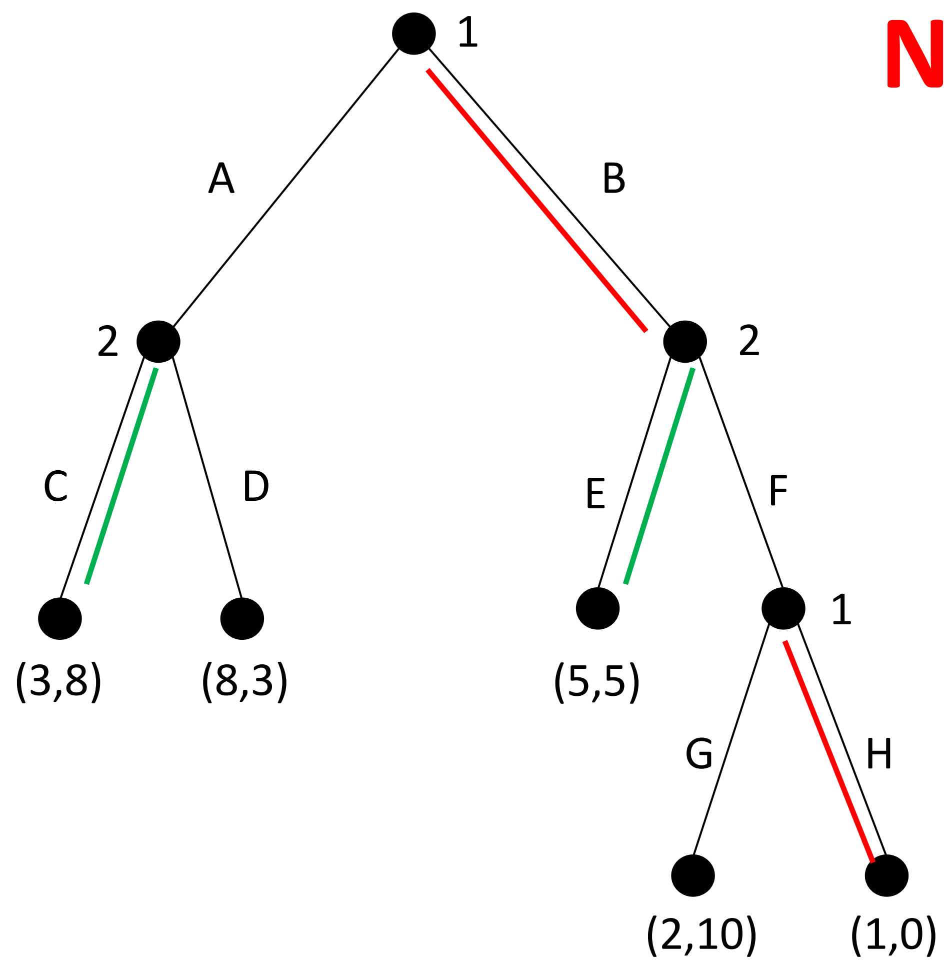
- Consider the NE {(*B,H*),(*C,E*)}, is it a **subgame-perfect equilibrium**?



Is this subgame a NE? NO

Subgame-Perfect Equilibrium

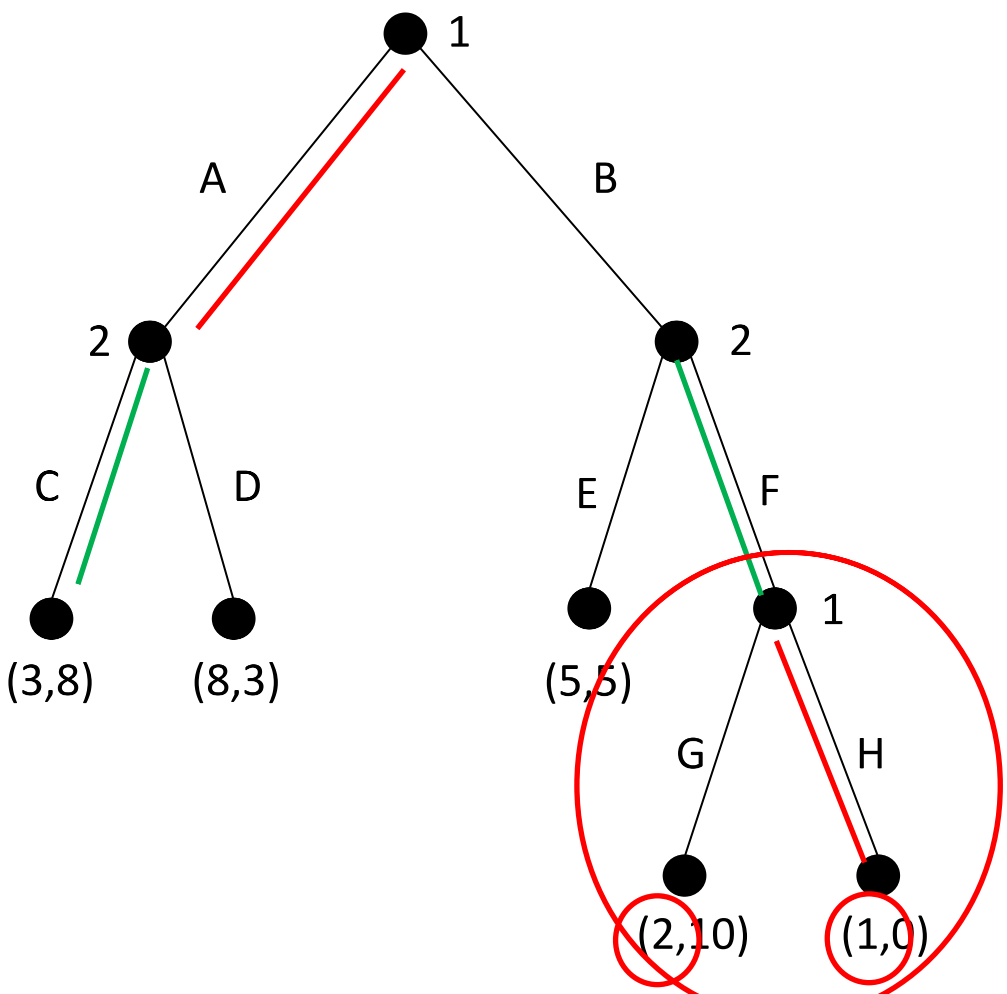
- Consider the NE {(*B,H*),(*C,E*)}, is it a **subgame-perfect equilibrium**?



NO

Subgame-Perfect Equilibrium

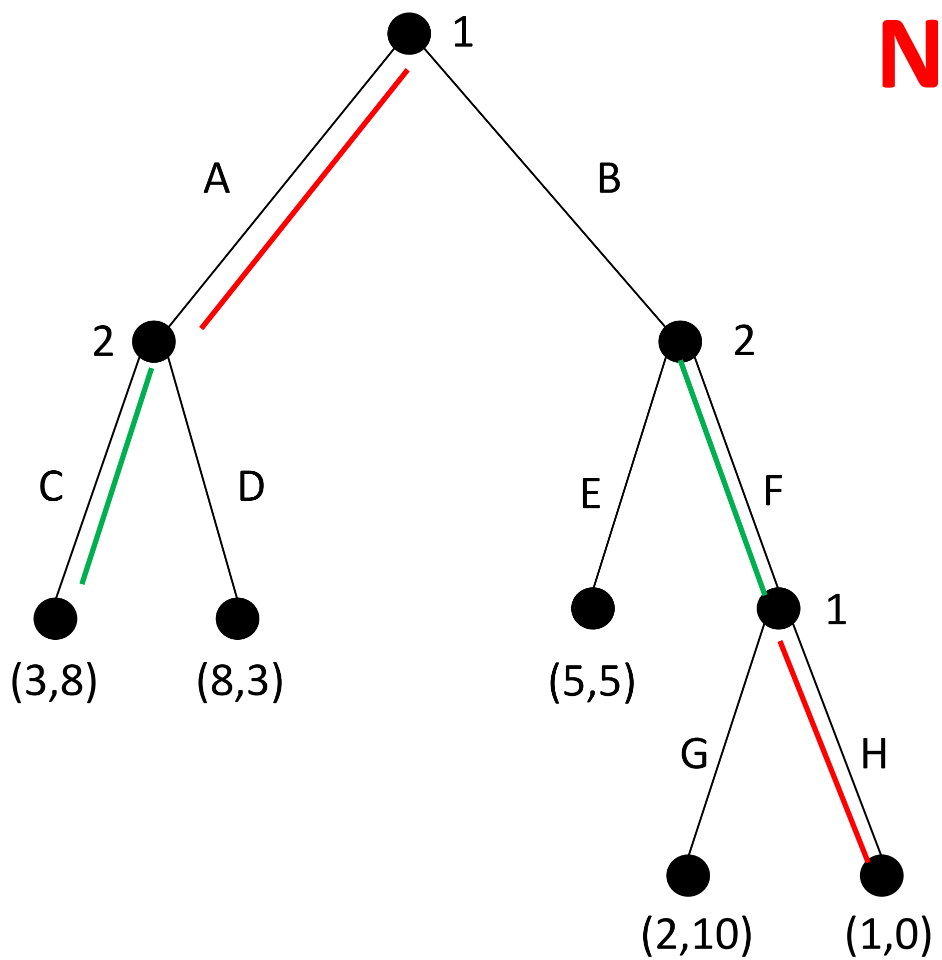
- Consider the NE {(*A*,*H*),(*C*,*F*)}, is it a **subgame-perfect equilibrium**?



