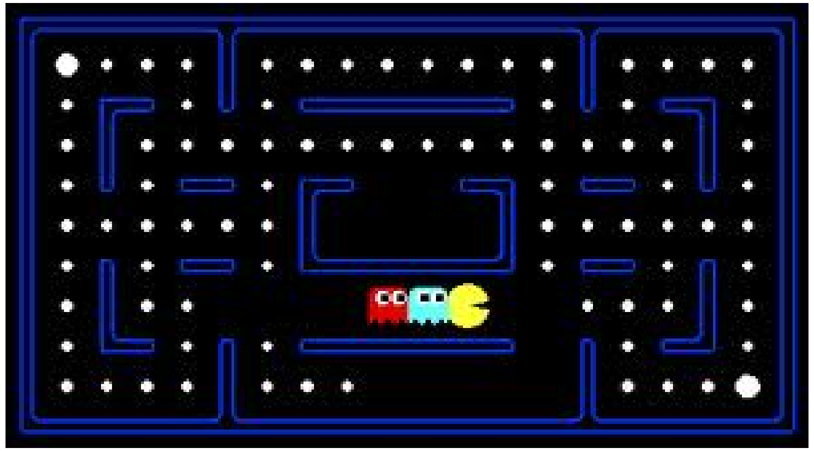
Games: minimax

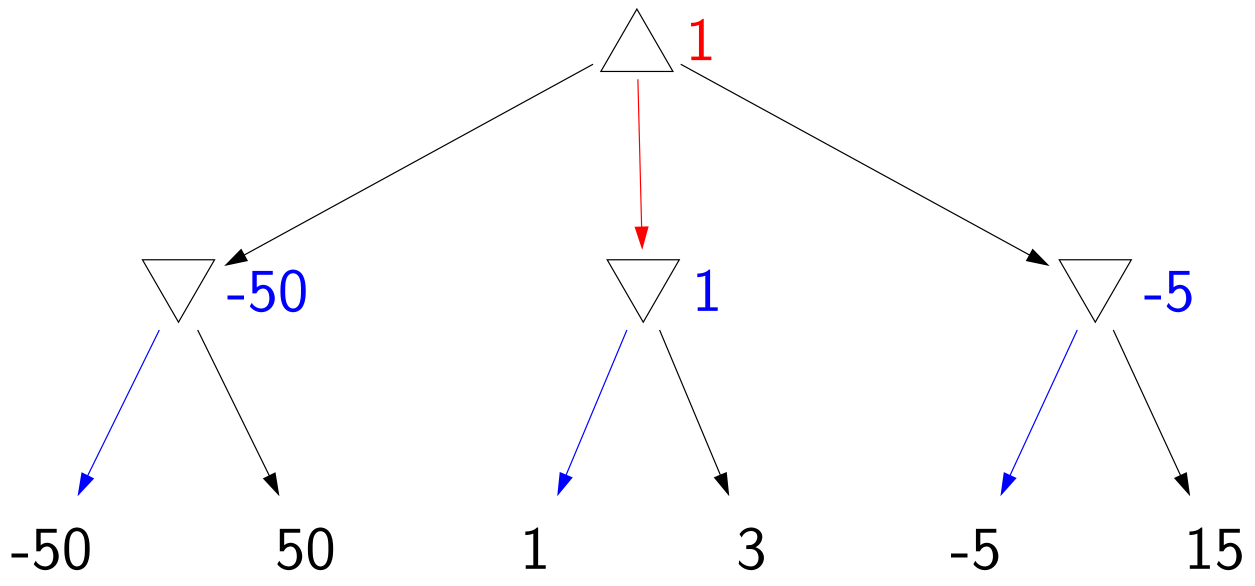


Problem: don't know opponent's policy

Approach: assume the worst case

Minimax example

Example: minimax

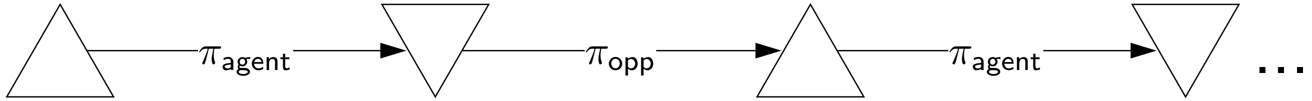

1
-50
1
-5
-50
50
1
3
-5
15

$$\boxed{V_{\text{minmax}}(s_{\text{start}}) = 1}$$

- If we could perform some mind-reading and discover the opponent's policy, then we could maximally exploit it. However, in practice, we don't know the opponent's policy. So our solution is to assume the **worst case**, that is, the opponent is doing everything to minimize the agent's utility.
- In the game tree, we use an upside-down triangle to represent **min nodes**, in which the player minimizes the value over possible actions.
- Note that the policy for the agent changes from choosing the rightmost action (expectimax) to the middle action. Why is this?

