Zero-Sum Games Are Special

CMPUT 366: Intelligent Systems

S&LB §3.4.1

Lecture Outline

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

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 where the agents are in **pure competition**

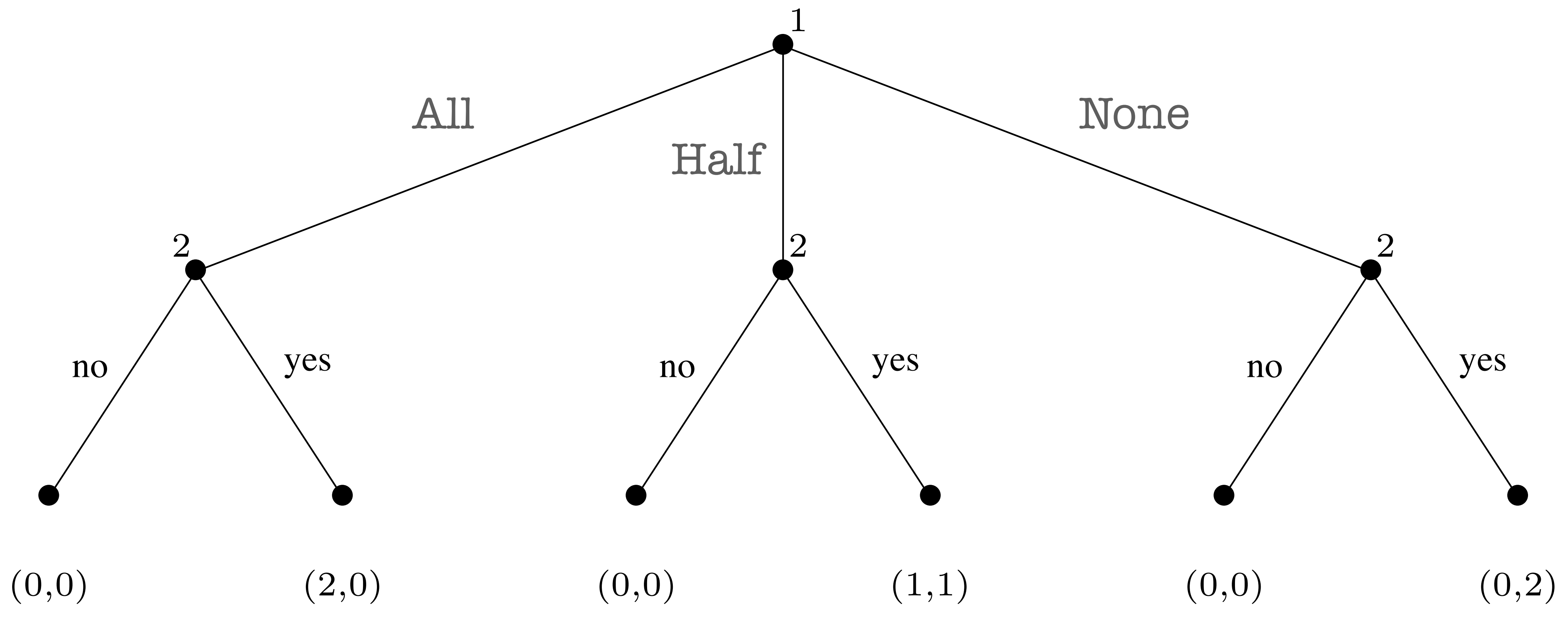
	Ballet	Soccer
Ballet	2, 1	0, 0
Soccer	0, 0	1, 2

	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**,
- $u = (u_1, u_2, ..., u_n)$ is a **utility function** for each player, $u_i : Z \to \mathbb{R}$



Maxmin Strategies

What is the maximum amount that an agent can **guarantee** themselves in expectation?

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 a 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 finite, two-player, zero-sum game, in any Nash equilibrium, each player receives an expected utility v_i equal to both their maxmin and their minmax value.

Proof sketch:

1. Suppose that $v_i < \overline{v}_i$. But then i could guarantee a higher payoff by playing their maxmin strategy. So $v_i \geq \overline{v}_i$.

