Zero-Sum Games Are Special

CMPUT 366: Intelligent Systems

S&LB §3.4.1

Lecture Outline

1. Recap & Logistics
2. Maxmin Strategies and Equilibrium
3. Alpha-Beta Search

Logistics

- **Assignment 4** is due **Friday April 15** at 11:59pm
- **USRIs** are now available for this course:
  - You should have gotten an email
  - Can also access at: https://p20.courseval.net/etw/ets/et.asp?nxappid=UA2&nxmid=start
  - Survey is available until **this Friday (April 8)** at 11:59pm
- **Assignment 3** marks should be available by the end of the week
- **Solutions** to midterm and assignment 3 are available on eClass

Recap: Game Theory

- Game theory studies the **interactions of rational agents**
    - Canonical representation is the **normal form game**
- Game theory uses **solution concepts** rather than optimal behaviour
    - "Optimal behaviour" is not clear-cut in multiagent settings
    - **Pareto optimal:** no agent can be made better off without making some other agent worse off
    - **Nash equilibrium:** no agent regrets their strategy given the choice of the other agents' strategies
- **Zero-sum games** are games in which agents are in "pure competition"

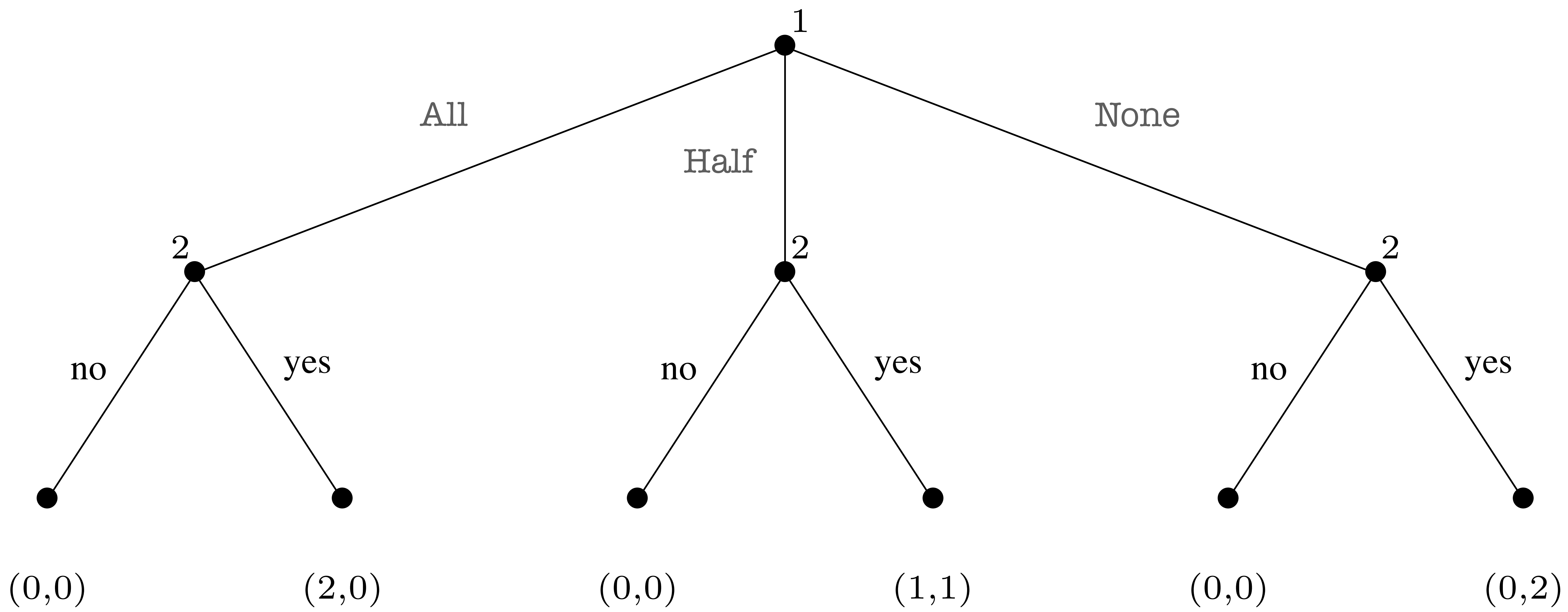
	Ballet	Soccer
Ballet	2, 1	0, 0
Soccer	0, 0	1, 2

	Heads	Tails
Heads	1,-1	-1,1
Tails	-1,1	1,-1

Recap: Perfect Information Extensive Form Game

Definition:
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 n **players**,
- 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),
- $\chi : H \to 2^A$ is the **action function**,
- $\rho : H \to N$ is the **player function**,
- $\sigma : H \times A \to H \cup Z$ is the **successor function**,
- $u = (u_1, u_2, \ldots, u_n)$ is a **utility function** for each player, $u_i : Z \to \mathbb{R}$



Maxmin Strategies

What is the maximum expected utility that an agent can **guarantee** themselves?

