Artificial Intelligence

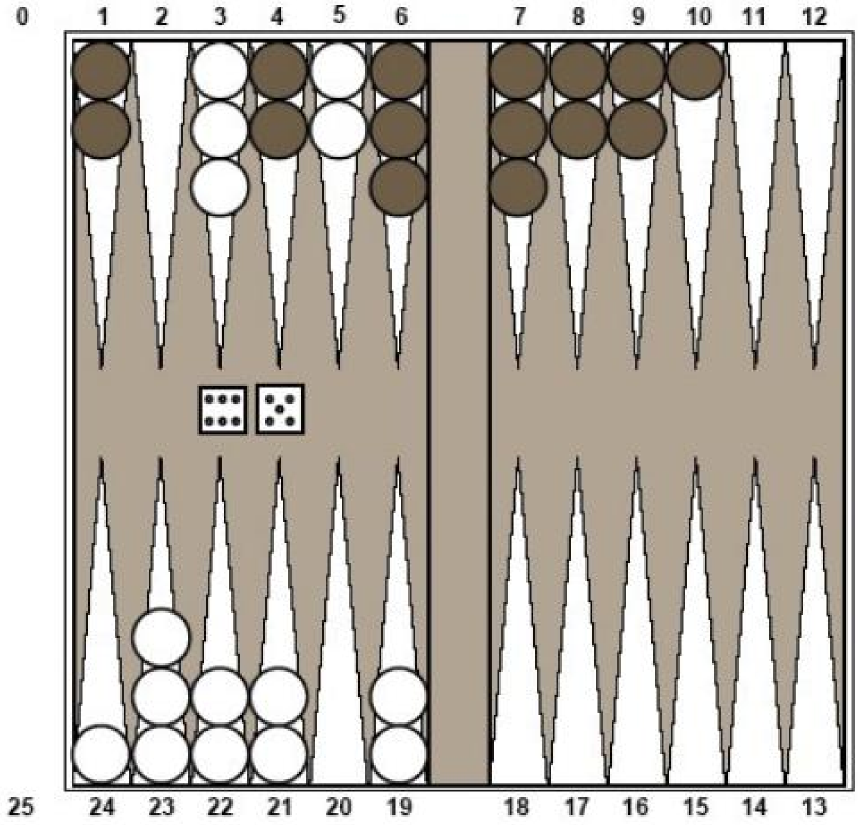
CS4365 --- Fall 2022

Games of Chance

Instructor: Yunhui Guo

Backgammon -- Board

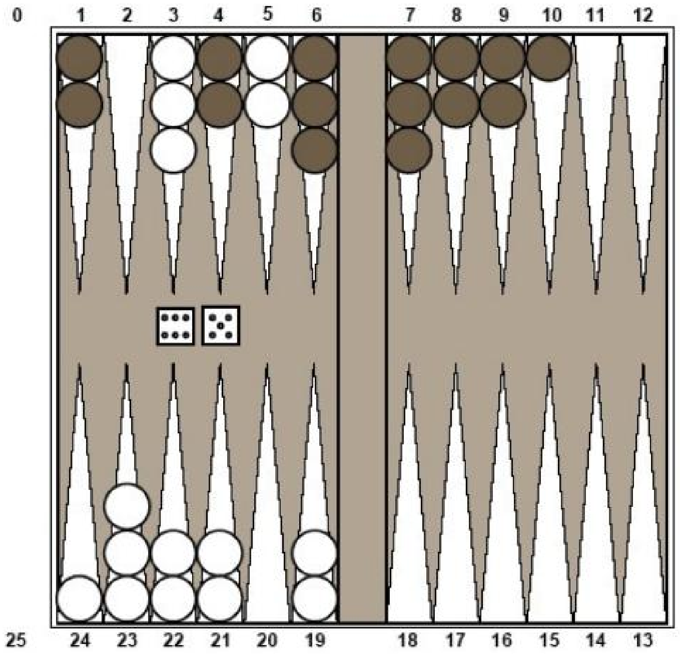
- Goal: move all of your pieces off the board before your opponent does.
- Black moves counterclockwise toward 0.
- White moves clockwise toward 25.
- The number of steps depend on the outcomes of tossing two dices.
- A piece can move to any position except one **where there are two or more of the opponent's pieces.**