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 $v_i = \min_{s_{-i}} u_i(s_i^*, s_{-i}) \leq \max_{s_i} \min_{s_{-i}} u_i(s_i, s_{-i}) = \overline{v}_i$. ∎

Minimax Theorem Implications

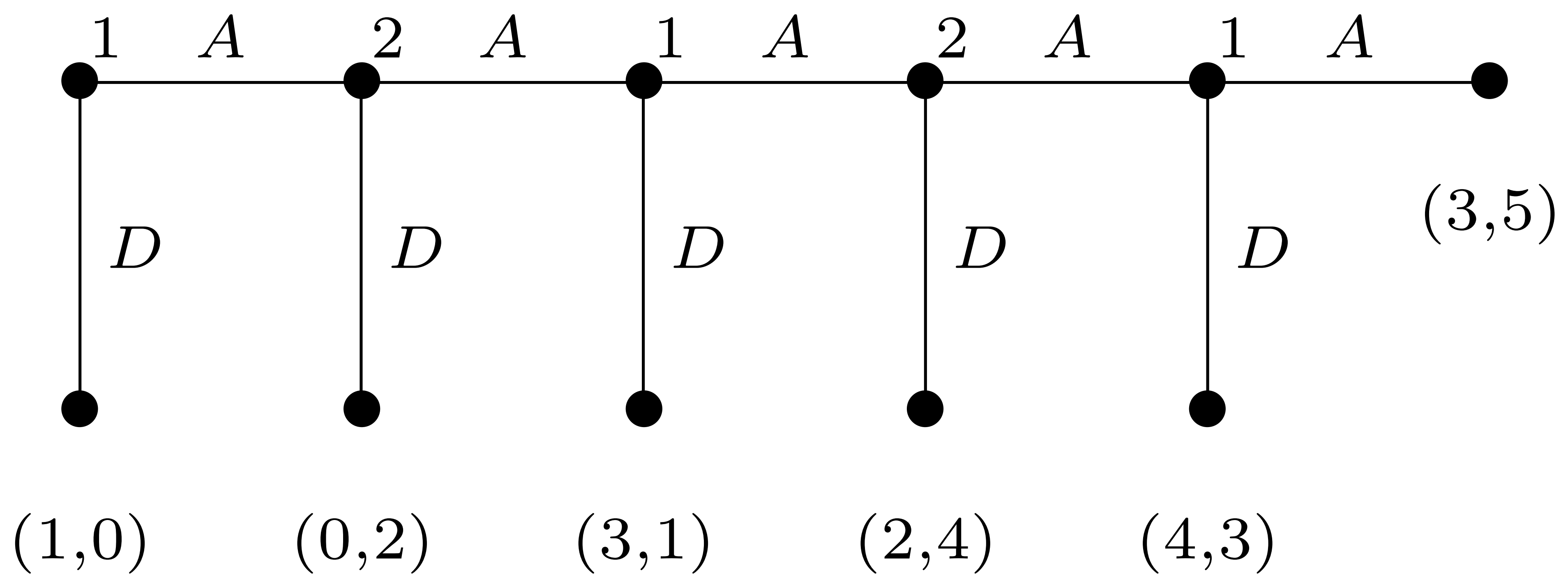
In any **zero-sum** game:

