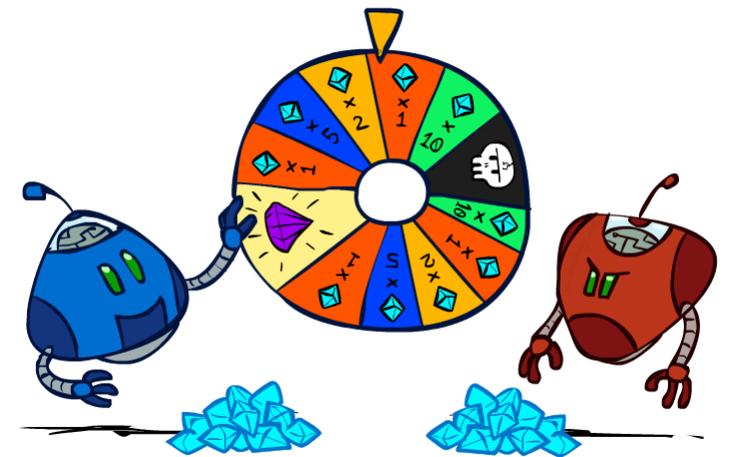# Reminder

- PS1 Due this Friday
- HW1 Due next Friday
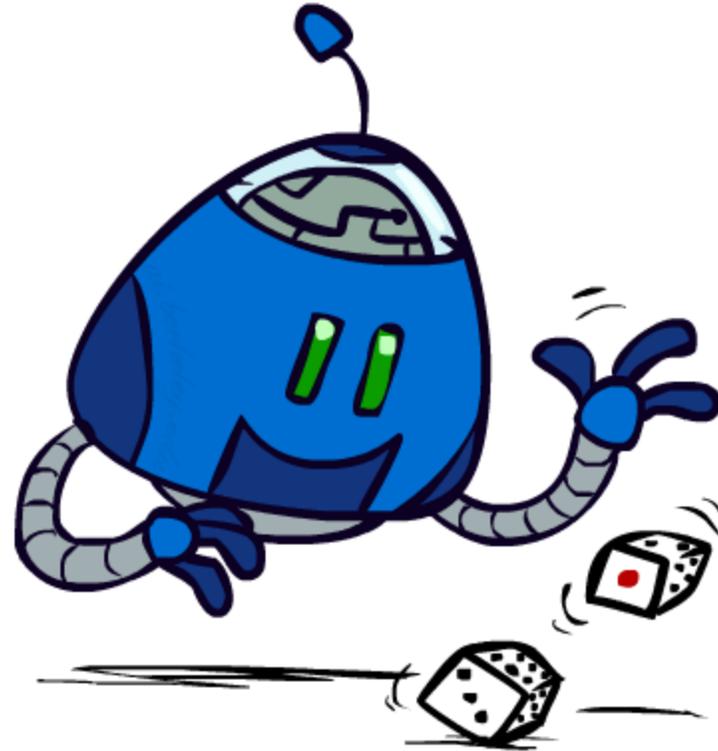
CSE 573:
Artificial Intelligence
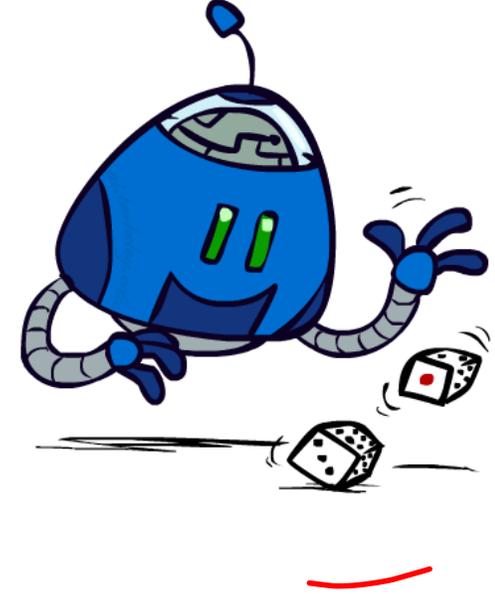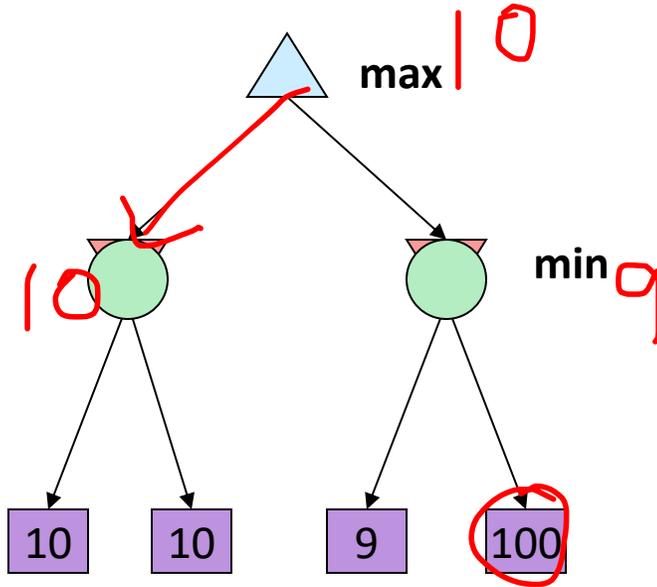
Hanna Hajishirzi

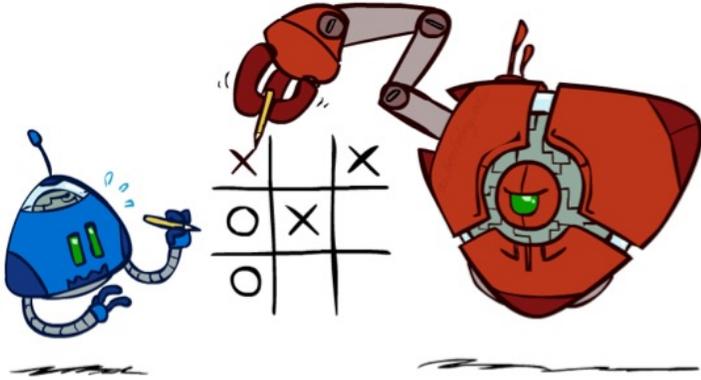## Expectimax – Complex Games

slides adapted from
Dan Klein, Pieter Abbeel ai.berkeley.edu
And Dan Weld, Luke Zettlemoyer

# Uncertain Outcomes

Worst-Case vs. Average Case

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

Expectimax Search

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

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

o Expectimax search: compute the average score under optimal play
  o Max nodes as in minimax search
  o Chance nodes are like min nodes but the outcome is uncertain
  o Calculate their expected utilities
  o I.e. take weighted average (expectation) of children