Definition:
A **maxmin strategy** for i is a strategy $\bar{s}_i$ that maximizes i's worst-case payoff:

$$\bar{s}_i = \arg\max_{s_i \in S_i} \left[\min_{s_{-i} \in S_i} u_i(s_i, s_{-i}) \right]$$

Definition:
The **maxmin value** of a game for i is the value $\bar{v}_i$ guaranteed by a maxmin strategy:

$$\bar{v}_i = \max_{s_i \in S_i} \left[\min_{s_{-i} \in S_i} u_i(s_i, s_{-i}) \right]$$

Question:

1. Does a maxmin strategy always **exist**?
2. Is an agent's maxmin strategy always **unique**?
3. Why would an agent **want** to play a maxmin strategy?

Minimax Theorem

Theorem: [von Neumann, 1928]
In any **Nash equilibrium** s^* of any **finite, two-player, zero-sum game**, each player receives an expected utility v_i equal to *both* their maxmin and their minmax value.

Proof sketch:

1. Suppose that $v_i < \bar{v}_i$. But then i could guarantee a higher payoff by playing their maxmin strategy. So $v_i \geq \bar{v}_i$.
2. $-i$'s equilibrium payoff is $v_{-i} = \max_{s_{-i}} u_{-i}(s_i^*, s_{-i})$
3. Equivalently, $v_i = \min_{s_{-i}} u_i(s_i^*, s_{-i})$, since the game is zero sum.
4. So $v_i = \min_{s_{-i}} u_i(s_i^*, s_{-i}) \leq \max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i}) = \bar{v}_i$. ∎

Because:
$u_{-i}(s) = -u_i(s)$, so
$v_i = -v_{-i}$ and
$-v_i = \max_{s_i} \left[-u_i(s_i^*, s_{-i})\right]$, and
$-v_i = -\left[\min_{s_i} u_i(s_i^*, s_{-i})\right]$.

Minimax Theorem Implications

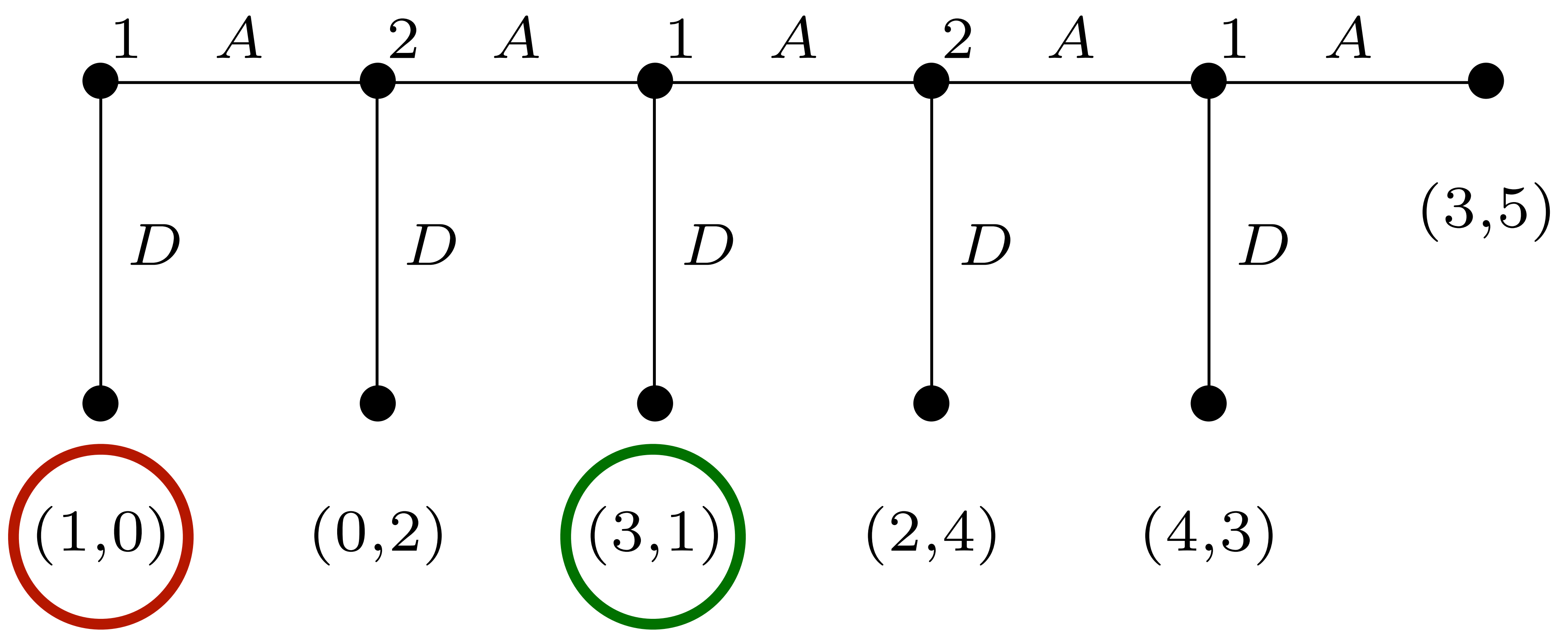
In any **zero-sum** game:

