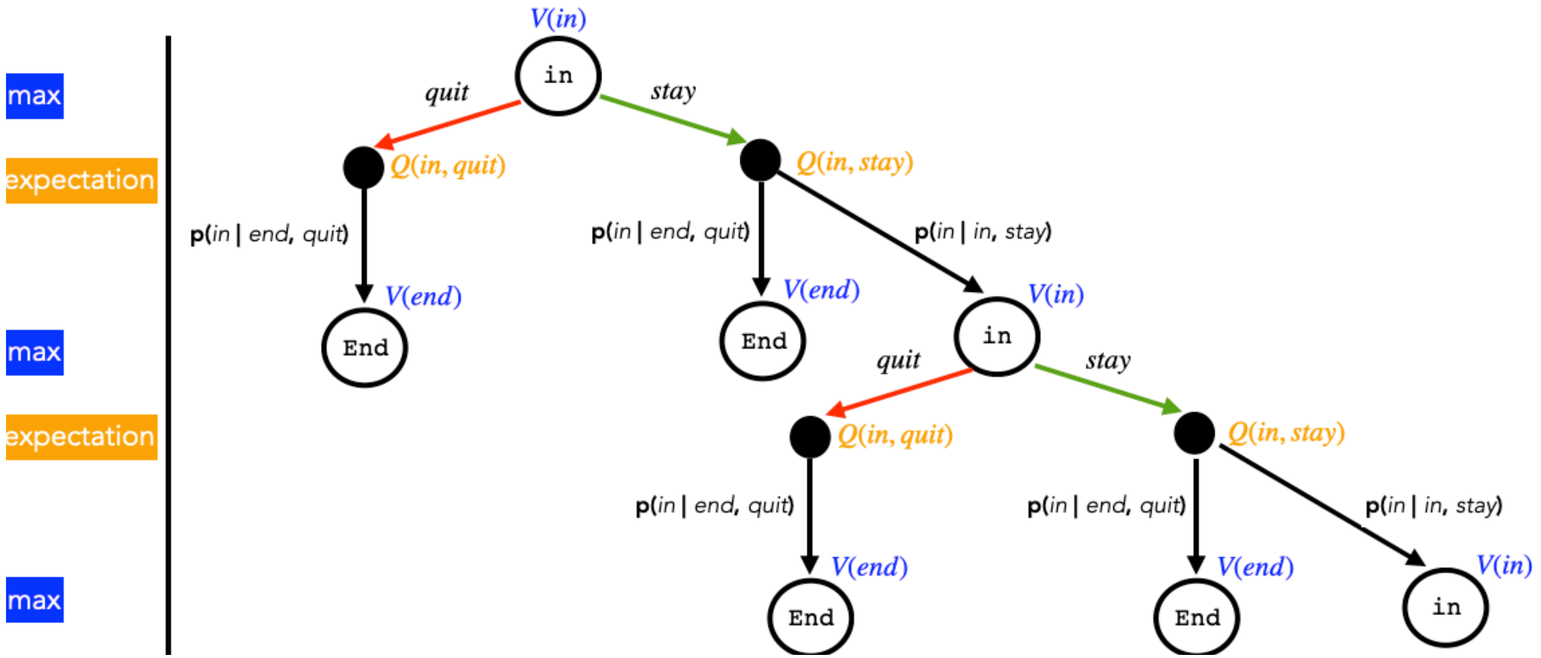
CS143: Artificial Intelligence



Value Iteration Algorithm

Drake UNIVERSITY

What is the MDP for Frozen Lake?

- The transition function encodes the movements in the Lake
  - In Frozen Lake, model or dynamics is known
  - $P(s'|a,s)$ is read as agent transitions from state s to s' by taking action a

- How many states are there?
  - How can you show the probability transitions $P(s'|a,s)$ from each state to the next?

- Let's simplify and consider a simpler problem where there only a couple of states
  - Let's consider a simple dice game with 2 states only

MDP for a Simple Dice Game with 2 States

MDP for a Simple Dice Game

- For each round t = 1,2,3, …
  - You choose stay or quit
    - If quit, you get $10 and we end the game
    - If stay, you get $4 and then I roll a 6-sided dice
      - If the dice results in 1 or 2, we end the game
      - Otherwise, if the dice results in 3, 4, 5, or 6, we continue to the next round

Dice Game: Rewards for following policy ‘stay’

- For each round t = 1,2,3,
  - You choose stay or quit
    - If quit, you get $10 and we end the game
    - If stay, you get $4 and then I roll a 6-sided dice
      - If the dice results in 1 or 2, we end the game
      - Otherwise, if the dice results in 3, 4, 5, or 6, we continue to the next round

Dice Game: Rewards for following policy ‘stay’

- You choose stay or quit
  - If quit, you get $10 and we end the game
  - If stay, you get $4 and then I roll a 6-sided dice
    - If the dice results in 1 or 2, we end the game
    - Otherwise, if the dice results in 3, 4, 5, or 6, we continue to the next round

- Expected reward if you follow policy stay
  - round r=1: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 2
  - round r=2: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 3
  - round r=3: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 4
  - …
- In other words, we need to compute the expected reward for the following sequence of action selections

stay , stay , stay , … stay

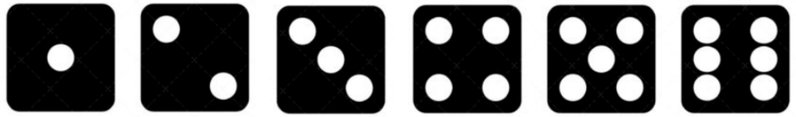
Dice Game: Rewards for following policy ‘stay’

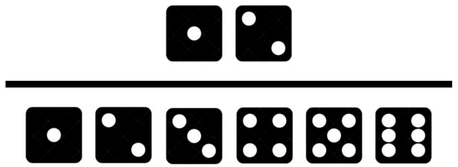
- You choose stay or quit
  - If quit, you get $10 and we end the game
  - If stay, you get $4 and then I roll a 6-sided dice
    - If the dice results in 1 or 2, we end the game
    - Otherwise, if the dice results in 3, 4, 5, or 6, we continue to the next round

- Let’s find the expected reward for the first two rounds if we choose to **stay**
  - round r=1: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 2

- How many ways can we roll the dice so that we end up exiting from round 1?