Minimax recurrence

No analogy in MDPs:


π_{agent}
π_{opp}
π_{agent}
...

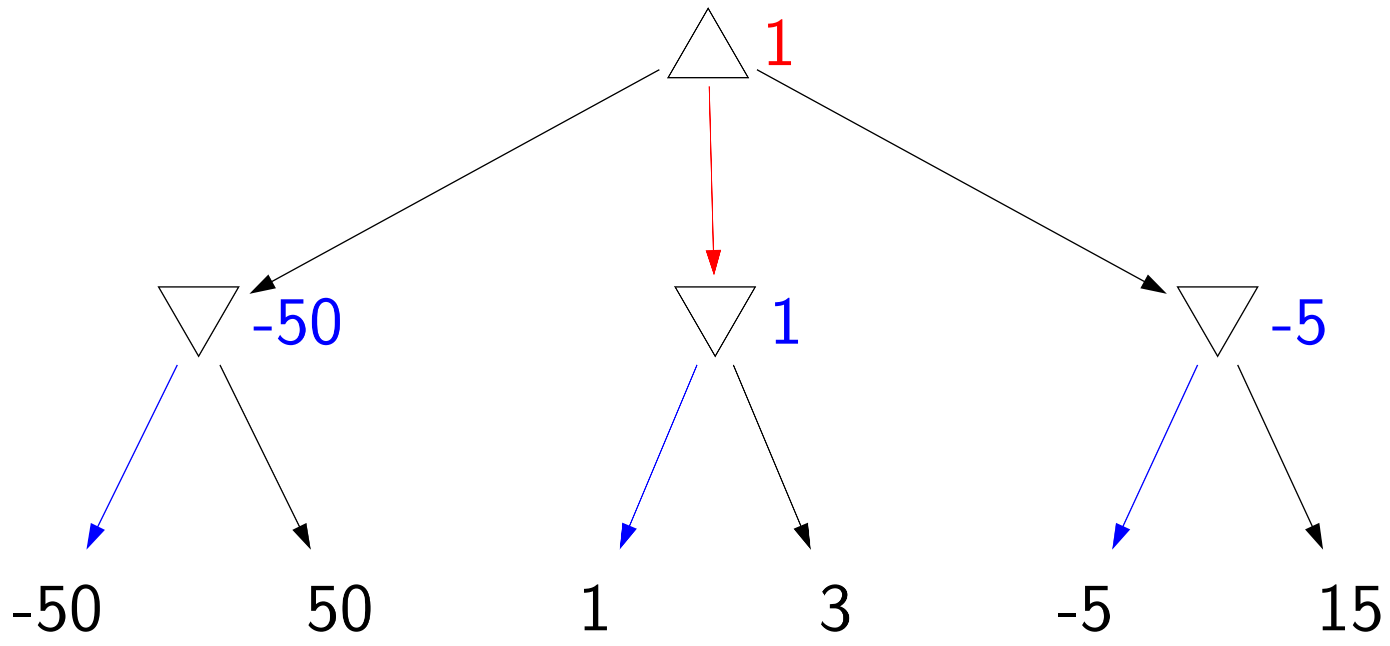
$$V_{\text{minmax}}(s) = \begin{cases} \text{Utility}(s) & \text{IsEnd}(s) \\ \color{red}{\max_{a \in \text{Actions}(s)}} V_{\text{minmax}}(\text{Succ}(s,a)) & \text{Player}(s) = \text{agent} \\ \color{blue}{\min_{a \in \text{Actions}(s)}} V_{\text{minmax}}(\text{Succ}(s,a)) & \text{Player}(s) = \text{opp} \end{cases}$$

- The general recurrence for the minimax value is the same as expectimax, except that the expectation over the opponent's policy is replaced with a minimum over the opponent's possible actions. Note that the minimax value does not depend on any policies at all: it's just the agent and opponent playing optimally with respect to each other.