Is this subgame a NE? NO

Subgame-Perfect Equilibrium

- Consider the NE {(*A*,*H*),(*C*,*F*)}, is it a **subgame-perfect equilibrium**?



NO

Subgame-Perfect Equilibrium

- Since G is its own subgame, **every subgame-perfect equilibrium is also a Nash equilibrium**

- Subgame-perfect equilibrium is a **stronger concept** than Nash equilibrium
  - Every SPE is a NE
  - Not every NE is a SPE

- Every perfect-information extensive-form game has **at least one subgame-perfect equilibrium**

Outline

- Perfect-information games in extensive form
- Strategies and equilibria
- Subgame-perfect equilibrium
- **Backward induction**
- Example

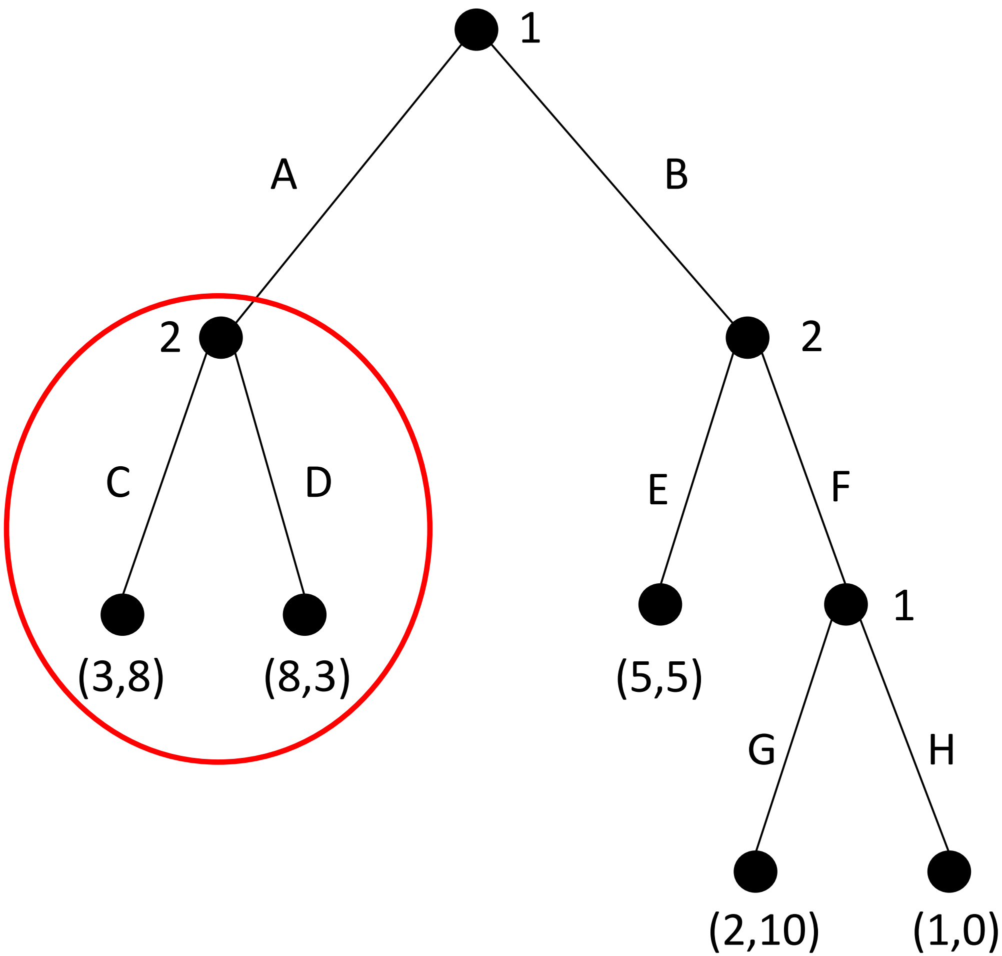
Backward Induction

- **Backward induction** is a procedure to compute the **subgame-perfect equilibrium**
- General idea:
  - Identify the equilibria in the “bottom-most” subgame trees
  - Consider that these equilibria will be played
  - Back up and consider increasingly larger trees

Backward Induction

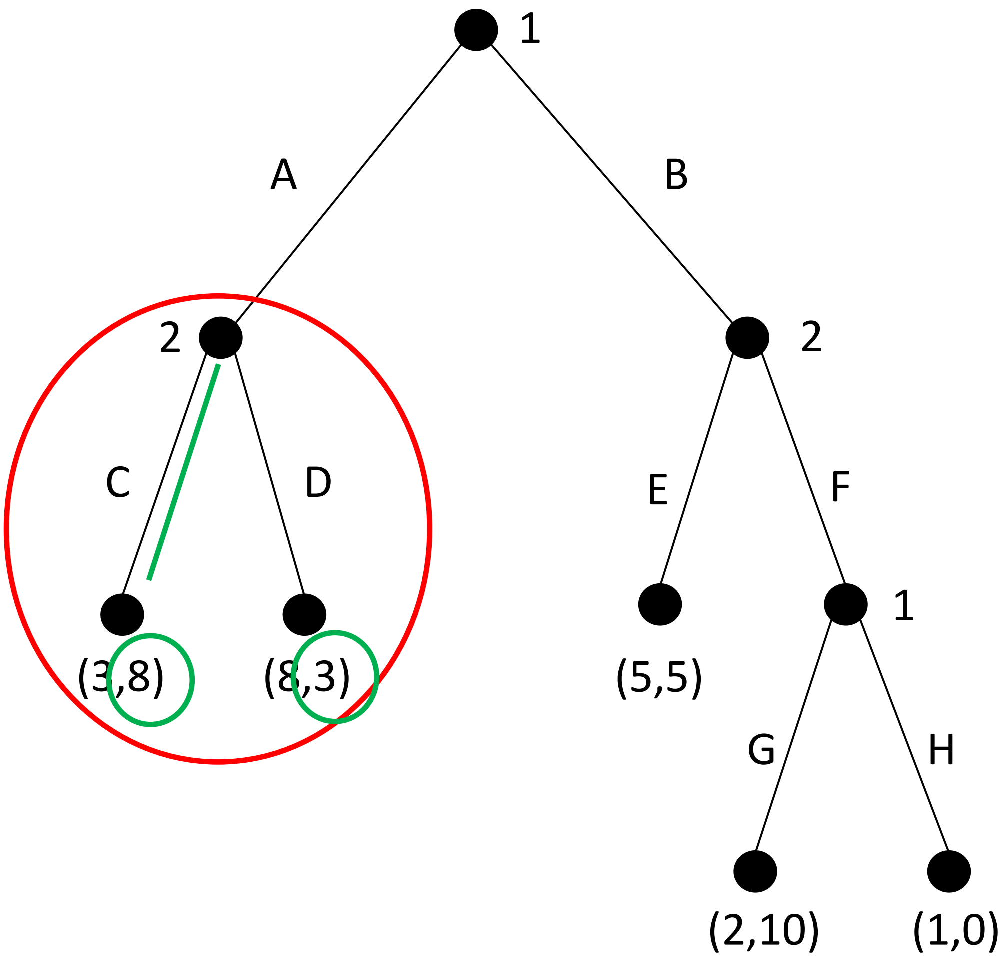
- Consider the extensive-form game below. **Let us use backward induction.**

What is the NE?



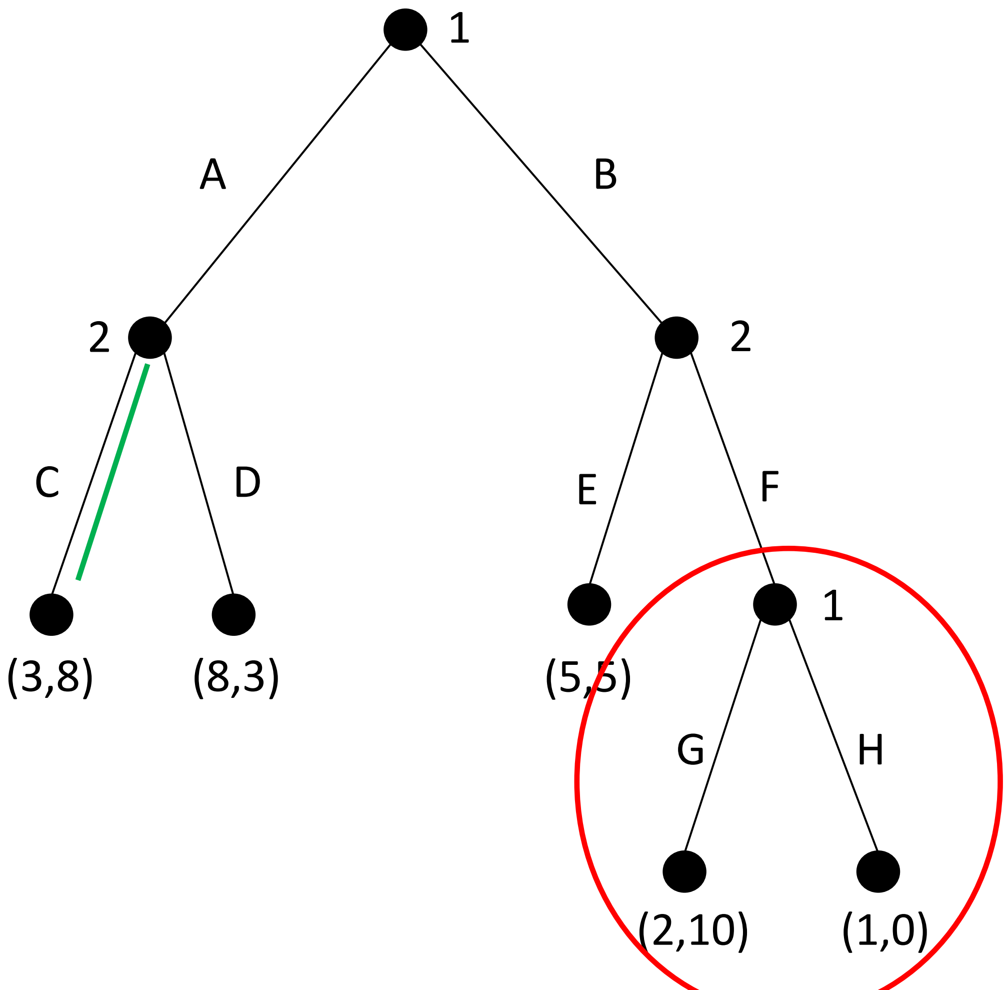
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