- How many sides are there in the dice?



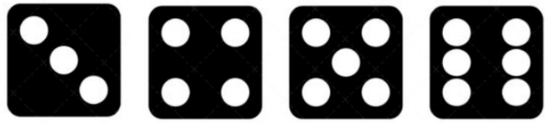
- Probability of exiting from round 1: $P_{Round_1} =$  $= \frac{2}{6} = \frac{1}{3}$

Dice Game: Rewards for following policy ‘stay’

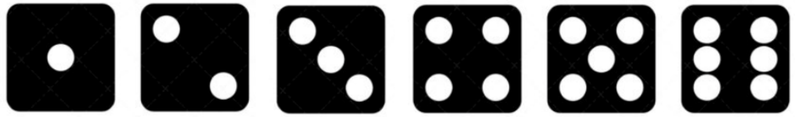
- You choose stay or quit
  - If quit, you get $10 and we end the game
  - If stay, you get $4 and then I roll a 6-sided dice
    - If the dice results in 1 or 2, we end the game
    - Otherwise, if the dice results in 3, 4, 5, or 6, we continue to the next round

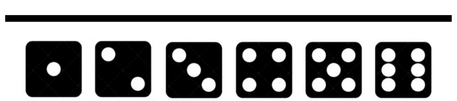
- Let’s find the expected reward for the first two rounds if we to choose to **stay**
  - round r=1: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 2

- How many ways can we roll the dice so that we end up in round 2?



- How many sides are there in the dice?



- Probability of going to round 2: $P_{Round_2} =$  $=$  $= \frac{4}{6} = \frac{2}{3}$

Expected Value (recall from expectiminimax)

- The ***expected value*** of a random event is the average value produced by that event with respect to the probability of each possible outcome
- In terms of equation, the expected value of a (discrete) random variable X

$$E[X] = \sum_{x \in Val(X)} xP(x)$$

Expected Values

- **Example 1:** The expected number of dots showing on a six-sided die with random variable X is denoted by E[X]:

$$\left(1 \cdot \frac{1}{6}\right) + \left(2 \cdot \frac{1}{6}\right) + \left(3 \cdot \frac{1}{6}\right) + \left(4 \cdot \frac{1}{6}\right) + \left(5 \cdot \frac{1}{6}\right) + \left(6 \cdot \frac{1}{6}\right) = 3.5$$

Dice Game: Rewards for following policy ‘stay’

- Expected reward for the first two rounds if we choose to stay
  - round r=1: let’s say the dice results in either 1, or 2, we exit from the round 1
  - round r=1: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 2

- Expected reward after round 1 and round 2

$$P_{Round_1} * reward_{Round_1} + P_{Round_2} * (reward_{Round_1} + reward_{Round_2})$$

$$= \frac{1}{3}(4) + \frac{2}{3}(4 + reward_{Round_2})$$

Dice Game: Rewards for following policy ‘stay’

- Expected reward if you follow policy stay
  - round r=1: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 2
  - round r=2: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 3
  - round r=3: let’s say the dice results in either 3, 4, 5, or 6, we continue to the round 4
  - …
- If we continue the same process for the remaining rounds, the expected reward for the following sequence of action selections would be

stay , stay , stay , … stay

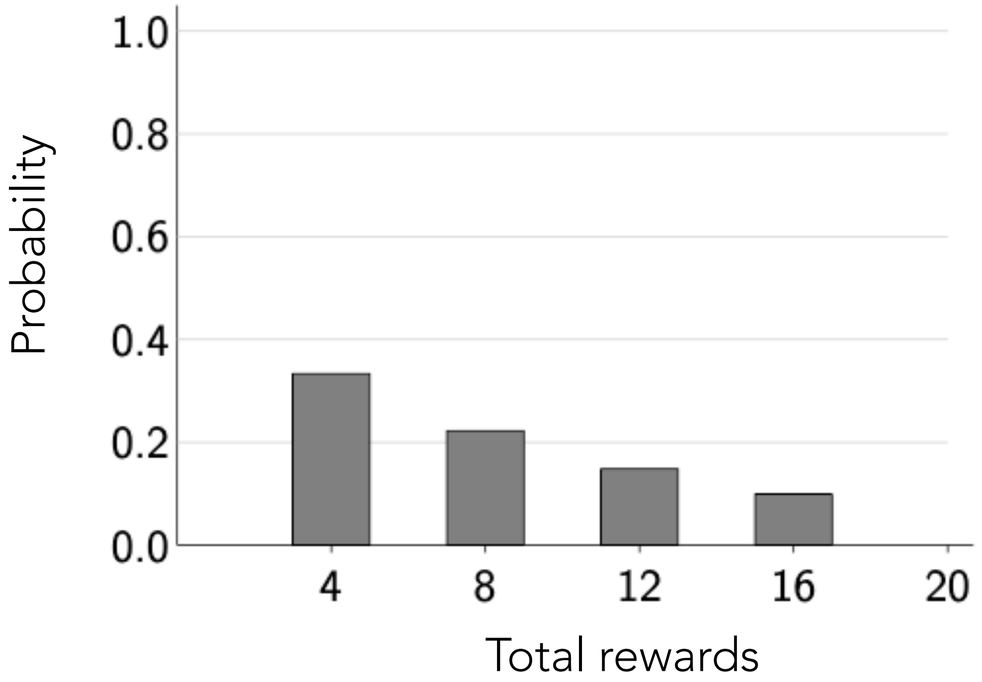
$$= \frac{1}{3}(4) + \frac{2}{3}.\frac{1}{3}.(4+4) + \frac{2}{3}.\frac{2}{3}.\frac{1}{3}.(4+4+4) + \ldots$$

$$= \frac{1}{3}(4) + \frac{2}{3}.\frac{1}{3}.(8) + \frac{2}{3}.\frac{2}{3}.\frac{1}{3}.(12) + \ldots$$

$$= 12$$

Summary: Rewards for following policy ‘stay’

