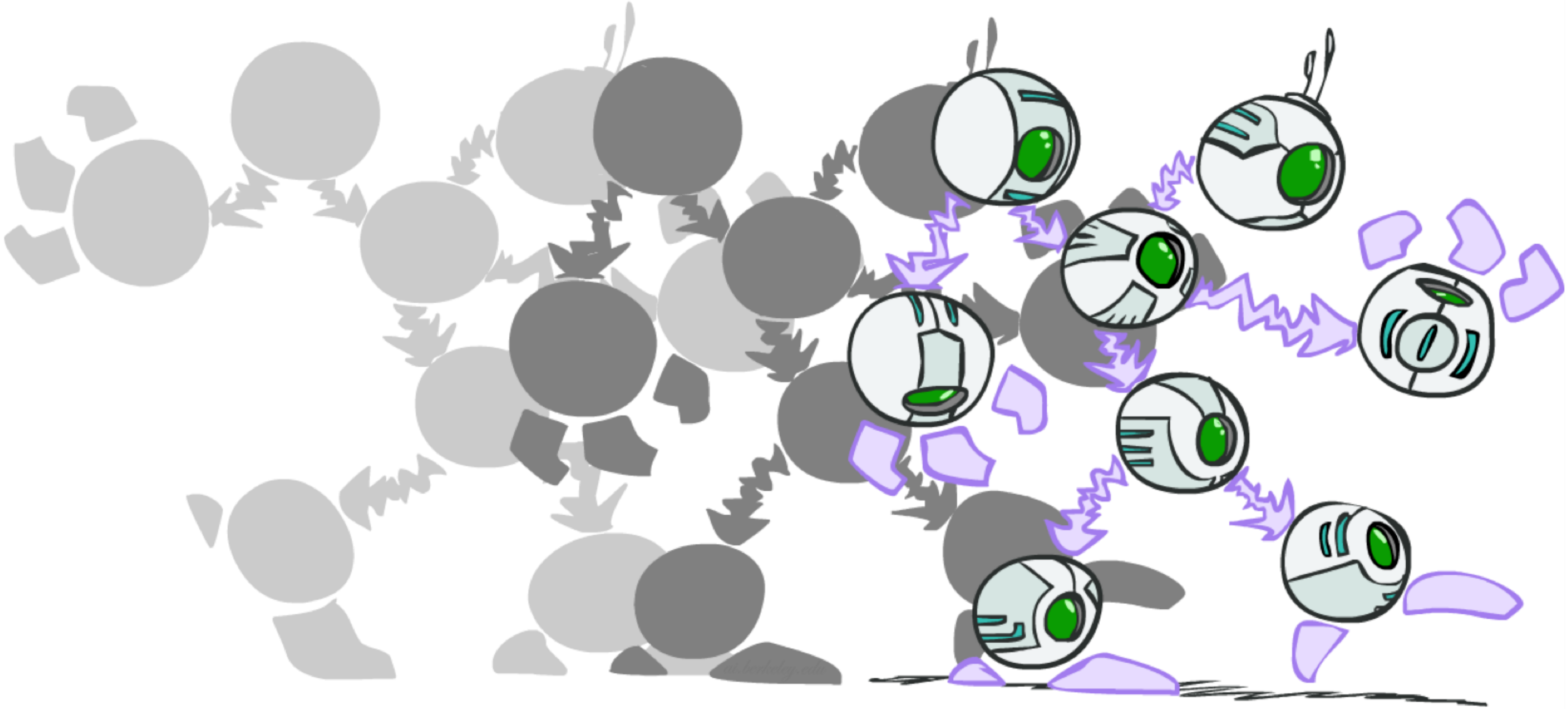
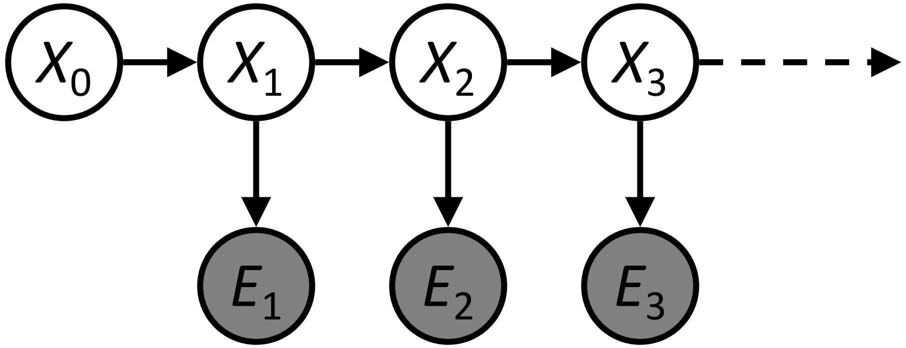
Dynamic Bayes Nets