Backward Induction

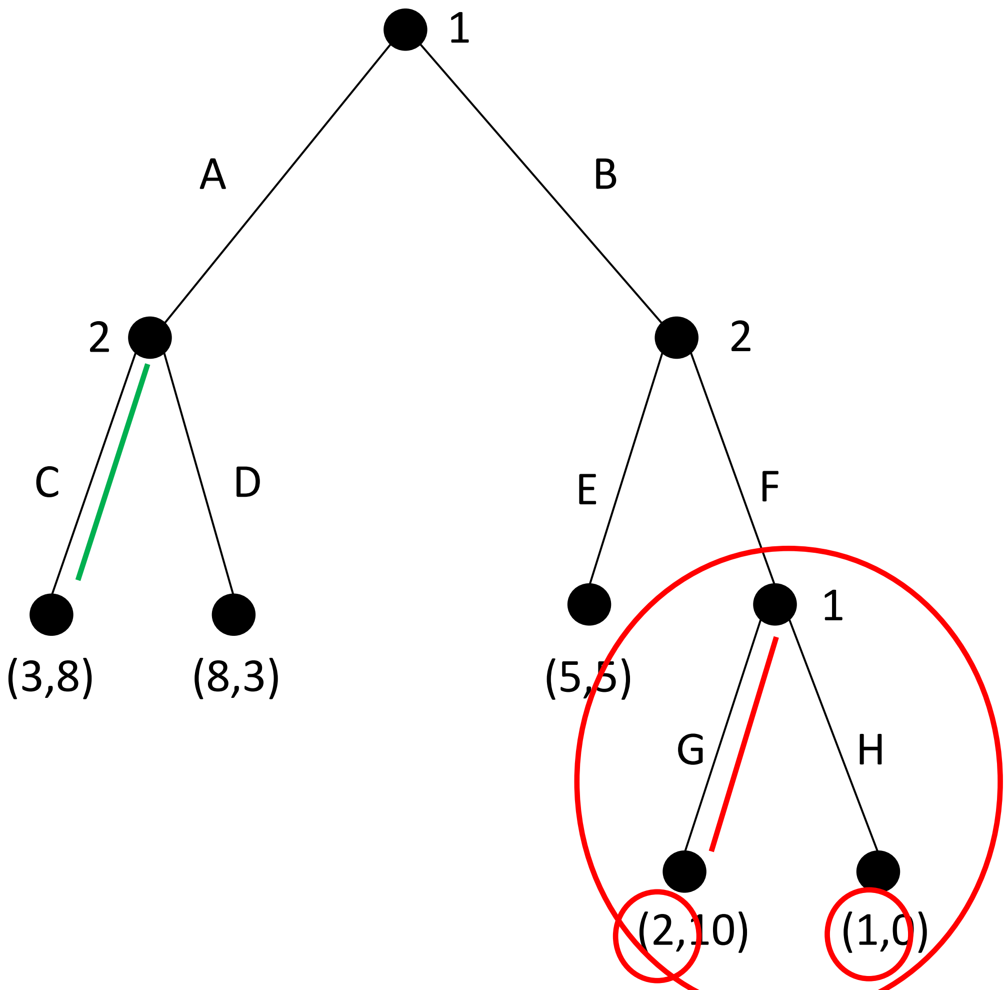
- Consider the extensive-form game below. **Let us use backward induction.**



What is the NE?

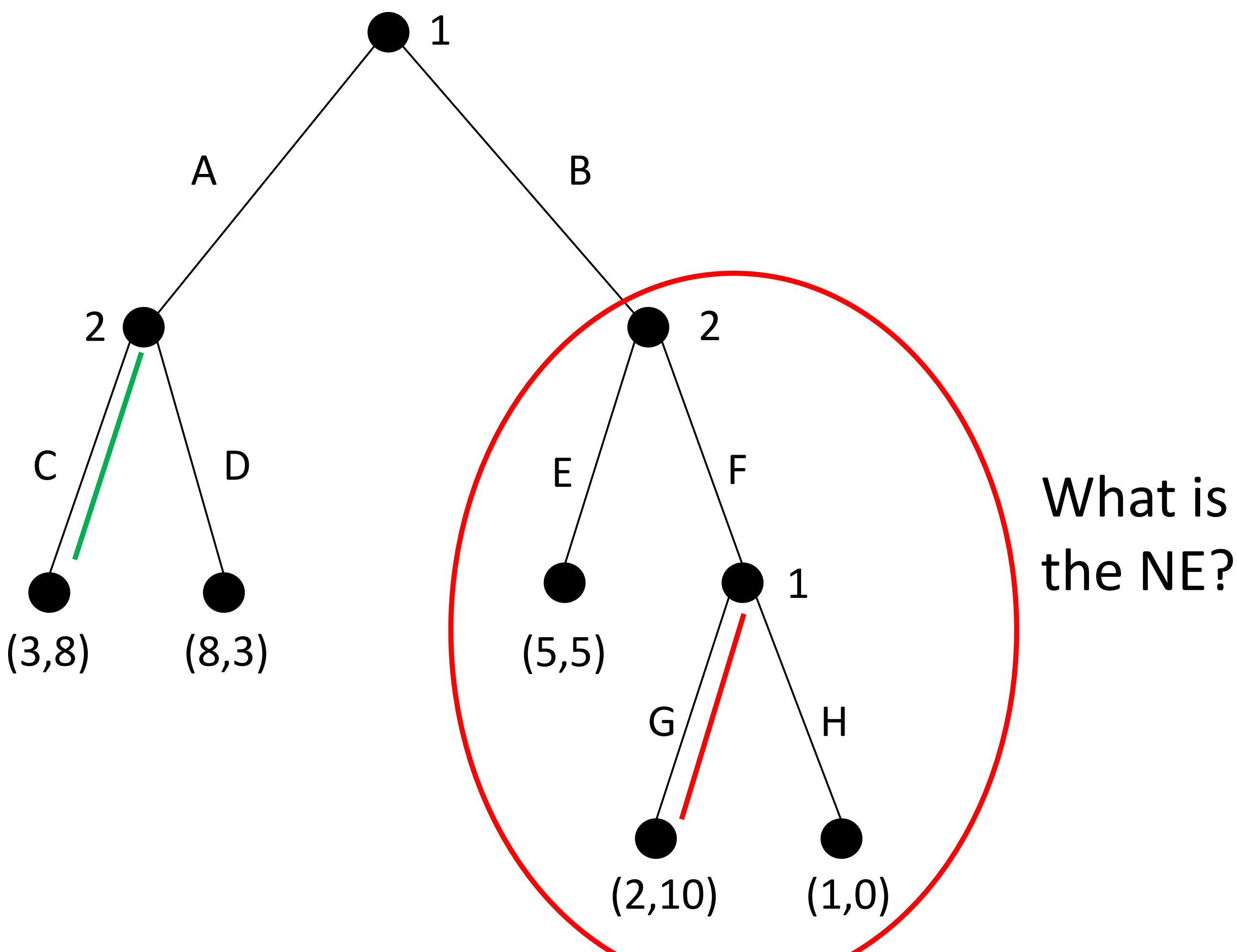
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