- For following the policy stay



- Expected rewards

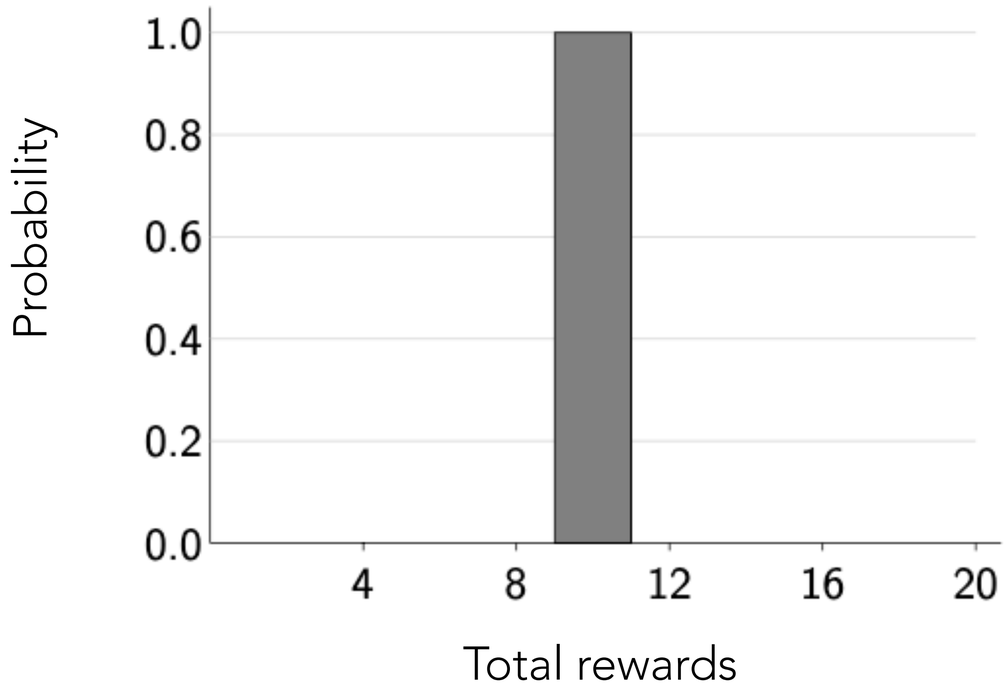
$$\begin{aligned} &\text{stay} \quad , \quad \text{stay} \quad , \quad \text{stay} \quad , \ldots \quad \text{stay} \\ &= \frac{1}{3}(4) + \frac{2}{3}.\frac{1}{3}.(8) + \frac{2}{3}.\frac{2}{3}.\frac{1}{3}.(12) + \ldots \\ &= 12 \end{aligned}$$

Dice Game: Rewards for following policy ‘quit’

- For each round t = 1,2,3,
  - You choose stay or quit
    - If quit, you get $10 and we end the game
    - If stay, you get $4 and then I roll a 6-sided dice
      - If the dice results in 1 or 2, we end the game
      - Otherwise, if the dice results in 3, 4, 5, or 6, we continue to the next round

Dice Game: Rewards for following policy ‘quit’

- For following the policy quit



- Expected rewards

quit

$$= 1.0 * (10)$$
$$= 10$$

Expected rewards: 10.0

MDP for a Simple Dice Game

- You choose stay or quit
  - If quit, you get $10 and we end the game
  - If stay, you get $4 and then I roll a 6-sided dice
    - If the dice results in 1 or 2, we end the game
    - Otherwise, if the dice results in 3, 4, 5, or 6, we continue to the next round

WHAT IS THE BEST STRATEGY FOR THIS GAME?

stay

Expected rewards:

12.0

quit

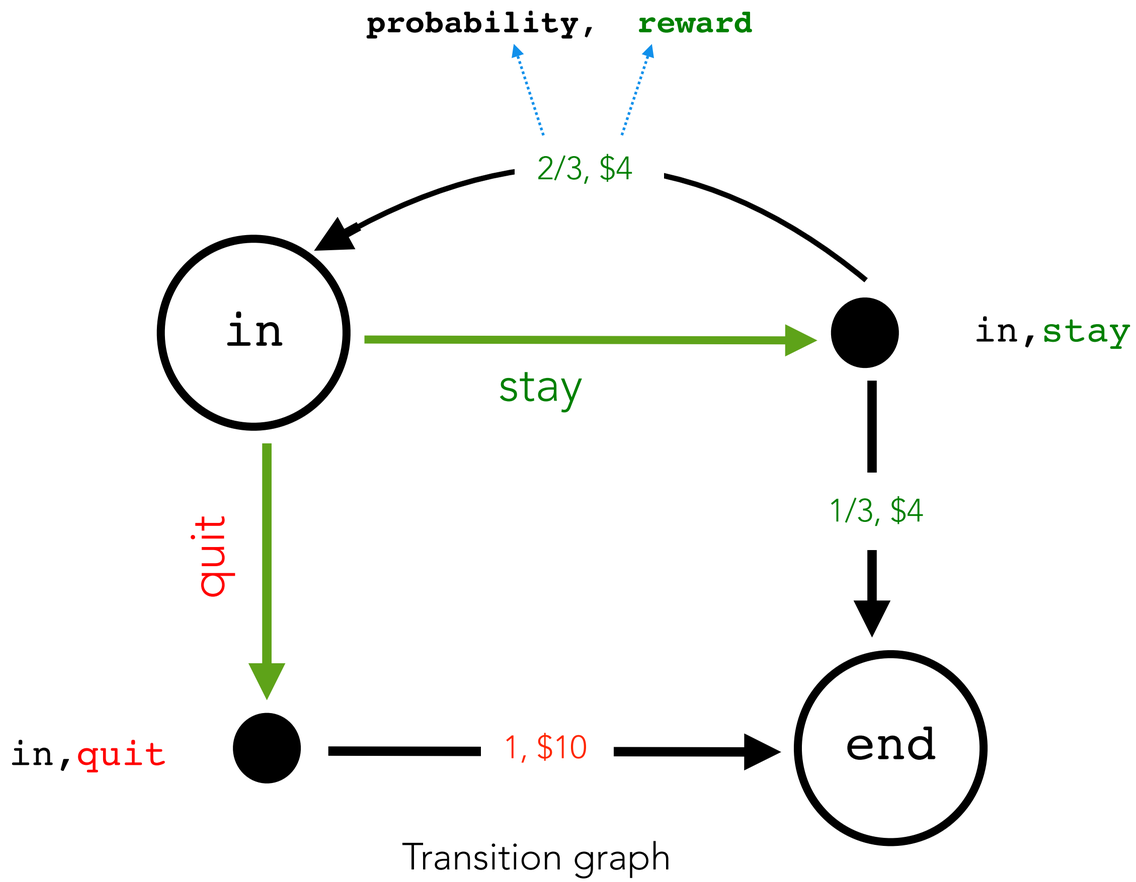
Expected rewards:

10.0

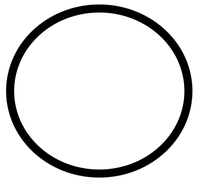
If you quit, then you'll get a reward of $10. Therefore, the stay strategy is preferred, even though sometimes you'll get less than $10.