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

max

chance

10

$\frac{9+100}{2} = 54.5$

10   10   9   100
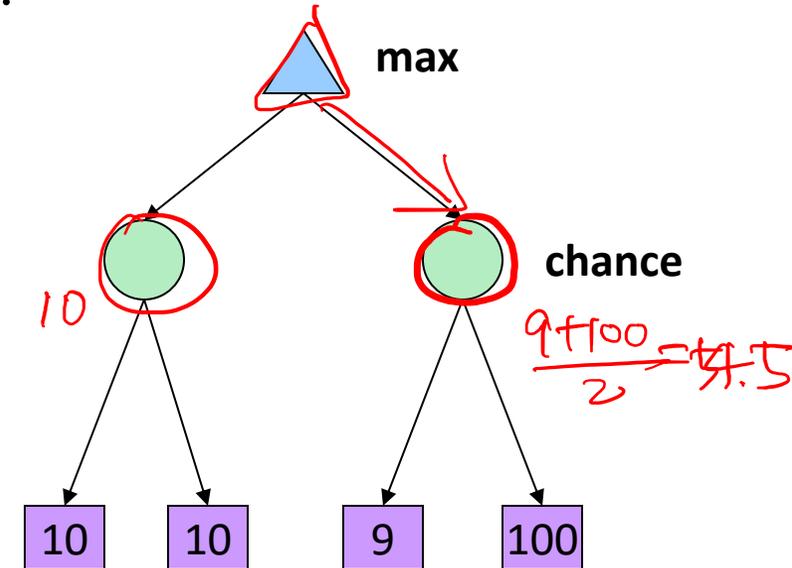
Video of Demo Min vs. Exp (Min)
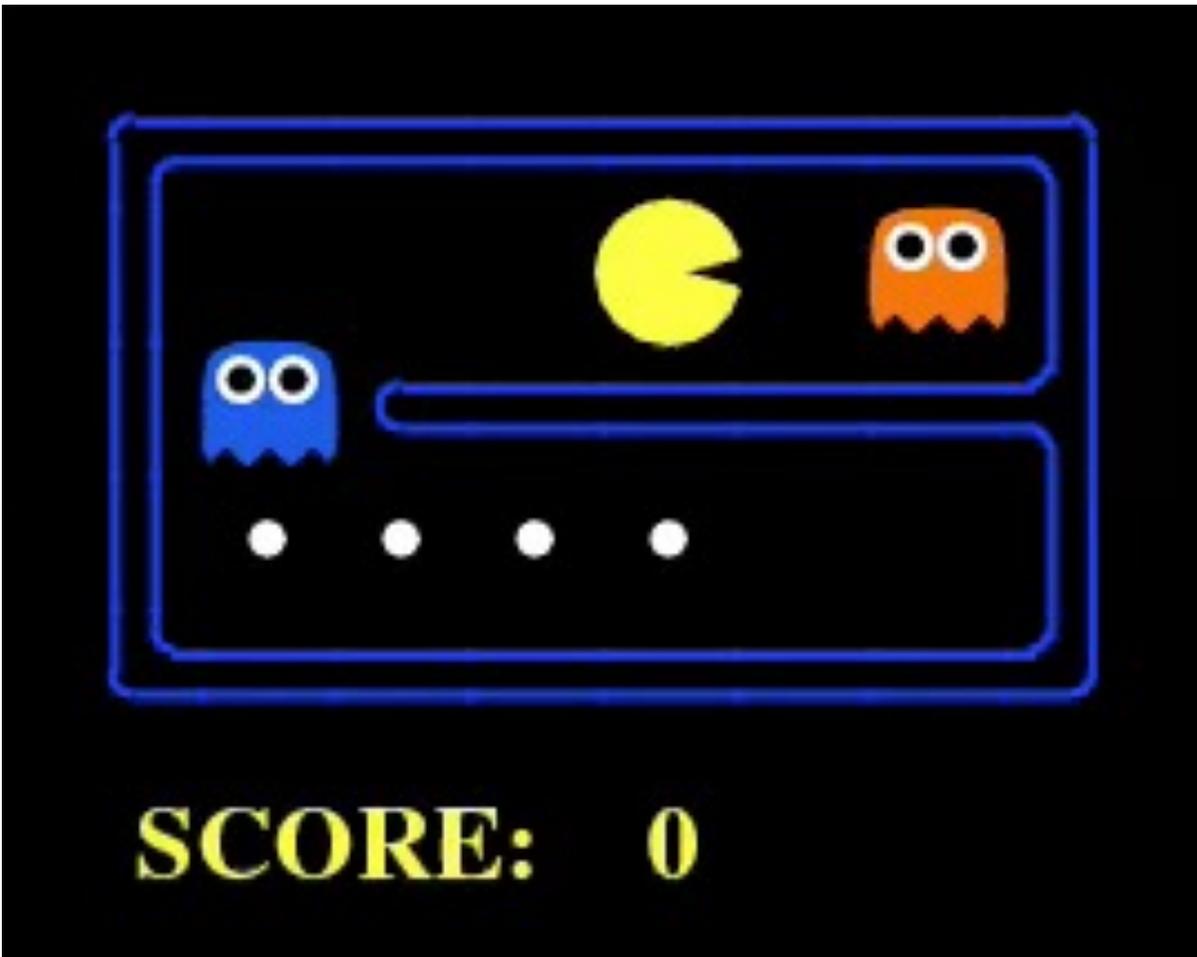
Expectimax Pseudocode

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)
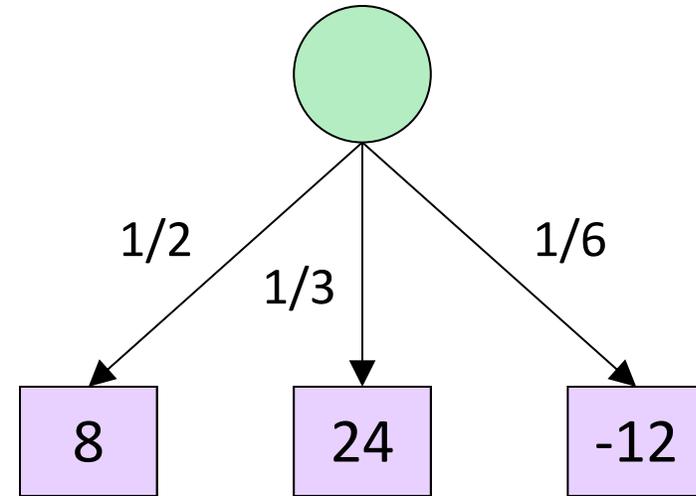   if the next agent is EXP: return exp-value(state)