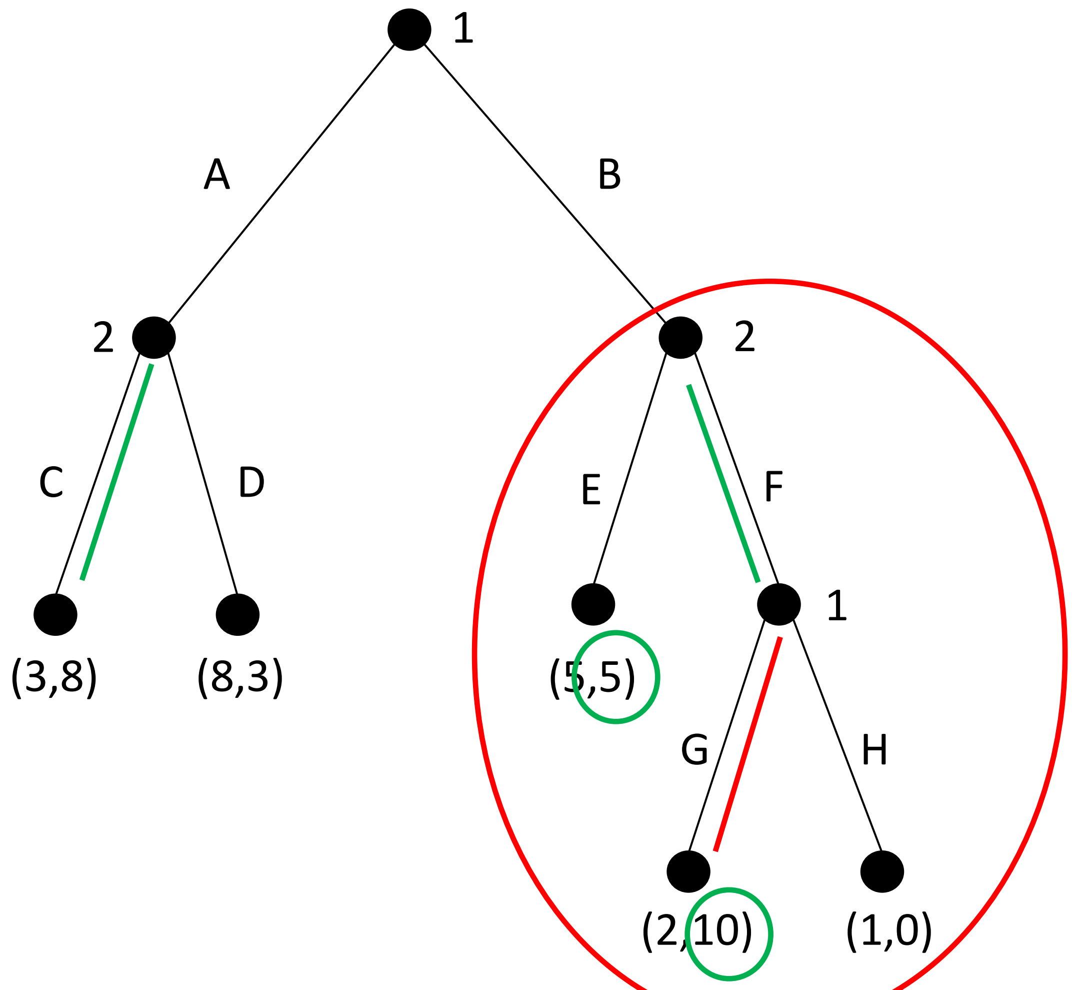
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



Backward Induction

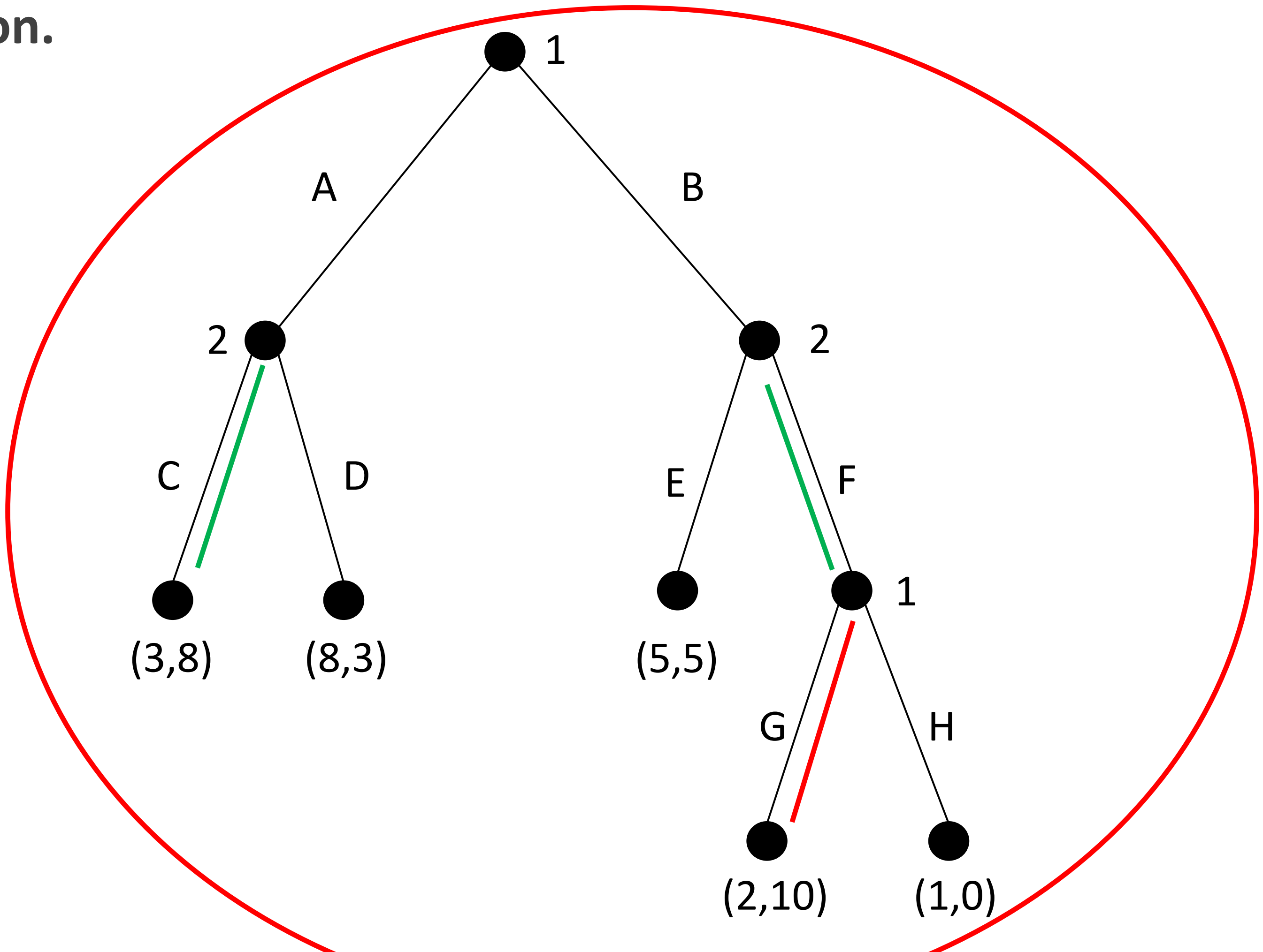
- Consider the extensive-form game below. **Let us use backward induction.**



Backward Induction

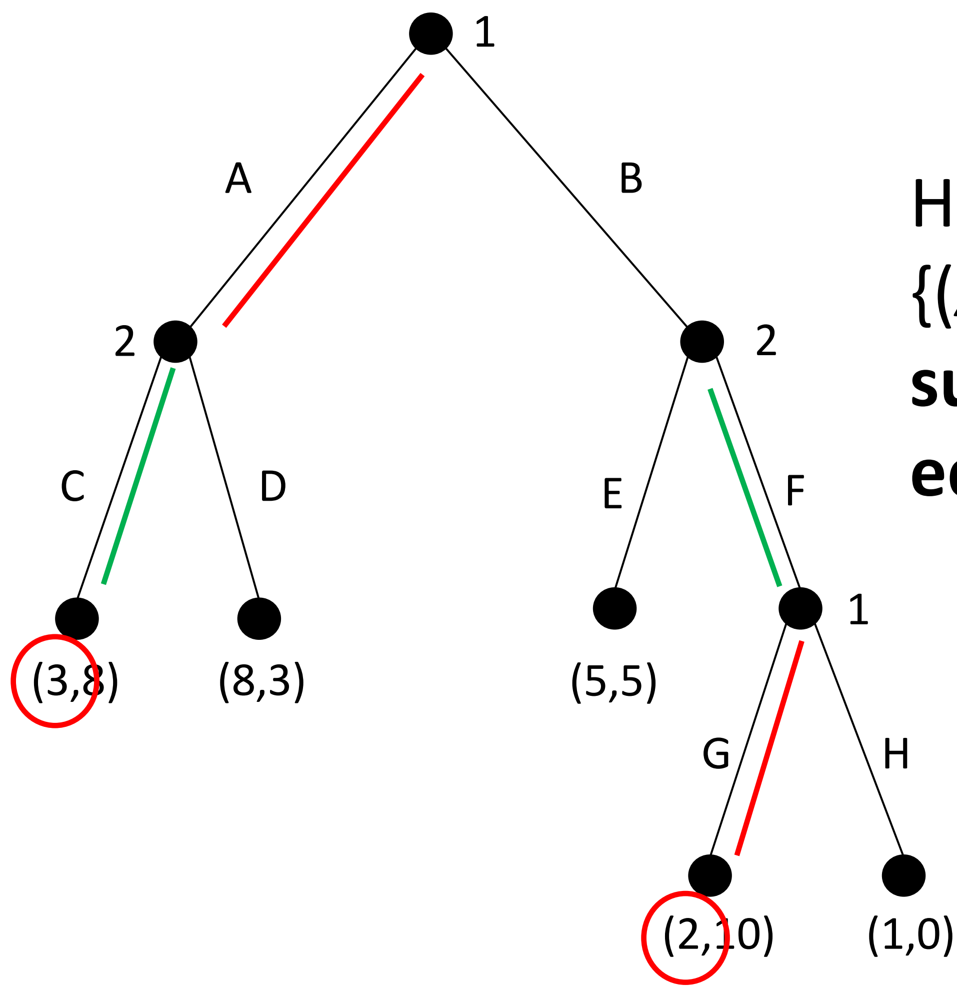
- Consider the extensive-form game below. **Let us use backward induction.**

What is the NE?



Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



Hence, {(A,G),(C,F)}, is a **subgame-perfect equilibrium**

Backward Induction

function $\textsc{BackwardInduction}$ (node h) **returns** $u(h)$
if $h \in Z$ **then**
  return $u(h)$ // h is a terminal node
$best\_util \leftarrow -\infty$
forall $a \in \chi(h)$ **do**
  $util\_at\_child \leftarrow \textsc{BackwardInduction}(\sigma(h, a))$
  if $util\_at\_child_{\rho(h)} > best\_util_{\rho(h)}$ **then**
    $best\_util \leftarrow util\_at\_child$
return $best\_util$

This procedure is a single depth-first traversal of the game tree. Thus, it requires time linear in the size of the game representation.

Backward Induction

function BACKWARDINDUCTION (node h) **returns** $u(h)$
if $h \in Z$ **then**
  return $u(h)$ // h is a terminal node
$best\_util \leftarrow -\infty$
forall $a \in \chi(h)$ **do**
  $util\_at\_child \leftarrow$ BACKWARDINDUCTION$(\sigma(h, a))$
  if $util\_at\_child_{\rho(h)} > best\_util_{\rho(h)}$ **then**
    $best\_util \leftarrow util\_at\_child$
return $best\_util$

Returns the payoff when it reaches a terminal node (end of the recursion)

Keeps track of the best payoff of a node

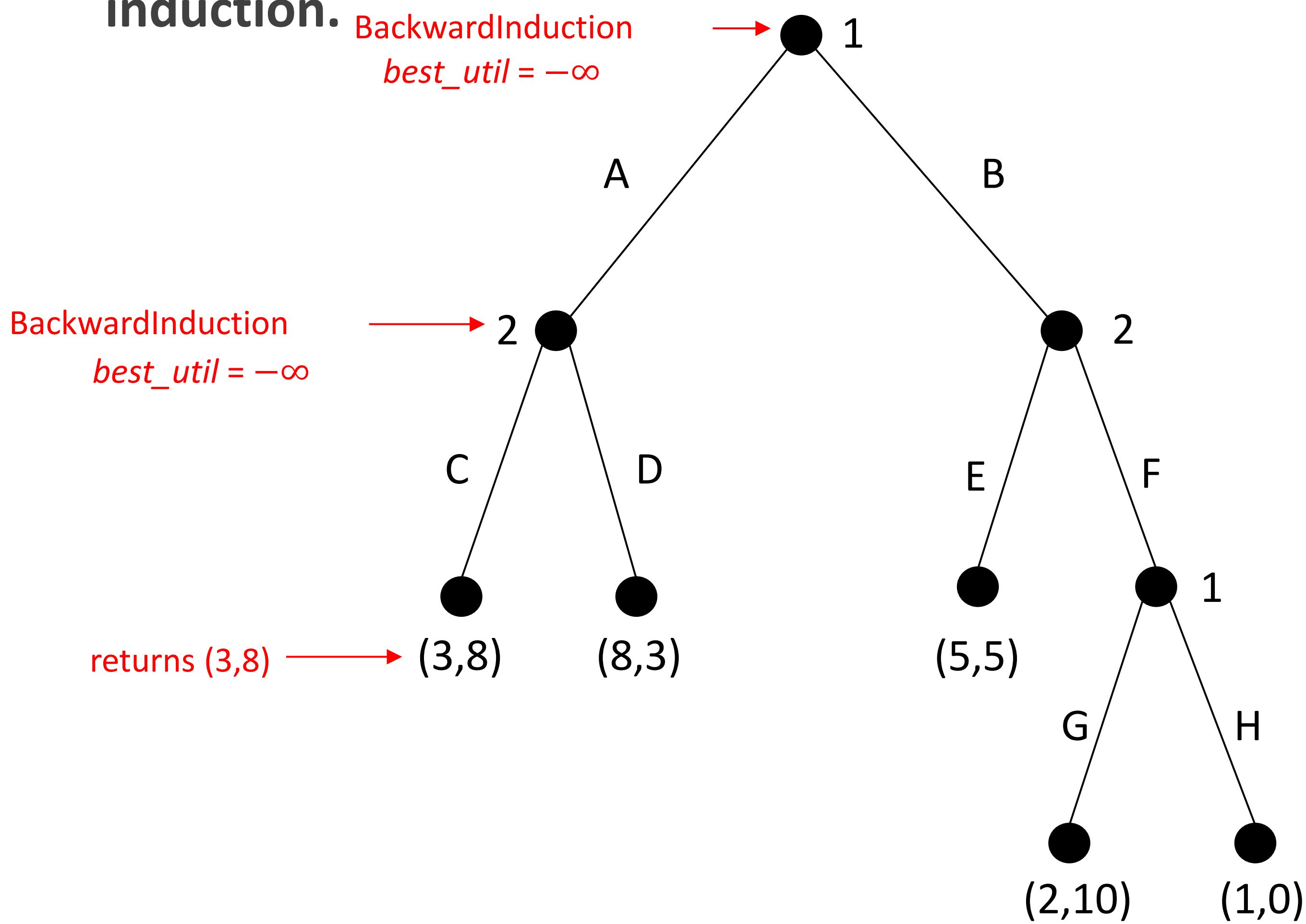
Iterates over all actions of a node

Recursively calls the function for a child node

Recall that each non-terminal node has an associated agent. If the agent's payoff in the child node is greater than agent's payoff in *best_util* then update *best_util*

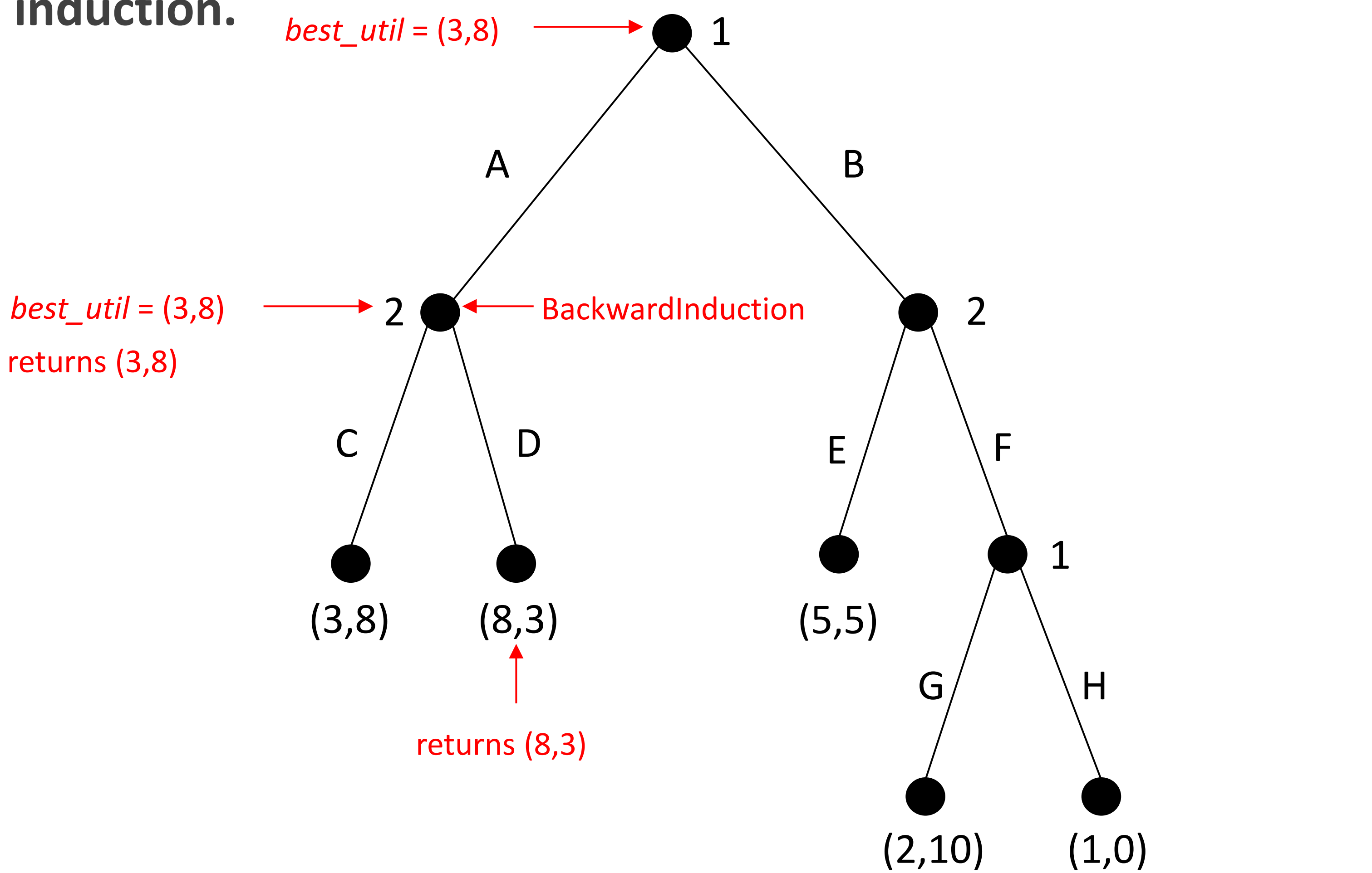
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