Extracting minimax policies

$$\pi_{\max}(s) = \arg\max_{a \in \mathsf{Actions}(s)} V_{\mathsf{minmax}}(\mathsf{Succ}(s,a))$$

$$\pi_{\min}(s) = \arg\min_{a \in \mathsf{Actions}(s)} V_{\mathsf{minmax}}(\mathsf{Succ}(s,a))$$



- Having computed the minimax value $V_{\textsf{minmax}}$, we can extract the minimax policies $\pi_{\max}$ and $\pi_{\min}$ by just taking the action that leads to the state with the maximum (or minimum) value.
- In general, having a value function tells you which states are good, from which it's easy to set the policy to move to those states (provided you know the transition structure, which we assume we know here).

The halving game

Problem: halving game

Start with a number N.

Players take turns either decrementing N or replacing it with $\lfloor \frac{N}{2} \rfloor$.

The player that is left with 0 wins.

[semi-live solution: `minimaxPolicy`]

Face off

Recurrences produces policies:

$$\begin{aligned} V_{\text{exptmax}} \quad &\Rightarrow \quad \pi_{\text{exptmax(7)}}, \pi_7 \text{ (some opponent)} \\ V_{\text{minmax}} \quad &\Rightarrow \quad \pi_{\max}, \pi_{\min} \end{aligned}$$

Play policies against each other:

	$\pi_{\min}$	π_7
$\pi_{\max}$	$V(\pi_{\max}, \pi_{\min})$	$V(\pi_{\max}, \pi_7)$
$\pi_{\text{exptmax(7)}}$	$V(\pi_{\text{exptmax(7)}}, \pi_{\min})$	$V(\pi_{\text{exptmax(7)}}, \pi_7)$

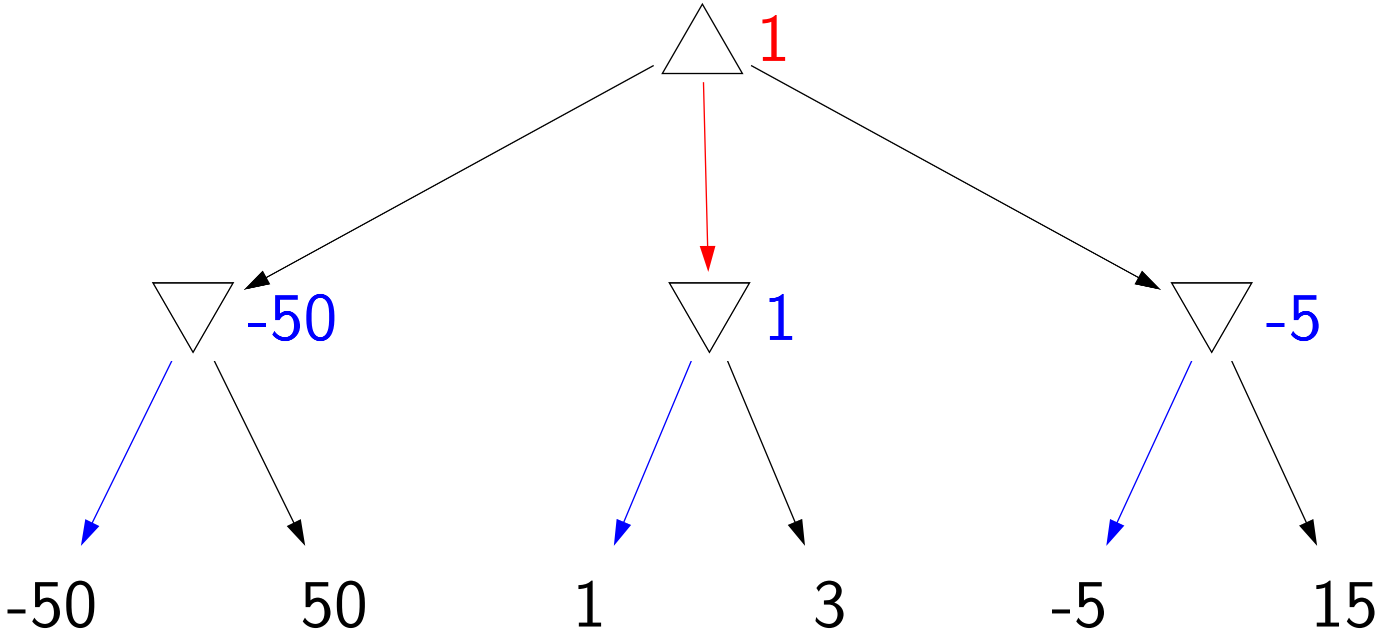
What's the relationship between these values?