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)
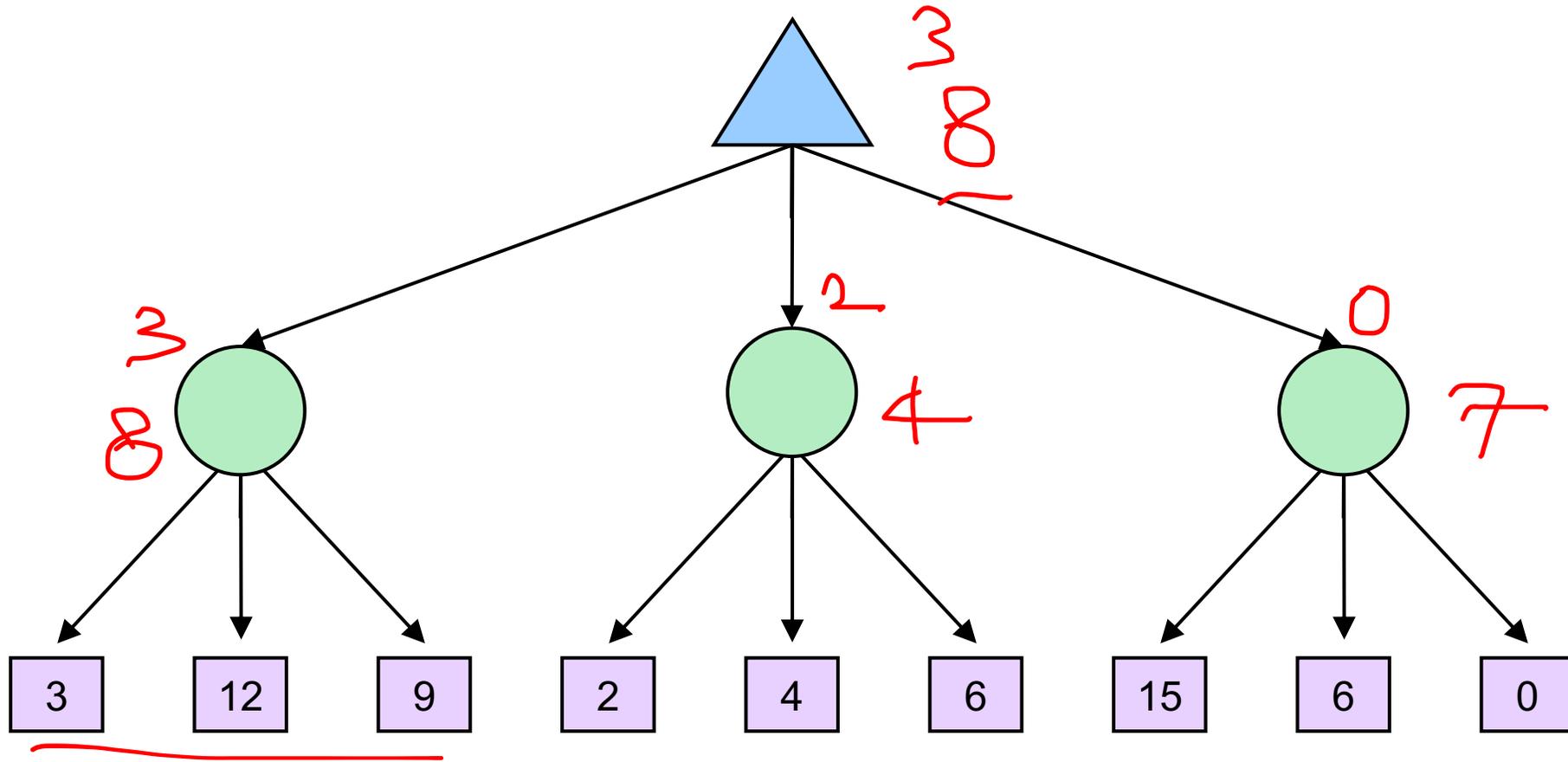   v += p * value(successor)
    return v

Expectimax Pseudocode

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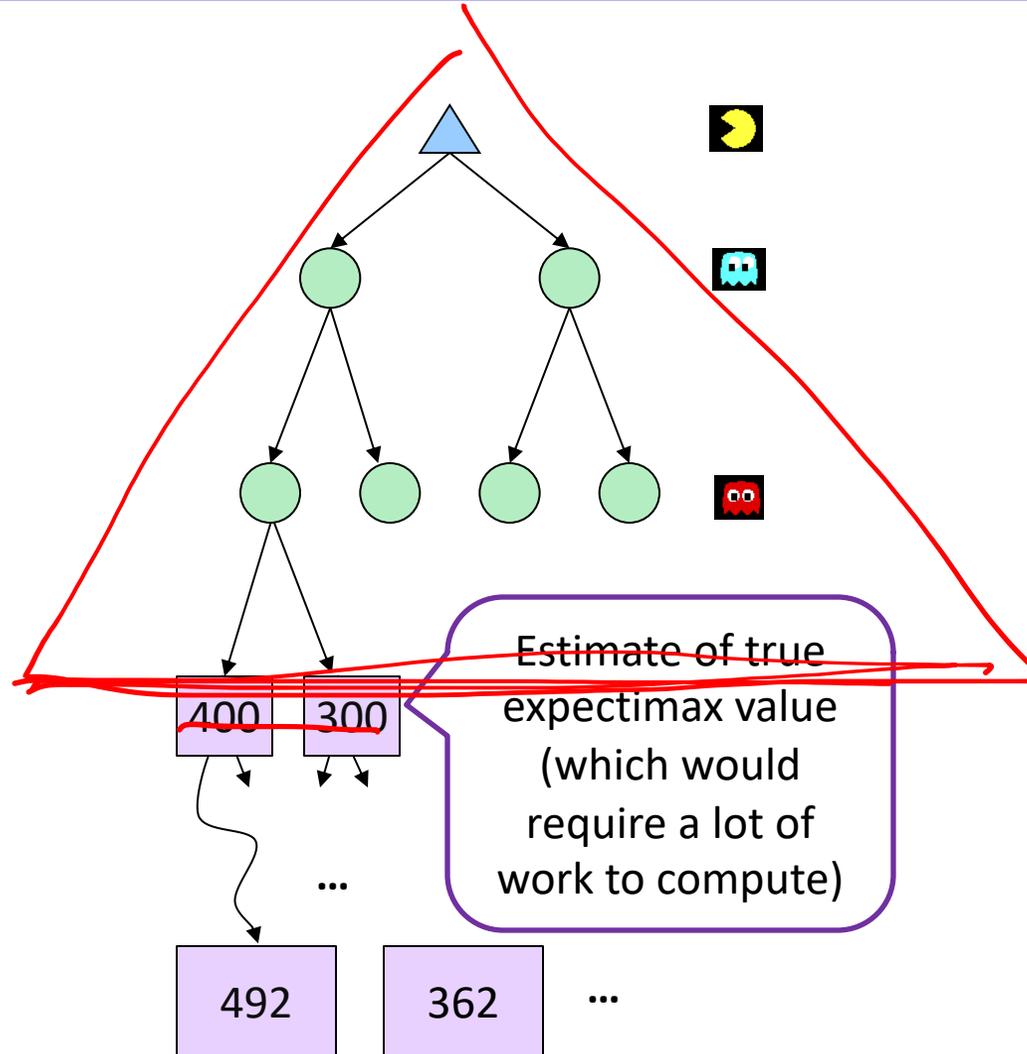
$$v = (1/2)\,(8) + (1/3)\,(24) + (1/6)\,(-12) = 10$$
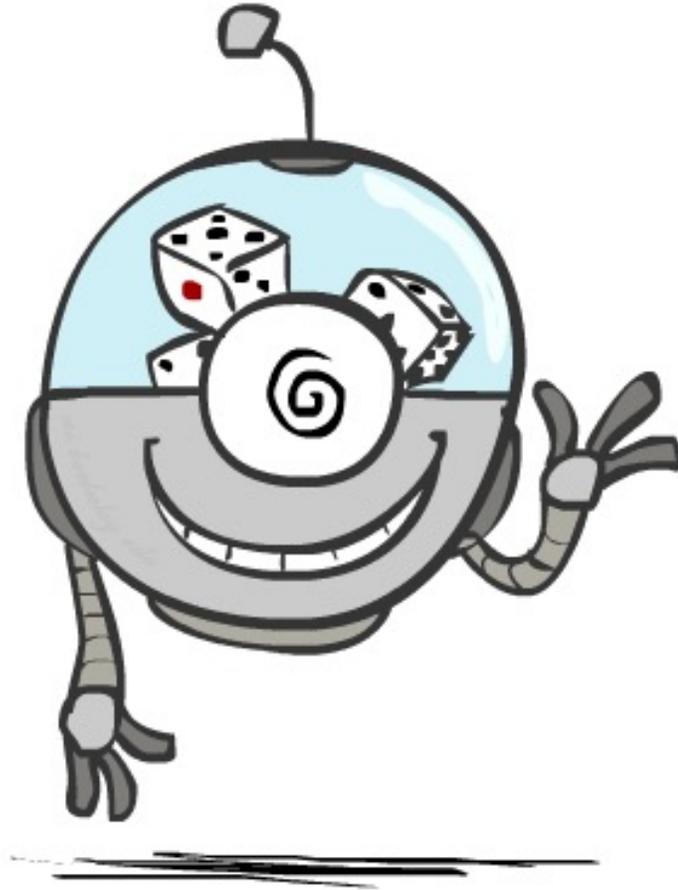
Expectimax Example

Expectimax Pruning?