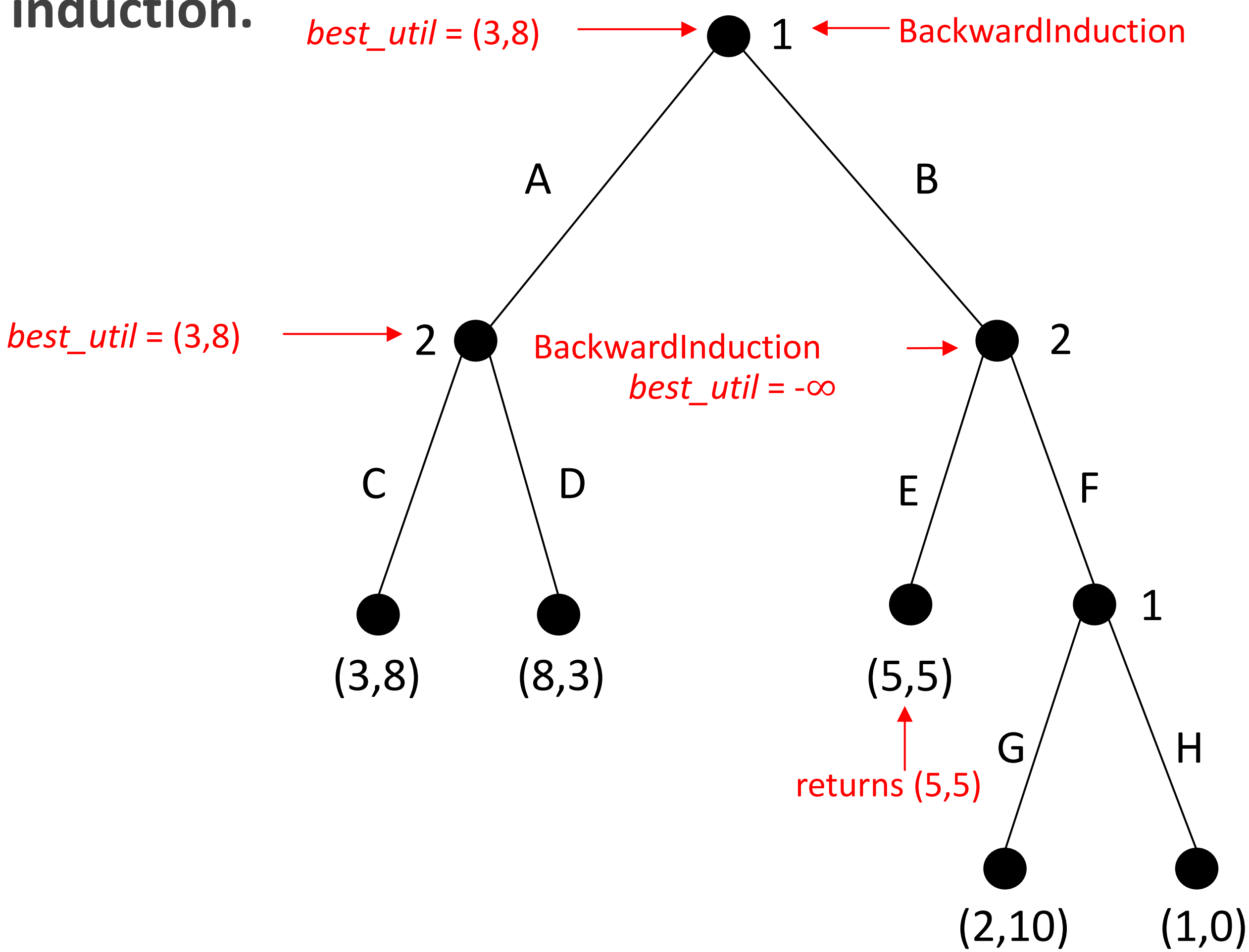
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



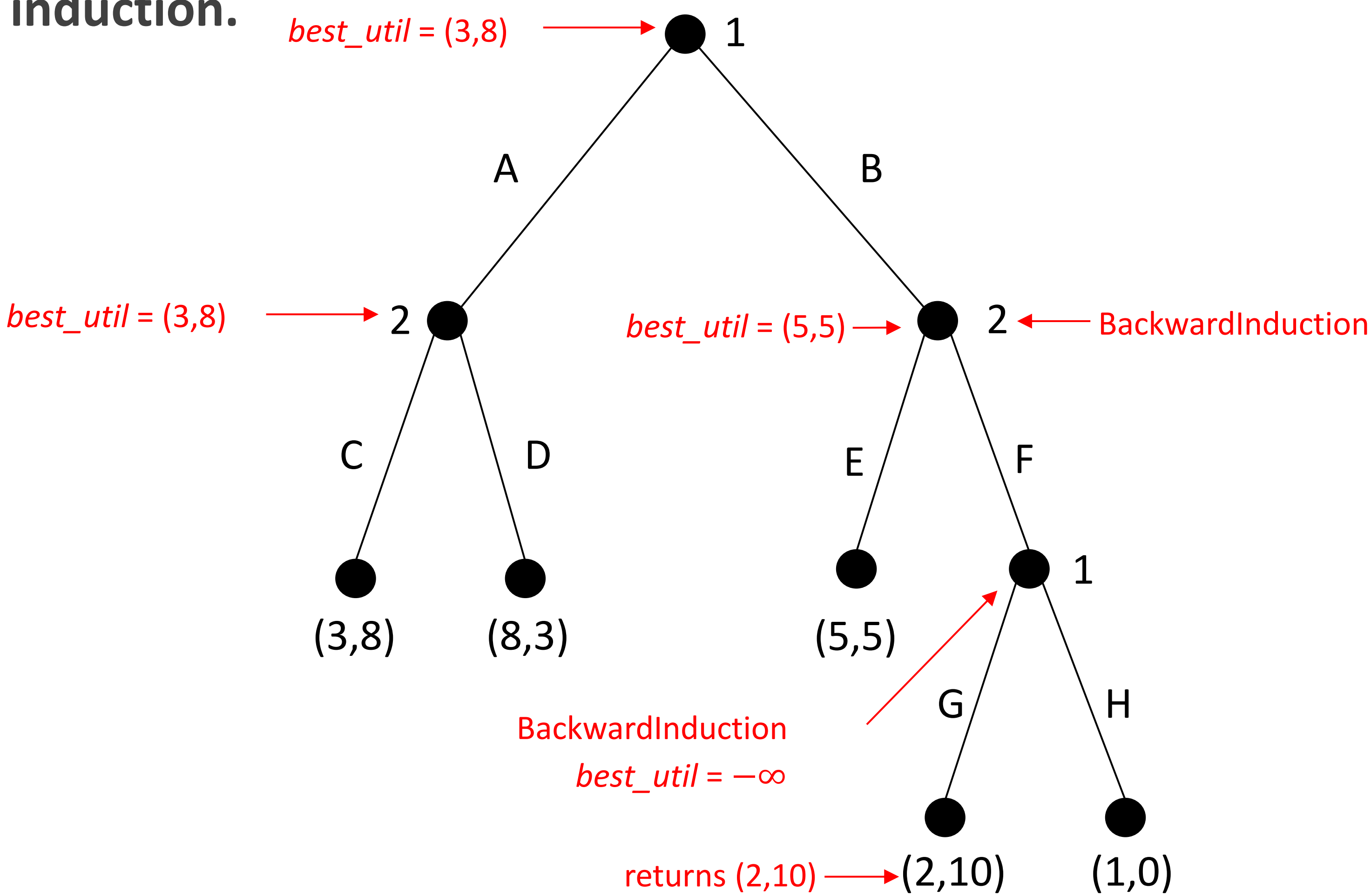
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