1. Each player's maxmin value is equal to their minmax value (i.e., $\overline{v}_i = \underline{v}_i$). We call this the **value of the game**.

2. For both players, the maxmin strategies and the Nash equilibrium strategies are the **same sets**.

3. Any **maxmin strategy profile** (a profile in which both agents are playing maxmin strategies) is a Nash equilibrium. Therefore, each player gets the same payoff in every Nash equilibrium (namely, their value for the game).

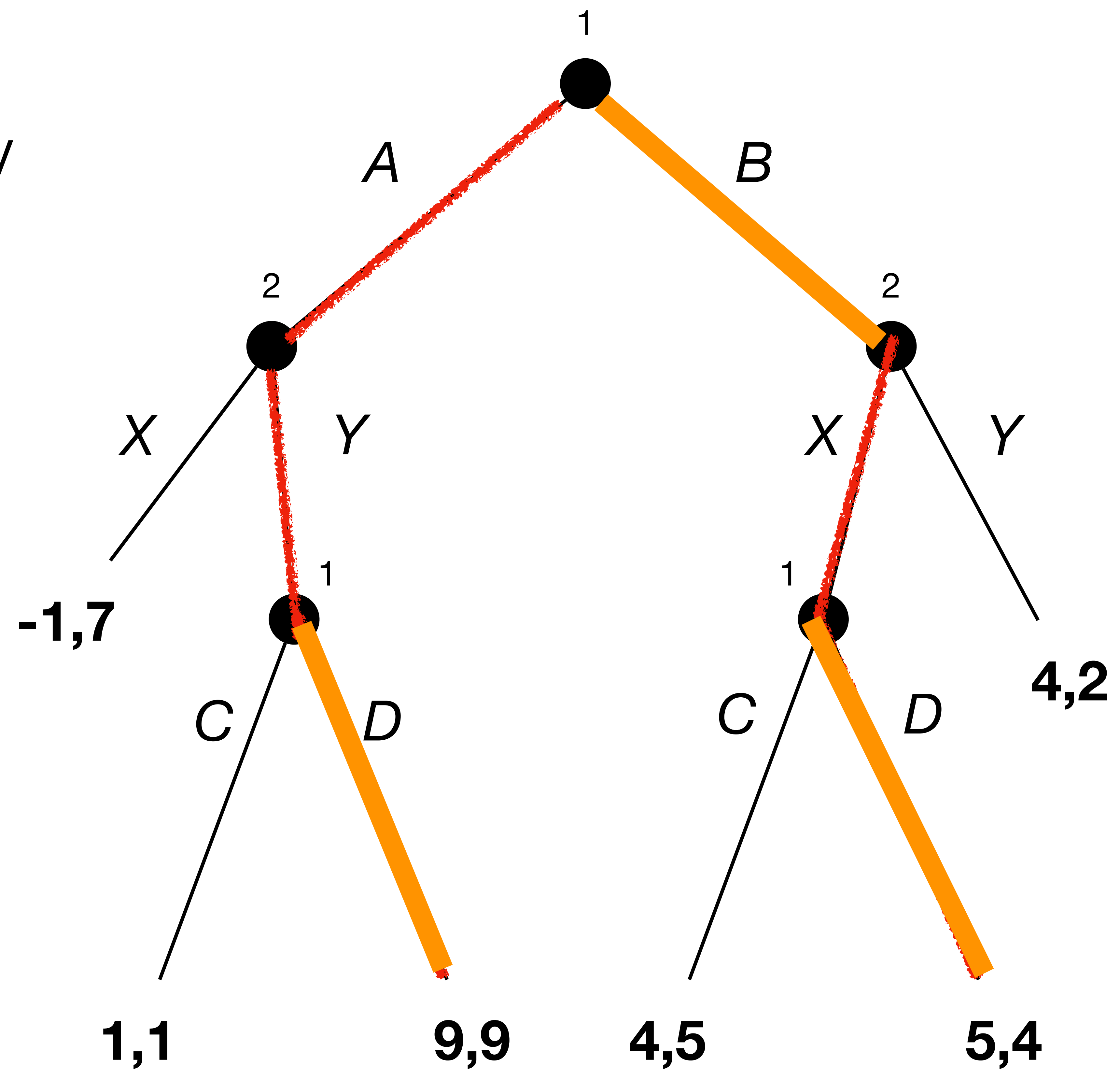
Nash Equilibrium Safety



- Perfect-information extensive form games: Straightforward to compute Nash equilibrium using **backward induction**
  - In Centipede, the unique equilibrium is for all players to play D at *every* choice node
- In the Centipede game, the equilibrium outcome is **Pareto dominated**
- **Question:** Can player 2 ever **regret** playing a Nash equilibrium strategy against a non-Nash strategy for player 1 in Centipede?

Nash Equilibrium Safety: General Sum Games

- In a **general-sum** game, a **Nash equilibrium** strategy is not always a **maxmin** strategy
- **Question:** What is a **Nash equilibrium** of this game?
- **Question:** What is player 1's **maxmin strategy**?
- **Question:** Can player 1 ever **regret** playing a Nash equilibrium against a **non-equilibrium** player?