Depth-Limited Expectimax

400 300

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

...

492   362   ...

Probabilities

Reminder: Probabilities

o   A random variable represents an event whose outcome is unknown
o   A probability distribution is an assignment of weights to outcomes
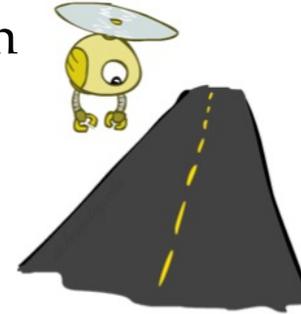
0.25 → .1

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

o   Some laws of probability (more later):
  o   Probabilities are always non-negative
  o   Probabilities over all possible outcomes sum to one

0.50 → .3

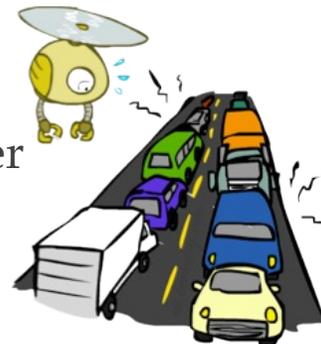o   As we get more evidence, probabilities may change:
  o   P(T=heavy) = 0.25, P(T=heavy | Hour=8am) = 0.60
  o   We'll talk about methods for reasoning and updating probabilities later
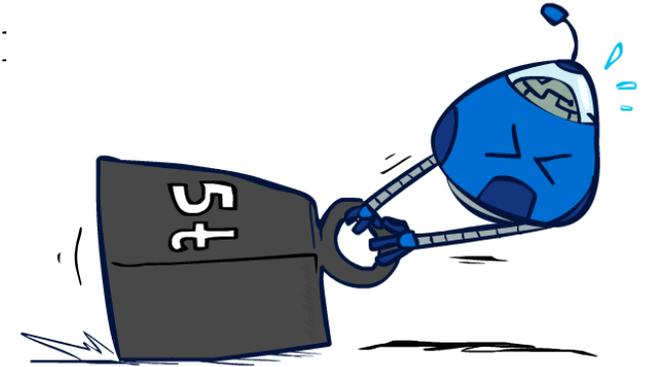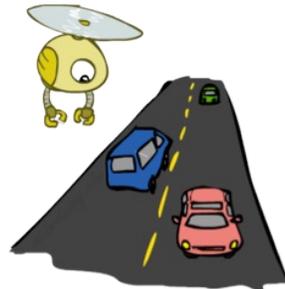
0.25 → .6

Reminder: Expectations

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

o Example: How long to get to the airport?

| Time:	20 min		30 min		60 min		35 min
	x	+	x	+	x	⟶	
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
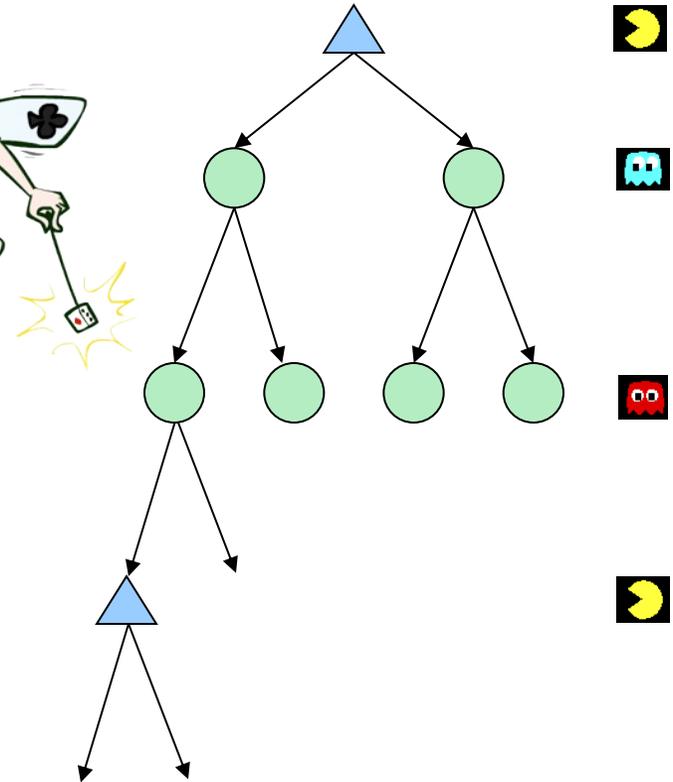
Having a probabilistic belief about another agent's action does not mean that the agent is flipping any coins!

Quiz: Informed Probabilities

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

o Question: What tree search should you use?
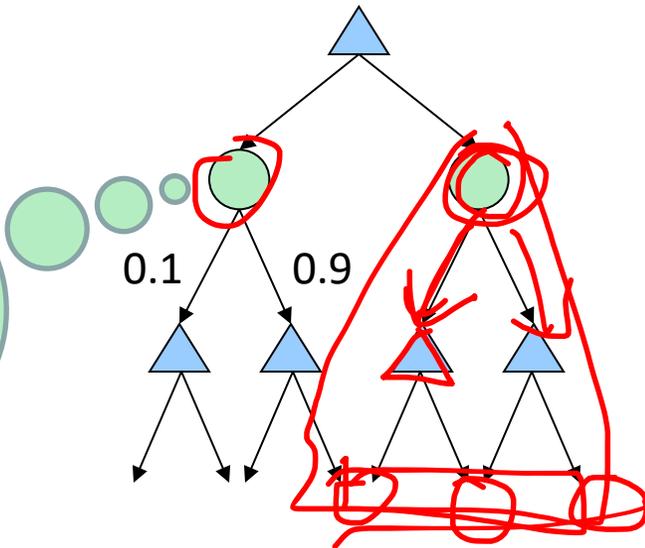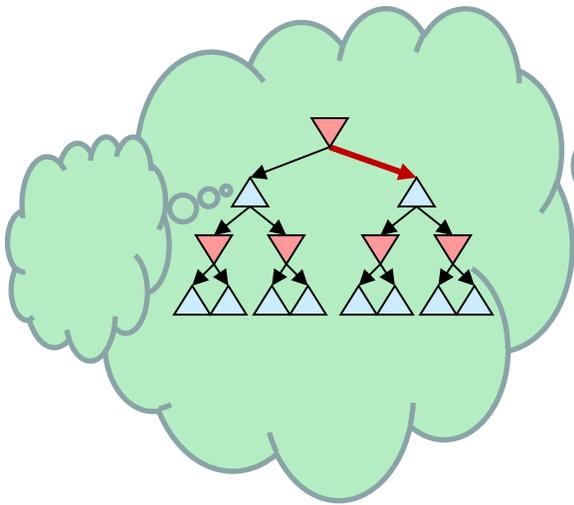
■ **Answer: Expectimax!**