MDP for a Simple Dice Game

- Let us formalize the dice game as a Markov decision process (MDP)



Transition graph



state node

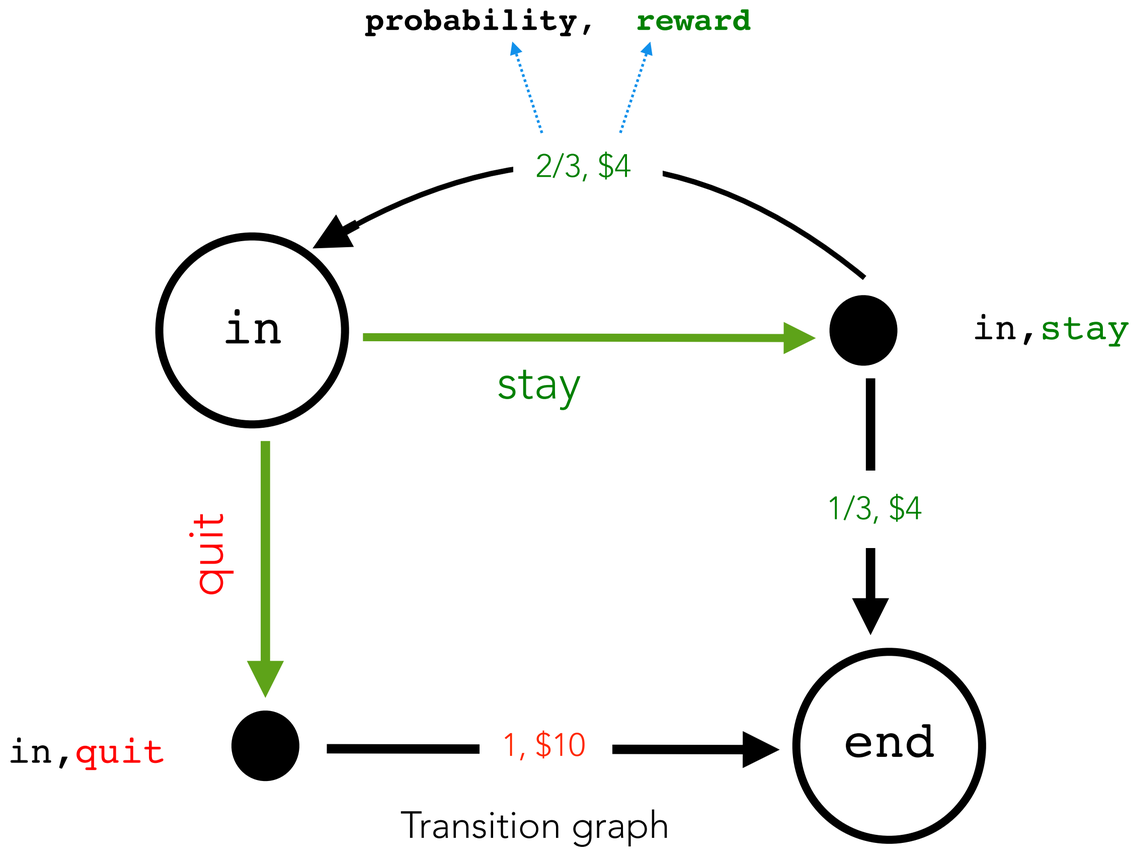
- Edges coming out of a state node are the possible actions from that state node leading to a action node

action node

- Edges coming out of a action node are the possible random outcomes of that action, which end up back in state nodes
- Sum of probabilities coming out of a action node is 1.0

