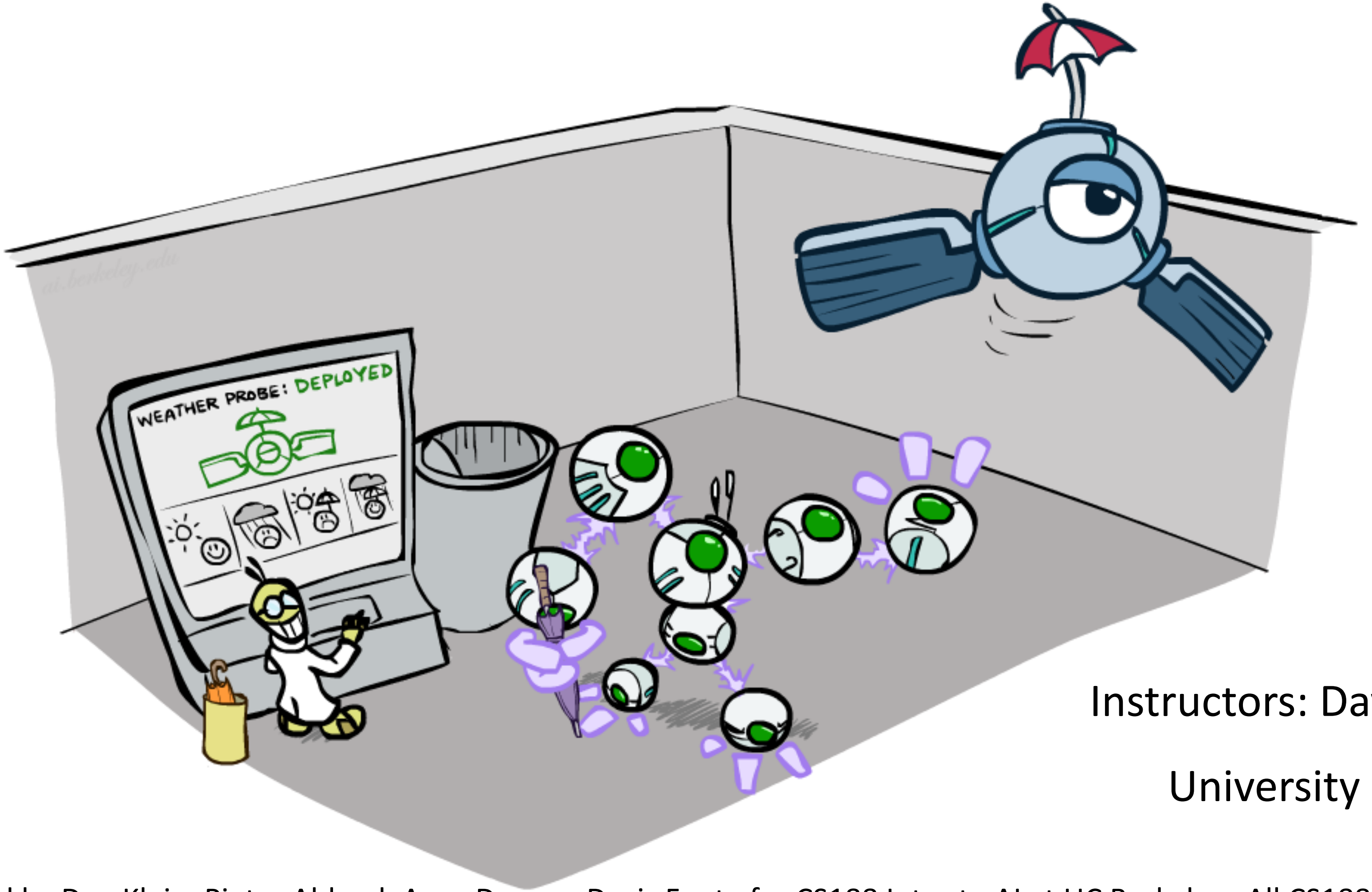
CS 188: Artificial Intelligence

Decision Networks and Value of Perfect Information

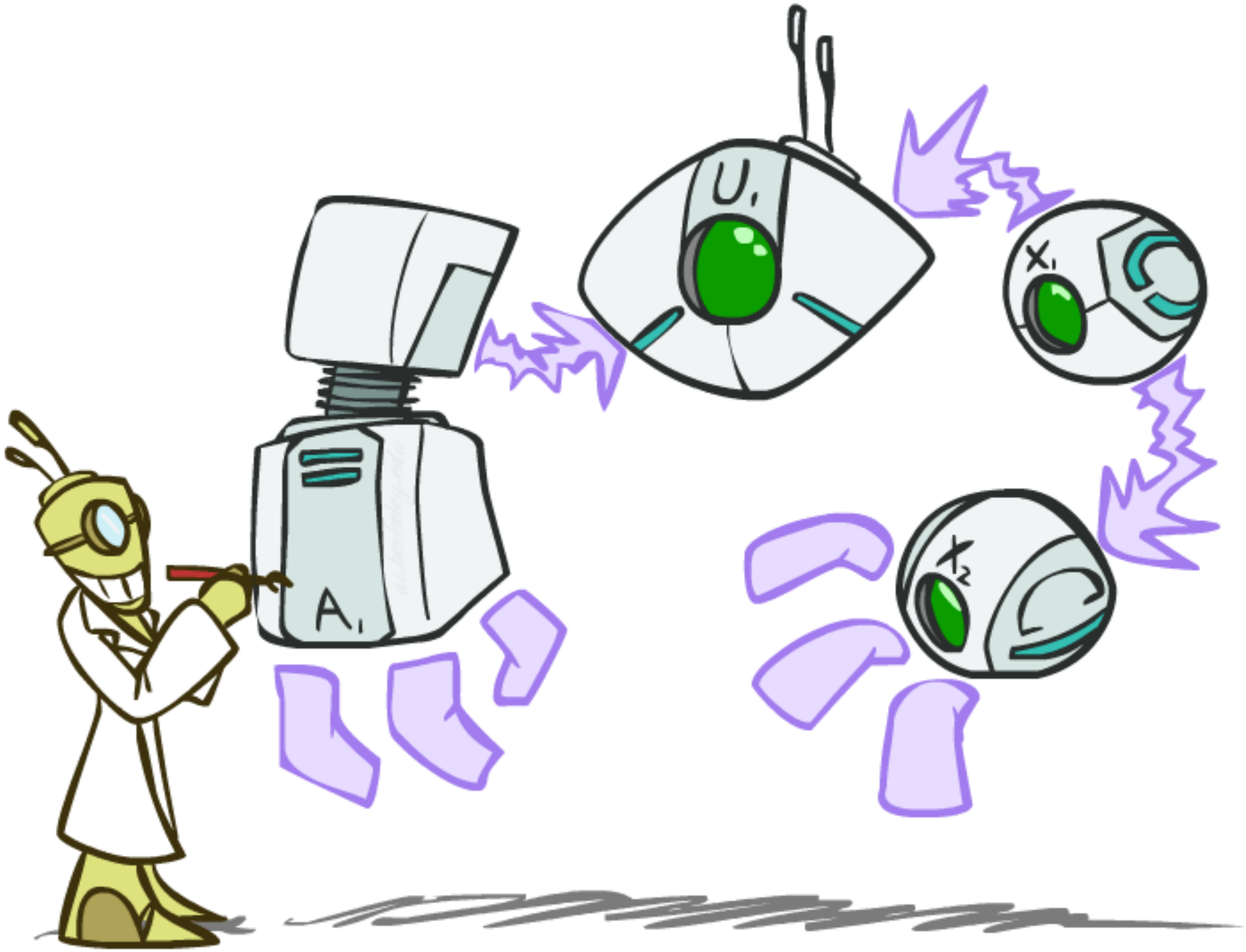


Instructors: Davis Foote & Jacob Andreas

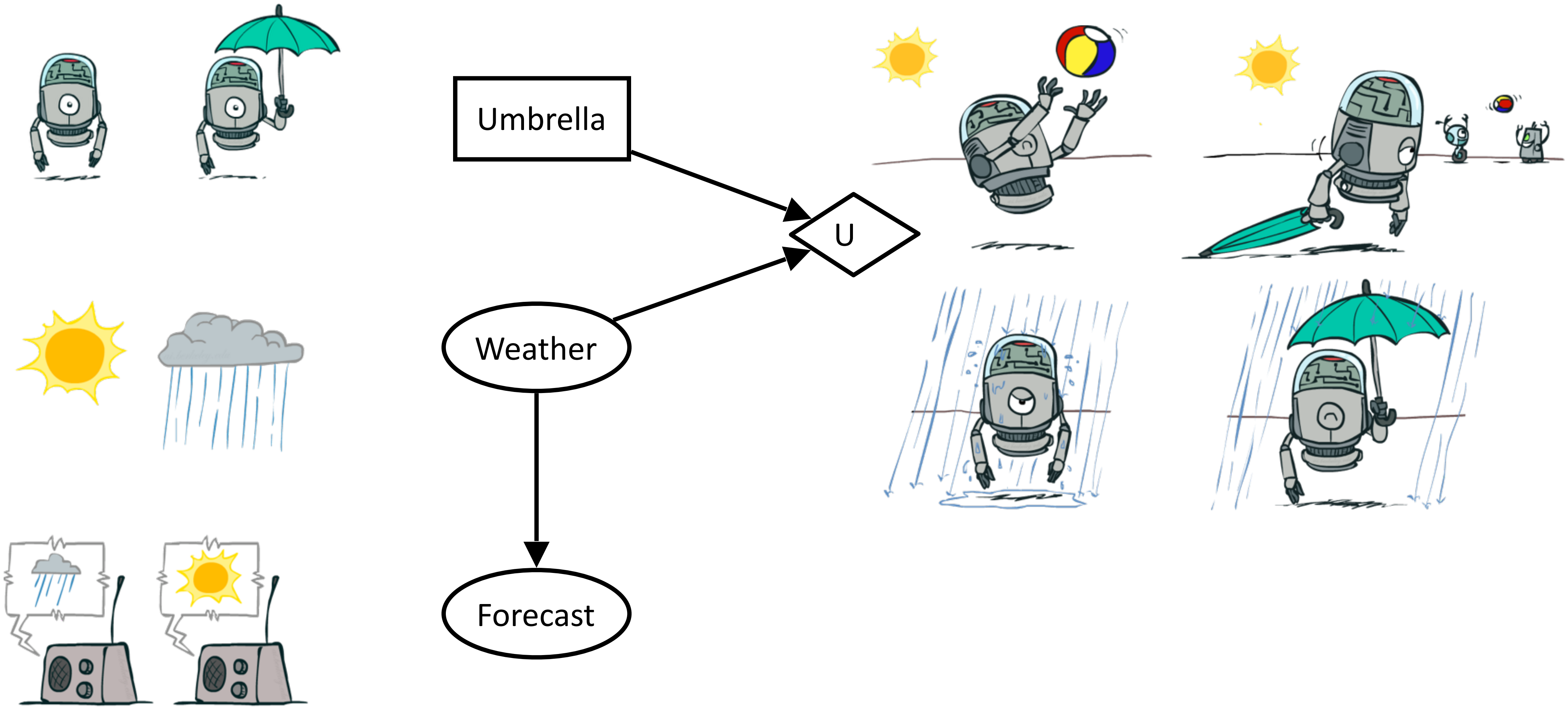
University of California, Berkeley

[Slides created by Dan Klein, Pieter Abbeel, Anca Dragan, Davis Foote for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

Decision Networks



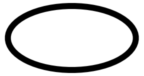
Decision Networks



Decision Networks

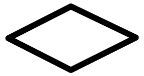
- **MEU: choose the action which maximizes the expected utility given the evidence**

  - Can directly operationalize this with decision networks
    - Bayes nets with nodes for utility and actions
    - Lets us calculate the expected utility for each action
  - New node types:

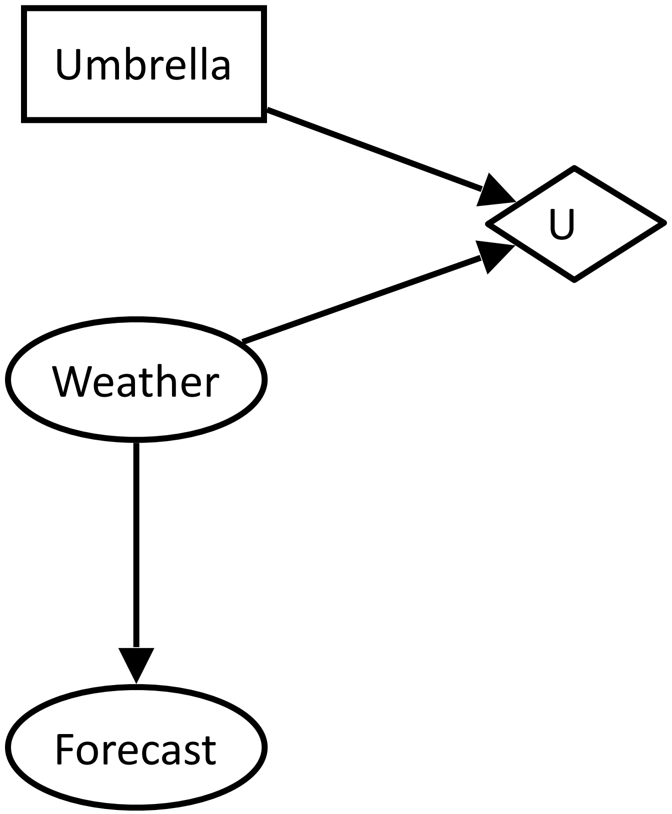


  - Chance nodes (just like BNs)

  - Actions (rectangles, cannot have parents, act as observed evidence)

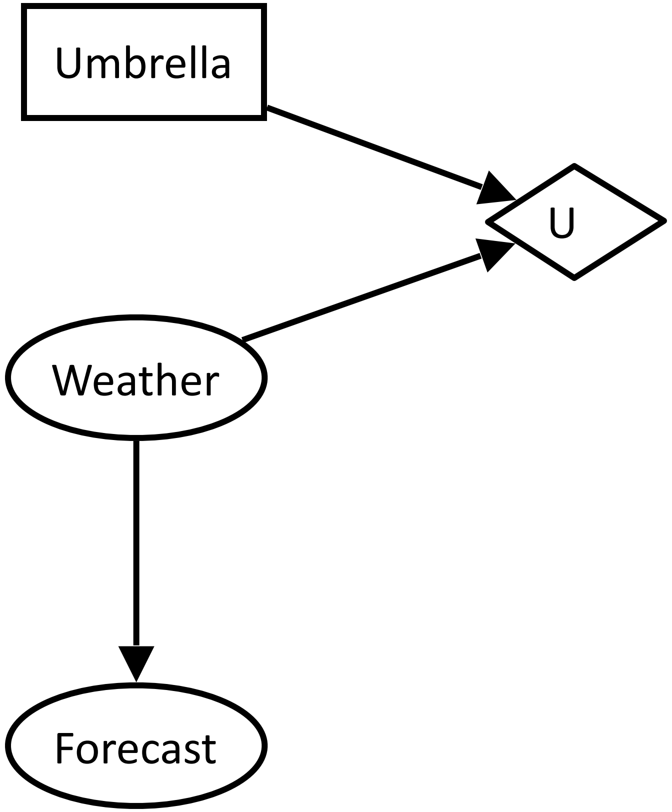


  - Utility node (diamond, depends on action and chance nodes)



Decision Networks

- Action selection
  - Instantiate all evidence
  - Set action node(s) each possible way
  - Calculate posterior for all parents of utility node, given the evidence
  - Calculate expected utility for each action
  - Choose maximizing action



Decision Networks

Umbrella = leave

$$\mathrm{EU}(\text{leave}) = \sum_{w} P(w)U(\text{leave}, w)$$

$$= 0.7 \cdot 100 + 0.3 \cdot 0 = 70$$

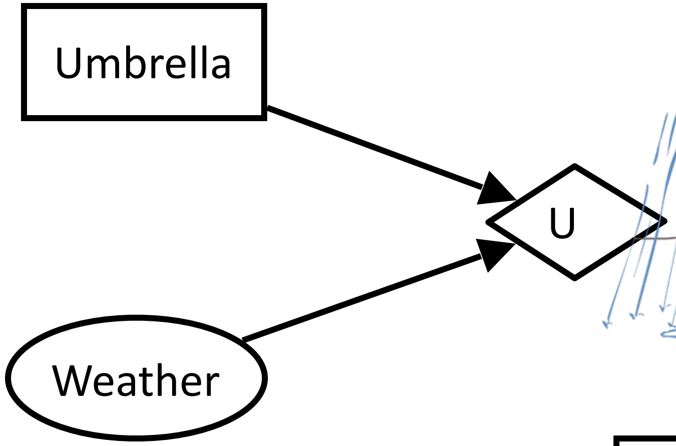
Umbrella = take

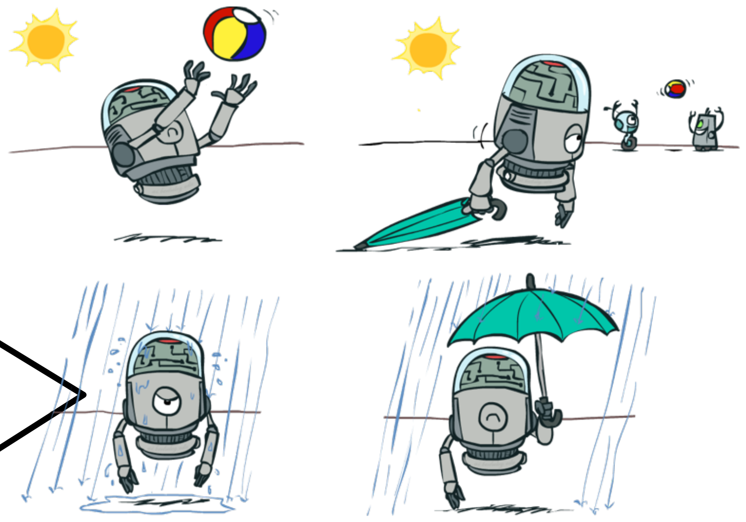
$$\mathrm{EU}(\text{take}) = \sum_{w} P(w)U(\text{take}, w)$$