1. Each player's maxmin value is equal to their minmax value. 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

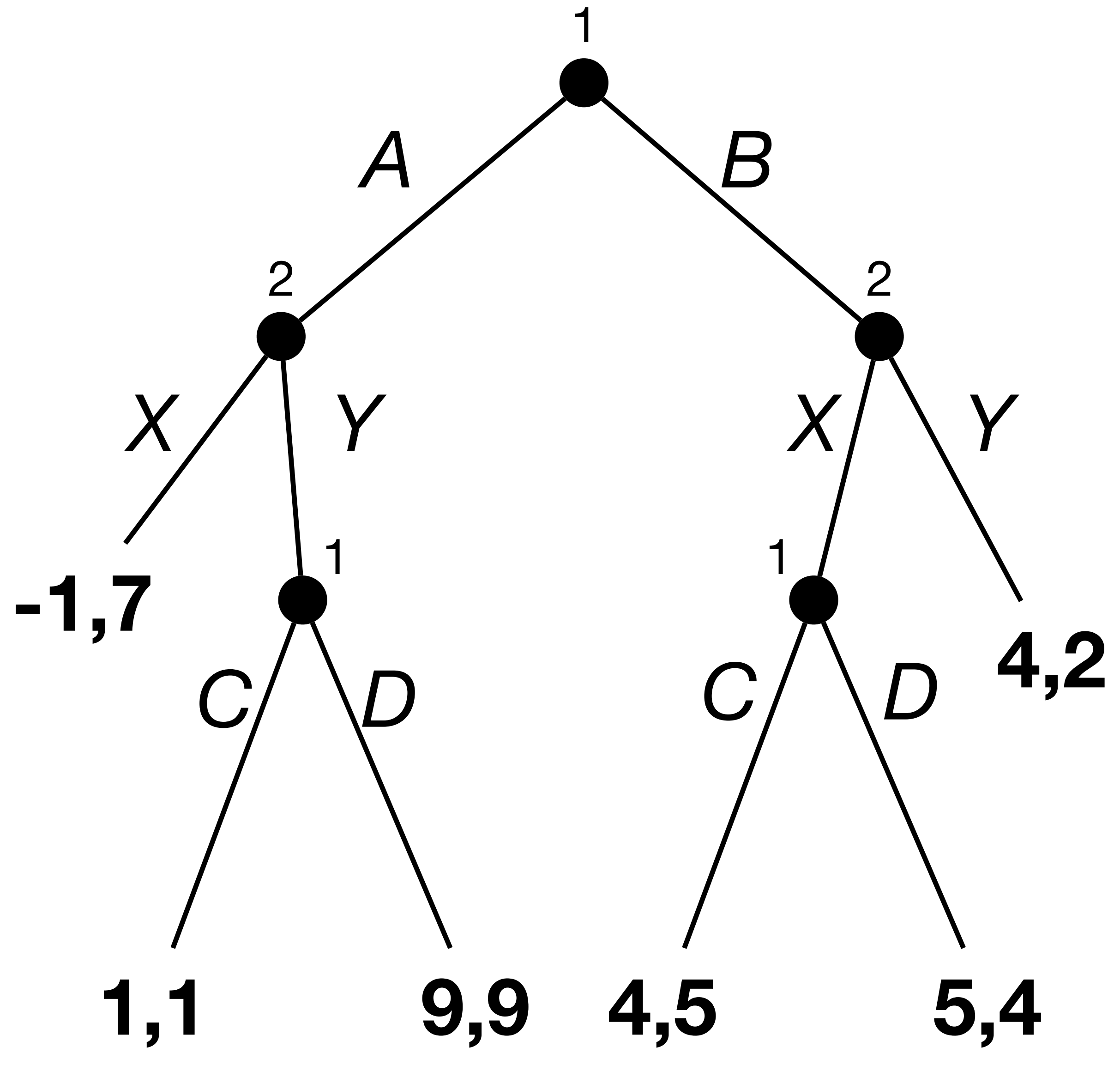


- Perfect-information extensive form games: Straightforward to compute Nash equilibrium using **backward induction**
- In the Centipede game, the equilibrium outcome is **Pareto dominated**
- **Question:** Can player 2 ever regret playing a Nash equilibrium strategy against a suboptimal player 1 in Centipede?

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? $[(A,D,D),(Y,X)]$
- **Question:** What is player 1's **maxmin strategy**?
  (B,D,D)
- **Question:** Can player 1 ever **regret** playing a Nash equilibrium against a **suboptimal** player?