Backgammon -- Rules

- If you roll doubles you take 4 moves (example: roll 5,5,make moves 5,5,5,5).
- Moves can be made by one or two pieces (in the case of doubles by 1, 2, 3 or 4 pieces)
- And a few other rules that concern bearing off and forced moves

Backgammon -- Rules



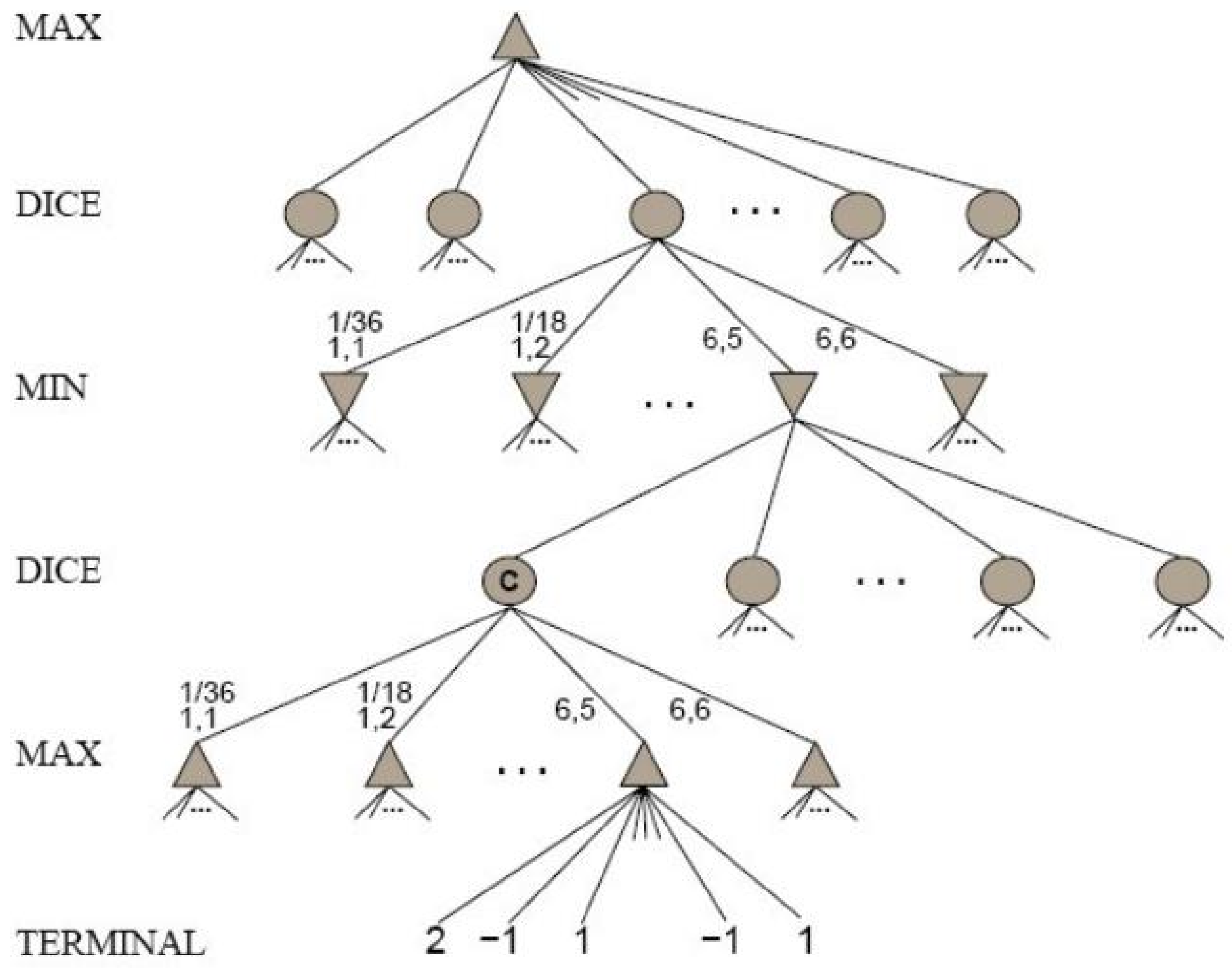
- White has rolled 6-5 and has 4 legal moves: (5-10,5-11), (5-11,19-24), (5-10,10-16) and (5-11,11-16).

