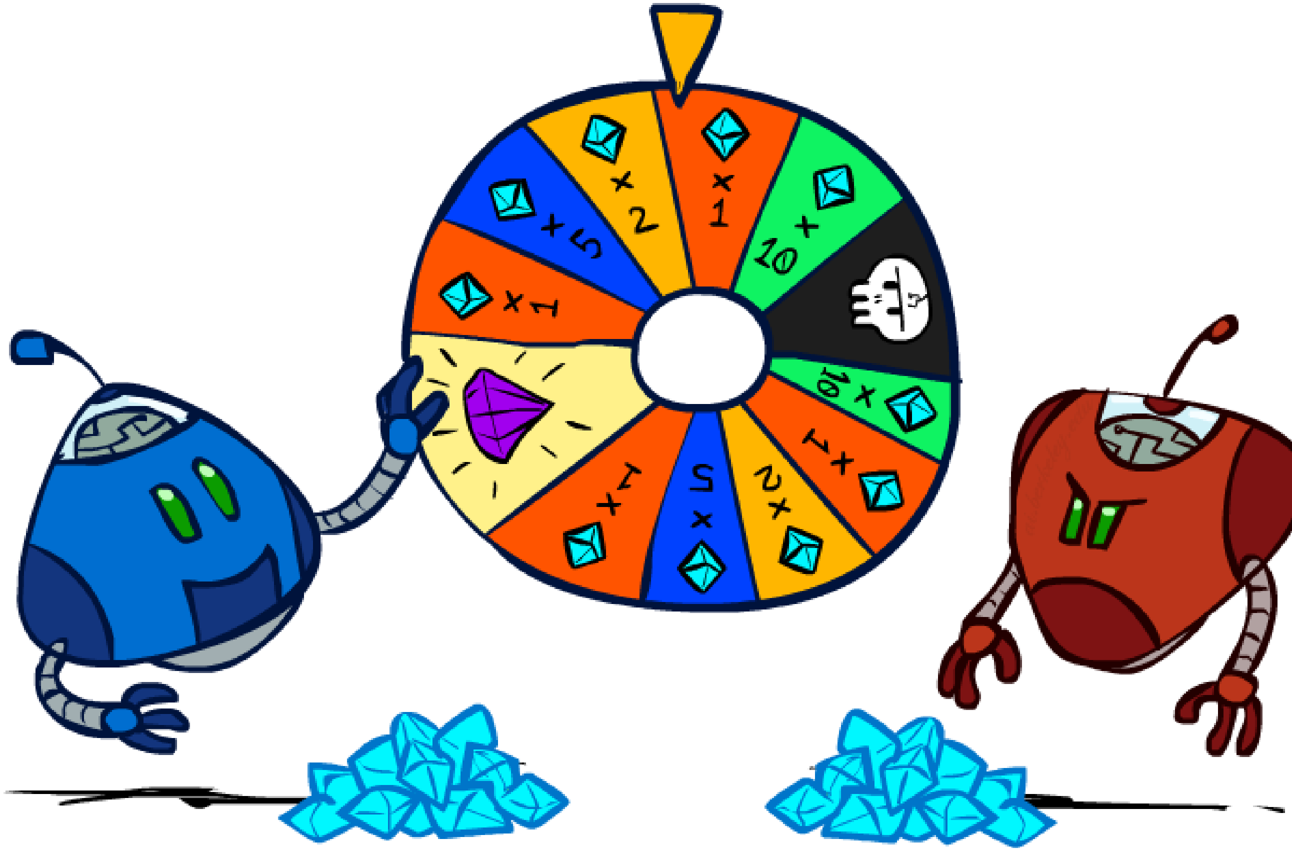
CS 188: Artificial Intelligence

Uncertainty and Utilities

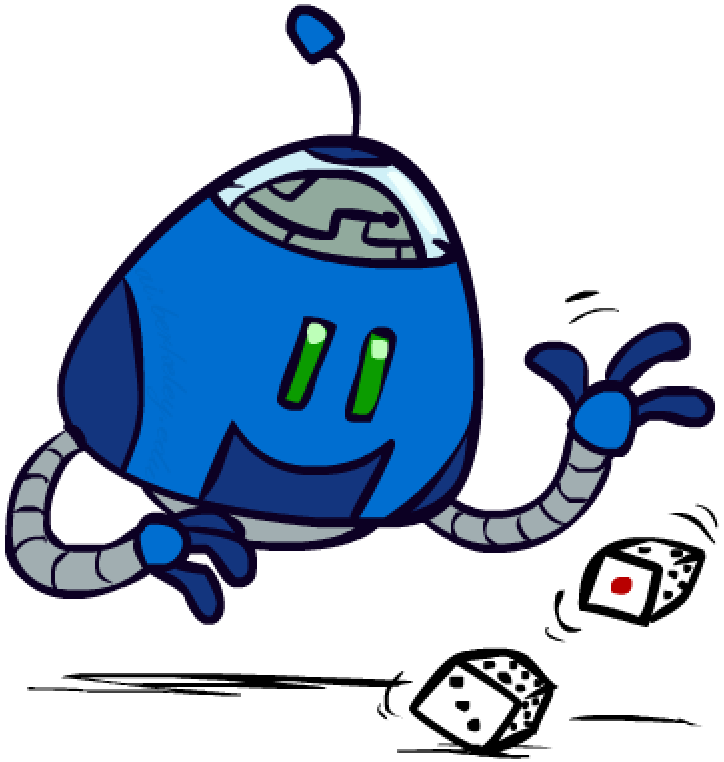


Instructors: Fatemeh Mansoori

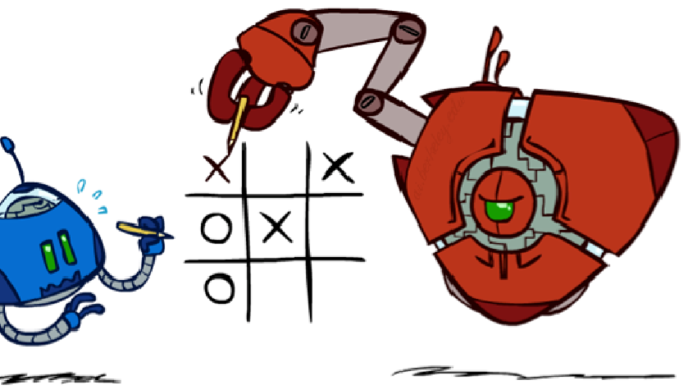
University Isfahan

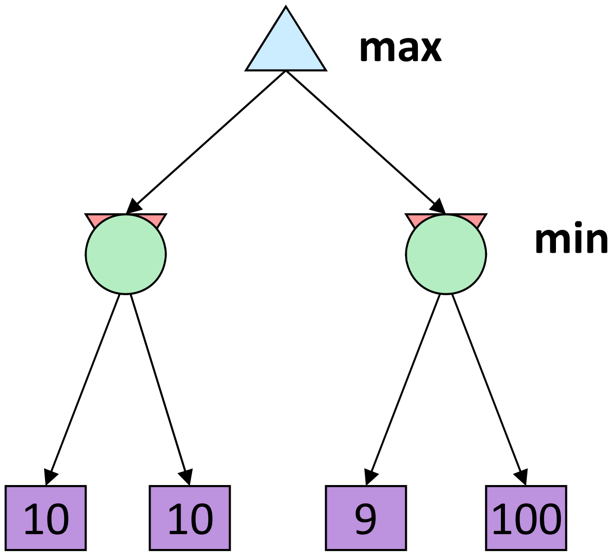
[These slides were created by Dan Klein, Pieter Abbeel for CS188 Intro to AI at UC Berkeley (ai.berkeley.edu).]

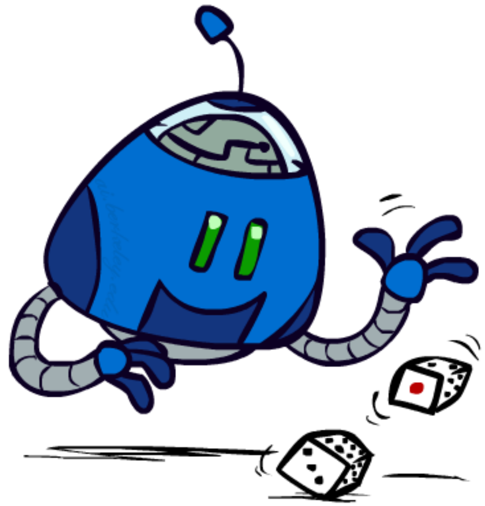
Uncertain Outcomes



Worst-Case vs. Average Case







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

Expectimax Search

Why wouldn't we know what the result of an action will be?

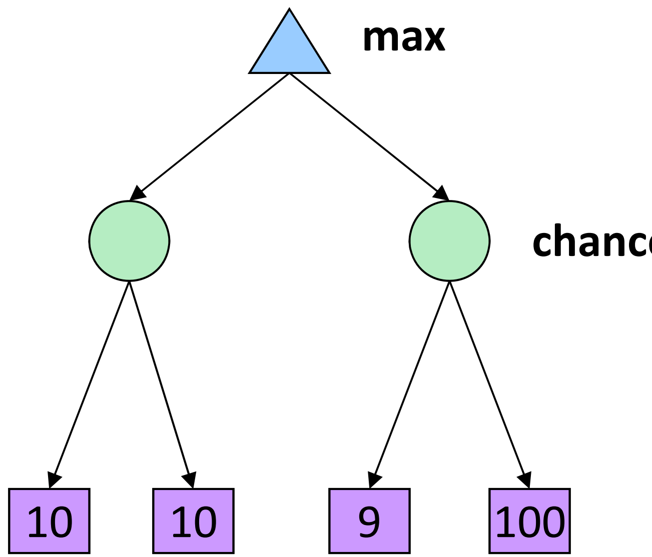
- Explicit randomness: rolling dice
- Unpredictable opponents: the ghosts respond randomly
