Implementation: Value Iteration

In the previous concept, you learned about **value iteration**. In this algorithm, each sweep over the state space effectively performs both policy evaluation and policy improvement. Value iteration is guaranteed to find the optimal policy π_* for any finite MDP.

The pseudocode can be found below.

Value Iteration

Input: MDP, small positive number θ
Output: policy $\pi \approx \pi_*$
Initialize V arbitrarily (e.g., $V(s) = 0$ for all $s \in \mathcal{S}^+$)
repeat
  $\Delta \leftarrow 0$
  for $s \in \mathcal{S}$ **do**
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a)(r + \gamma V(s'))$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
  end
until $\Delta < \theta$;
$\pi \leftarrow$ **Policy_Improvement**(MDP, V)
return π

Note that the stopping criterion is satisfied when the difference between successive value function estimates is sufficiently small. In particular, the loop terminates if the difference is less than θ for each state. And, the closer we want the final value function estimate to be to the optimal value function, the smaller we need to set the value of θ.