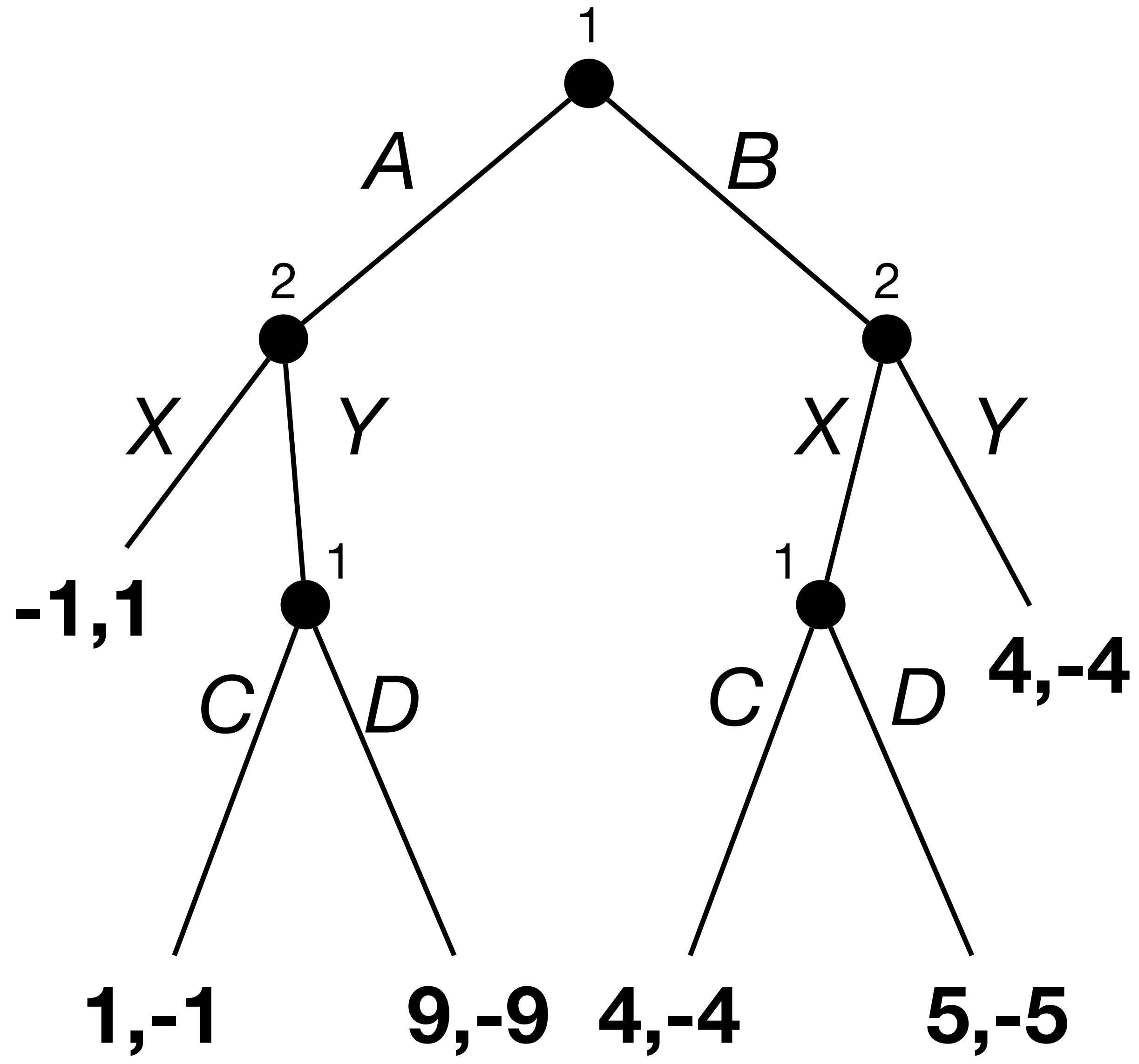
Yes, because if player 2 does not follow the same Nash equilibrium, player 1 could get -1 (the worst payoff in the game).



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 for this game? 4 (same as previous game)
- **Question:** Can player 1 ever regret playing a Nash equilibrium strategy against a suboptimal player?

No, because player 1's equilibrium strategy is also their maxmin strategy.



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, $-\infty$, ∞)
  return $a \in \chi(h)$ such that MAXVALUE($\sigma(h,a)$) = v

MAXVALUE(choice node h, max value α, min value β):
  if $h \in Z$: **return** $u(h)$
  $v \leftarrow -\infty$
  for $h' \in \{ h' \mid a \in \chi(h) \text{ and } \sigma(h,a) = h' \}$:
    $v \leftarrow$ **max**(v, MINVALUE(h′, α, β))
    if $v \geq \beta$: **return** v
    $\alpha \leftarrow$ **max**(α, v)
  return v

MINVALUE(h, α, β):
  if $h \in Z$: **return** $u(h)$
  $v \leftarrow +\infty$
  for $h' \in \{ h' \mid a \in \chi(h) \text{ and } \sigma(h,a) = h' \}$:
    $v \leftarrow$ **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?
  No, because it needs to get to the "bottom" of the tree before it can start pruning
- **Question:** Can this algorithm work for **non-zero-sum** games?
  No, it relies on the fact that player 1 and player 2 are maximizing and minimizing the **same quantity**.

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