$$= 0.7 \cdot 20 + 0.3 \cdot 70 = 35$$

Optimal decision = leave

$$\mathrm{MEU}(\emptyset) = \max_{a} \mathrm{EU}(a) = 70$$

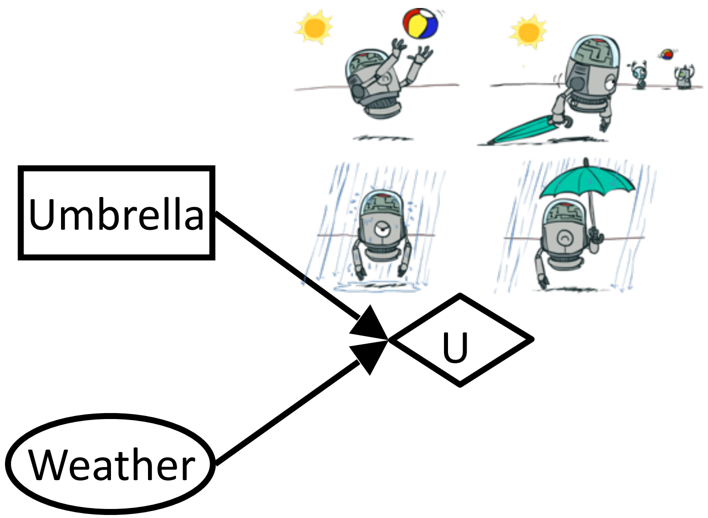


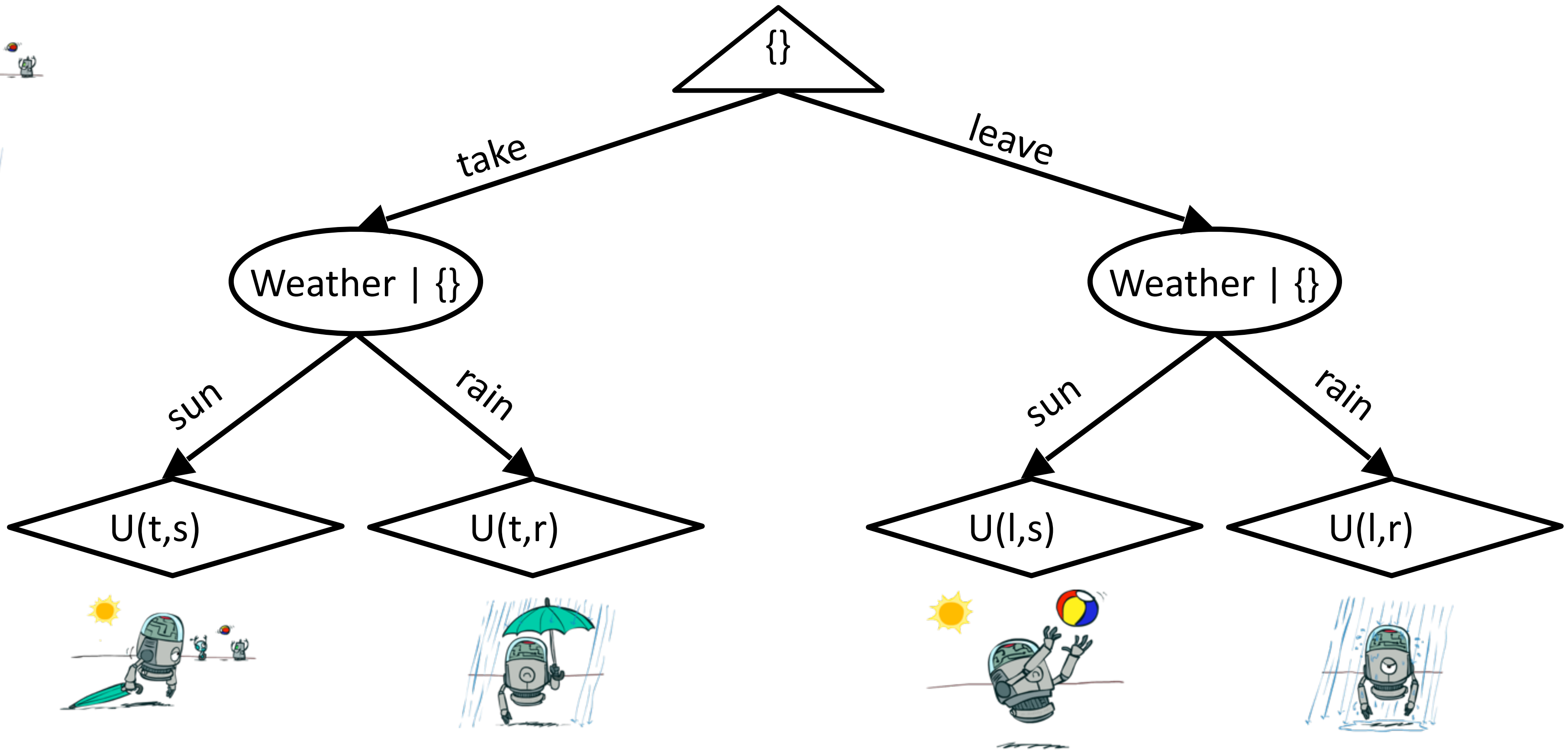


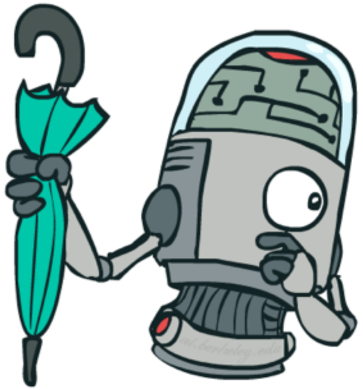
W	P(W)
sun	0.7
rain	0.3

A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

Decisions as Outcome Trees







- Almost exactly like expectimax / MDPs
- What's changed?

Example: Decision Networks

Umbrella = leave

$$\mathrm{EU}(\mathrm{leave}|\mathrm{bad}) = \sum_{w} P(w|\mathrm{bad})U(\mathrm{leave}, w)$$

$$= 0.34 \cdot 100 + 0.66 \cdot 0 = 34$$

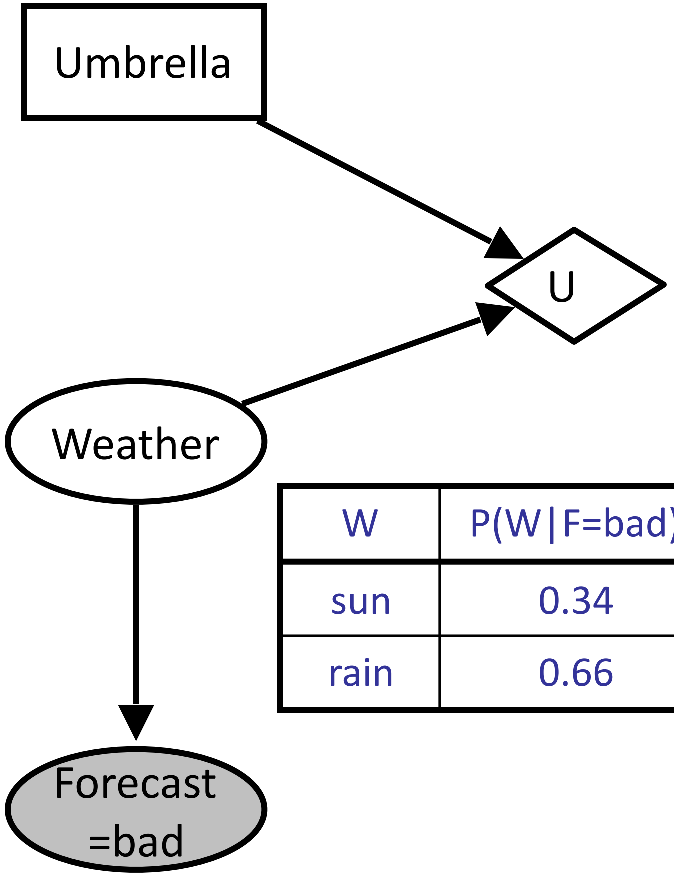
Umbrella = take

$$\mathrm{EU}(\mathrm{take}|\mathrm{bad}) = \sum_{w} P(w|\mathrm{bad})U(\mathrm{take}, w)$$

$$= 0.34 \cdot 20 + 0.66 \cdot 70 = 53$$

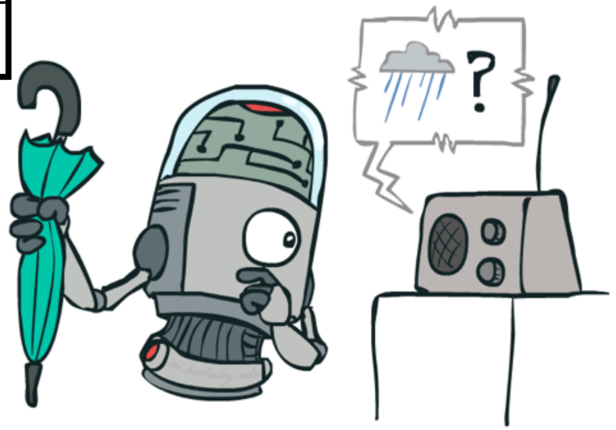
Optimal decision = take

$$\mathrm{MEU}(F = \mathrm{bad}) = \max_{a} \mathrm{EU}(a|\mathrm{bad}) = 53$$

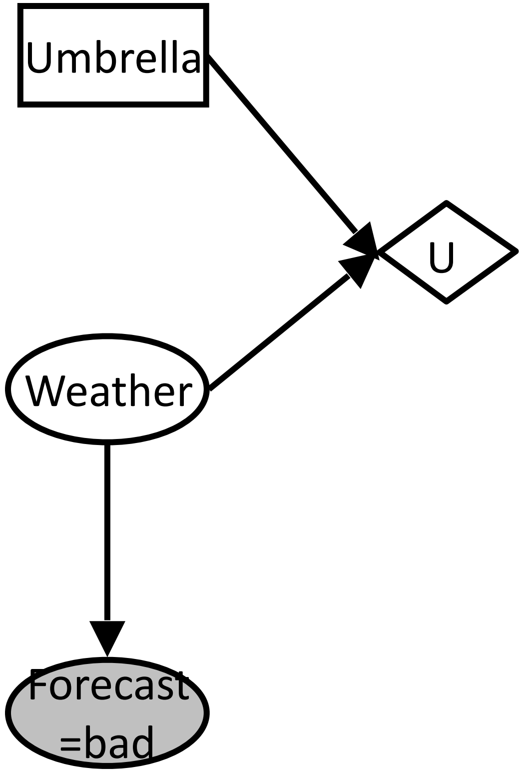


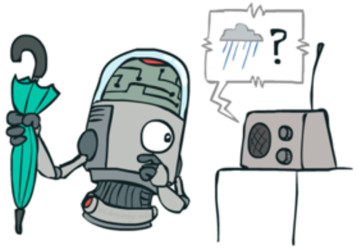
A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