- Actions can fail: when moving a robot, wheels might slip

Values should 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

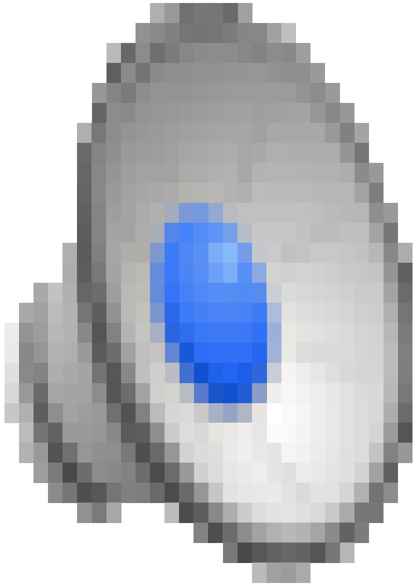
Later, we'll learn how to formalize the underlying uncertain-result problems as Markov Decision Processes



[Demo: min vs exp (L

Video of Demo Minimax vs Expectimax (Min)

Video of Demo Minimax vs Expectimax (Exp)



Expectimax Pseudocode

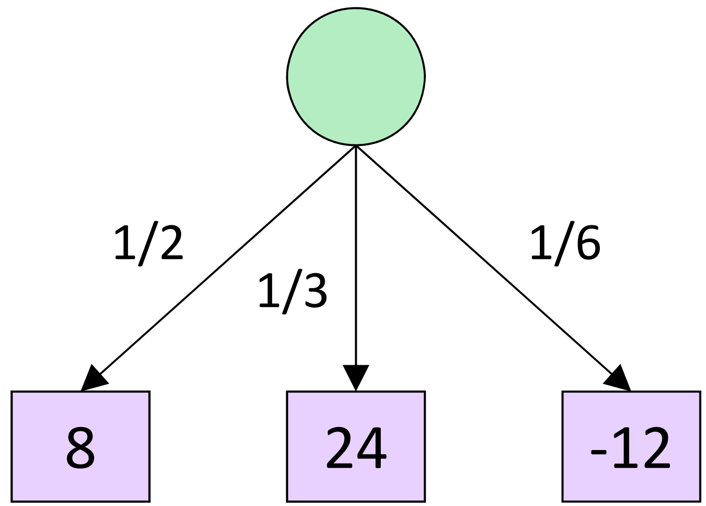
```
def value(state):
    if the state is a terminal state: return the state's utility
    if the next agent is MAX: return max-value(state)
    if the next agent is EXP: return exp-value(state)
```

```
f 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
```