Backgammon -- Rules

- The player tosses two dices
- Based on the outcomes to make certain moves
- The opponent tosses the dices
- Based on the outcomes to make certain moves

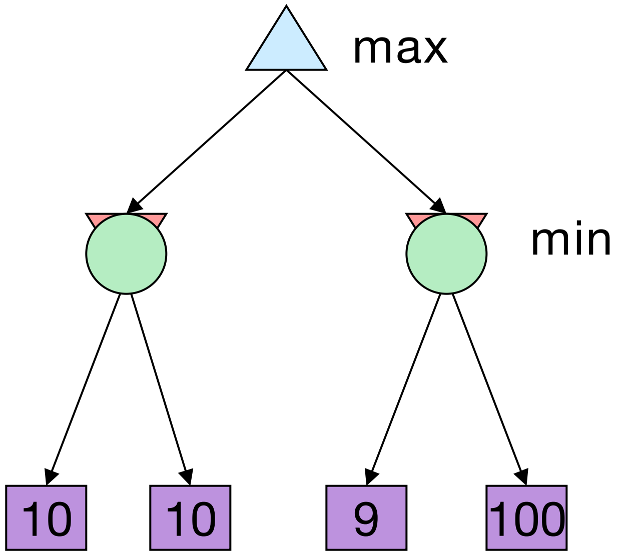
...

Game Tree for Backgammon



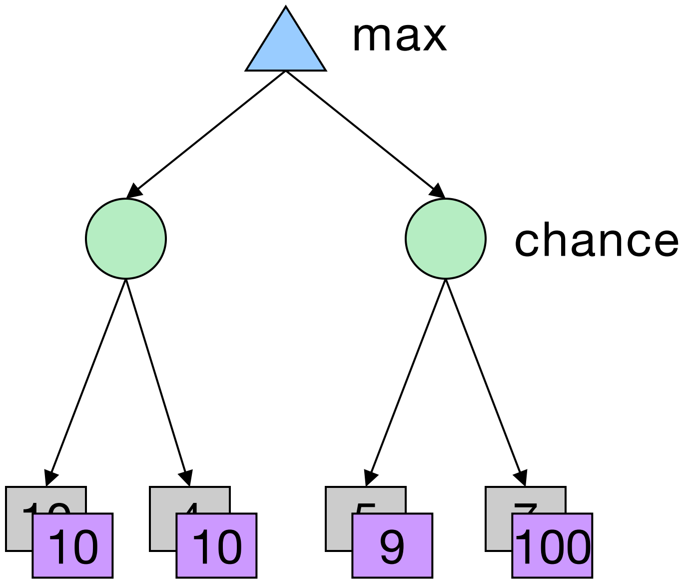
Uncertain Outcome

- Idea: Uncertain outcomes controlled by chance, not an adversary!