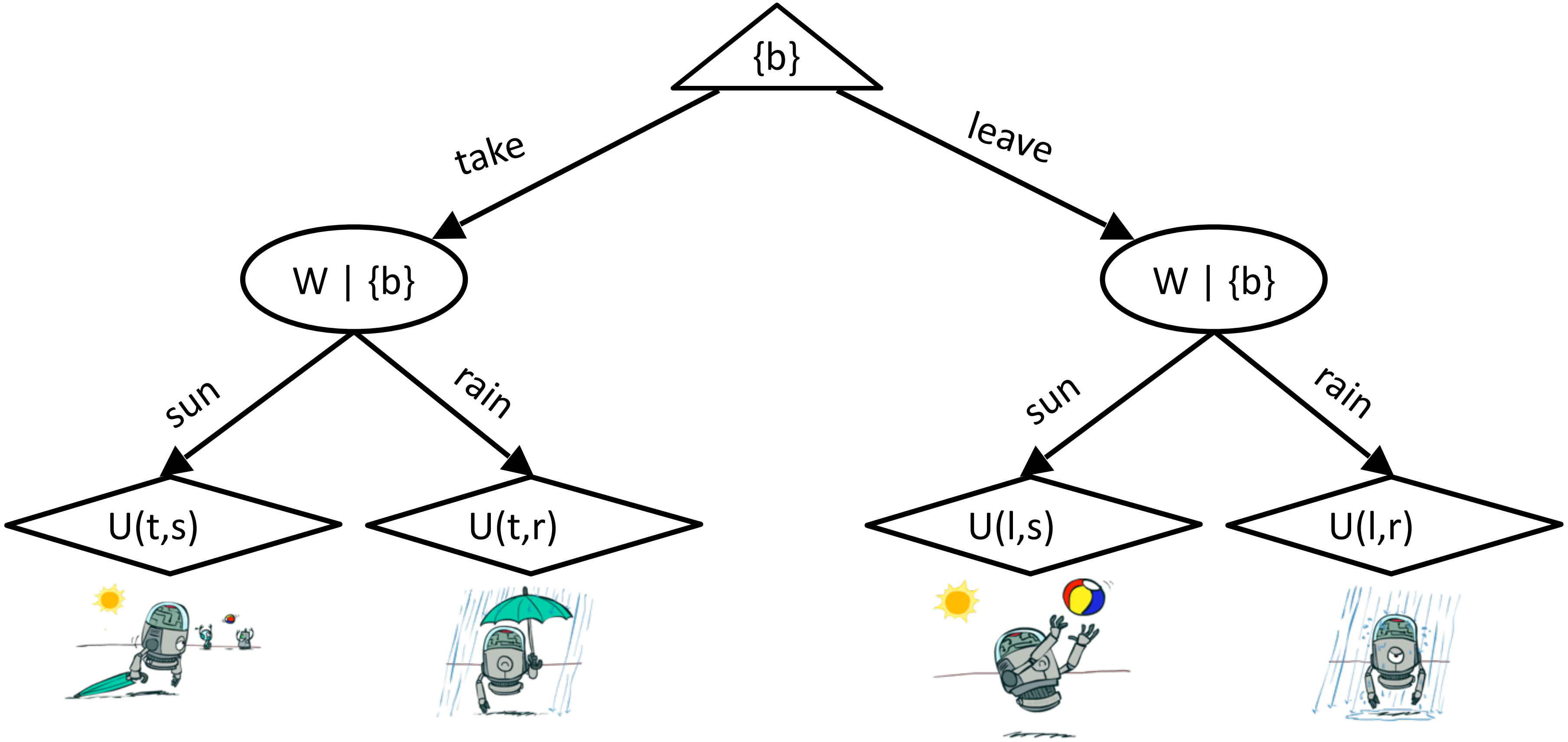
W	P(W|F=bad)
sun	0.34
rain	0.66



Decisions as Outcome Trees

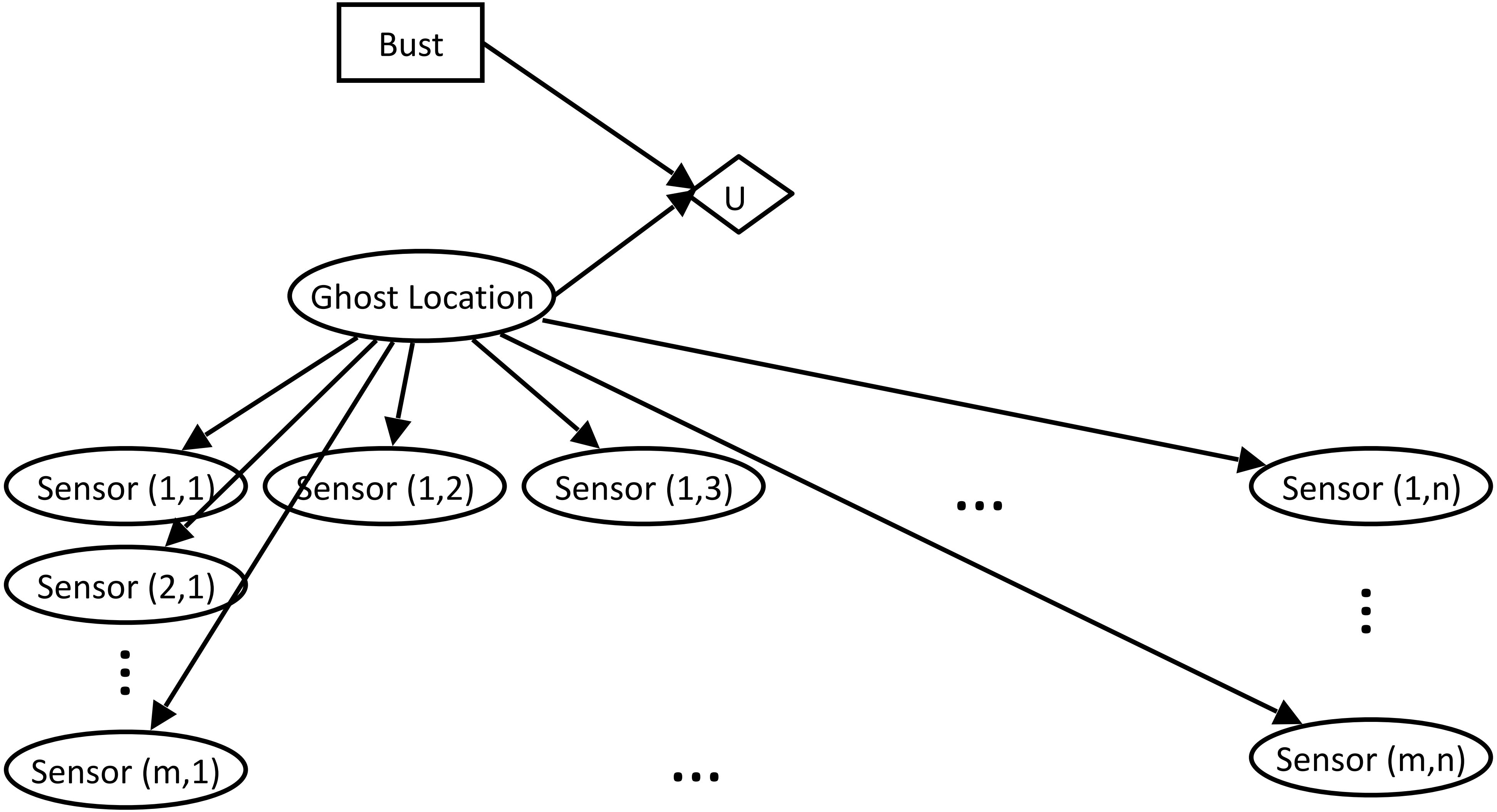




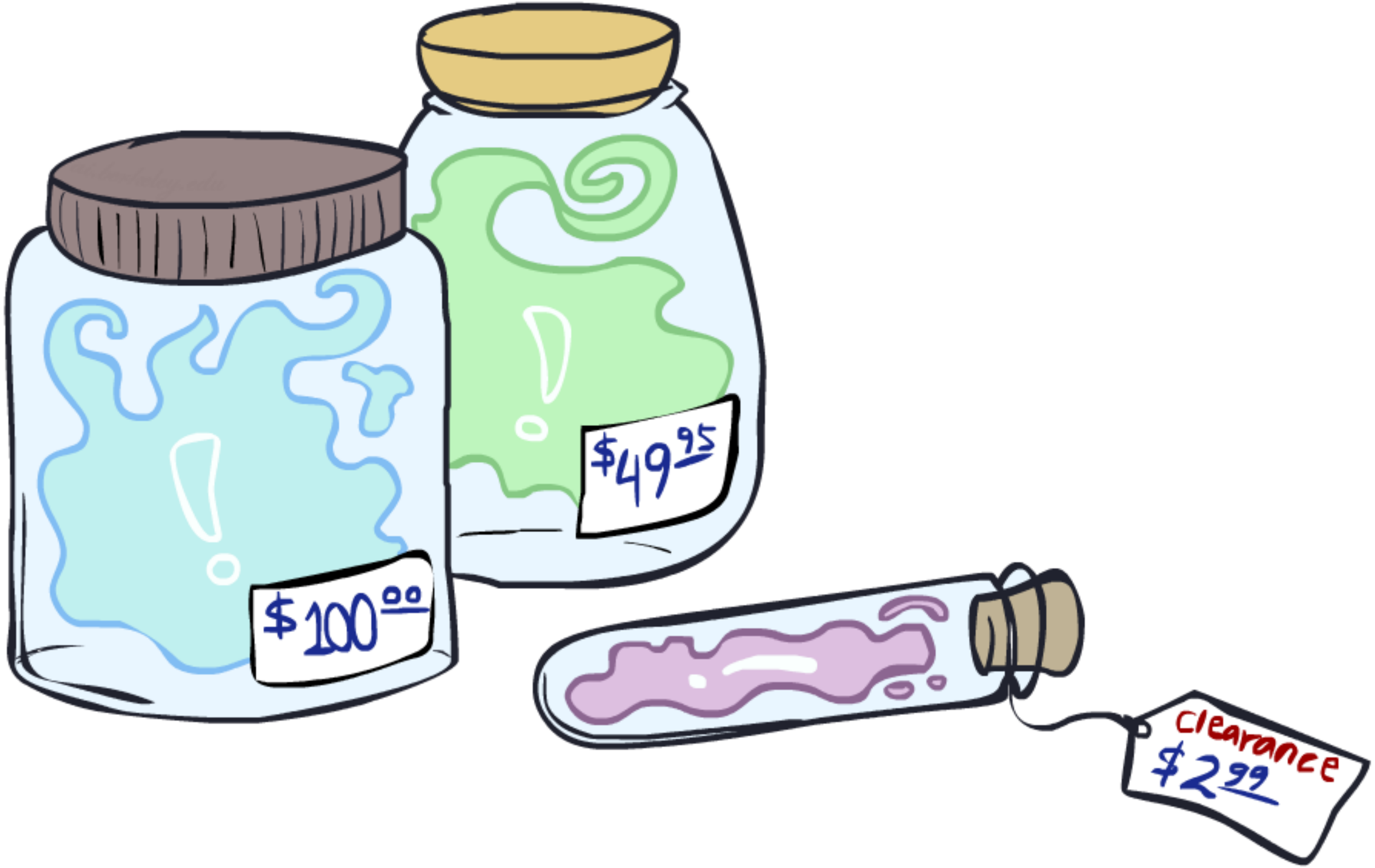


Ghostbusters Decision Network

Demo: Ghostbusters with probability



Value of Information



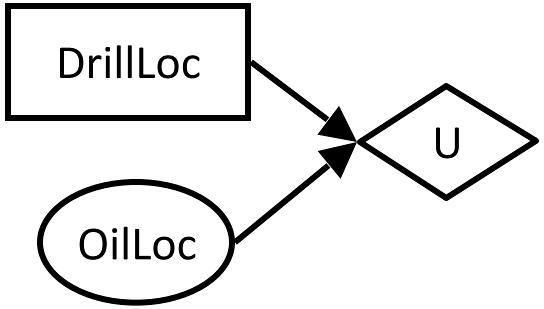
Value of Information

- Idea: compute value of acquiring evidence
  - Can be done directly from decision network

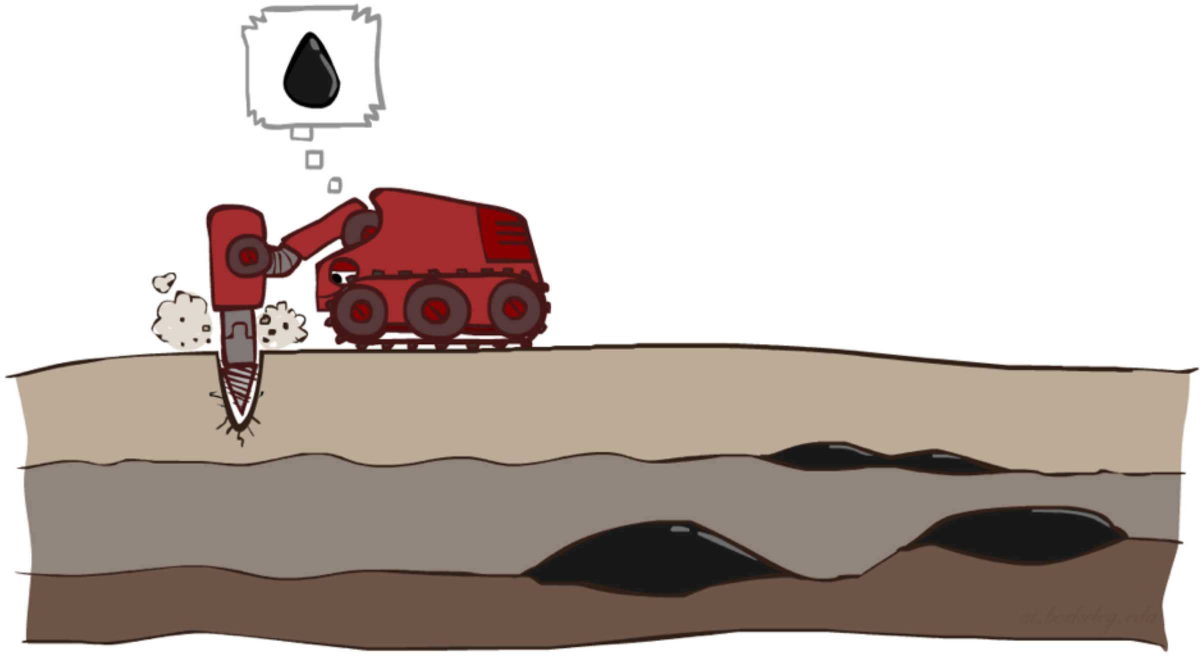
- Example: buying oil drilling rights
  - Two blocks A and B, exactly one has oil, worth k
  - You can drill in one location
  - Prior probabilities 0.5 each, & mutually exclusive
  - Drilling in either A or B has EU = k/2, MEU = k/2

- Question: what’s the value of information of O?
  - Value of knowing which of A or B has oil
  - Value is expected gain in MEU from new info
  - Survey may say “oil in a” or “oil in b,” prob 0.5 each
  - If we know OilLoc, MEU is k (either way)