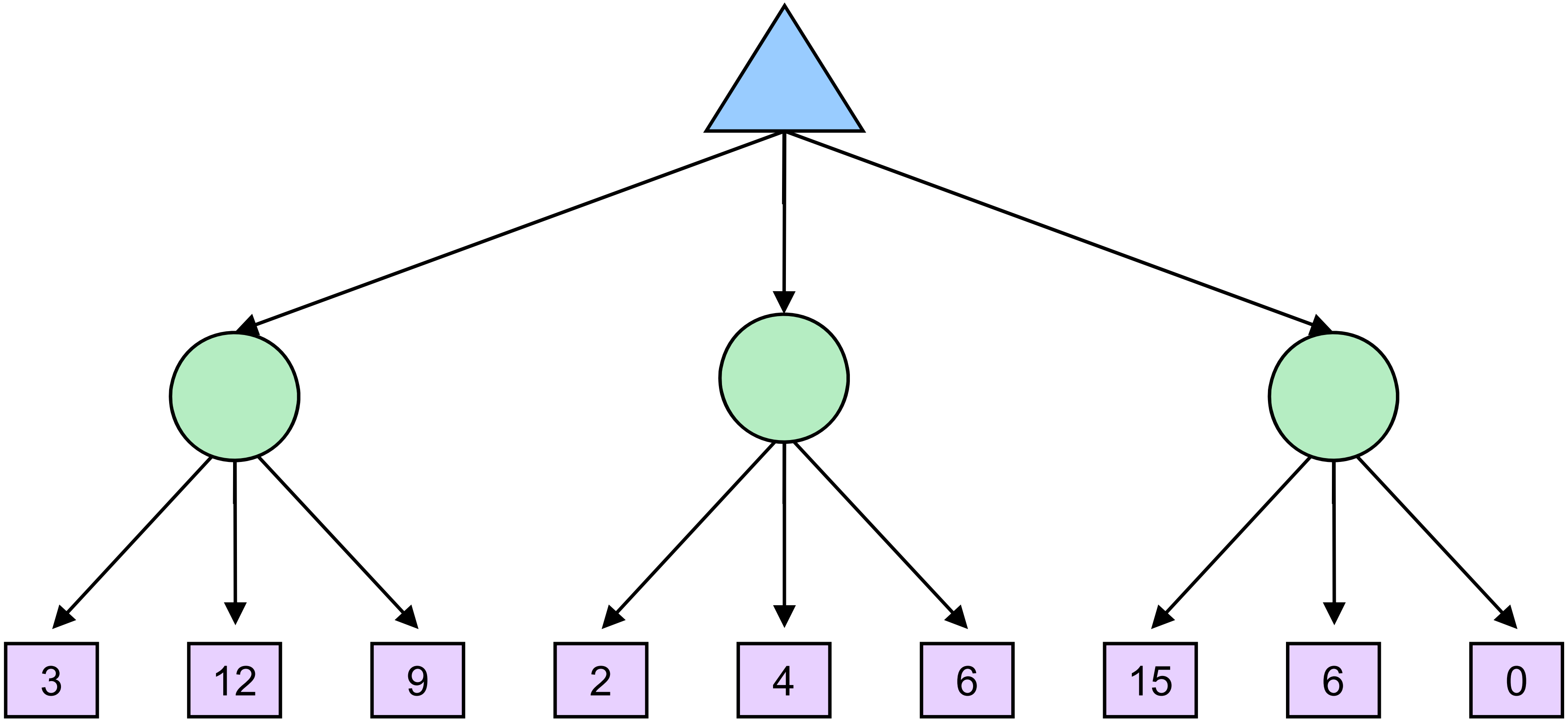
Expectimax Pseudocode

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

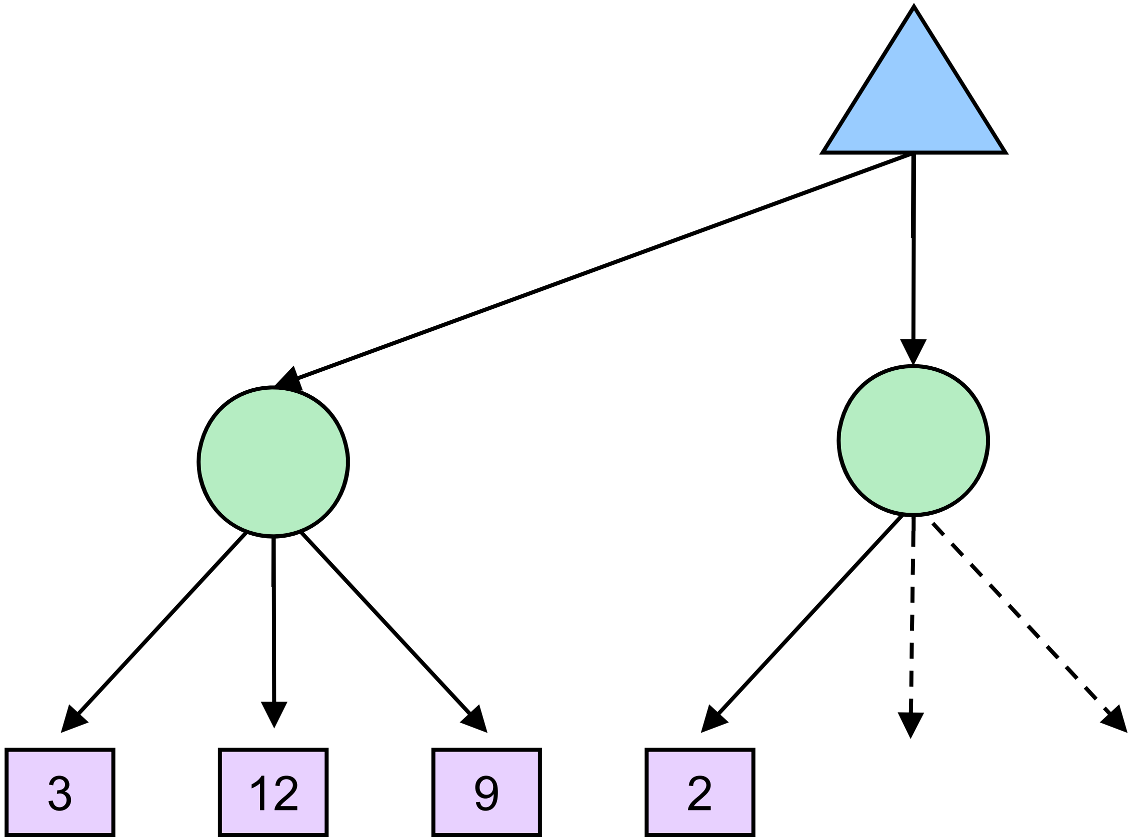


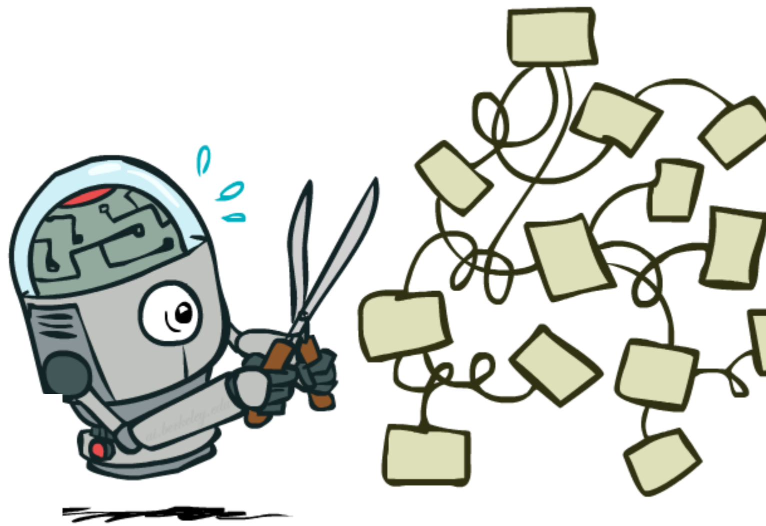
v = (1/2) (8) + (1/3) (24) + (1/6) (-12) = 10