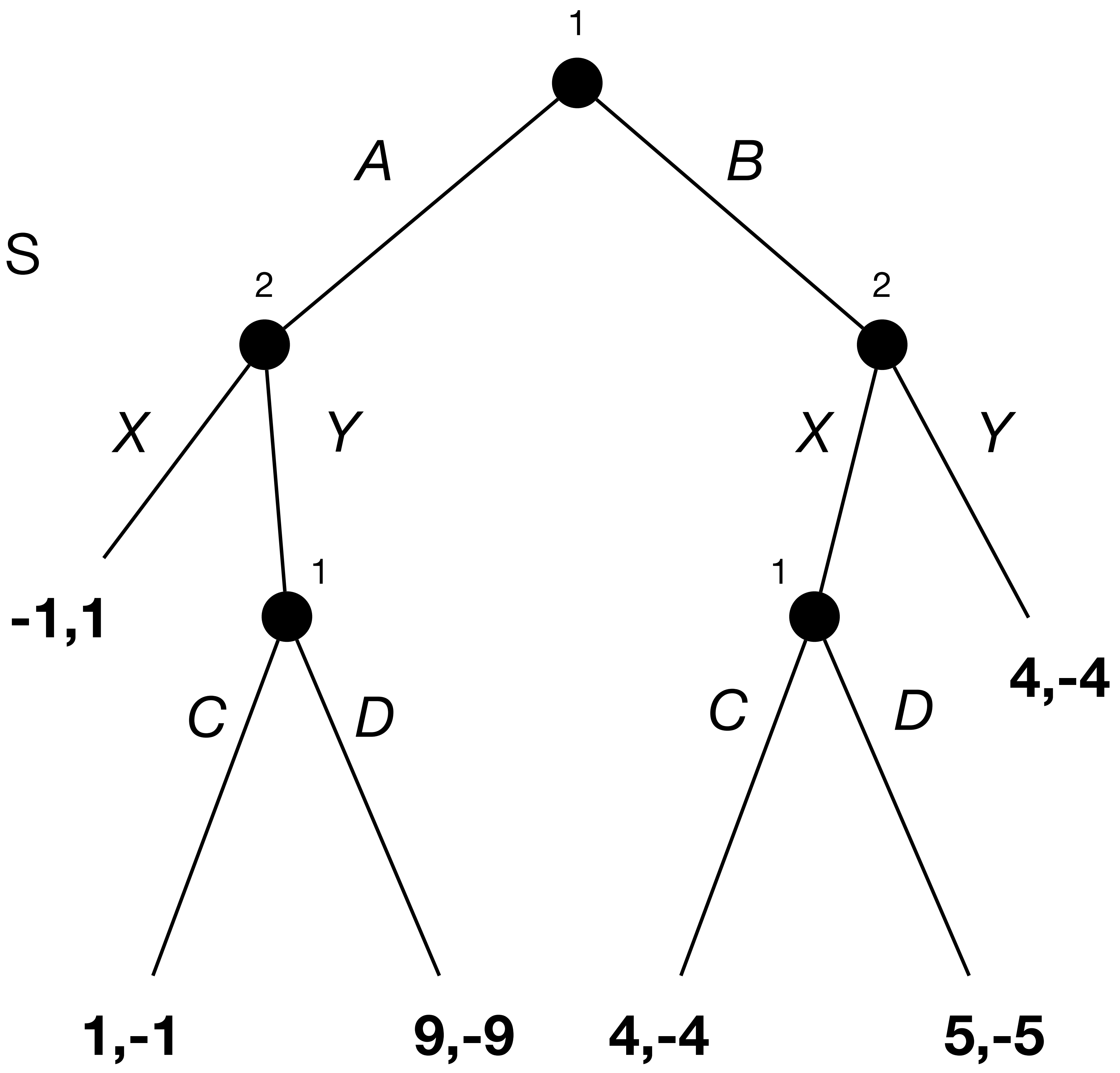


Nash Equilibrium Safety: Zero-sum Games

- In a zero-sum game, every **Nash equilibrium** strategy is **also** a maxmin strategy

- **Question:** What is player 1's **maxmin value**?

- **Question:** Can player 1 ever regret playing a Nash equilibrium strategy against a **non-equilibrium** player?



Efficient Equilibrium Computation

- Backward induction requires us to examine **every leaf node**
- However, in a zero-sum game, we can do better by **pruning** some sub-trees
    - Special case of **branch and bound**
- **Intuition:** If a player can guarantee **at least** x starting from a given subtree h, but their opponent can guarantee them getting less than x in an earlier subtree, then the opponent will never allow the player to reach h

Algorithm: Alpha-Beta Search

AlphaBetaSearch(a choice node h):

  $v \leftarrow$ MaxValue$(h, -\infty, +\infty)$

  return $a \in \chi(h)$ such that MaxValue($\sigma(h,a)$) = v

MaxValue(choice node h, bound α, bound β):

  if $h \in Z$: **return** $u_i(h)$

  $v \leftarrow -\infty$

  for $h' \in \{h'' \mid a \in \chi(h)$ *and* $\sigma(h,a) = h''\}$:

    $v \leftarrow$ **max**(v, MinValue(h', α, β))

    if $v \geq \beta$: **return** v

    $\alpha \leftarrow$ **max**(α, v)

  return v

