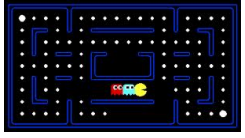




Games: game evaluation



Policies

Deterministic policies: $\pi_p(s) \in \text{Actions}(s)$

action that player p takes in state s

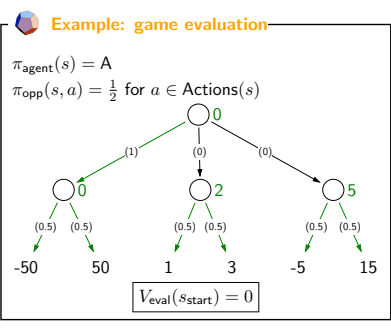
Stochastic policies $\pi_p(s, a) \in [0, 1]$:

probability of player p taking action a in state s

[semi-live solution: humanPolicy]

- Following our presentation of MDPs, we revisit the notion of a **policy**. Instead of having a single policy π , we have a policy π_p for each player $p \in \text{Players}$. We require that π_p only be defined when it's p 's turn; that is, for states s such that $\text{Player}(s) = p$.
- It will be convenient to allow policies to be stochastic. In this case, we will use $\pi_p(s, a)$ to denote the probability of player p choosing action a in state s .
- We can think of an MDP as a game between the agent and nature. The states of the game are all MDP states s and all chance nodes (s, a) . It's the agent's turn on the MDP states s , and the agent acts according to π_{agent} . It's nature's turn on the chance nodes. Here, the actions are successor states s' , and nature chooses s' with probability given by the transition probabilities of the MDP: $\pi_{\text{nature}}((s, a), s') = T(s, a, s')$.

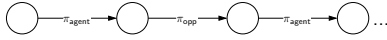
Game evaluation example



- Given two policies π_{agent} and π_{opp} , what is the (agent's) expected utility? That is, if the agent and the opponent were to play their (possibly stochastic) policies a large number of times, what would be the average utility? Remember, since we are working with zero-sum games, the opponent's utility is the negative of the agent's utility.
- Given the game tree, we can recursively compute the value (expected utility) of each node in the tree. The value of a node is the weighted average of the values of the children where the weights are given by the probabilities of taking various actions given by the policy at that node.

Game evaluation recurrence

Analogy: recurrence for policy evaluation in MDPs



Value of the game:

$$V_{\text{eval}}(s) = \begin{cases} \text{Utility}(s) & \text{IsEnd}(s) \\ \sum_{a \in \text{Actions}(s)} \pi_{\text{agent}}(s, a) V_{\text{eval}}(\text{Succ}(s, a)) & \text{Player}(s) = \text{agent} \\ \sum_{a \in \text{Actions}(s)} \pi_{\text{opp}}(s, a) V_{\text{eval}}(\text{Succ}(s, a)) & \text{Player}(s) = \text{opp} \end{cases}$$

- More generally, we can write down a recurrence for $V_{\text{eval}}(s)$, which is the **value** (expected utility) of the game at state s .
- There are three cases: If the game is over ($\text{IsEnd}(s)$), then the value is just the utility $\text{Utility}(s)$. If it's the agent's turn, then we compute the expectation over the value of the successor resulting from the agent choosing an action according to $\pi_{\text{agent}}(s, a)$. If it's the opponent's turn, we compute the expectation with respect to π_{opp} instead.