- ■ To figure out EACH chance node's probabilities, you have to run a simulation of your opponent
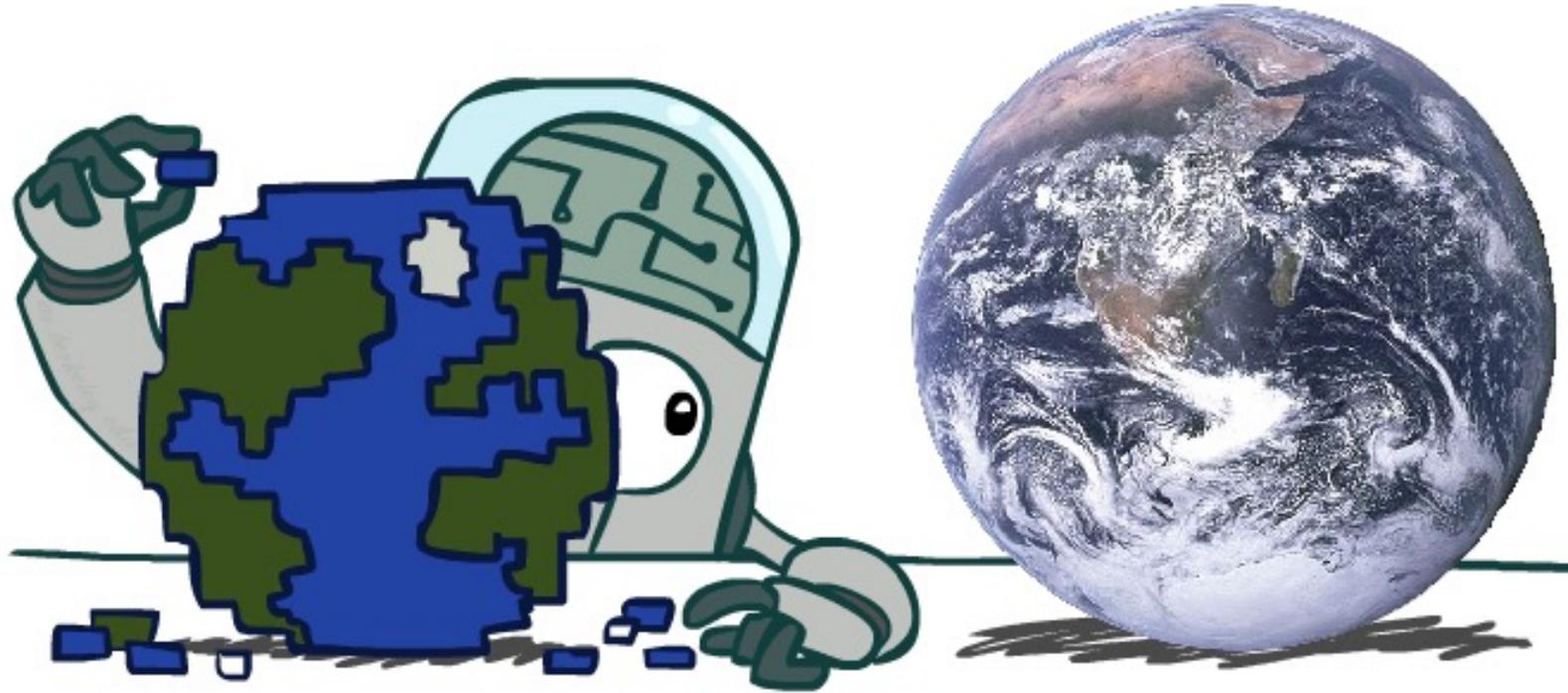- ■ This kind of thing gets very slow very quickly
- ■ Even worse if you have to simulate your opponent simulating you…
- ■ … except for minimax and maximax, which have the nice property that it all collapses into one game tree

80% L 100% R 20%

20% L 50% R 25%

L: 0.8·1 + 0.2·0.5

R: 10%

16

Modeling Assumptions

The Dangers of Optimism and Pessimism

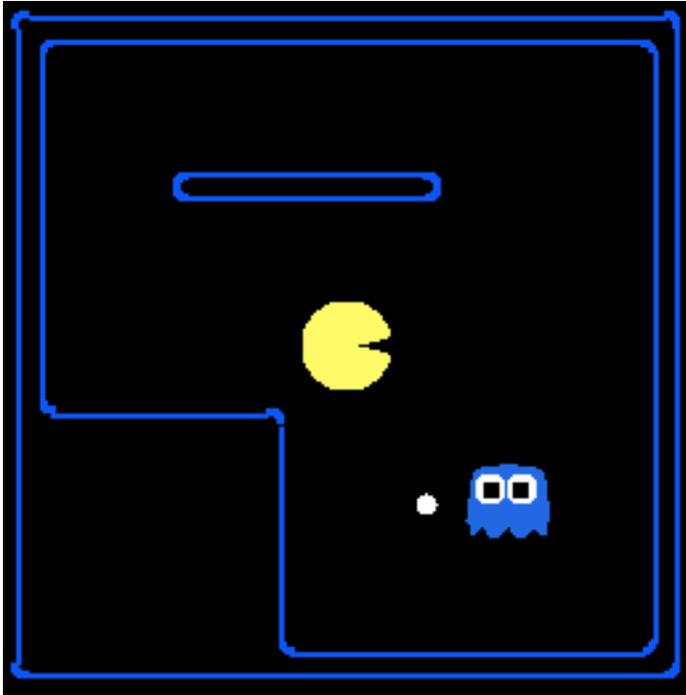### Dangerous Optimism
Assuming chance when the world is adversarial

