# Expectimax Search

- Why wouldn't we know what the result of an action will be?
  - Explicit randomness: rolling dice
  - Unpredictable opponents: the ghosts respond randomly
  - Actions can fail: when moving a robot, wheels might slip

- Values should now reflect average-case (expectimax) outcomes, not worst-case (minimax) outcomes

- Expectimax search: compute the average score under optimal play
  - Max nodes as in minimax search
  - Chance nodes are like min nodes but the outcome is uncertain
  - Calculate their expected utilities
  - I.e. take weighted average (expectation) of children
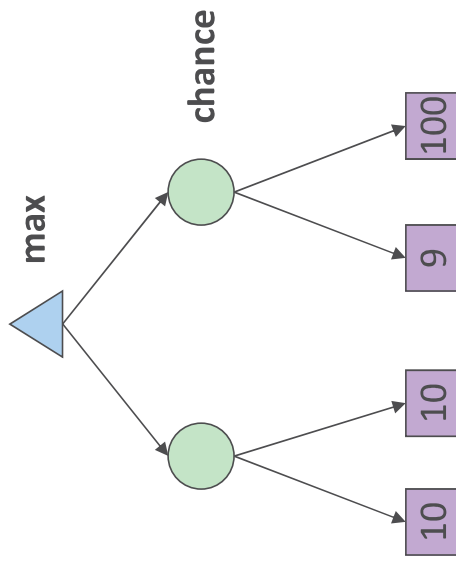
- Later, we'll learn how to formalize the underlying uncertain-result problems as Markov Decision Processes

max

chance

100

9

10

10

Expectimax Pseudocode

def value(state):
  if the state is a terminal state: return the state's utility
  if the next agent is MAX: return max-value(state)
  if the next agent is EXP: return exp-value(state)

def max-value(state):
  initialize v = -∞
  for each successor of state:
    v = max(v, value(successor))
  return v