- So far, we have seen how expectimax and minimax recurrences produce policies.
- The expectimax recurrence computes the best policy $\pi_{\mathsf{exptmax(7)}}$ against a fixed opponent policy (say π_7 for concreteness).
- The minimax recurrence computes the best policy $\pi_{\max}$ against the best opponent policy $\pi_{\min}$.
- Now, whenever we take an agent policy π_{agent} and an opponent policy π_{opp}, we can play them against each other, which produces an expected utility via game evaluation, which we denote as $V(\pi_{\mathsf{agent}}, \pi_{\mathsf{opp}})$.
- How do the four game values of different combination of policies relate to each other?

Minimax property 1

Proposition: best against minimax opponent

$V(\pi_{\max}, \pi_{\min}) \geq V(\pi_{\text{agent}}, \pi_{\min})$ for all π_{agent}

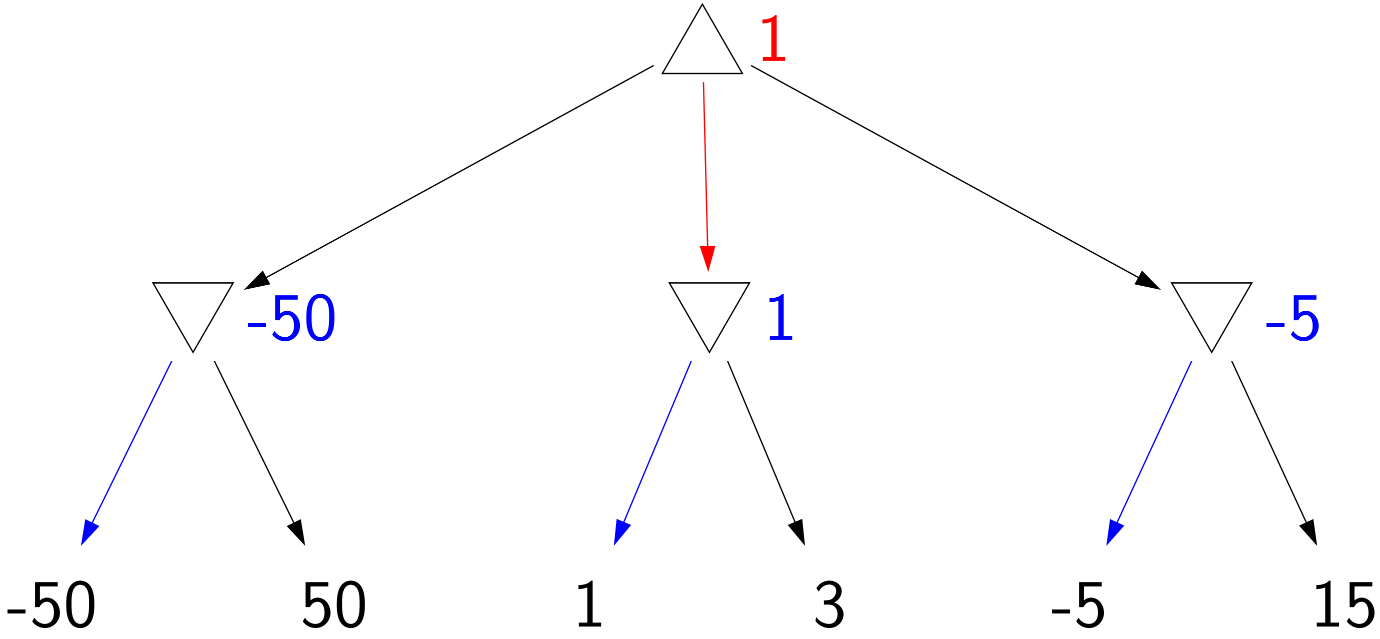

1
-50
1
-5
-50
50
1
3
-5
15

- Recall that $\pi_{\max}$ and $\pi_{\min}$ are the minimax agent and opponent policies, respectively. The first property is if the agent were to change her policy to any π_{agent}, then the agent would be no better off (and in general, worse off).