Expectimax Example

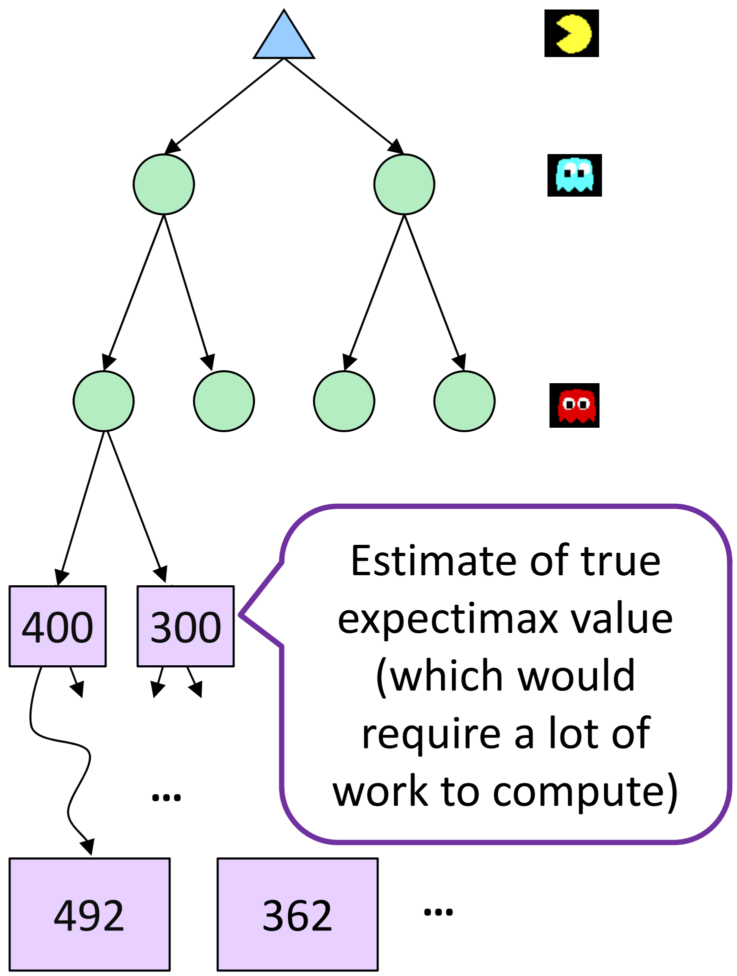


Expectimax Pruning?

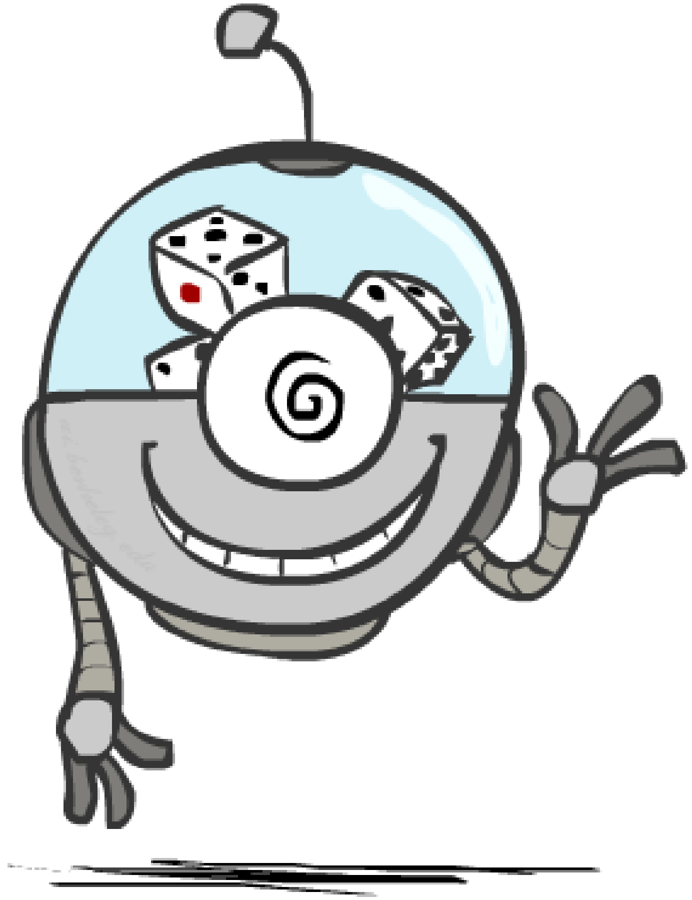




Depth-Limited Expectimax



Probabilities



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

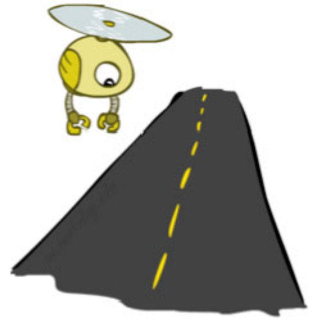
- Random variable: T = whether there's traffic
- Outcomes: T in {none, light, heavy}
- Distribution: P(T=none) = 0.25, P(T=light) = 0.50, P(T=heavy) = 0.25

Some laws of probability (more later):

- Probabilities are always non-negative
- Probabilities over all possible outcomes sum to one

As we get more evidence, probabilities may change:

- P(T=heavy) = 0.25, P(T=heavy | Hour=8am) = 0.60
- We'll talk about methods for reasoning and updating probabilities later