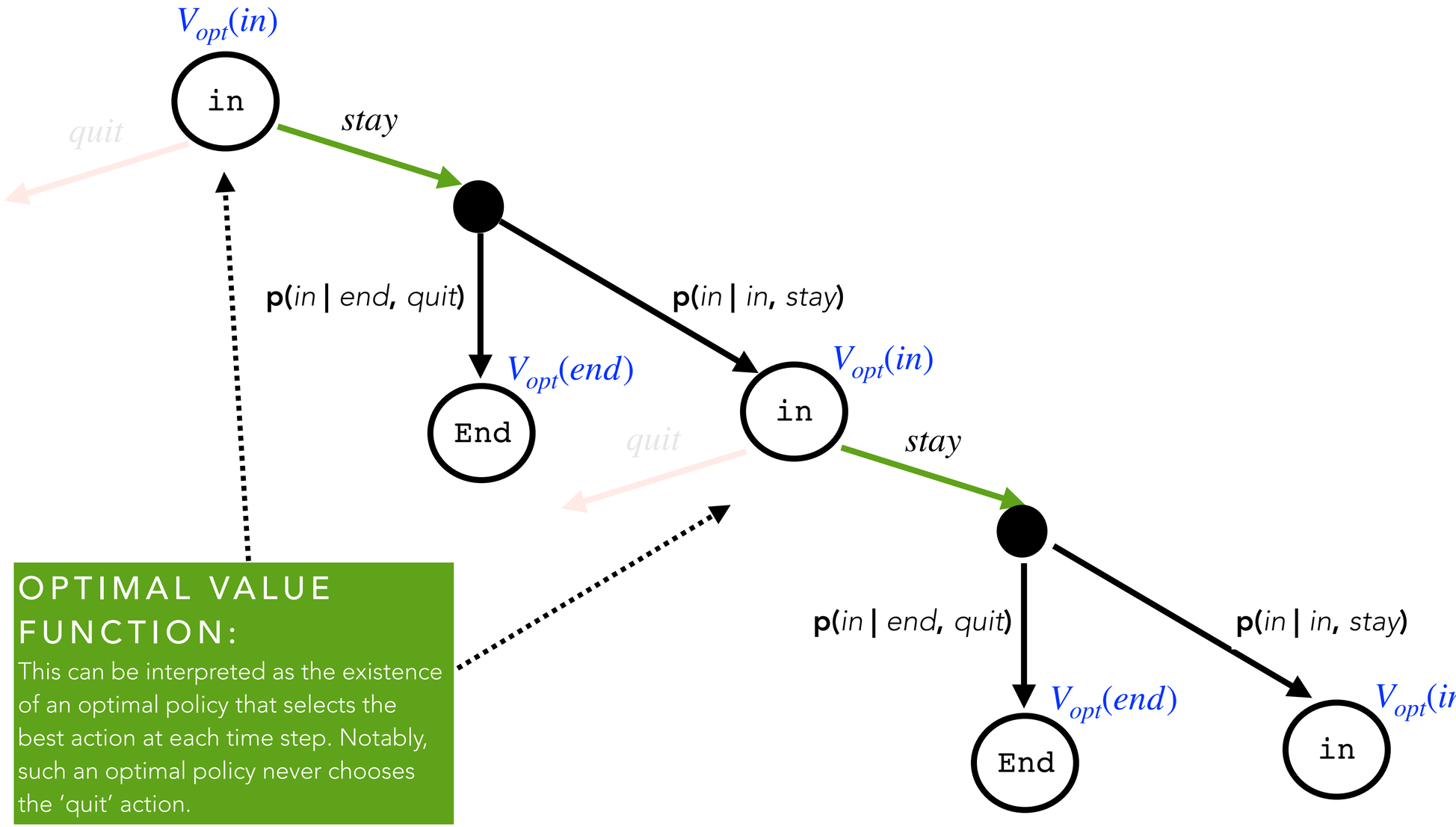
Optimal state-value or value function

- **value function** $V_{opt}(s)$: estimate of *how good* it is to be in a given state ***s*** and acting optimally
- $V_{opt}(s)$ is the expected reward received when starting in state ***s*** and following optimal policy π_{opt} thereafter



Transition graph

Optimal state-value or value function

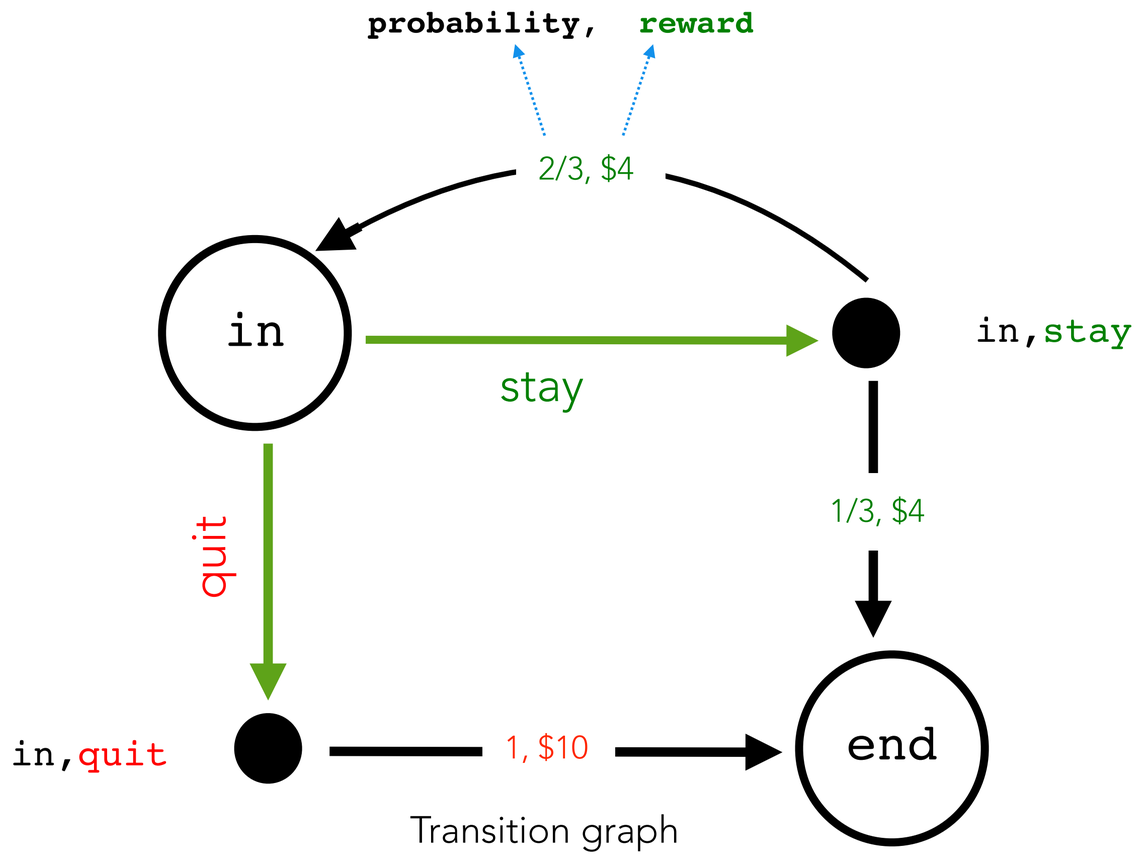


OPTIMAL VALUE FUNCTION:

This can be interpreted as the existence of an optimal policy that selects the best action at each time step. Notably, such an optimal policy never chooses the 'quit' action.