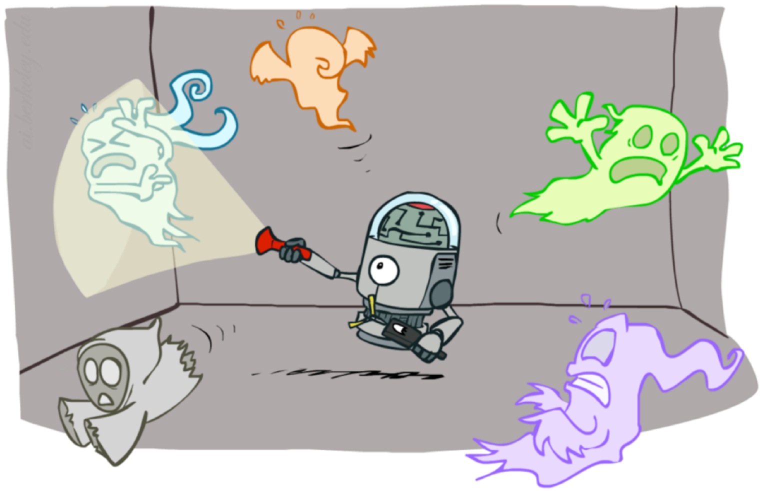
Hidden Markov Models

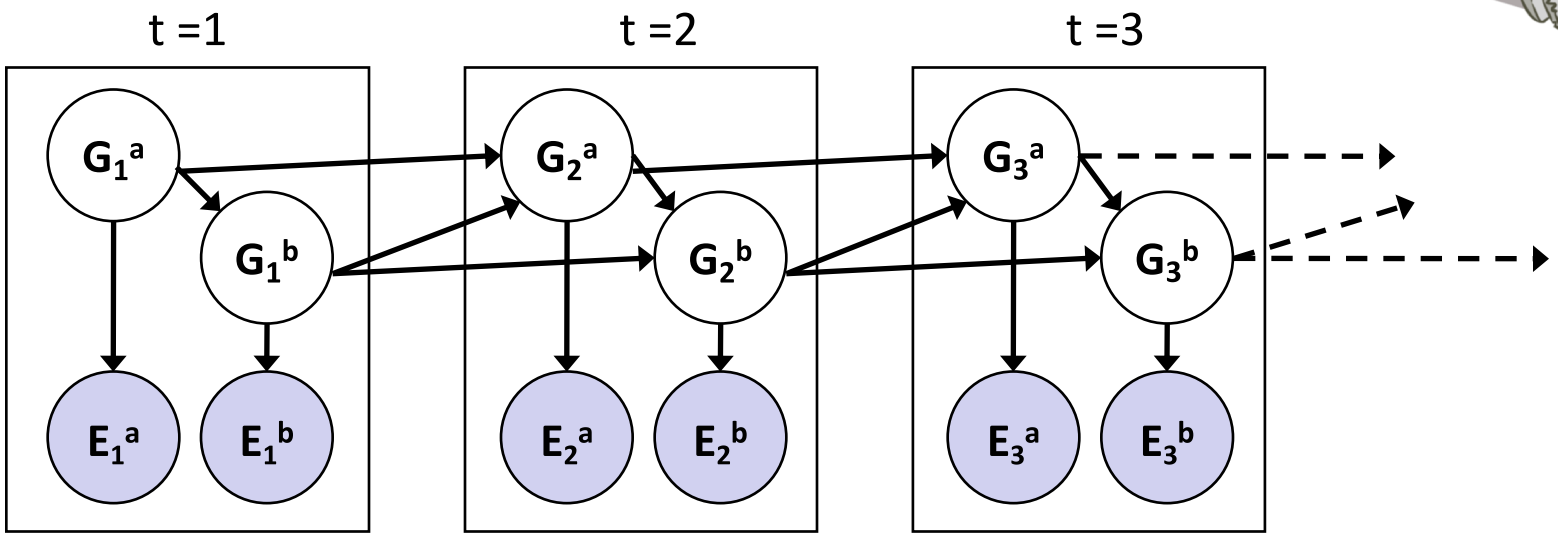


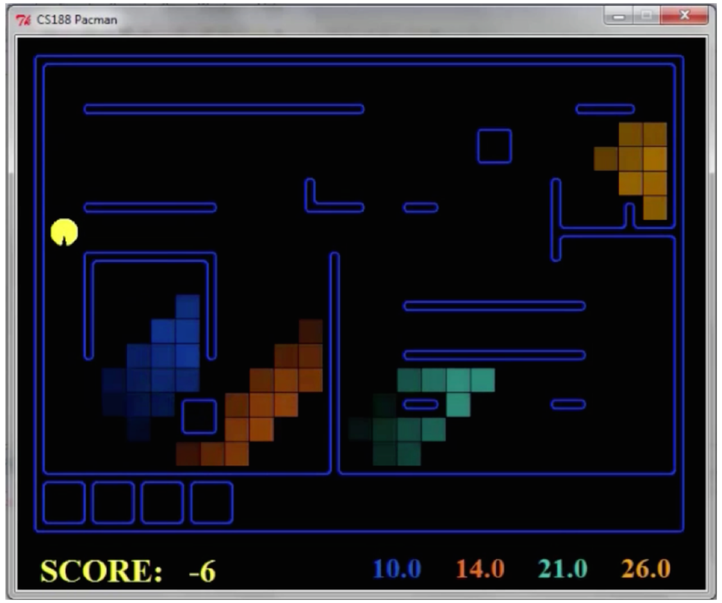
- Sensor models are the same at all times
- Current evidence is independent of everything else given the current state
- Filtering: calculate the distribution $\boldsymbol{f}_{1:t} = P(X_t | e_{1:t})$.
- Forward Algorithm: Predict (Time Elapse), Update, Normalize.
- Forward Algorithm: $P(X_{t+1} | e_{1:t+1}) = \alpha\, P(e_{t+1} | X_{t+1}) \sum_{x_t} P(x_t \mid e_{1:t})\, P(X_{t+1} | x_t)$
- Equivalently: $\boldsymbol{f}_{1:t+1} = \alpha\, O_{t+1} T^{\top} \boldsymbol{f}_{1:t}$
- Most likely explanation: $\arg\max_{x_{1:t}} P(x_{1:t} \mid e_{1:t})$

Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables at time *t* can have parents at time *t-1*