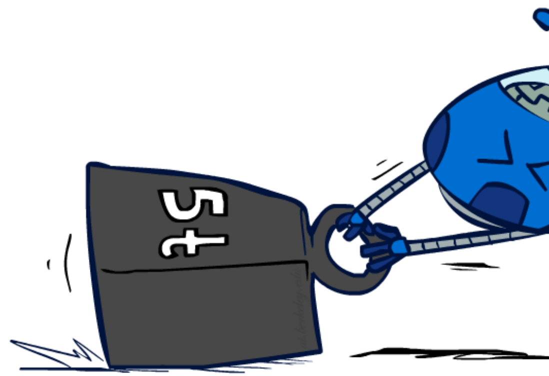
0.25

0.50

0.25

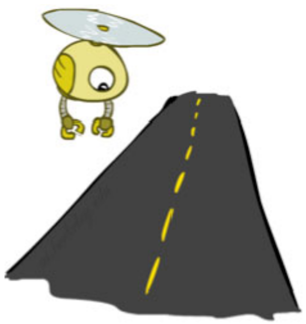
Reminder: Expectations

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		

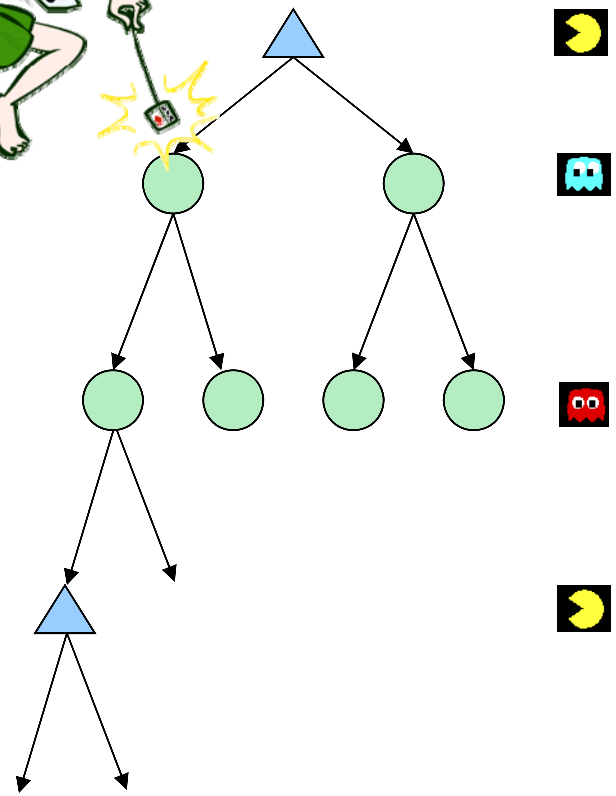


What Probabilities to Use?

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

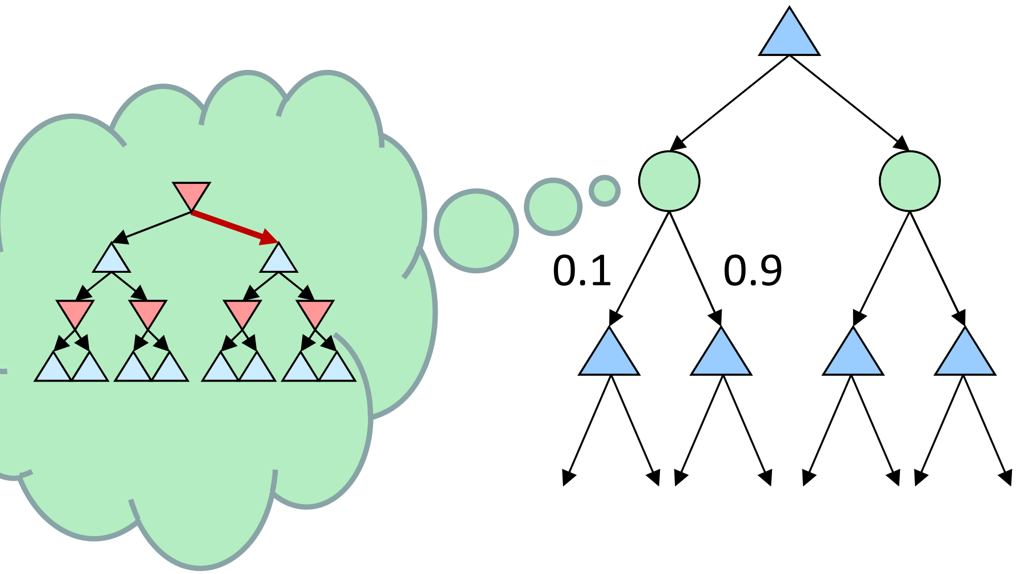


Having a probabilistic belief abo
another agent's action does not m
that the agent is flipping any coi

Quiz: Informed Probabilities

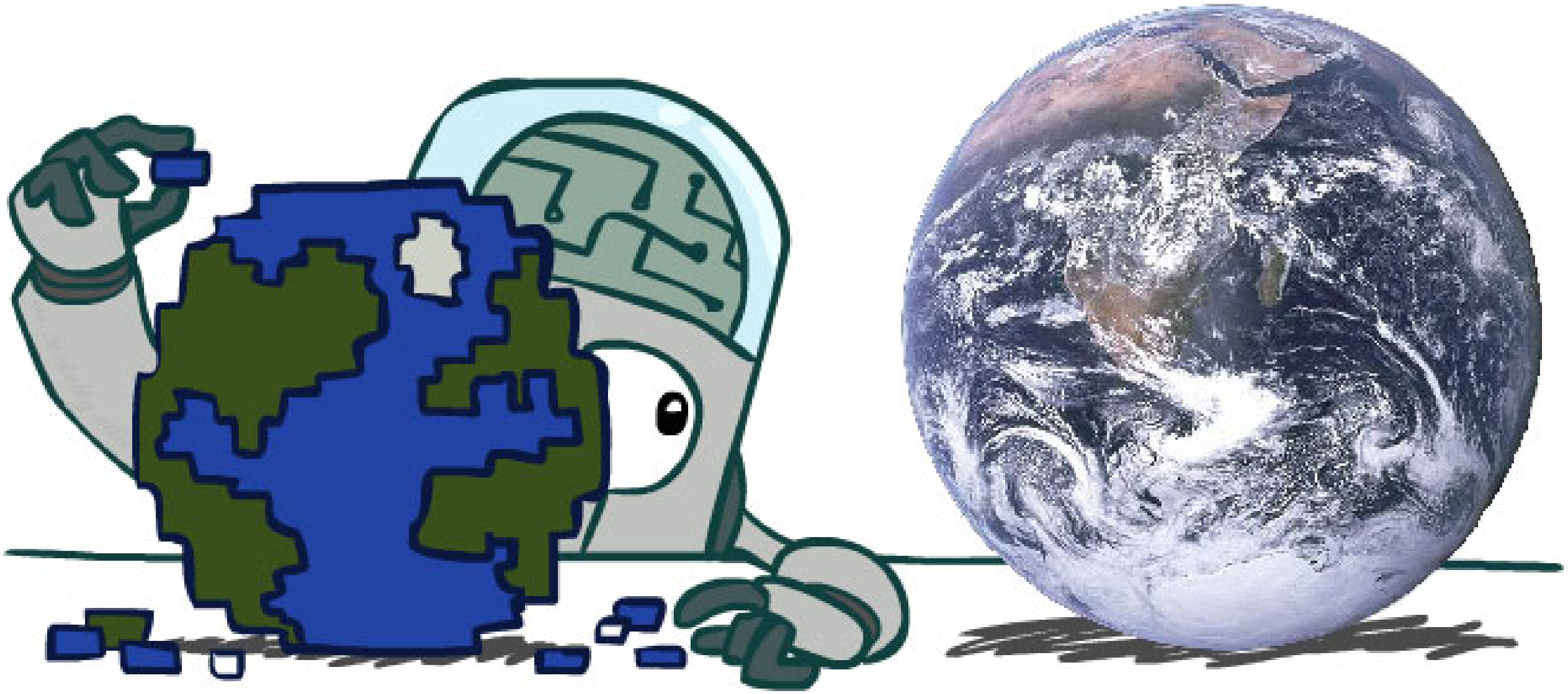
Let's say you know that your opponent is actually running a depth 2 minimax, using th
result 80% of the time, and moving randomly otherwise