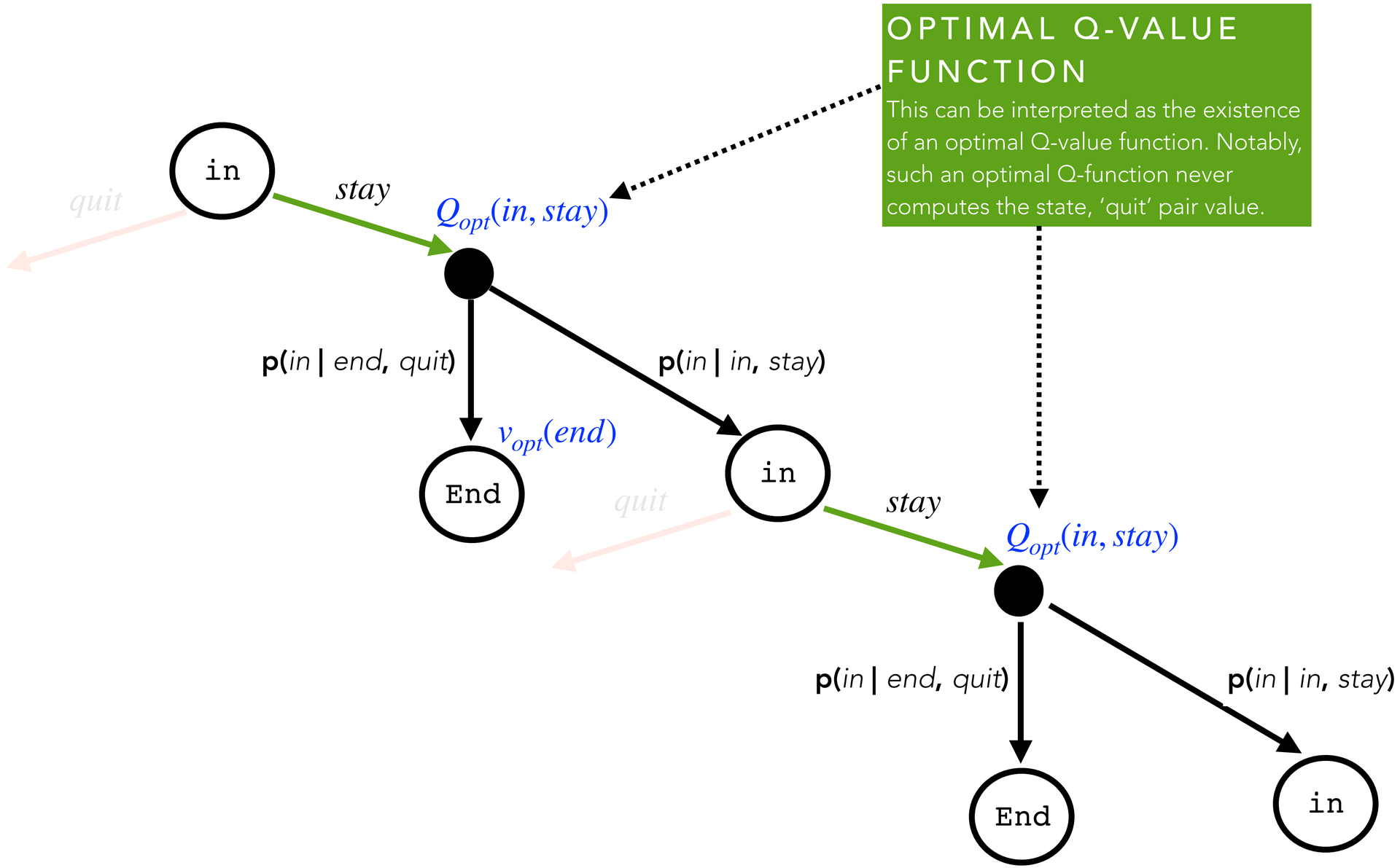
Optimal action-value or q-value function

- **q-value function** $Q_{opt}(s, a)$: estimate of the expected return obtained by taking action ***a*** in state ***s***, and thereafter following the optimal policy π_{opt} optimally thereafter



Transition graph

Optimal action-value or q-value function

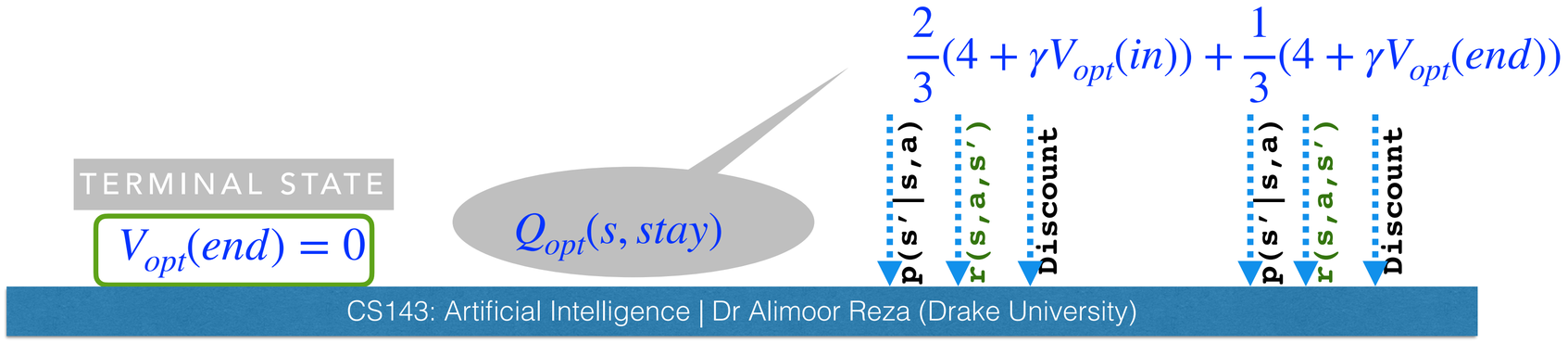


Relationship: value and q-value functions

- Looks like from the previous to visualizations, the relationship between the **value function** and the **Q-value** function can be expressed recursively. For example, from the previous tree graph, it can be seen that the value function can be computed by taking the maximum over multiple Q-values corresponding to different actions. Mathematically, this can be written as follows:

$$V_{opt}(s) = \max \begin{Bmatrix} Q_{opt}(s, quit) \\ Q_{opt}(s, stay) \end{Bmatrix}$$

$$= \max \begin{Bmatrix} P(in \mid s, quit)[R(s, quit, in) + \gamma V_{opt}(in)] + P(end \mid s, quit)[R(s, quit, end) + \gamma V_{opt}(end)] \\ P(in \mid s, stay)[R(s, stay, in) + \gamma V_{opt}(in)] + P(end \mid s, stay)[R(s, stay, end) + \gamma V_{opt}(end)] \end{Bmatrix}$$



Relationship: value and q-value functions

- In general, we can write the value function can be computed by taking the maximum over multiple Q-values corresponding to different actions. Mathematically, this can be written as follows:

$$V_{opt}(s) = \max_a Q_{opt}(s,a)$$
$$= \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V_{opt}(s')]$$

Then how do you solve an MDP?

