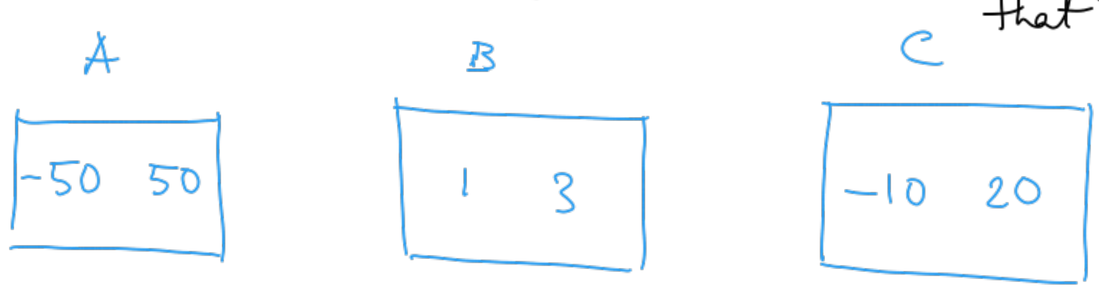
- Classical AI: Single agent problem
- Multi-agent AI: Rationality → Act with reasons
  - → loss function minimize
  - → Equilibrium

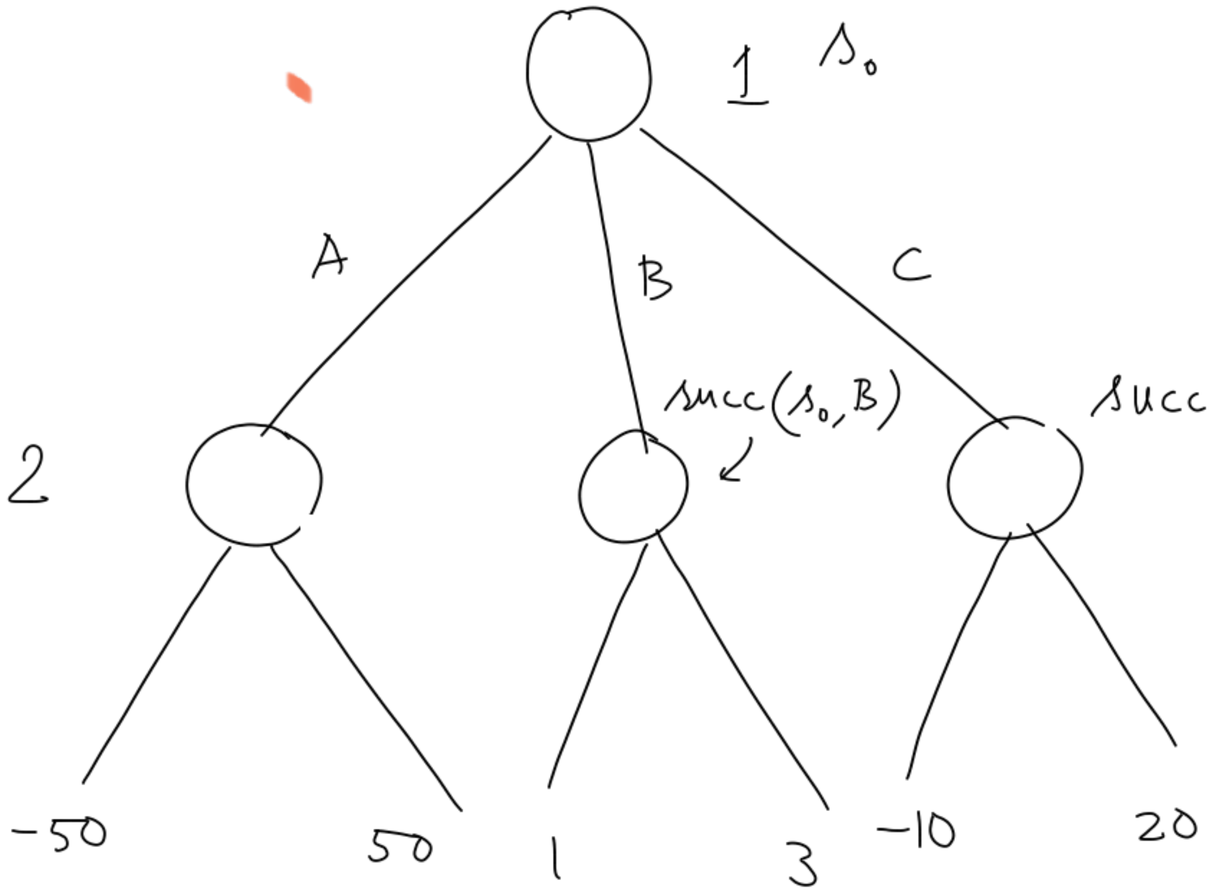
Example 2-player game

Player 1: Chooses the box

Player 2: Chooses the ball from that box



numbers represent the happiness of player 1.
(utility)



We need player modeling

Case 1: player 2 is a chance $\left(\frac{1}{2}, \frac{1}{2}\right)$ player

→ then choose C

Case 2: player 2 is a min player

→ choose B.

Notation: $\text{Players} = \{\text{agent}, \text{opp}\}$

s : state of the game, starting state s_0

$\text{actions}(s)$ = possible actions at s

$\text{succ}(s, a)$ = successor of s when a action is taken

$\text{isEnd}(s)$ = whether s is a terminal node

$\text{utility}(s)$ = agent's utility at a terminal node s

$\text{player}(s)$ = the current player at intermediate state s.