Dangerous Pessimism
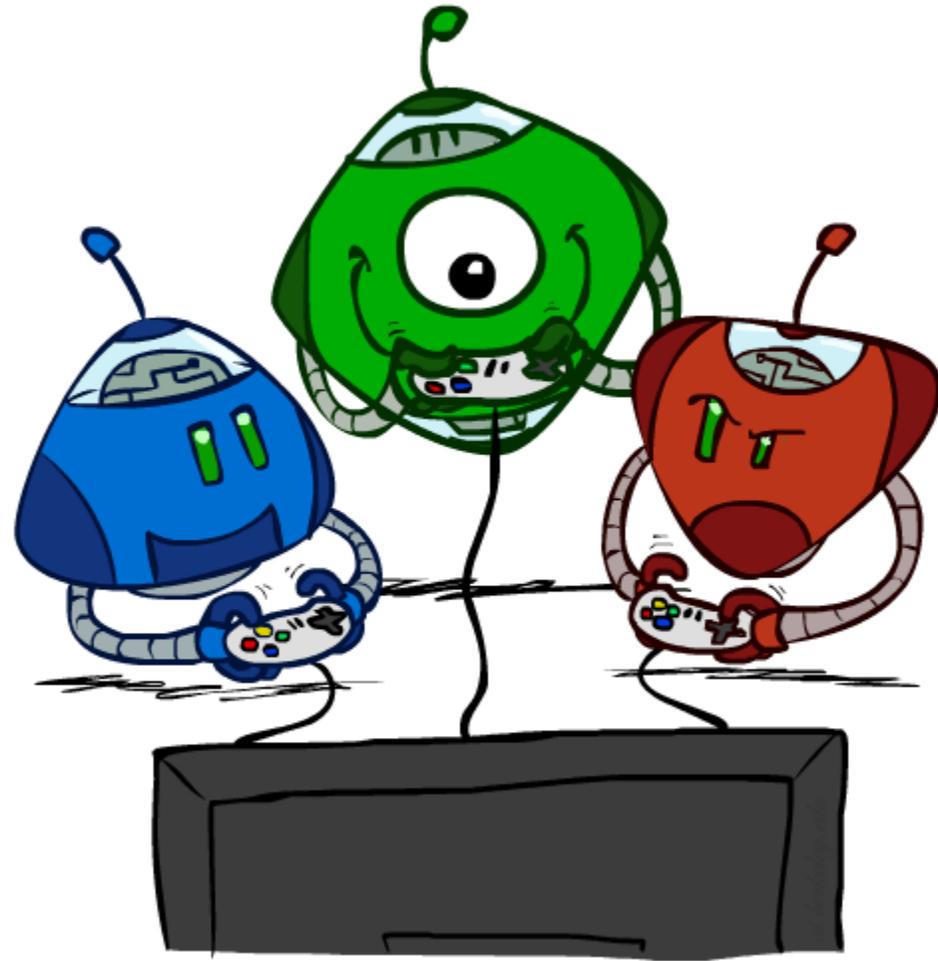Assuming the worst case when it's not likely

Assumptions vs. Reality

|	Adversarial Ghost	Random Ghost
Minimax Pacman		
Expectimax Pacman		

Results from playing 5 games

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

Other Game Types

Mixed Layer Types

o E.g. Backgammon

o Expecti-minimax

  o Environment is an extra "random agent" player that moves after each min/max agent

  o Each node computes the appropriate combination of its children
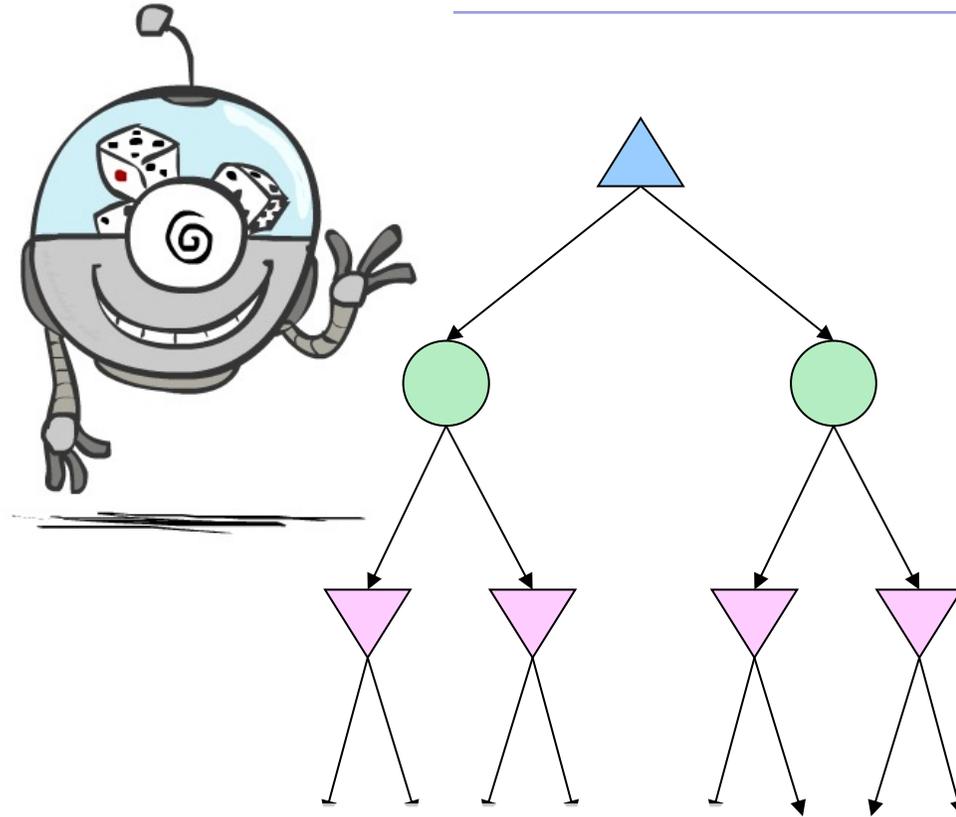
if *state* is a MAX node then
    return the highest EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)
if *state* is a MIN node then
    return the lowest EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)
if *state* is a chance node then
   return average of EXPECTIMINIMAX-VALUE of SUCCESSORS(*state*)

Example: Backgammon

o Dice rolls increase b: 21 possible rolls with 2 dice
  - o Backgammon $\approx$ 20 legal moves
  - o Depth 2 = 20 x (21 x 20)^3 = 1.2 x 10^9

o As depth increases, probability of reaching a given search node shrinks
  - o So usefulness of search is diminished
  - o So limiting depth is less damaging
  - o But pruning is trickier…

o Historic AI: TDGammon uses depth-2 search + very good evaluation function + reinforcement learning:
world-champion level play

o 1st AI world champion in any game!

Multi-Agent Utilities

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

o Generalization of minimax:
  o Terminals have utility tuples
  o Node values are also utility tuples
  o Each player maximizes its own component
  o Can give rise to cooperation and
    competition dynamically…