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



Reminder: Probabilities

- A random variable represents an event whose outcome is unknown
- A probability distribution is an assignment of weights to outcomes
- Example: Traffic on freeway
  - Random variable: T = whether there's traffic
  - Outcomes: T in {none, light, heavy}
  - Distribution: P(T=none) = 0.25, P(T=light) = 0.50, P(T=heavy) = 0.25

Reminder: Expectations

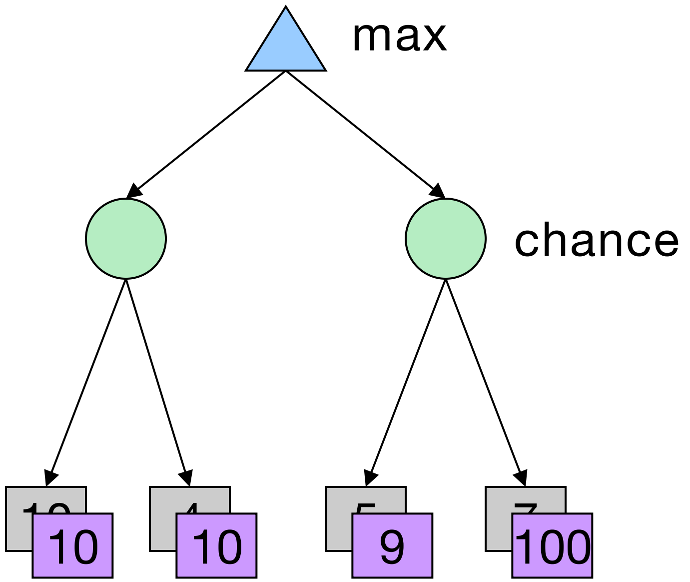
- The expected value of a function of a random variable is the average, weighted by the probability distribution over outcomes

- Example: How long to get to the airport?

Time:	20 min		30 min		60 min		
	x	+	x	+	x		35 min
Probability:	0.25		0.50		0.25		

What Probabilities to Use?

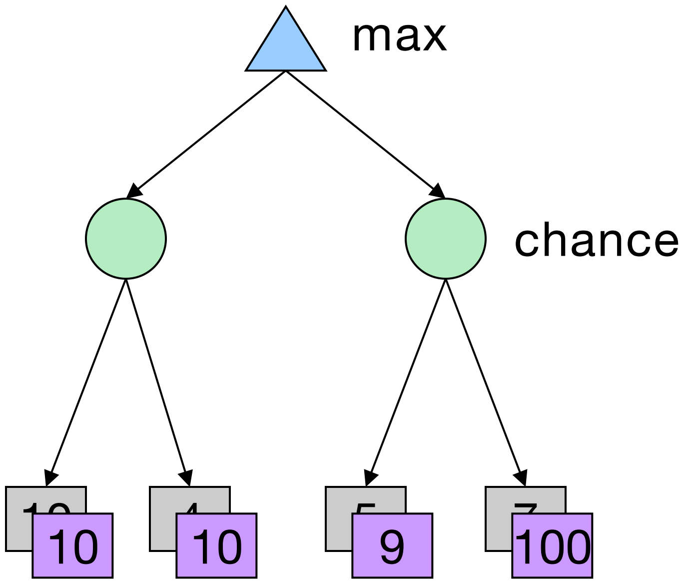
- In expectimax search, we have a probabilistic model of how the opponent (or environment) will behave in any state
  - Model could be a simple uniform distribution (roll a die)
  - Model could be sophisticated and require a great deal of computation
  - We have a chance node for any outcome out of our control: opponent or environment
  - The model might say that adversarial actions are likely!