- From the example, it's intuitive that this property should hold. To prove it, we can perform induction starting from the leaves of the game tree, and show that the minimax value of each node is the highest over all possible policies.

Minimax property 2

Proposition: lower bound against any opponent

$V(\pi_{\max}, \pi_{\min}) \leq V(\pi_{\max}, \pi_{\text{opp}})$ for all π_{opp}

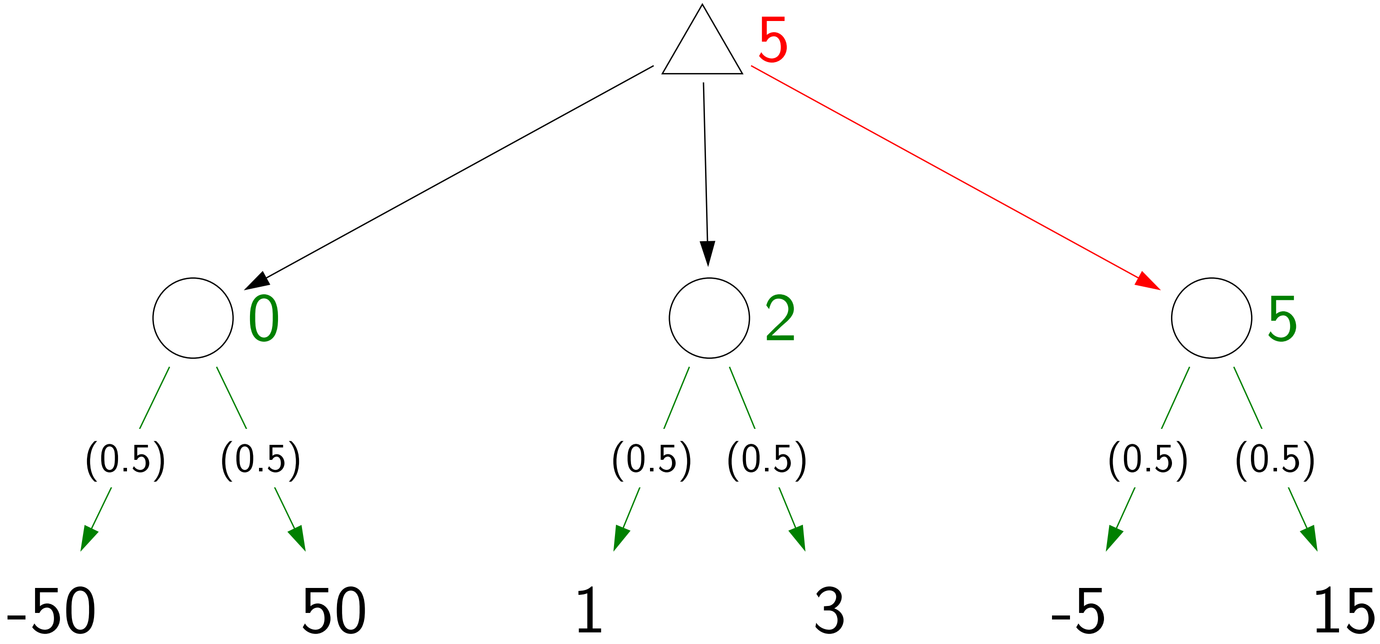

1
-50
1
-5
-50
50
1
3
-5
15

- The second property is the analogous statement for the opponent: if the opponent changes his policy from $\pi_{\min}$ to $\pi_{\textsf{opp}}$, then he will be no better off (the value of the game can only increase).
- From the point of the view of the agent, this can be interpreted as guarding against the worst case. In other words, if we get a minimax value of 1, that means no matter what the opponent does, the agent is guaranteed at least a value of 1. As a simple example, if the minimax value is $+\infty$, then the agent is guaranteed to win, provided it follows the minimax policy.