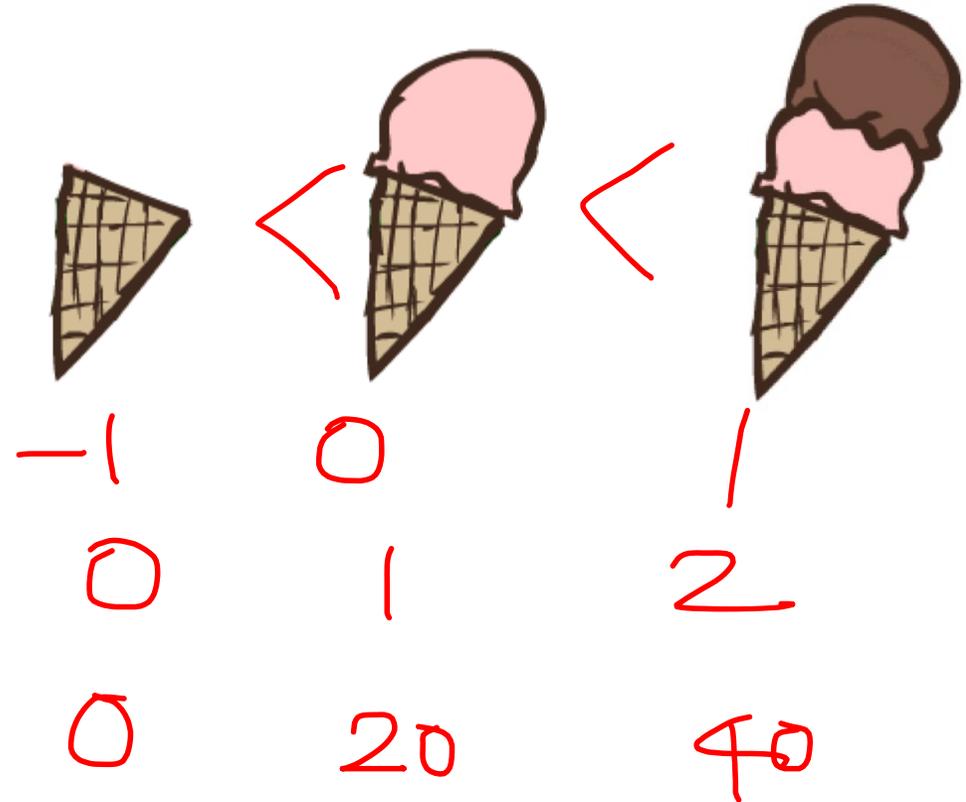| 1,6,6 | 7,1,2 | 6,1,2 | 7,2,1 | 5,1,7 | 1,5,2 | 7,7,1 | 5,2,5 |
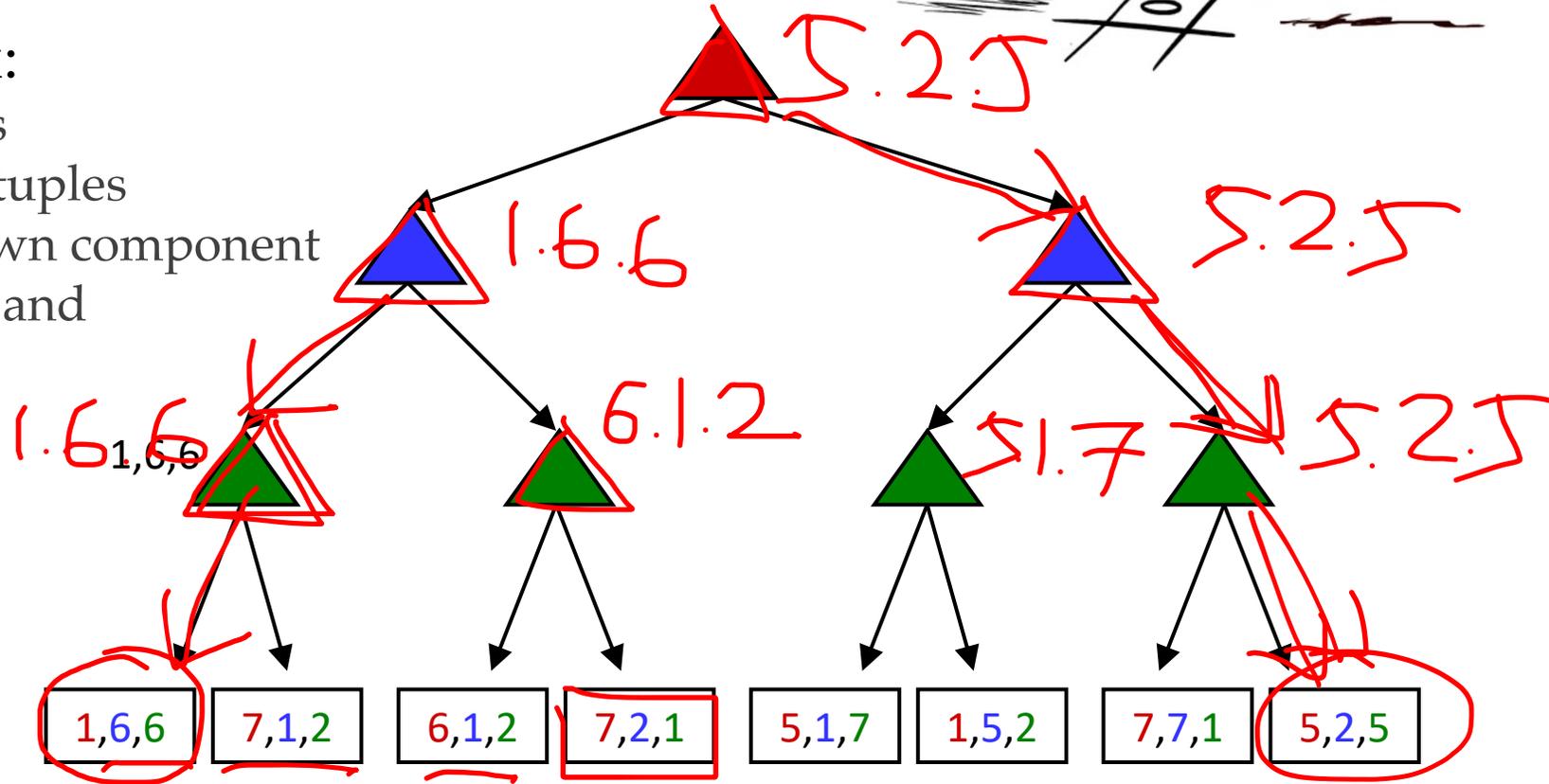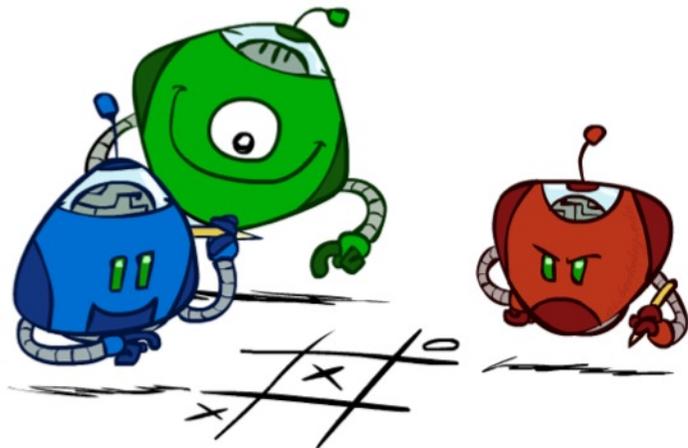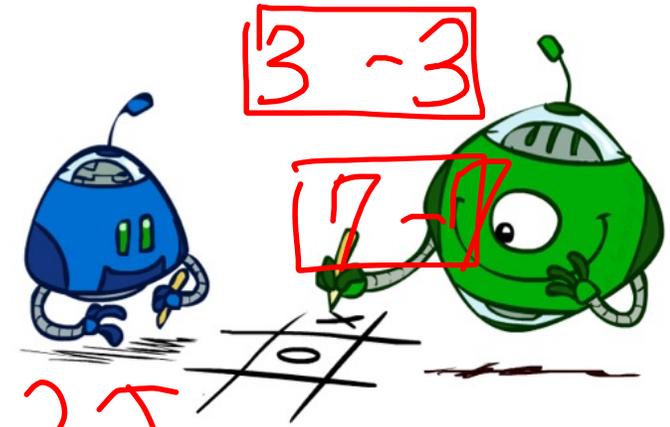
Utilities

o Utilities: values that we assign to every state

o Why should we average utilities? Why not minimax?