- For now, assume each chance node magically comes along with probabilities that specify the distribution over its outcomes

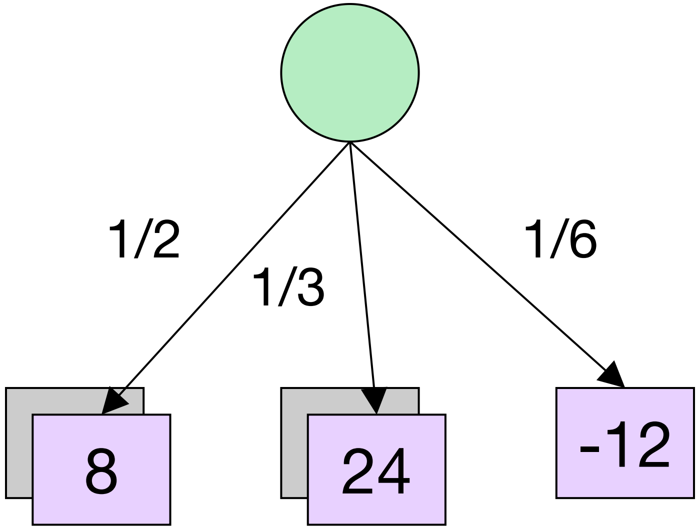
Expectimax Search

- 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



Expectimax Pseudocod

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



$$v = (1/2)\,(8) + (1/3)\,(24) + (1/6)\,(-12) = 10$$

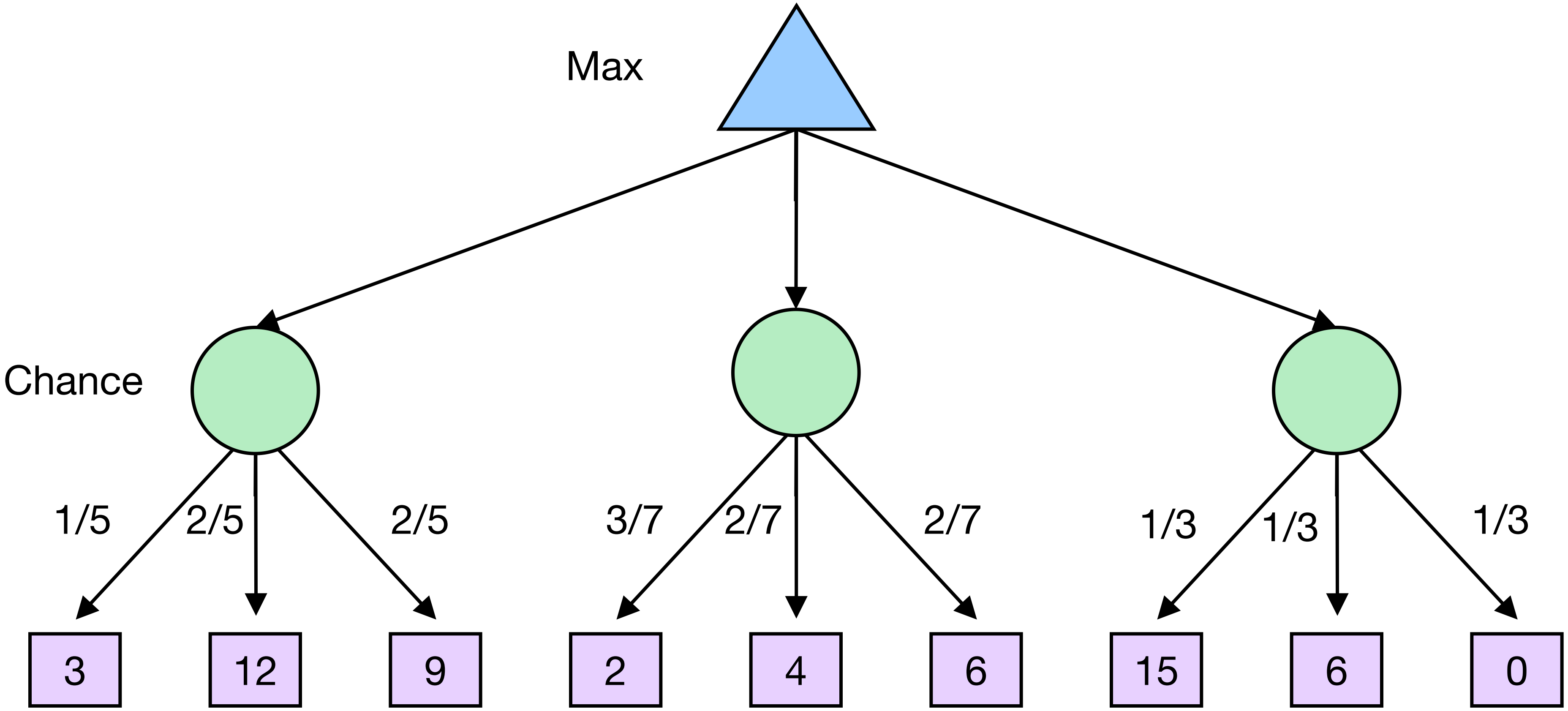
Expectimax Pseudocode

```
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)
```