Question: What tree search should you use?

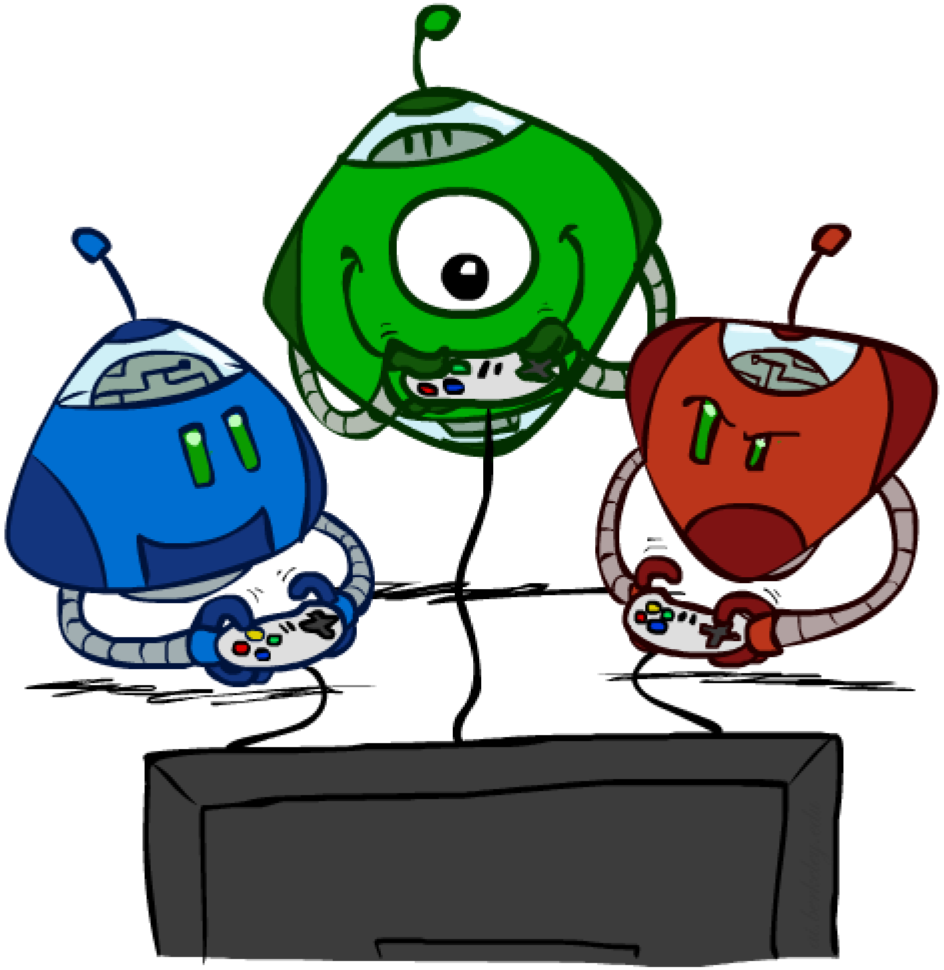


- Answer: Expectimax!
  - To figure out EACH chance node's probabiliti
    you have to run a simulation of your oppone
  - This kind of thing gets very slow very quickly