  - Gain in MEU from knowing OilLoc?
  - VPI(OilLoc) = k/2
  - Fair price of information: k/2

O	P
a	1/2
b	1/2



D	O	U
a	a	k
a	b	0
b	a	0
b	b	k



VPI Example: Weather

MEU with no evidence

$$\mathrm{MEU}(\phi) = \max_a \mathrm{EU}(a) = 70$$

MEU if forecast is bad

$$\mathrm{MEU}(F = \mathrm{bad}) = \max_a \mathrm{EU}(a|\mathrm{bad}) = 53$$

MEU if forecast is good

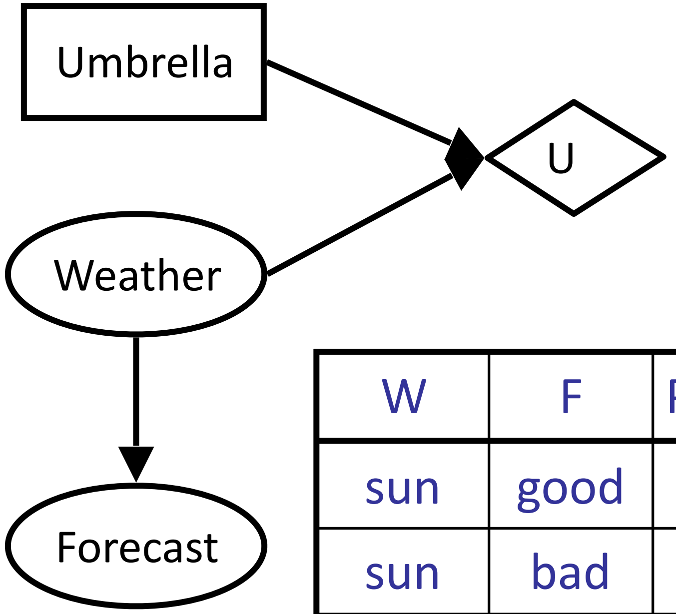
$$\mathrm{MEU}(F = \mathrm{good}) = \max_a \mathrm{EU}(a|\mathrm{good}) = 95$$

Marginal forecast distribution

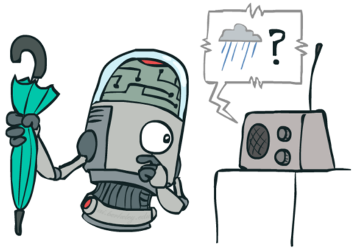
F	P(F)
good	0.59
bad	0.41

$$0.59 \cdot (95) + 0.41 \cdot (53) - 70$$
$$77.8 - 70 = 7.8$$

W	P(W)
sun	0.7
rain	0.3



W	F	P(F|W)
sun	good	0.8
sun	bad	0.2
rain	good	0.1
rain	bad	0.9



$$\mathrm{VPI}(E' \mid e) = \mathbb{E}_{e'|e}\left[\mathrm{MEU}(e,e') - \mathrm{MEU}(e)\right] = \left(\sum_{e'} P(e' \mid e)\mathrm{MEU}(e,e')\right) - \mathrm{MEU}(e)$$