MinValue(node h, bound α, bound β):

  if $h \in Z$: **return** $u_i(h)$

  $v \leftarrow +\infty$

  for $h' \in \{h'' \mid a \in \chi(h)$ *and* $\sigma(h,a) = h''\}$:

    $v \leftarrow$ **min**(v, MaxValue(h', α, β))

    if $v \leq \alpha$: **return** v

    $\beta \leftarrow$ **min**(β, v)

  return v

Randomness

- Sometimes a game will include elements of randomness in the environment
  - E.g., dice
- Can handle this by including **chance nodes** owned by **nature**
- Alpha-beta search can work in this setting, but it needs some tweaks
  - Take expectation at chance nodes instead of min/max
  - Pruning based on **bounds** on the expectation
- **Question:** What about randomness in the strategies of the **players**?

Alpha-Beta Search: Additional Considerations

- **Question:** Can this algorithm work with **arbitrarily deep** game trees?
- **Question:** Can this algorithm work for **non-zero-sum** games?

Summary

- **Maxmin** strategies maximize an agent's worst-case payoff
- **Nash equilibrium** strategies are different from maxmin strategies in general games
- In **zero-sum games**, they are the same thing
  - It is always **safe** to play an equilibrium strategy in a zero-sum game
  - **Alpha-beta search** computes equilibrium of zero-sum games more efficiently than backward induction