Modeling Assumptions



Other Game Types

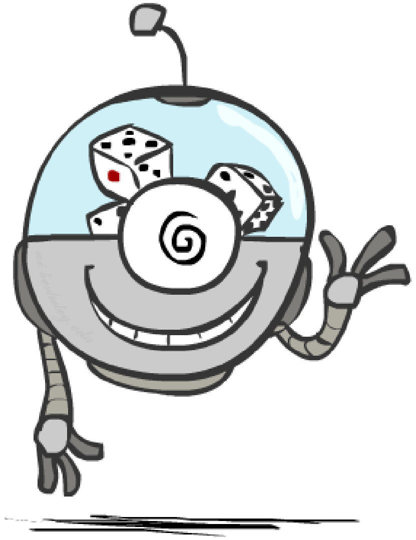


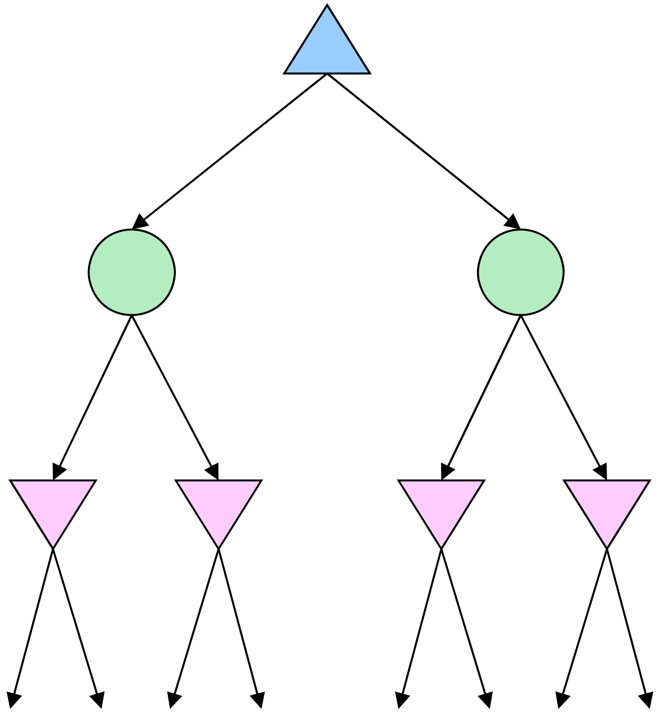
Mixed Layer Types

E.g. Backgammon

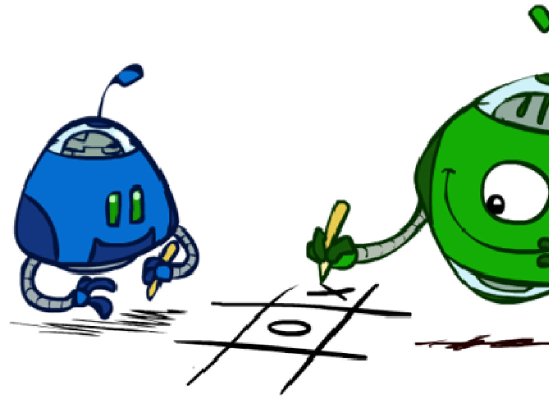
Expectiminimax

- Environment is an extra "random agent" player that moves after each min/max agent
- Each node computes the appropriate combination of its children





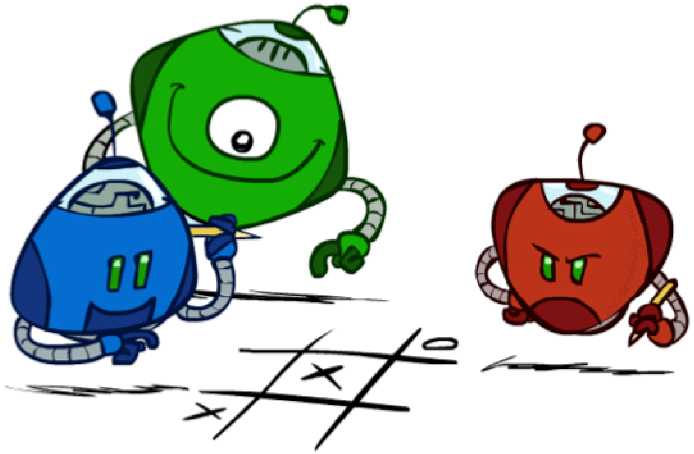
Multi-Agent Utilities

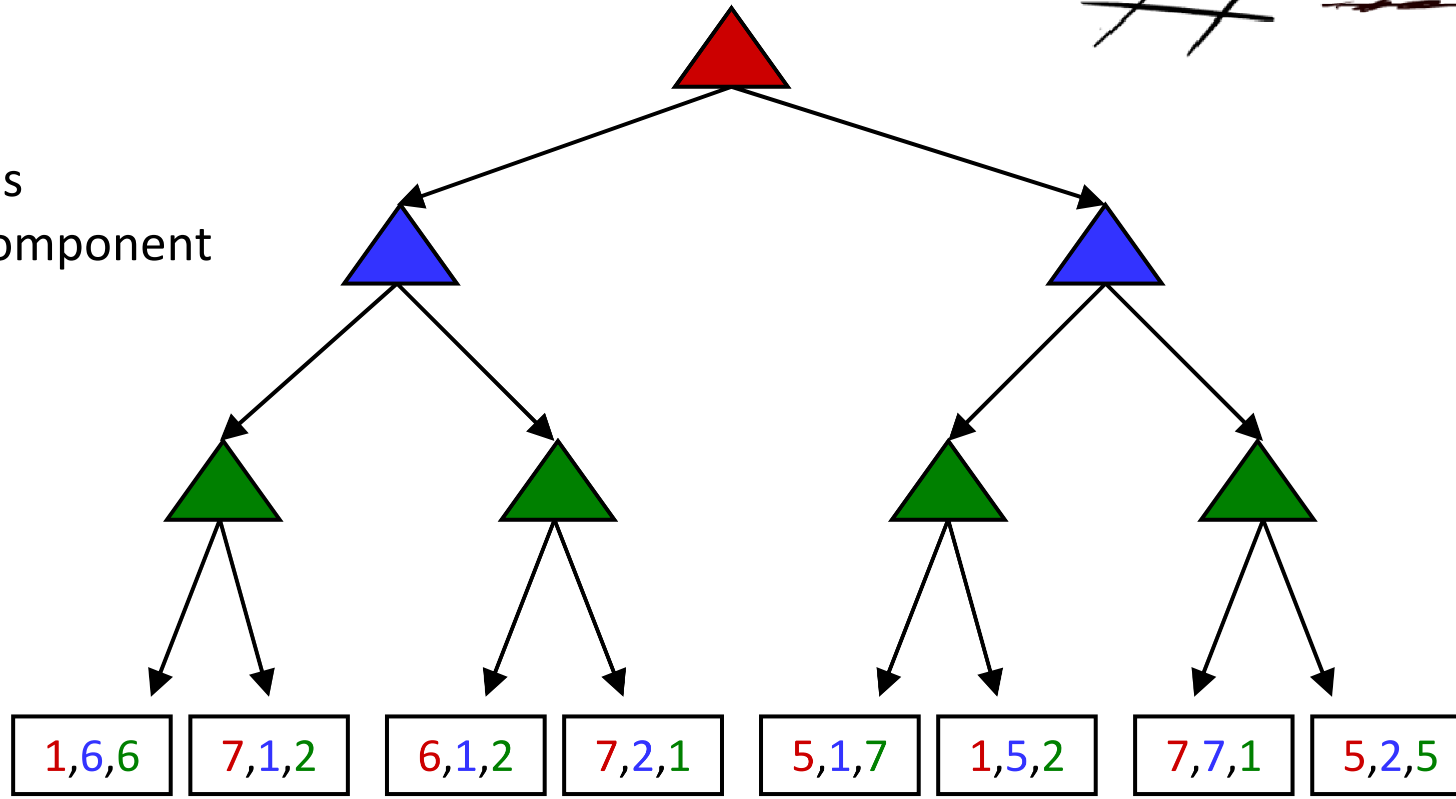


What if the game is not zero-sum, or has multiple players?