o Principle of maximum expected utility:
  o A rational agent should choose the action that maximizes its expected utility, given its knowledge
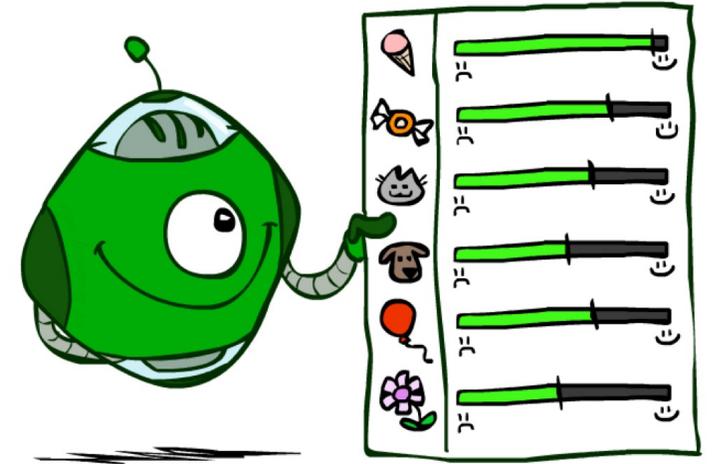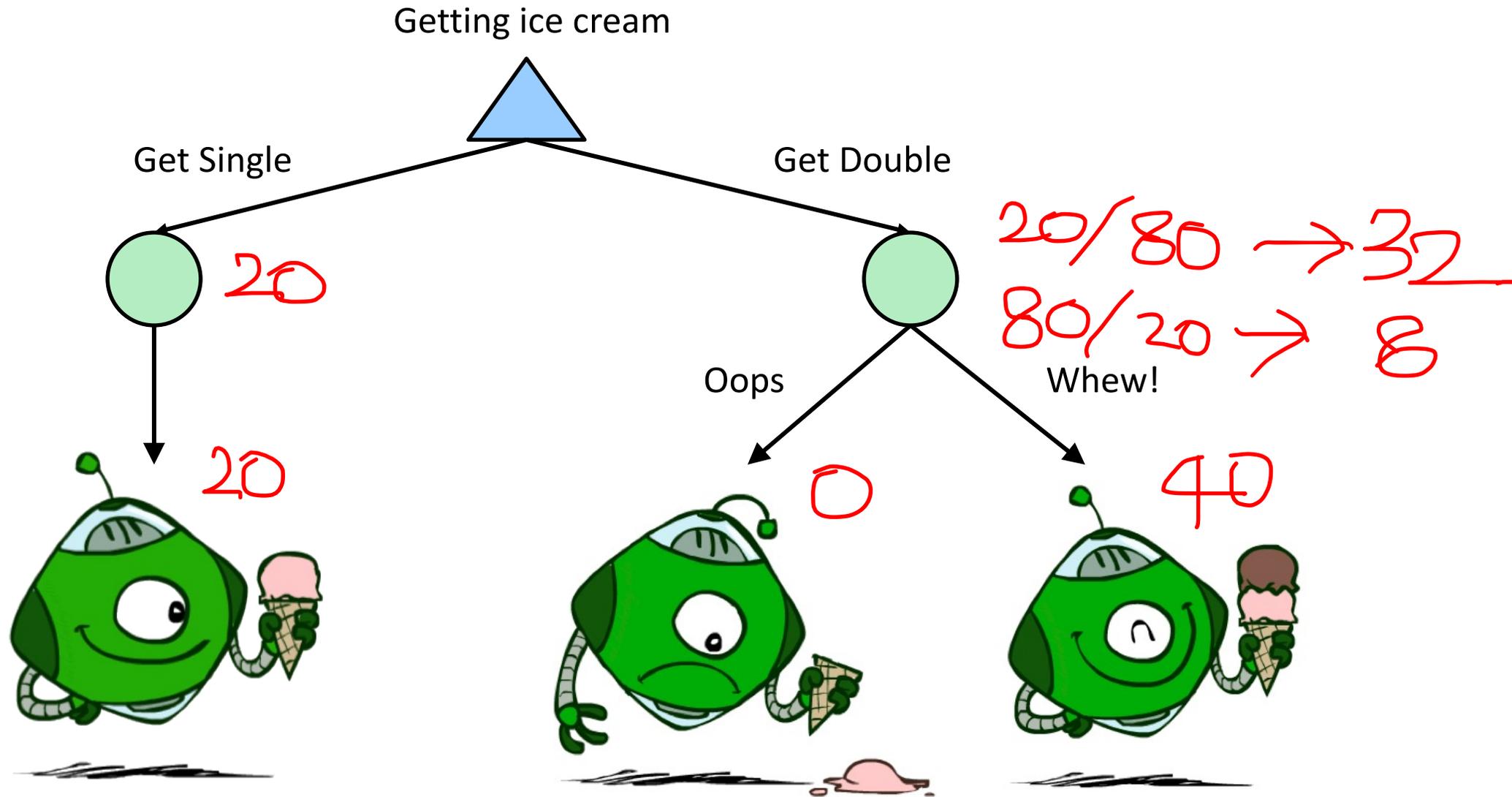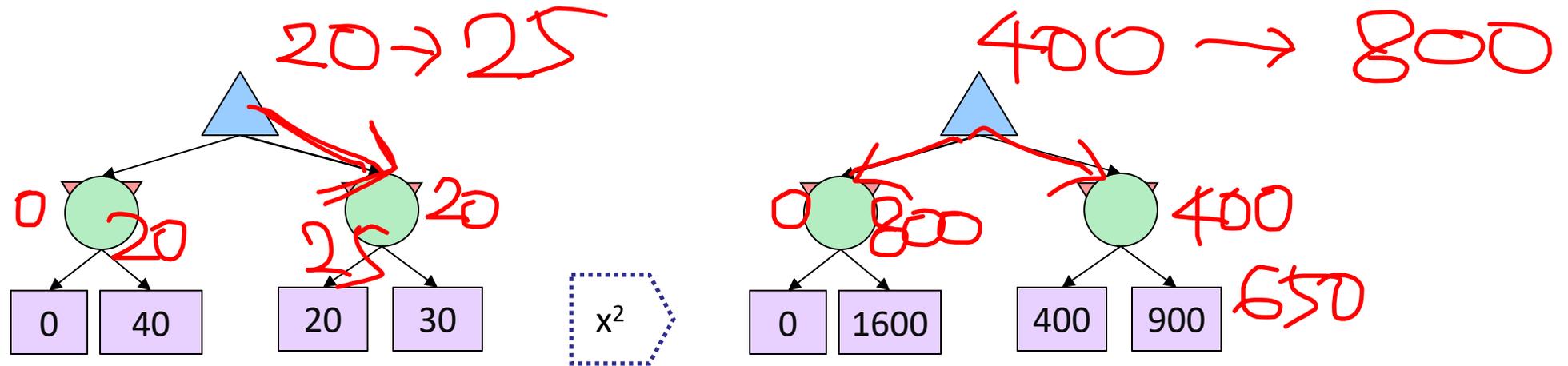
Utilities

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

o Where do utilities come from?
  o In a game, may be simple (+1/-1)
  o Utilities summarize the agent's goals

o We hard-wire utilities and let behaviors emerge

Utilities: Uncertain Outcomes

Getting ice cream

Get Single   Get Double

20   20/80 → 32
         80/20 → 8

Oops   Whew!

20   0   40

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

Next Topic: MDPs!