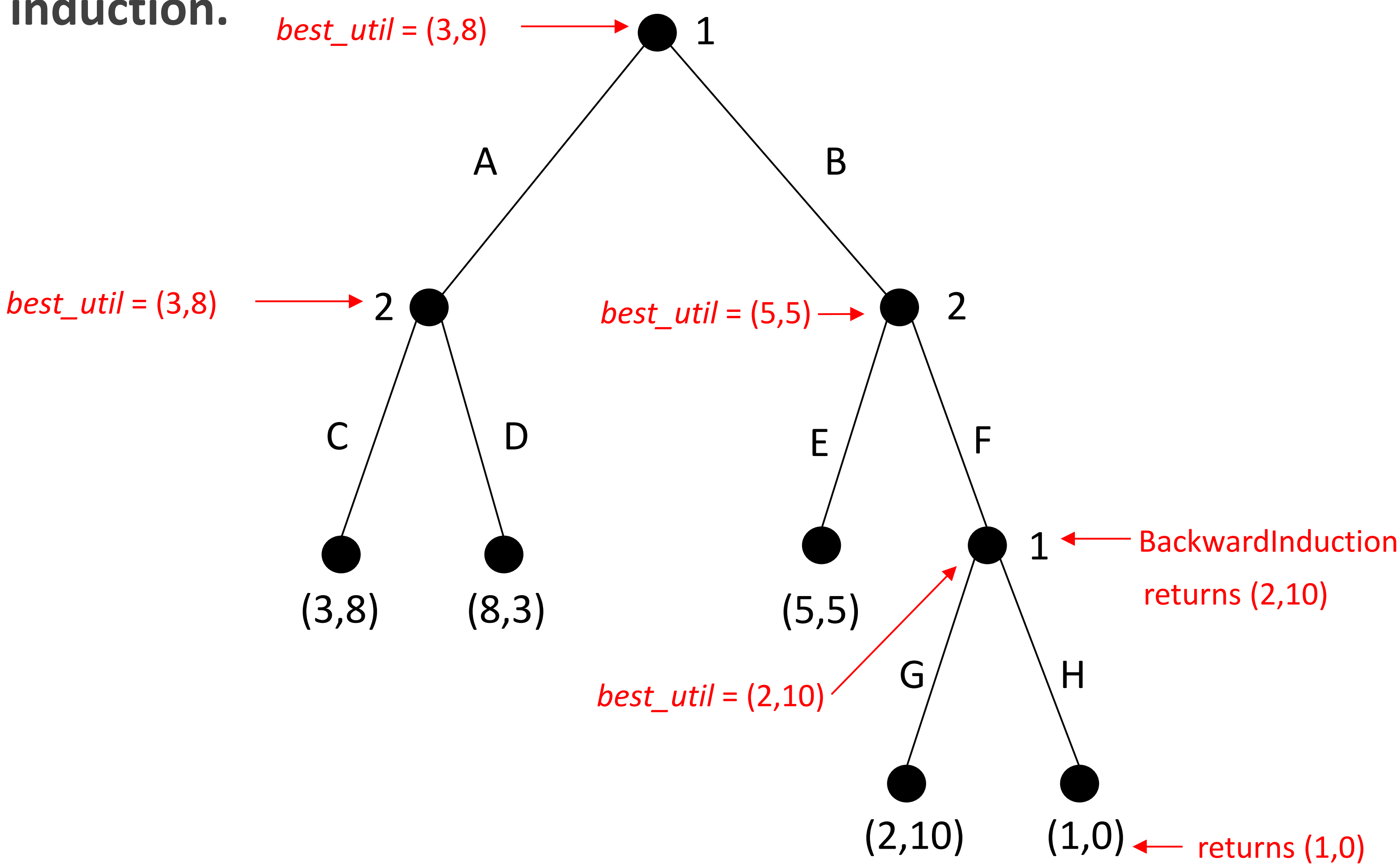
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



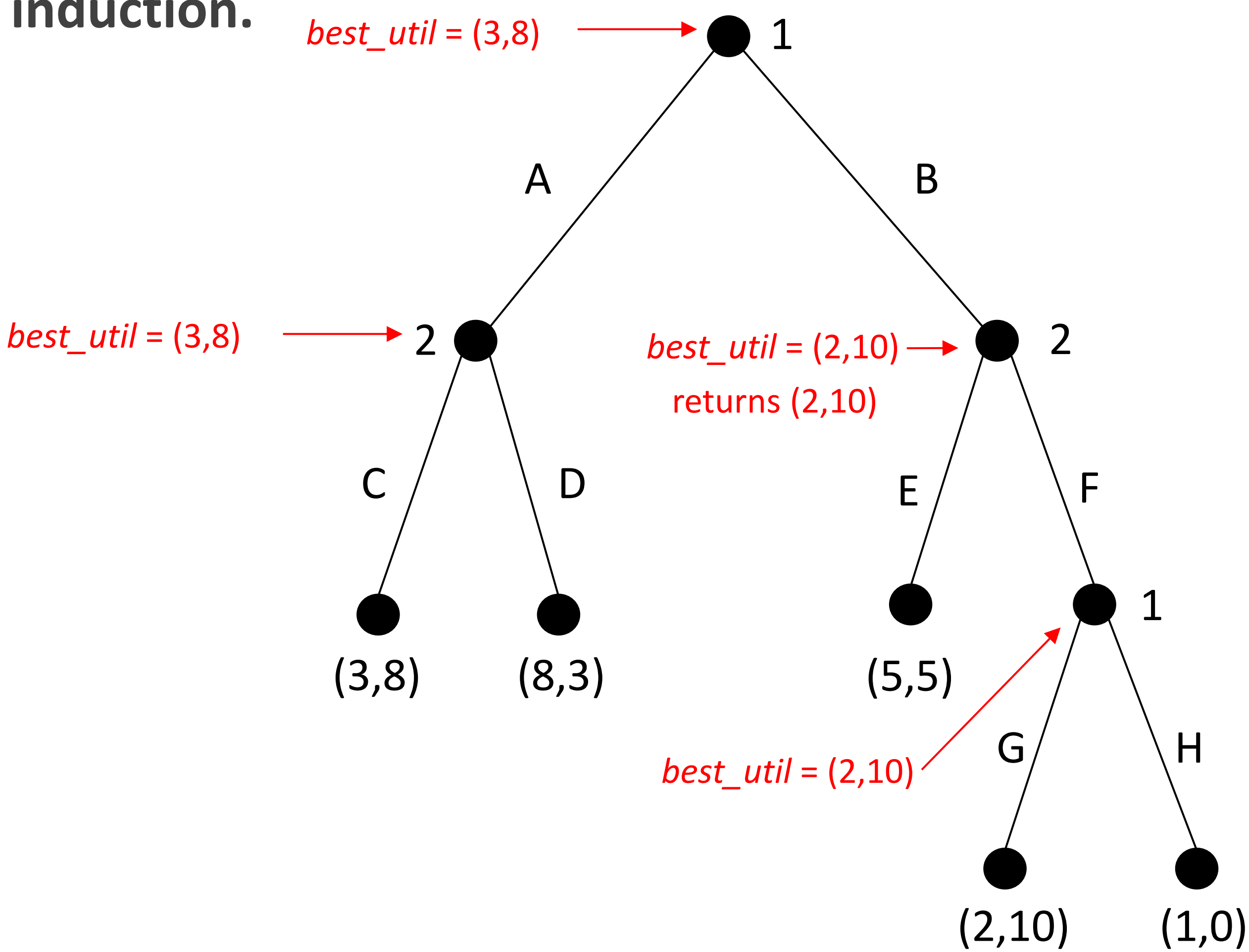
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



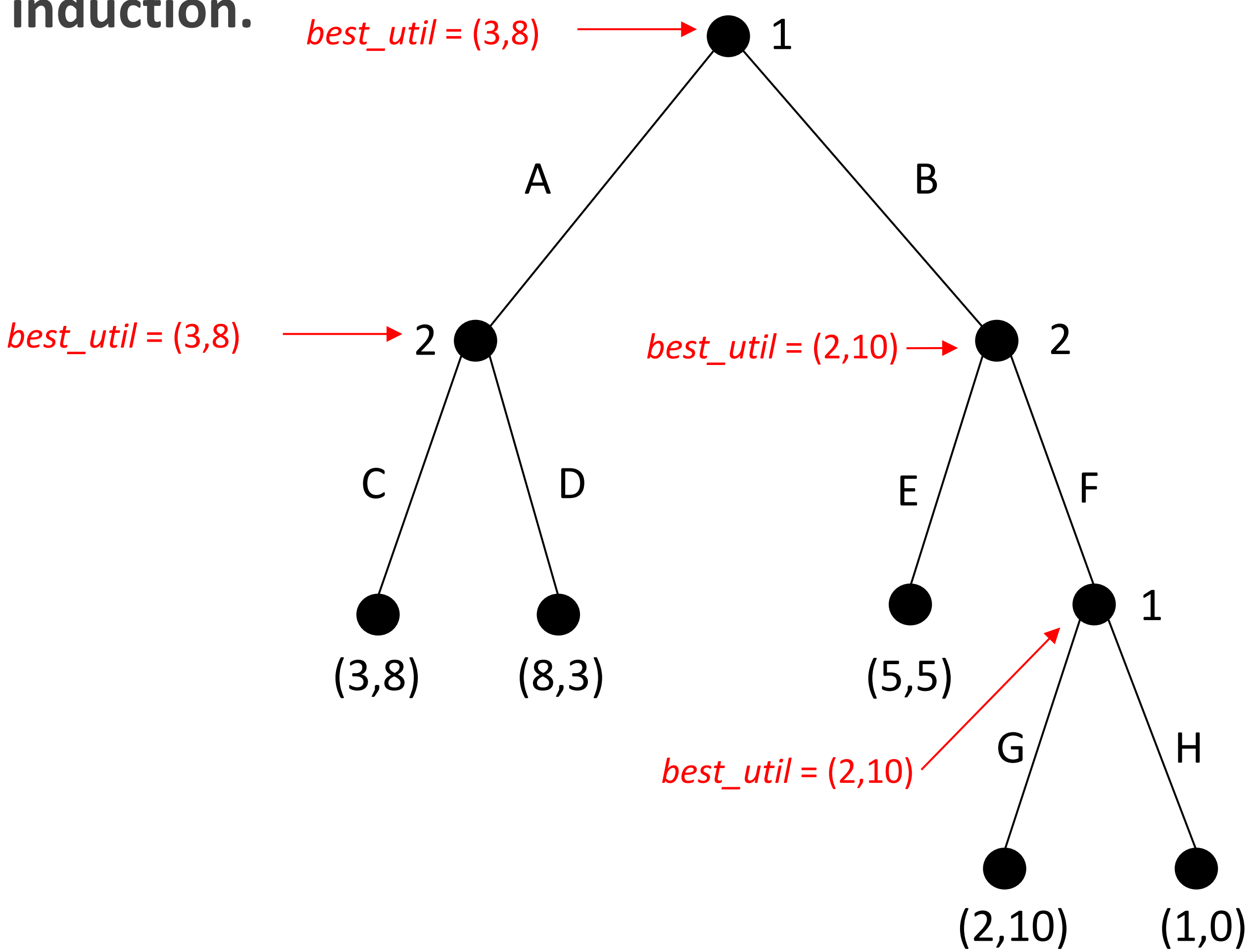
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