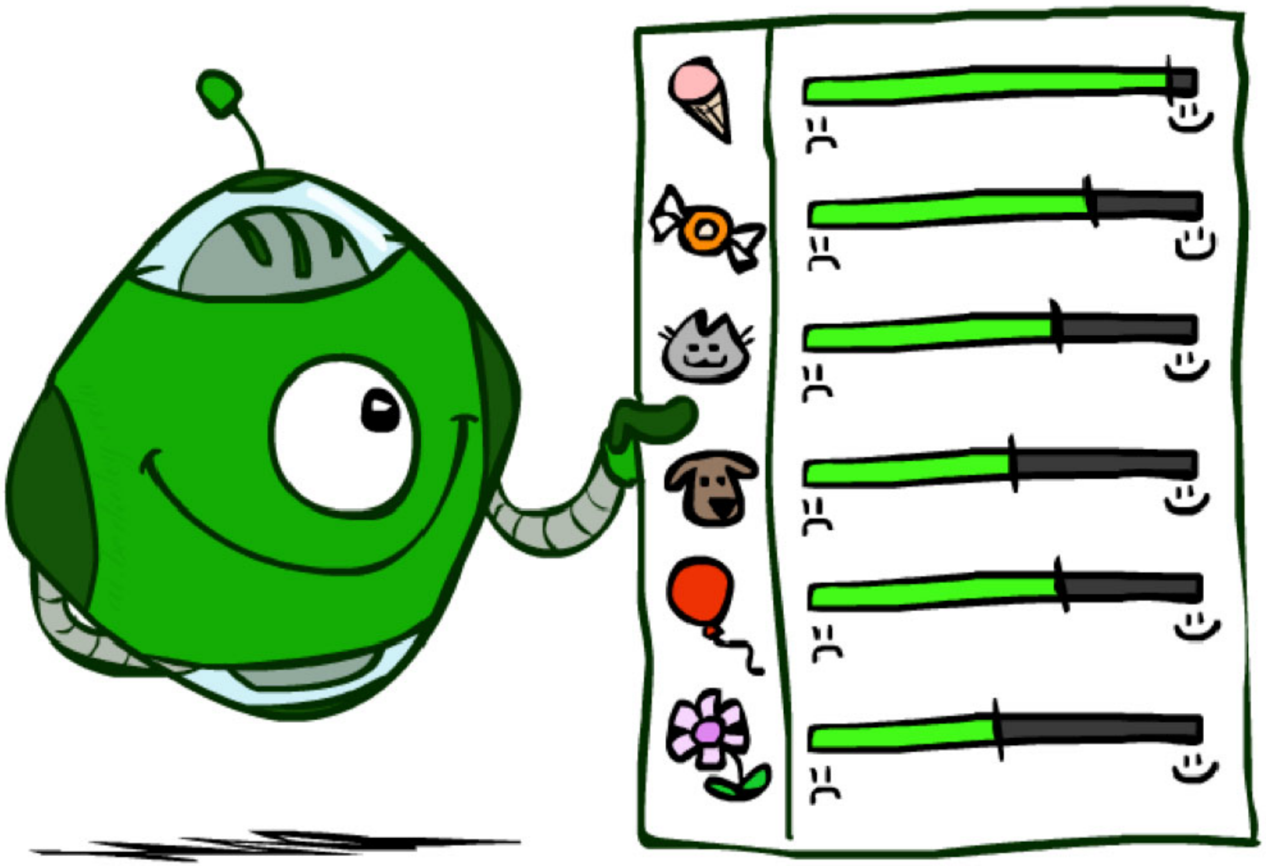
Generalization of minimax:

- Terminals have utility tuples
- Node values are also utility tuples
- Each player maximizes its own component
- Can give rise to cooperation and competition dynamically…





Utilities

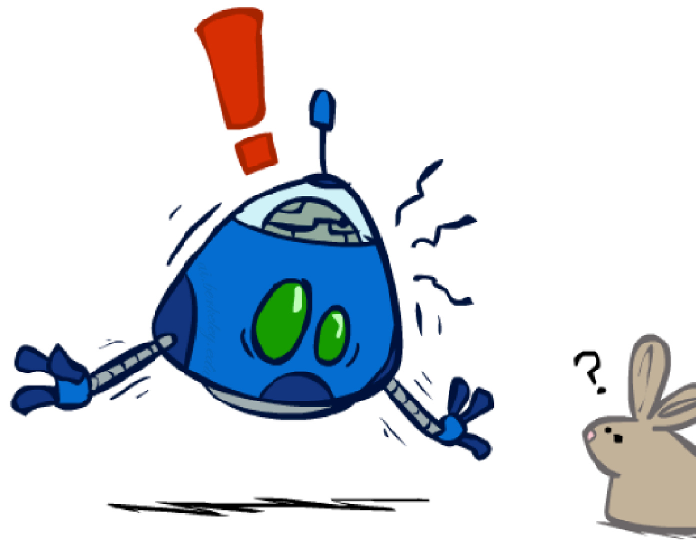


Maximum Expected Utility

Why should we average utilities? Why not minimax?

Principle of maximum expected utility:

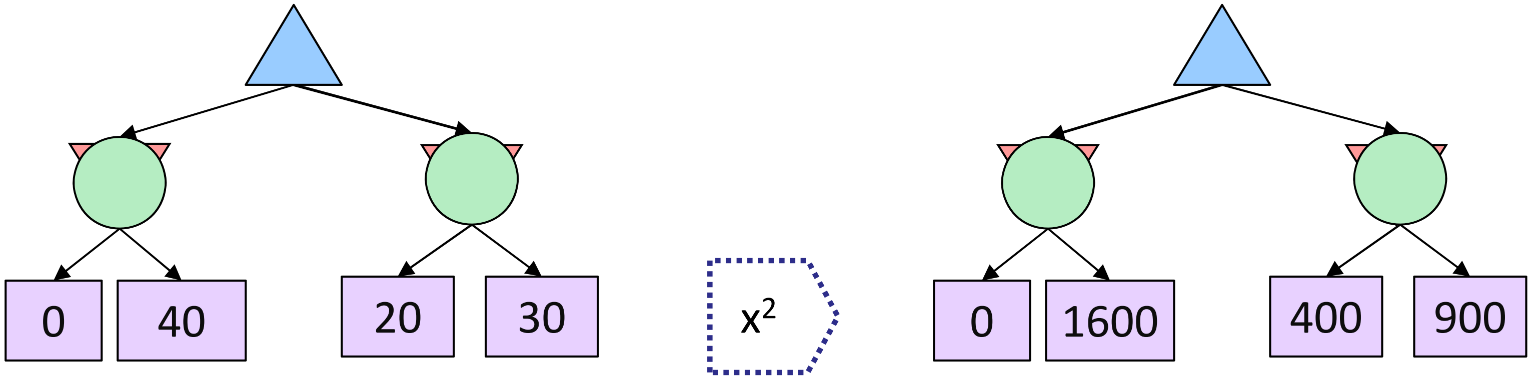
- A rational agent should chose the action that maximizes its expected utility, given its knowledge



Questions:

- Where do utilities come from?
- How do we know such utilities even exist?
- How do we know that averaging even makes sense?
- What if our behavior (preferences) can't be described by utilities?

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

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?

Utilities: Uncertain Outcomes