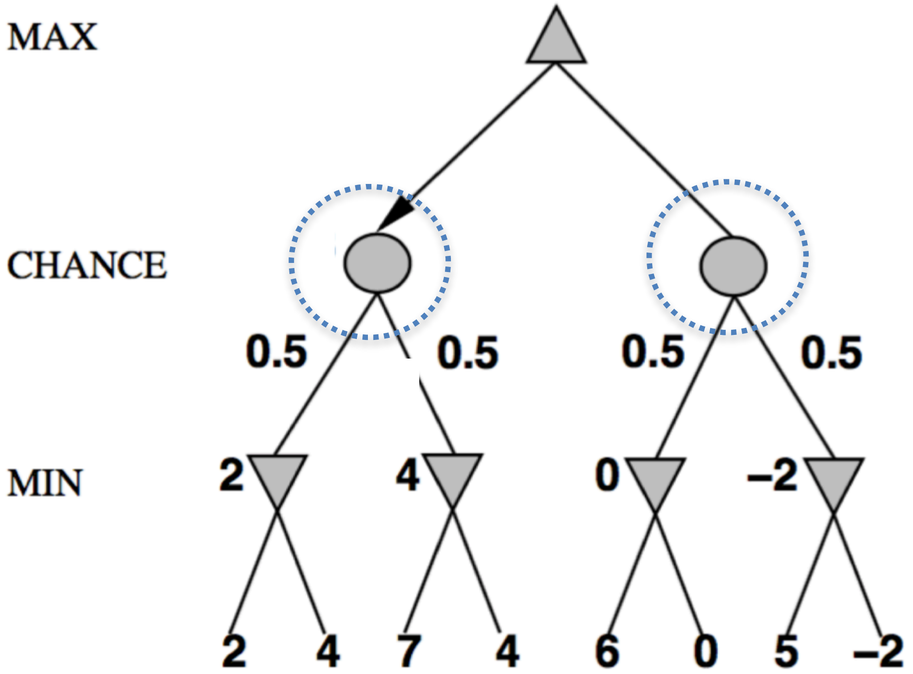
Solution to an MDP is called a **‘policy’**

Answer: Value Iteration Algorithm

An iterative algorithm to find the **optimal policy** (best solution) for an MDP

Recall: Expectiminimax

- We can apply a version of minimax by computing expected values at chance nodes



Recall: Expectiminimax Algorithm

- The utility of a MAX/MIN node in the game tree is the **max/min** of the utility values of its successor

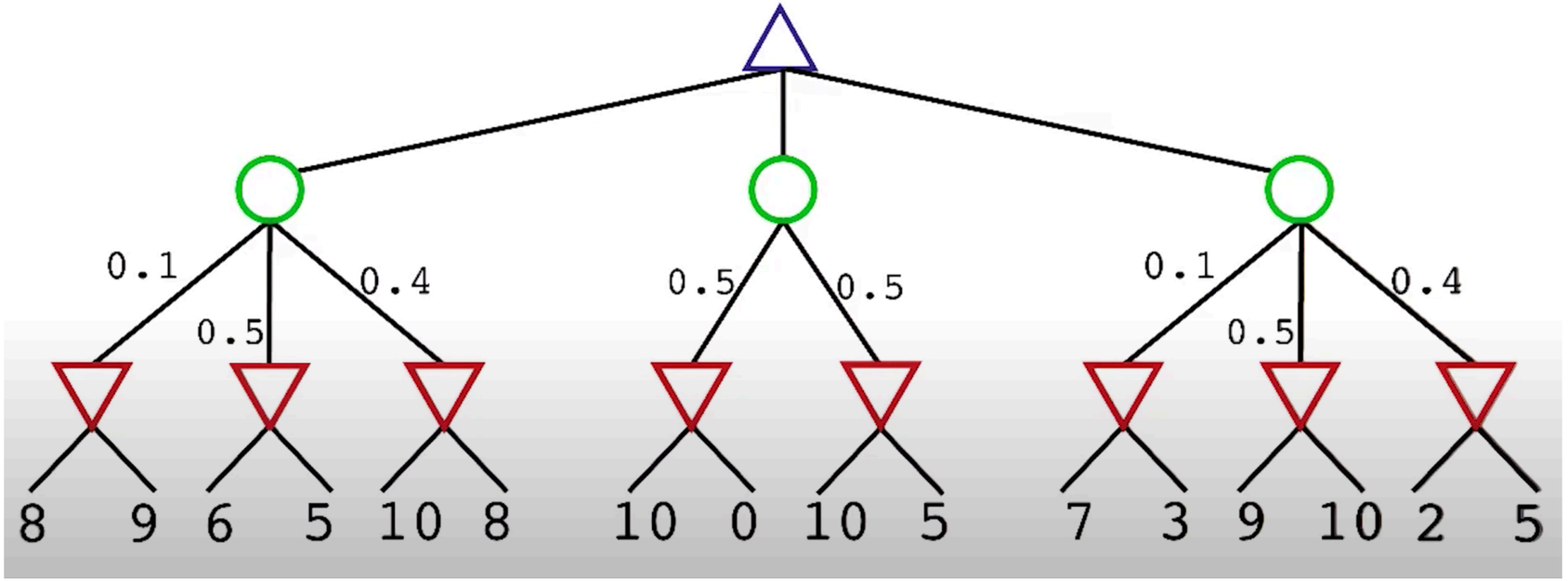
- The expected utility of a CHANCE node is the **expected value** of the utility values of its successors

$$\text{ExpectedValue}(s) = \sum_{s' \in \text{SUCC}(s)} \text{ExpectedValue}(s')\, P(s') \qquad \textbf{CHANCE nodes}$$