— opp's utility

Chess

$P = \{W, B\}$

fill the rest

Strategy of a player: a complete contingency plan of a player in a game.

Deterministic: $\pi_i(s) \in actions(s)$

function mapping the state to one of the actions

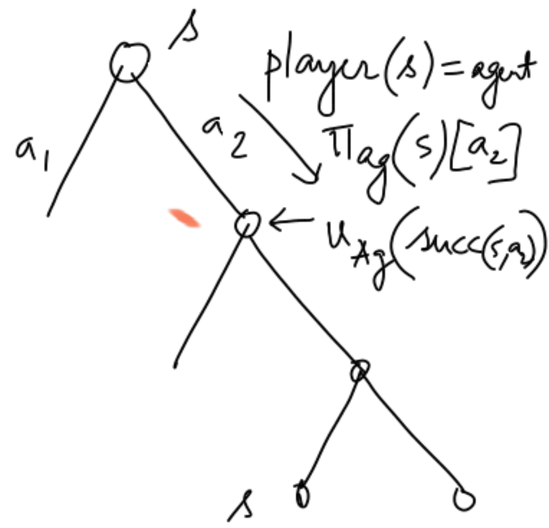
Randomized: $\pi_i(s) \in \Delta actions(s)$

ΔX = all possible prob. distributions over X.

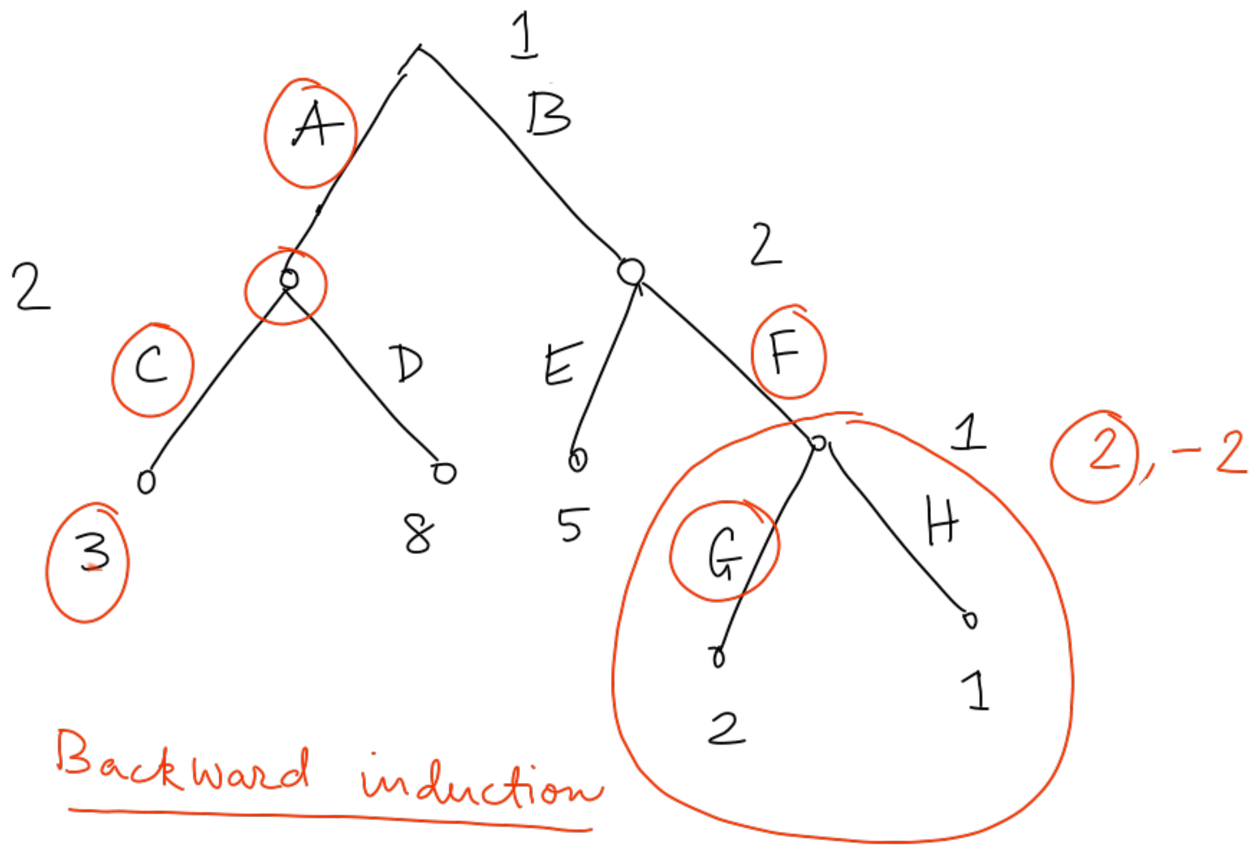
↑ set

$$u_{Ag}(s) = \begin{cases} \text{utility}(s), & \text{if } \text{isEnd}(s) = \text{TRUE} \\ \sum_{a \in \text{actions}(s)} \pi_{Ag}(s)[a]\, u_{Ag}(\text{succ}(s,a)), & \text{if player}(s) = \text{agent} \\ \sum_{a \in \text{actions}(s)} \pi_{opp}(s)[a]\, u_{Ag}(\text{succ}(s,a)), & \text{if player}(s) = \text{opp} \end{cases}$$

expected utility of agent at an intermediate node



Halving game actions → / → half the number (take floor)
− → reduce the number by 1



Checkers $\sim 10^{20}$ nodes

Chess $\sim 10^{40}$

Go $\sim 10^{170}$

Backward induction