Value of Information

- Assume we have evidence E=e. Value if we act now:

$$\text{MEU}(e) = \max_a \sum_s P(s|e)\, U(s,a)$$

- Assume we see that E' = e'. Value if we act then:

$$\text{MEU}(e,e') = \max_a \sum_s P(s|e,e')\, U(s,a)$$

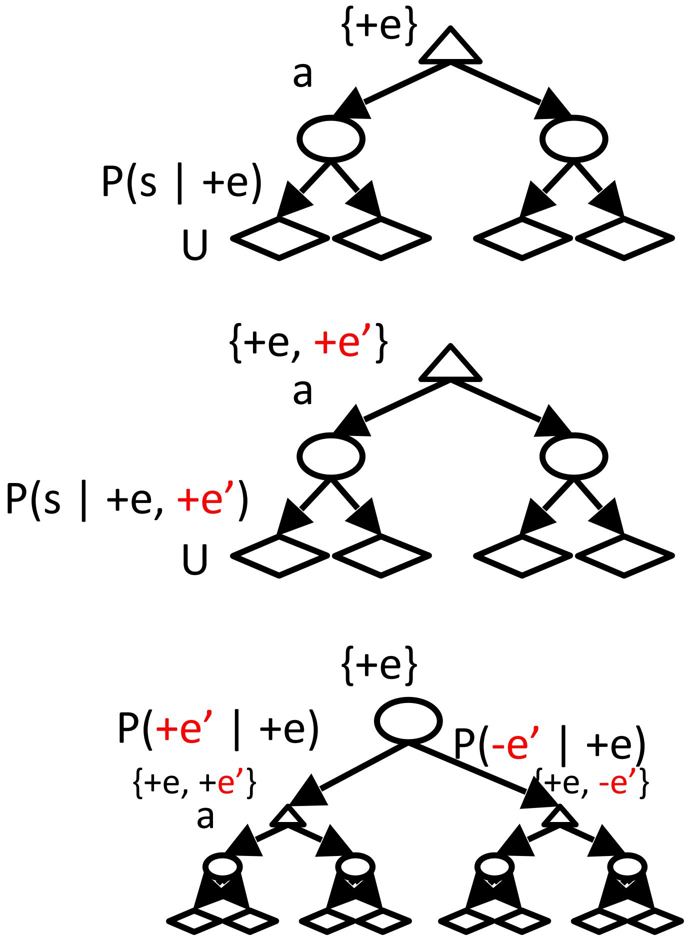
- BUT E' is a random variable whose value is unknown, so we don't know what e' will be

- Expected value if E' is revealed and then we act:

$$\text{MEU}(e,E') = \sum_{e'} P(e'|e)\text{MEU}(e,e')$$

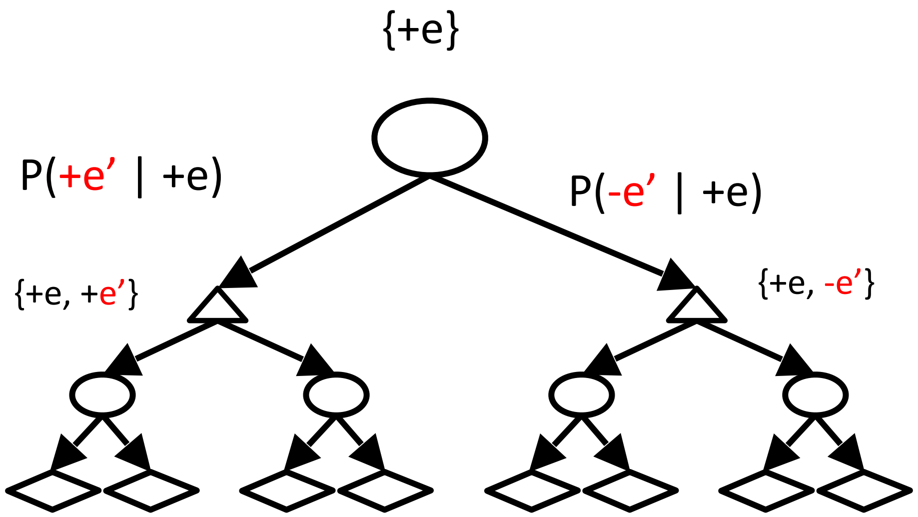
- Value of information: how much MEU goes up by revealing E' first then acting, over acting now:

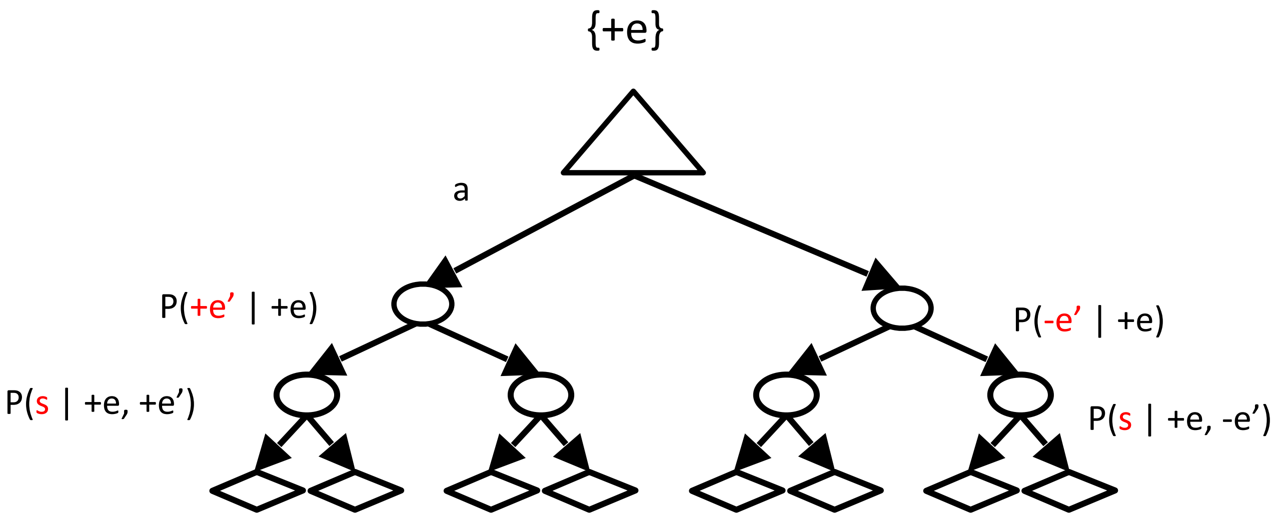
$$\text{VPI}(E'|e) = \text{MEU}(e,E') - \text{MEU}(e)$$



Nonnegativity

$$\text{VPI}(E'|e) = \text{MEU}(e, E') - \text{MEU}(e)$$





Nonnegativity Proof

$$\text{MEU}(e) = \max_a \sum_s P(s|e)\, U(s,a)$$

$$\text{MEU}(e,e') = \max_a \sum_s P(s|e,e')\, U(s,a)$$

$$\text{MEU}(e,E') = \sum_{e'} P(e'|e)\text{MEU}(e,e')$$

$$\text{VPI}(E'|e) = \text{MEU}(e,E') - \text{MEU}(e)$$

$$\text{MEU}(e,E') = \sum_{e'} P(e' \mid e) \max_a \sum_s P(s \mid e, e')U(s,a)$$

$$\begin{aligned}
\text{MEU}(e) &= \max_a \sum_s P(s \mid e)U(s,a) \\
&= \max_a \sum_{e'} P(e' \mid e) \sum_s P(s \mid e, e')U(s,a) \\
&= \sum_{e'} P(e' \mid e) \sum_s P(s \mid e, e')U(s,a_e^*)
\end{aligned}$$

$$\begin{aligned}
\text{VPI}(E' \mid e) &= \text{MEU}(e,E') - \text{MEU}(e) \\
&= \sum_{e'} P(e' \mid e) \left(\left(\max_a \sum_s P(s \mid e, e')U(s,a) \right) - \sum_s P(s \mid e, e')U(s,a_e^*) \right) \\
&= \sum_{e'} P(e' \mid e) \left(\max_a \sum_s P(s \mid e, e') (U(s,a) - U(s,a_e^*)) \right)
\end{aligned}$$

Since a^*_e is a potential choice for a, we can do at least as well as $U(s, a^*_e)$, so this term is nonnegative

VPI Properties

- Nonnegative

$$\forall E', e : \mathrm{VPI}(E'|e) \geq 0$$

- Nonadditive
  (think of observing E_j twice)

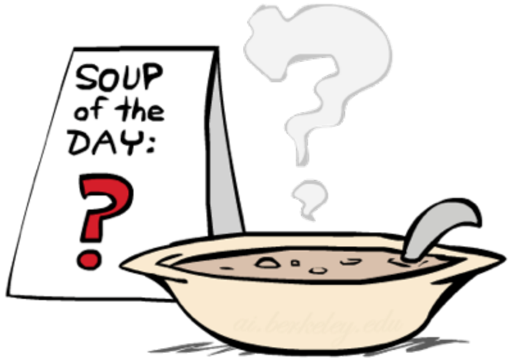
$$\mathrm{VPI}(E_j, E_k|e) \neq \mathrm{VPI}(E_j|e) + \mathrm{VPI}(E_k|e)$$