Minimax property 3

Proposition: not optimal if opponent is known

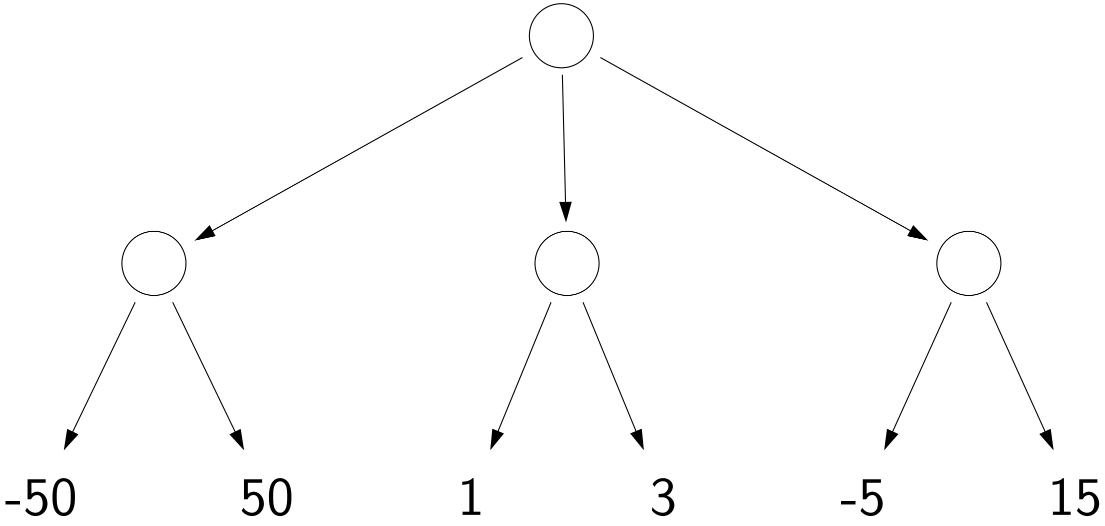
$V(\pi_{\max}, \pi_7) \leq V(\pi_{\text{exptmax}(7)}, \pi_7)$ for opponent π_7



5
0
2
5
(0.5)
(0.5)
(0.5)
(0.5)
(0.5)
(0.5)
-50
50
1
3
-5
15

- However, following the minimax policy might not be optimal for the agent if the opponent is known to be not playing the adversarial (minimax) policy.
- Consider the running example where the agent chooses A, B, or C and the opponent chooses a bin. Suppose the agent is playing $\pi_{\max}$, but the opponent is playing a stochastic policy π_7 corresponding to choosing an action uniformly at random.
- Then the game value here would be 2 (which is larger than the minimax value 1, as guaranteed by property 2). However, if we followed the expectimax $\pi_{\text{exptmax(7)}}$, then we would have gotten a value of 5, which is even higher.

Relationship between game values



$$\begin{array}{cccc}
 & \pi_{\min} & & \pi_7 \\
\pi_{\max} & \begin{array}{c} V(\pi_{\max}, \pi_{\min}) \\ 1 \end{array} & \leq & \begin{array}{c} V(\pi_{\max}, \pi_7) \\ 2 \end{array} \\
 & \geq & & \leq \\
\pi_{\text{exptmax(7)}} & \begin{array}{c} V(\pi_{\text{exptmax(7)}}, \pi_{\min}) \\ \text{-5} \end{array} & & \begin{array}{c} V(\pi_{\text{exptmax(7)}}, \pi_7) \\ 5 \end{array}
\end{array}$$

- Putting the three properties together, we obtain a chain of inequalities that allows us to relate all four game values.
- We can also compute these values concretely for the running example.