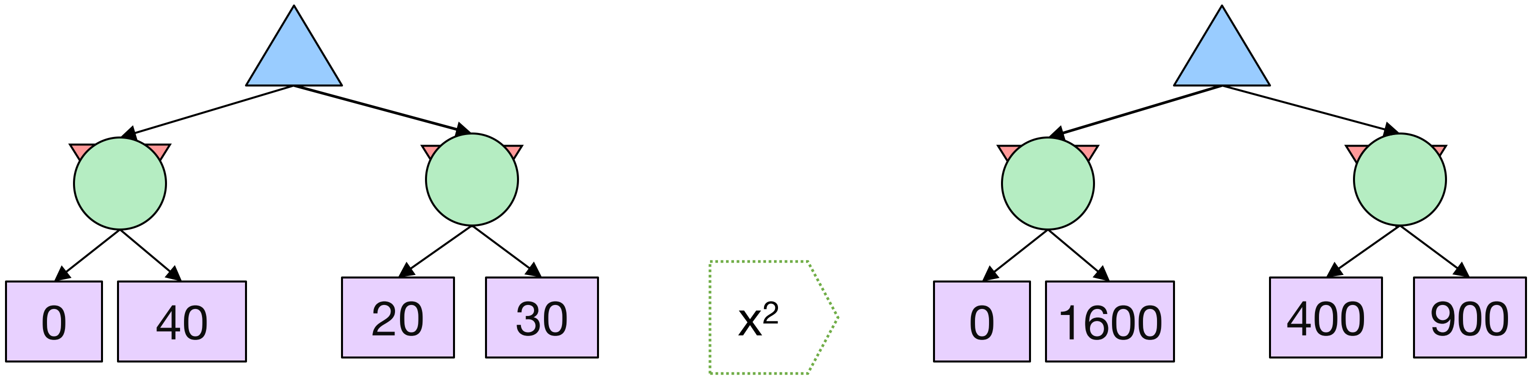
```
def max-value(state):
    initialize v = -∞
    for each successor of state:
        v = max(v, value(successor))
    return v
```