def exp-value(state):
  initialize v = 0
  for each successor of state:
    p = probability(successor)
    v += p * value(successor)
  return v
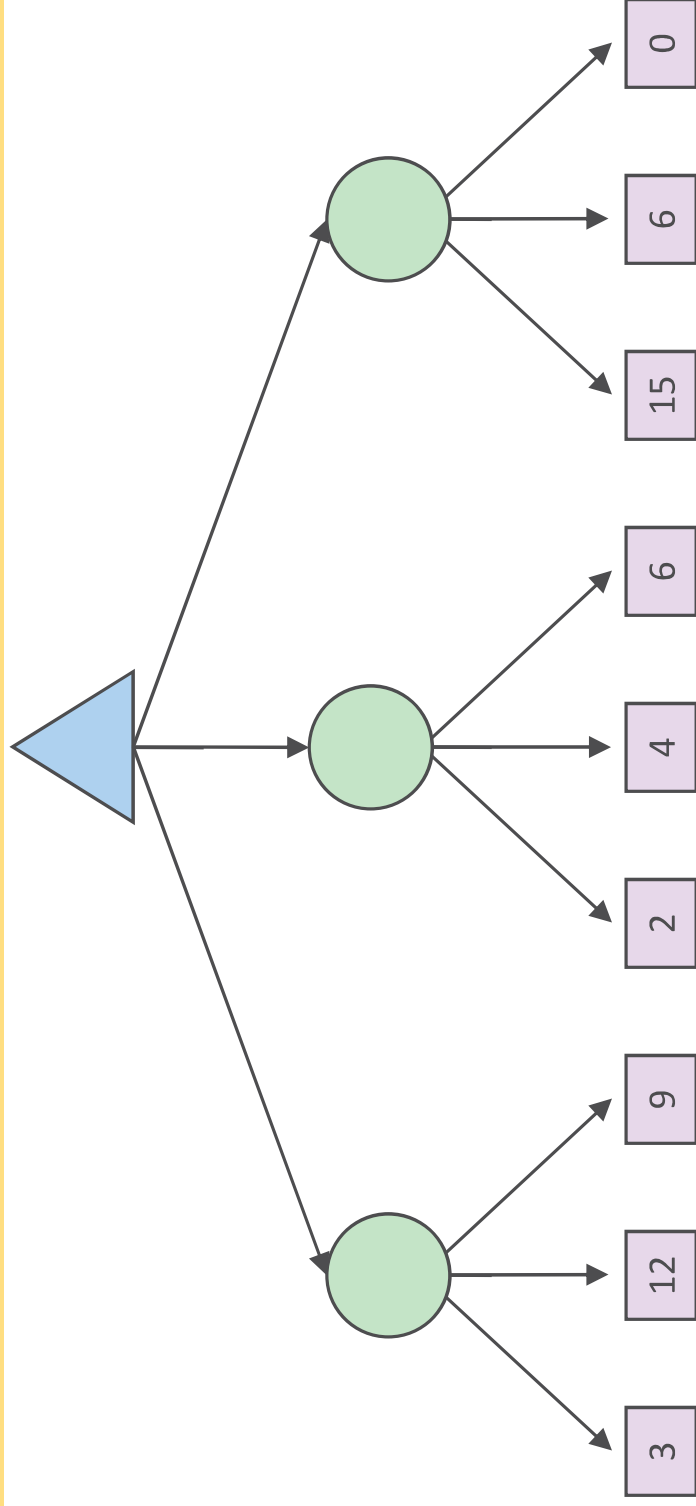
Expectimax Pseudocode

```
def exp-value(state):
    initialize v = 0
    for each successor of state:
        p = probability(successor)
        v += p * value(successor)
    return v
```
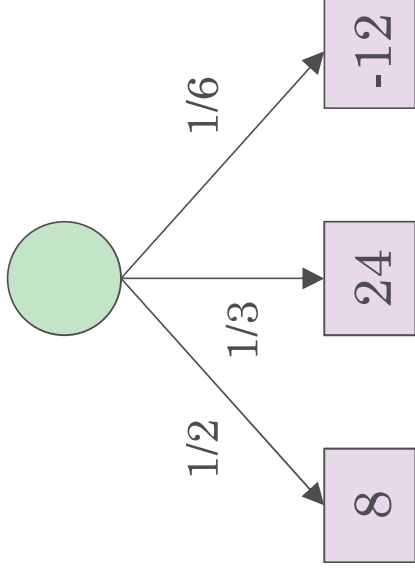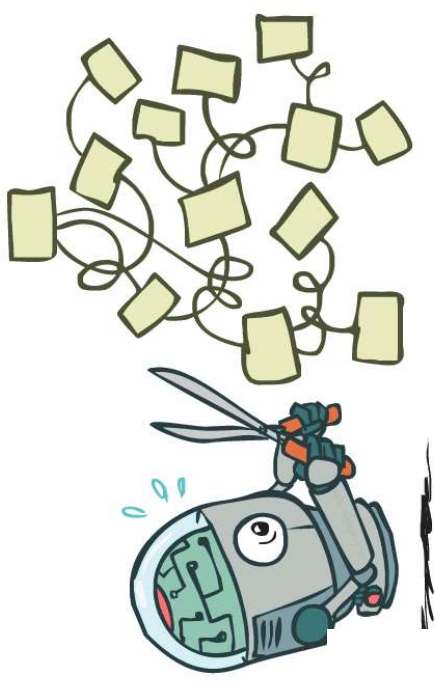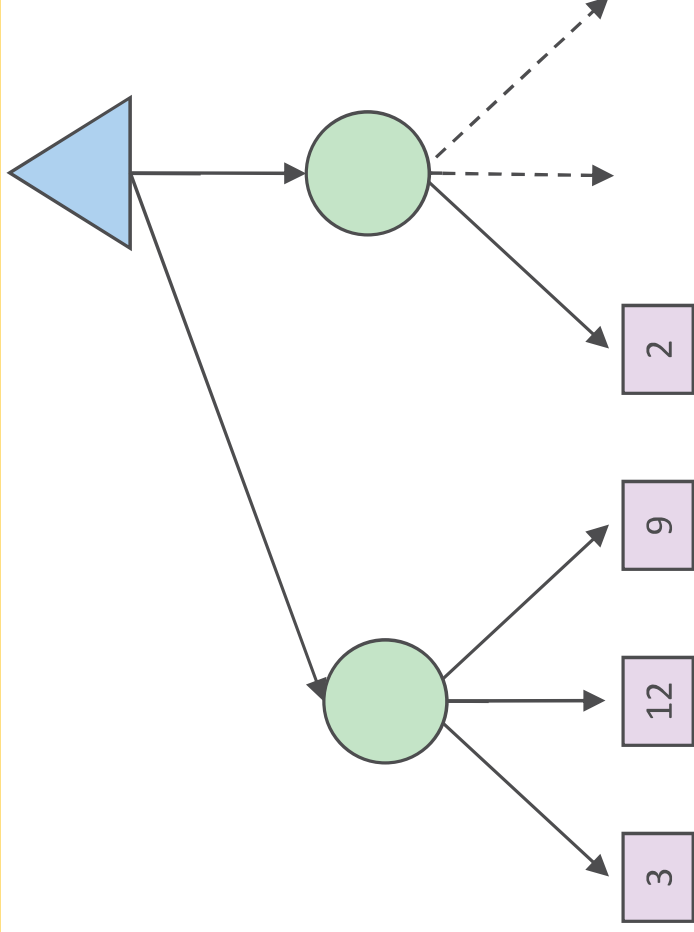
$$v = (1/2)\ (8) + (1/3)\ (24) + (1/6)\ (-12) = 10$$

Expectimax Example

Expectimax Pruning?

Depth–Limited Expectimax

Estimate of true
expectimax value
(which would require a
lot of work to compute)

300

400

492

362