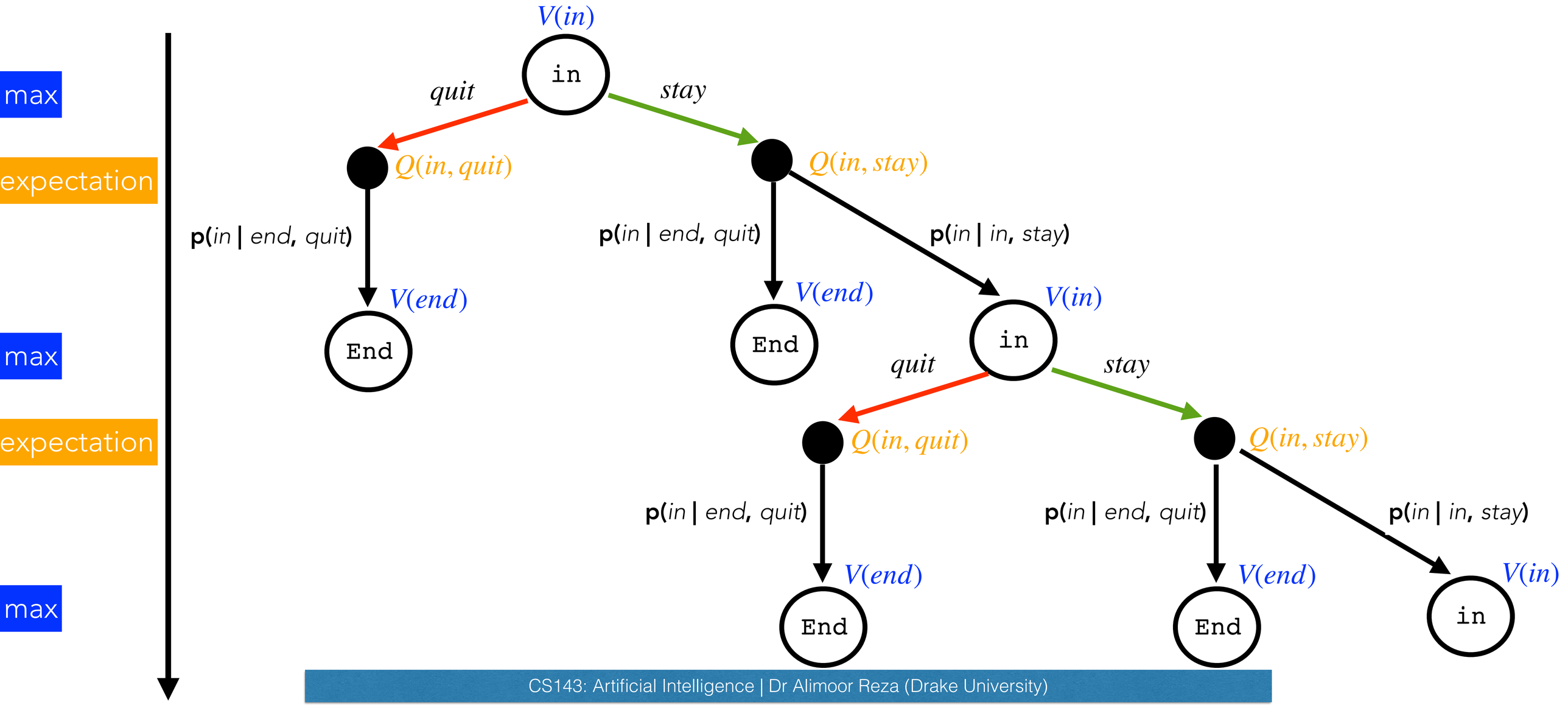
$$\text{MinimaxValue}(s) = \max_{s' \in \text{SUCC}(s)} \text{MinimaxValue}(s') \qquad \textbf{MAX nodes}$$

$$\text{MinimaxValue}(s) = \min_{s' \in \text{SUCC}(s)} \text{MinimaxValue}(s') \qquad \textbf{MIN nodes}$$

Recall: Expectiminimax Activity



Build an Expectiminimax-like search tree



Optimal value function

- The optimal (or best) value $V_{opt}(s)$ is the maximum value attained among all possible policies for a given MDP

- The optimal value from state s can be formulated as

$$V_{opt}(s) = \max_{a \in actions(s)} Q_{opt}(s, a)$$

$$= \max_{a \in actions(s)} \sum_{s'} p(s' \mid s, a)[r(s, a, s') + \gamma V_{opt}(s')]$$

Value iteration algorithm

INITIALIZE VALUES WITH SOME ROUGH ESTIMATES

REFINE IT OVER AND OVER AGAIN UNTIL THEIR VALUES DO NOT CHANGE

Algorithm 2: Value Iteration Algorithm

Initialize state-value $v^0_{opt}(s)$ for all the states s

//Repeat the iterative procedure for some iterations, let's say $T = 10000$

for $t = 1$ **to** T **do**

  for *each state* s **do**

    $v^t_{opt}(s) \leftarrow \max_{a \in actions(s)} \sum_{s'} p(s'|s,a)[r(s,a,s') + \gamma v^{t-1}_{opt}(s')]$

This is because the "value iteration algorithm" is now finding the best policy out of all possible candidate policies

Coding activity: Value iteration algorithm on our MDP

TERMINAL STATE

$$V_{opt}(end) = 0$$

NON-TERMINAL STATE

$$V^t_{opt}(in) = \max \left\{ \begin{array}{l} Q^t_{opt}(in, stay) \\ Q^t_{opt}(in, quit) \end{array} \right\}$$

$$= \max \left\{ \begin{array}{c} \frac{2}{3}(4 + \gamma V^{t-1}_{opt}(in)) + \frac{1}{3}(4 + \gamma V^{t-1}_{opt}(end)) \\ \frac{1}{1}(10 + \gamma V^{t-1}_{opt}(end)) \end{array} \right\}$$

Algorithm 2: Value Iteration Algorithm

Initialize state-value $v^0_{opt}(s)$ for all the states s

//Repeat the iterative procedure for some iterations, let's say $T = 10000$

for $t = 1$ **to** T **do**

  for *each state* s **do**

    $v^t_{opt}(s) \leftarrow \max_{a \in actions(s)} \sum_{s'} p(s'|s,a)[r(s,a,s') + \gamma v^{t-1}_{opt}(s')]$

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
$V_{opt}(end)$	0	0	0	0	0	0	0	0	0	0
$\pi_{opt}(end)$	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
$V_{opt}(in)$	0									
$\pi_{opt}(in)$										