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

Expectimax Example

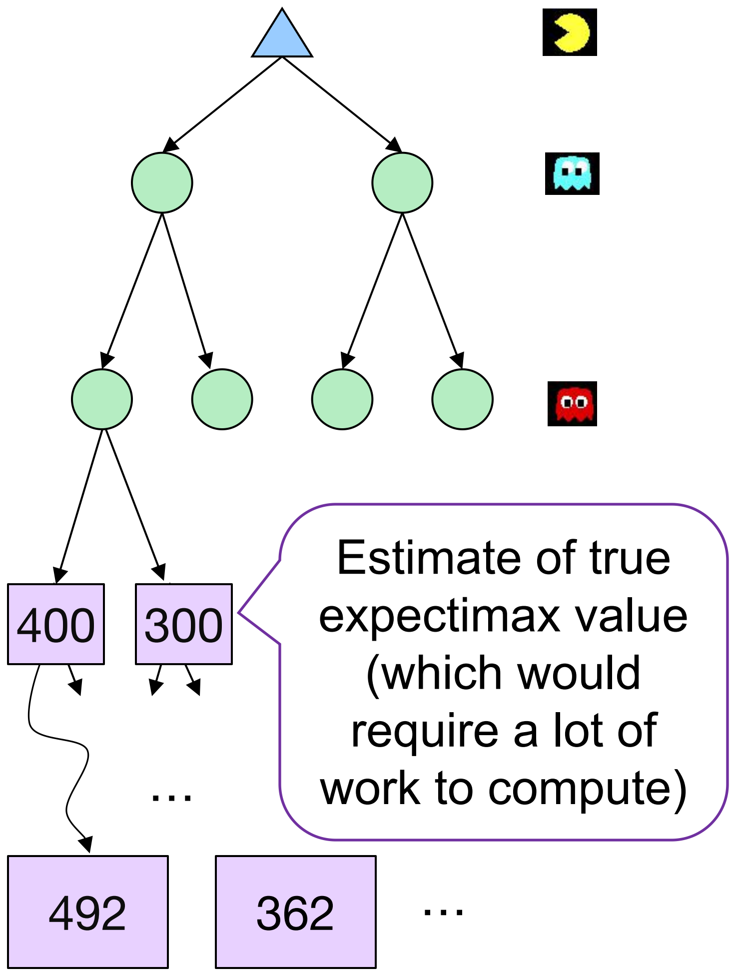


What Utilities to Use?



- For worst-case minimax reasoning, terminal function scale doesn't matter
  - We just want better states to have higher evaluations (get the ordering right)
  - We call this insensitivity to monotonic transformations
- For average-case expectimax reasoning, we need magnitudes to be meaningful

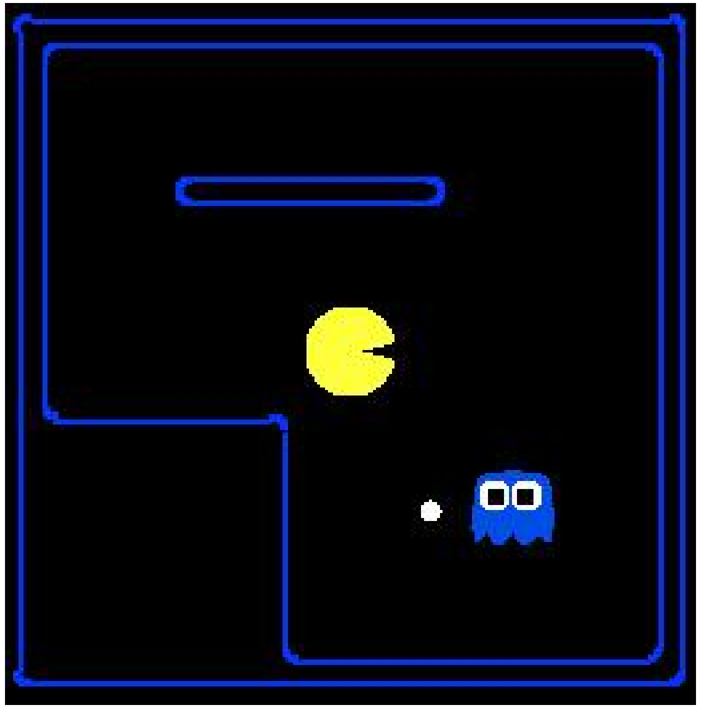
Depth-Limited Expectimax



The Dangers of Optimism and Pessimism

- Dangerous Optimism
  - Assuming chance when the world is adversarial
  - MiniMax
- Dangerous Pessimism
  - Assuming the worst case when it’s not likely
  - ExpectiMax

Assumptions vs. Reality



	Adversarial Ghost	Random Ghost
Minimax Pacman	Won 5/5 Avg. Score: 483	Won 5/5 Avg. Score: 493
Expectimax Pacman	Won 1/5 Avg. Score: -303	Won 5/5 Avg. Score: 503