- Order-independent

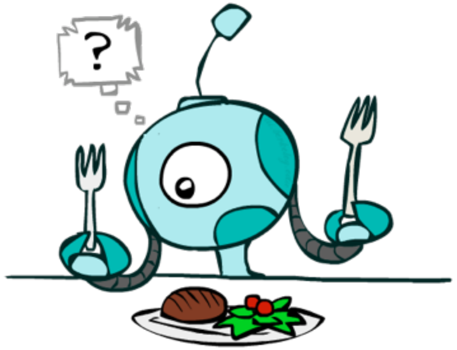
$$\mathrm{VPI}(E_j, E_k|e) = \mathrm{VPI}(E_j|e) + \mathrm{VPI}(E_k|e, E_j)$$
$$= \mathrm{VPI}(E_k|e) + \mathrm{VPI}(E_j|e, E_k)$$

Quick VPI Questions

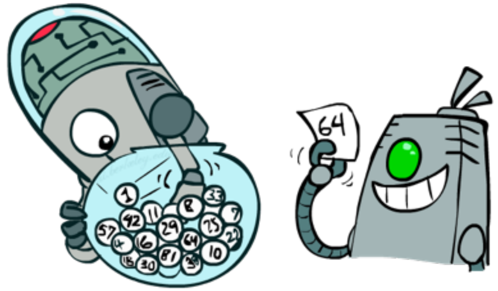
- The soup of the day is either clam chowder or split pea, but you wouldn't order either one. What's the value of knowing which it is?



- There are two kinds of plastic forks at a picnic. One kind is slightly sturdier. What's the value of knowing which?



- You're playing the lottery. The prize will be \$0 or \$100. You can play any number between 1 and 100 (chance of winning is 1%). What is the value of knowing the winning number?

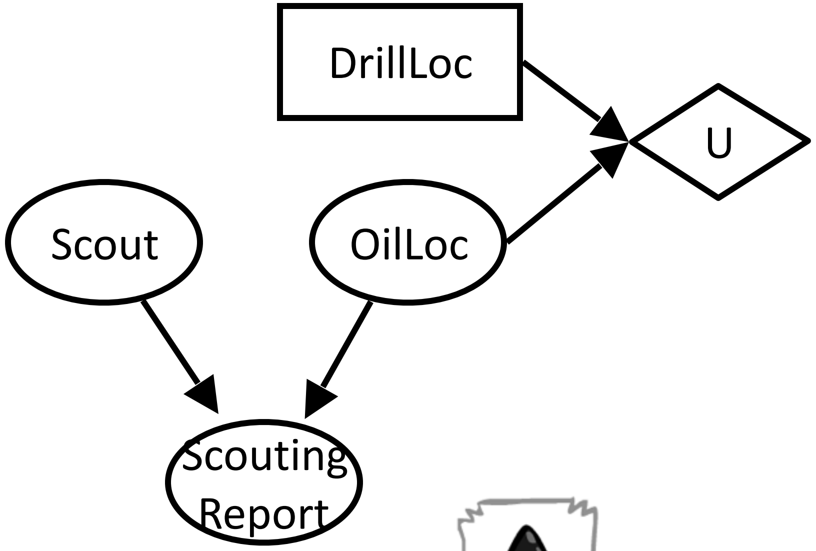


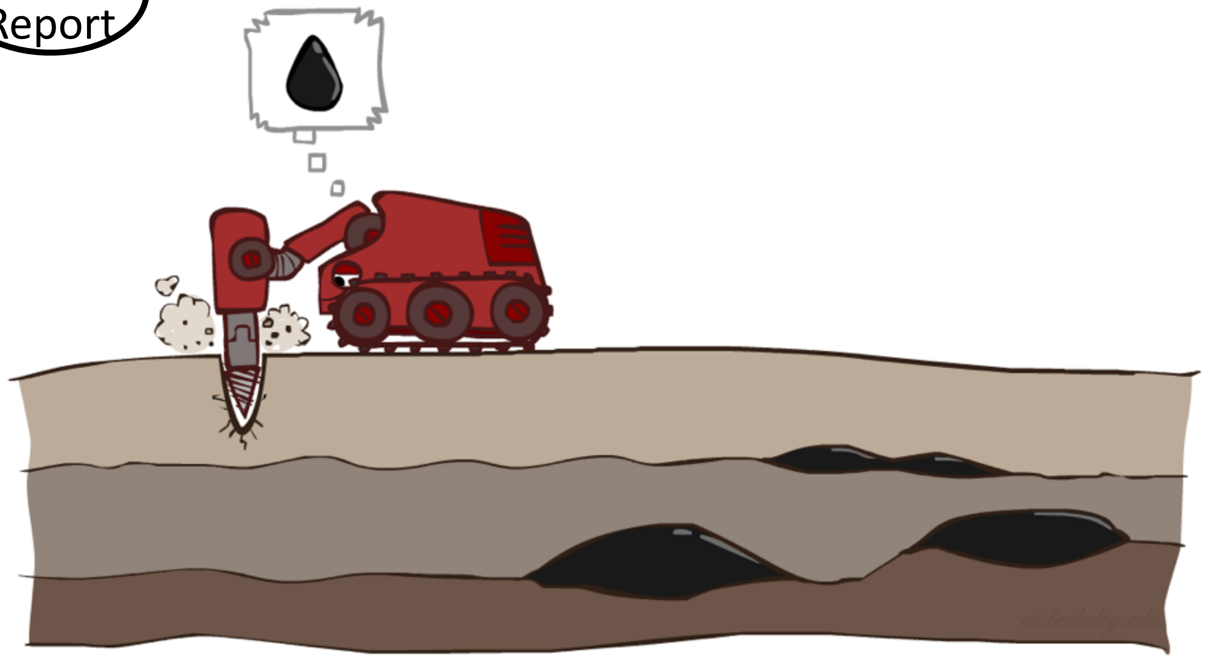
Value of Imperfect Information?

- No such thing
- Information corresponds to the observation of a node in the decision network
- If data is “noisy” that just means we don’t observe the original variable, but another variable which is a noisy version of the original one

VPI Question

- VPI(OilLoc) ?
- VPI(ScoutingReport) ?
- VPI(Scout) ?
- VPI(Scout | ScoutingReport) ?



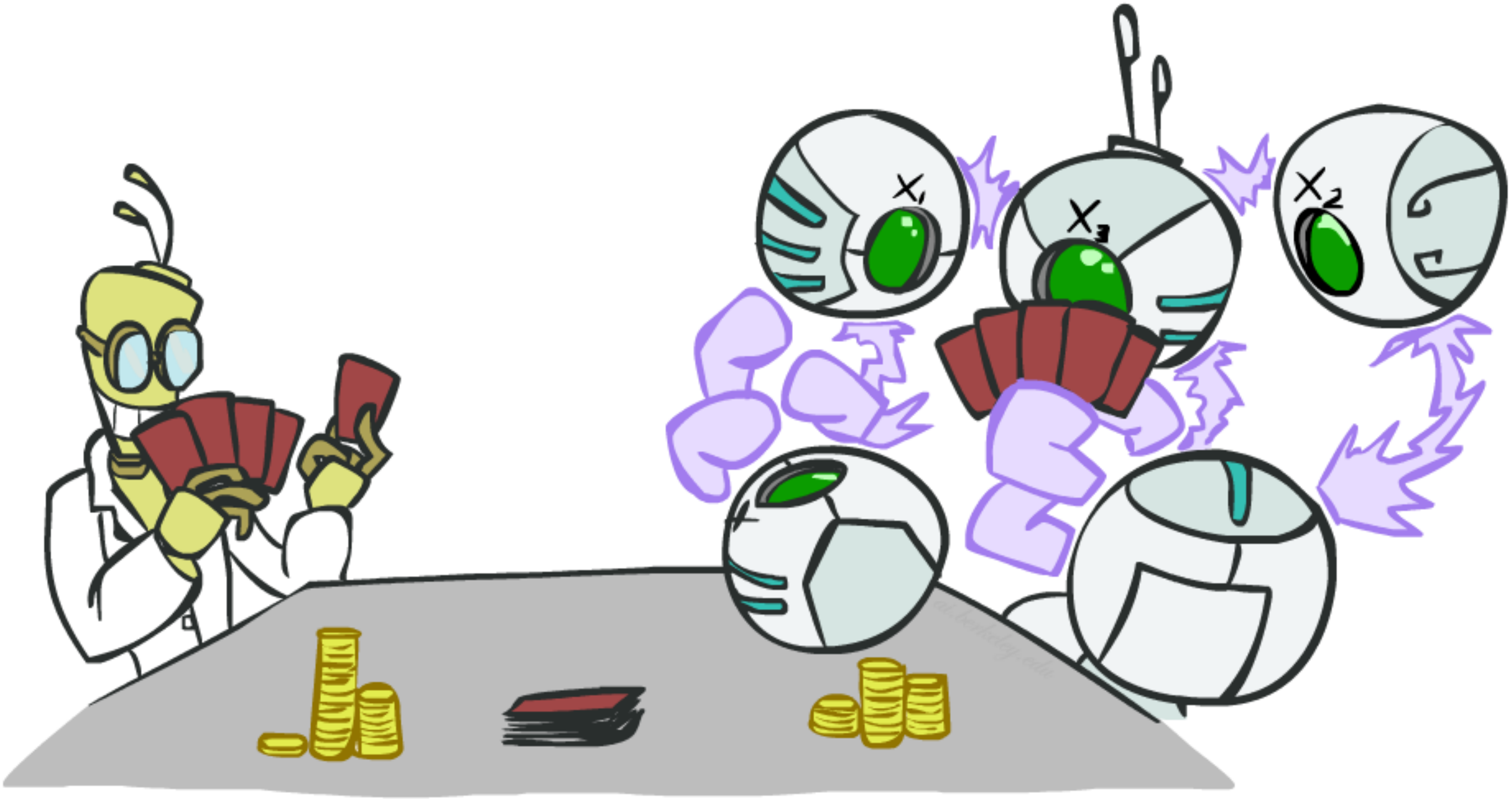


- Generally:

  If $\quad$ Parents(U) $\perp\!\!\!\perp$ Z | CurrentEvidence

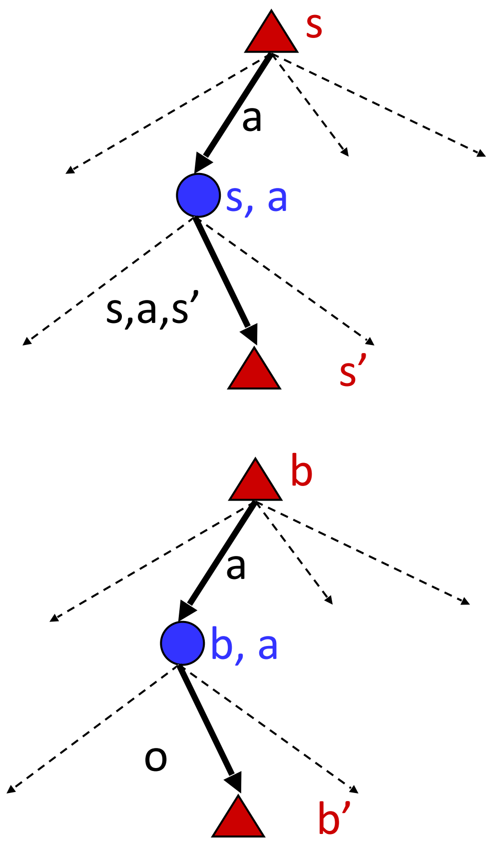
  Then $\quad$ VPI(Z | CurrentEvidence) = 0

POMDPs



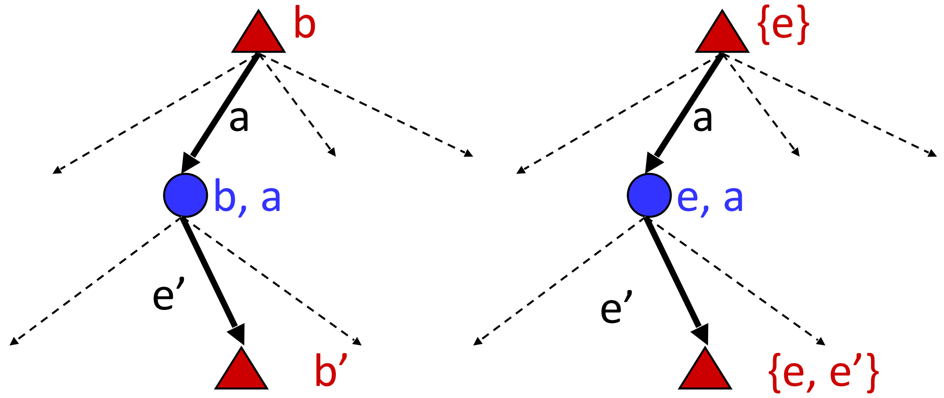
POMDPs

- MDPs have:
  - States S
  - Actions A
  - Transition function P(s'|s,a) (or T(s,a,s'))
  - Rewards R(s,a,s')
- POMDPs add:
  - Observations O
  - Observation function P(o|s) (or O(s,o))
- POMDPs are MDPs over belief states b (distributions over S)
- We'll be able to say more in a few lectures

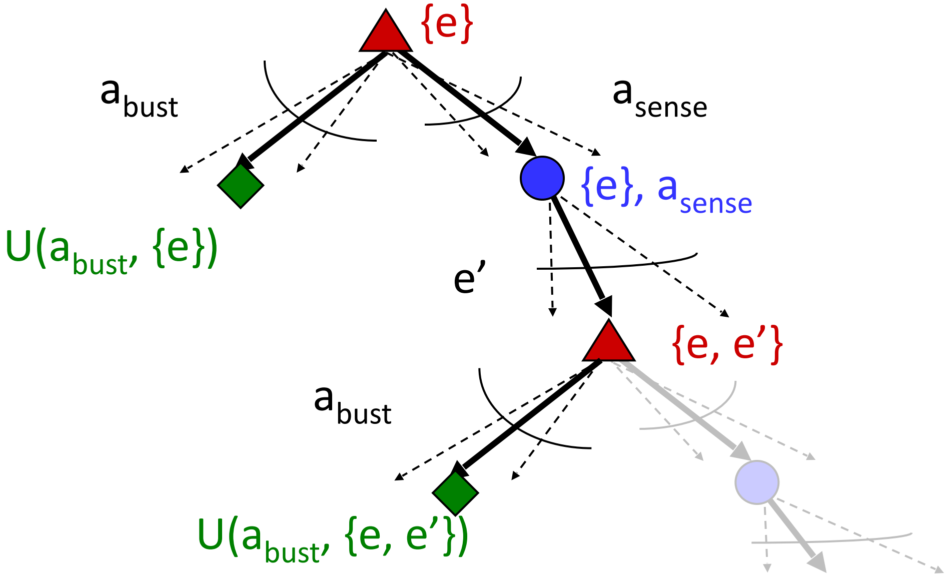


Example: Ghostbusters

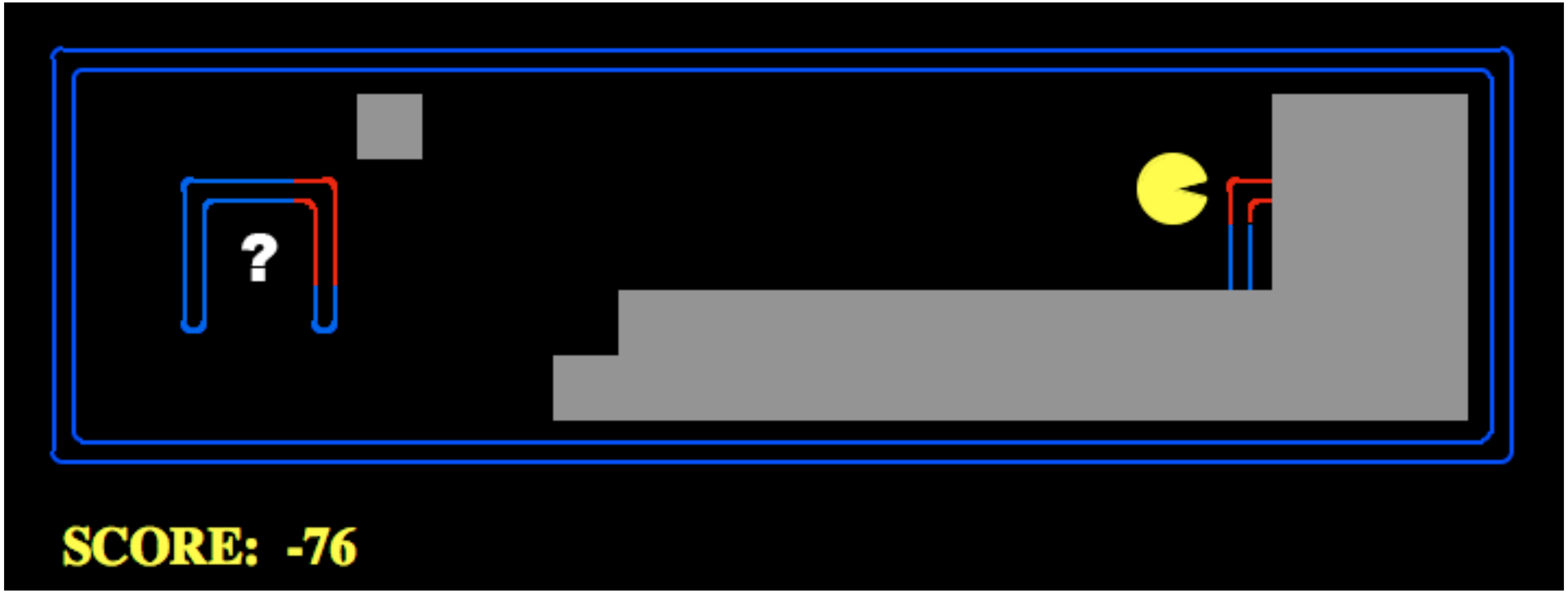
- In (static) Ghostbusters:
  - Belief state determined by evidence to date {e}
  - Tree really over evidence sets
  - Probabilistic reasoning needed to predict new evidence given past evidence



- Solving POMDPs
  - One way: use truncated expectimax to compute approximate value of actions
  - What if you only considered busting or one sense followed by a bust?
  - You get a VPI-based agent!



Project 4: Bayes' Nets and VPI



More Generally*

- General solutions map belief functions to actions
  - Can divide regions of belief space (set of belief functions) into policy regions (gets complex quickly)
  - Can build approximate policies using discretization methods
  - Can factor belief functions in various ways
- Overall, POMDPs are very (actually PSPACE-) hard
- Most real problems are POMDPs, but we can rarely solve then in general!