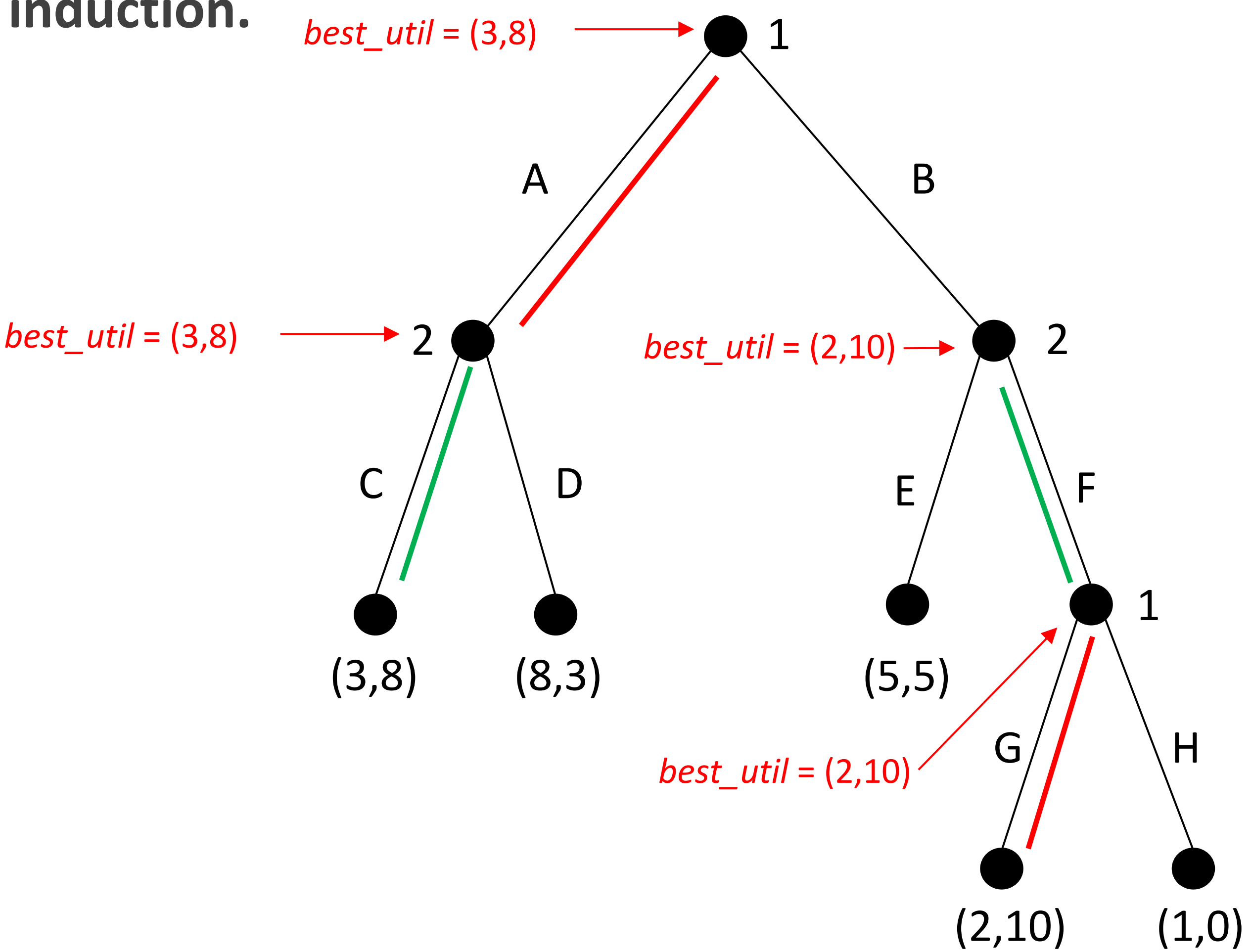
Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



Backward Induction

- Consider the extensive-form game below. **Let us use backward induction.**



Backward Induction

- For zero-sum games, Backward Induction has another name:

  the minimax algorithm

- In this case, it is enough to store one number per node

- It is possible to speed up things with the “alpha-beta pruning”

Exercise

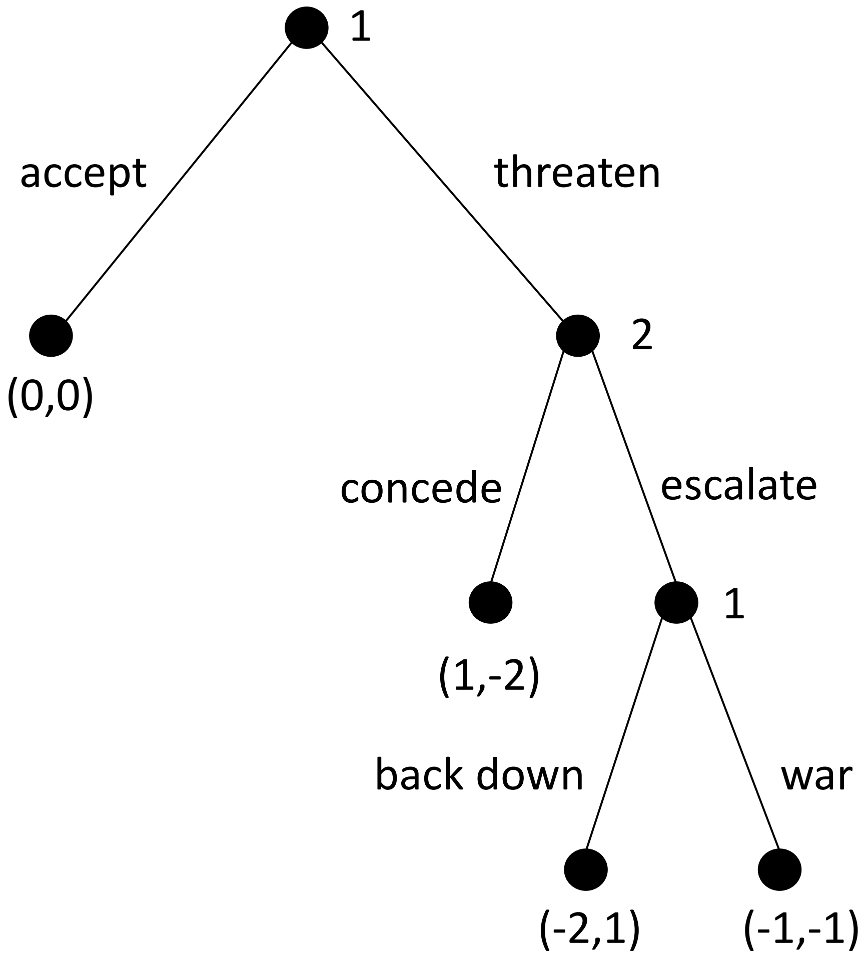
- **Escalation Game**:
  - Two countries are in the brink of war
  - Country 1 may accept the status quo or issue a threat
    - If Country 1 accepts the status quo, the game ends
  - If Country 1 threatens, Country 2 decides to either to concede or escalate the conflict
    - If Country 2 concedes, the game ends

Exercise

- **Escalation Game**:
  - If Country 2 escalates, Country 1 chooses whether to launch war or back down
    - Either way, the game ends

Exercise

- **Escalation Game**:



Exercise

- Use backward induction to find the subgame-perfect equilibrium

Exercise

Convert the game to an extensive form. Check the Nash equilibria and SPE

		Prisoner 2	
		Not confess	***Confess***
Prisoner 1	***Not confess***	-1, -1	-9, <u>0</u>
	Confess	0, -9	-6, -6

Exercise

Convert the game to an extensive form. Check the Nash equilibria and SPE

		Agent 2	
		Heads	***Tails***
Agent 1	***Heads***	-1, 1	1, -1
	Tails	1, -1	-1, 1

Thank You

rui.prada@tecnico.ulisboa.pt