Pacman used depth 4 search with an eval function that avoids trouble
Ghost used depth 2 search with an eval function that seeks Pacman

Utilities

- Utilities are functions from outcomes (states of the world) to real numbers that describe an agent's preferences

- Where do utilities come from?
  - In a game, may be simple (+1/-1)
  - Utilities summarize the agent's goals
  - **Theorem**: any "rational" preferences can be summarized as a utility function
- We hard-wire utilities and let behaviors emerge
  - Why don't we let agents pick utilities?
  - Why don't we prescribe behaviors?
  - Why maximize expected utility?
    - Expected utility theory is a theory about how to make optimal decisions under a given probability of utility.

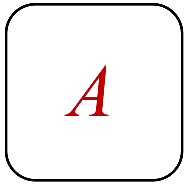
Preferences

- An agent must have preferences among:
  - Prizes: *A, B*, etc.
  - Lotteries: situations with uncertain prizes

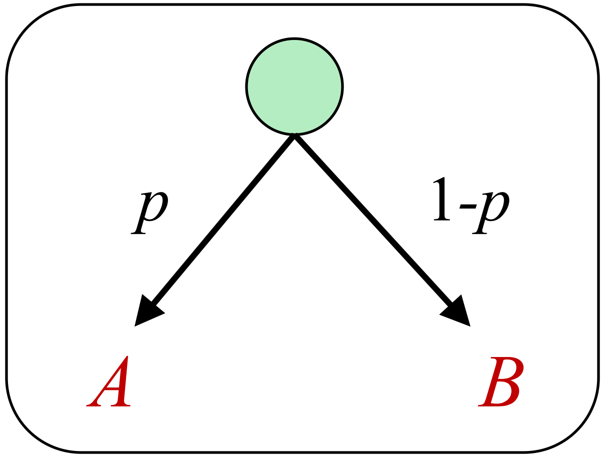
$$L = [p, A;\ \ (1-p), B]$$

- Notation:
  - Preference: $A \succ B$
  - Indifference: $A \sim B$

A Prize



A Lottery



Rational Preferences

- We want some constraints on preferences before we call them rational, such as:

$$\boxed{\text{Axiom of Transitivity: } (A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)}$$

- For example: an agent with intransitive preferences can be induced to give away all of its money
  - If B > C, then an agent with C would pay (say) 1 cent to get B
  - If A > B, then an agent with B would pay (say) 1 cent to get A
  - If C > A, then an agent with A would pay (say) 1 cent to get C

Rational Preferences

The Axioms of Rationality

Orderability

$$(A \succ B) \vee (B \succ A) \vee (A \sim B)$$

Transitivity

$$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$

Continuity

$$A \succ B \succ C \Rightarrow \exists p\ [p, A;\ 1-p, C] \sim B$$

Substitutability

$$A \sim B \Rightarrow [p, A;\ 1-p, C] \sim [p, B; 1-p, C]$$

Monotonicity

$$A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A;\ 1-p, B] \succeq [q, A;\ 1-q, B])$$

- If all the axioms are satisfied, the agent is said to be rational

MEU Principle

- Theorem [Ramsey, 1931; von Neumann & Morgenstern, 1944]
  - Given any preferences satisfying these constraints, there exists a real-valued function U such that:

$$U(A) \geq U(B) \;\Leftrightarrow\; A \succeq B$$

$$U([p_1, S_1;\; \ldots\; ;\; p_n, S_n]) = \sum_i p_i U(S_i)$$

  - I.e. values assigned by U preserve preferences of both prizes and lotteries!

- **Maximum expected utility (MEU) principle**:
  - Choose the action that maximizes expected utility
  - Note: an agent can be entirely rational (consistent with MEU) without ever representing or manipulating utilities and probabilities
  - E.g., a lookup table for perfect tic-tac-toe

Summary

- Minimax considers adversarial opponent
- ExpectiMax considers random opponent.
- Maximum expected utility leads to a rational agent.