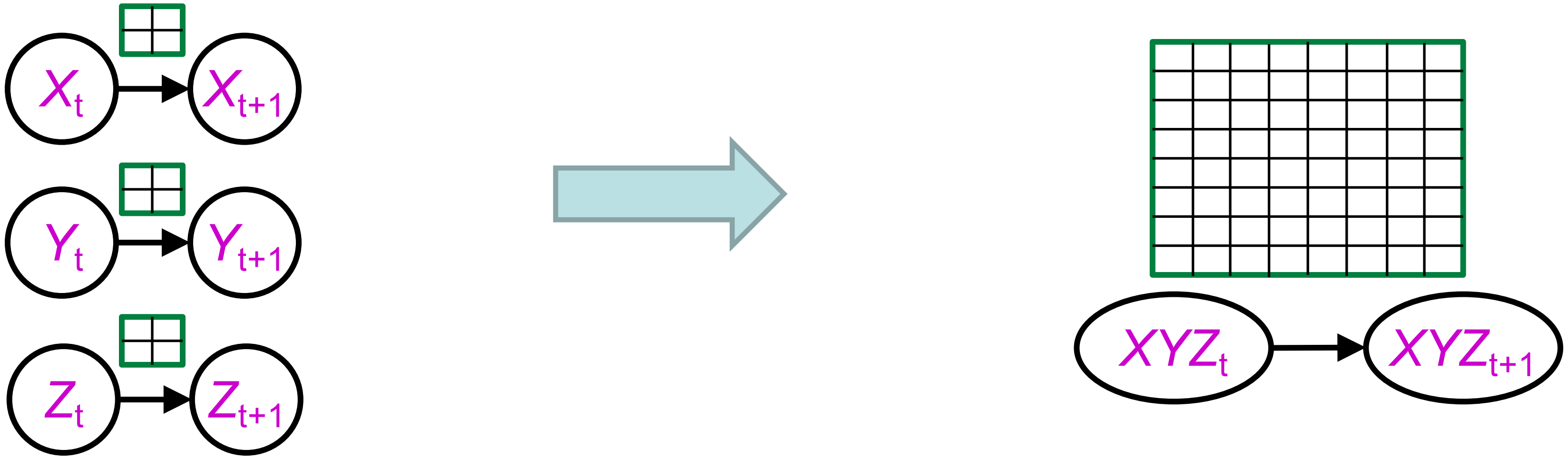






DBNs and HMMs

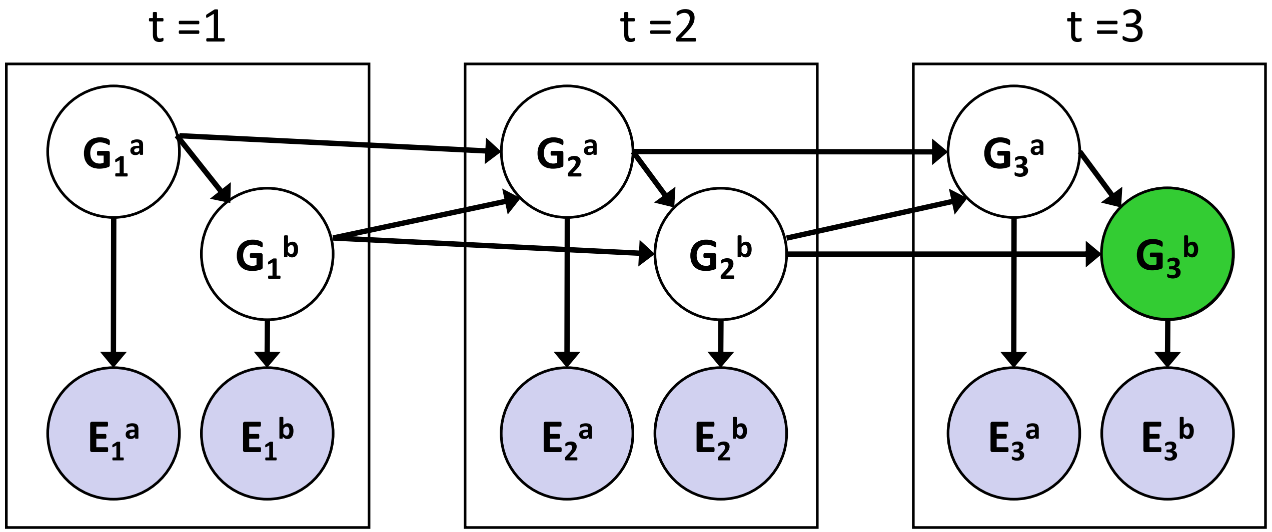
- Every HMM is a single-variable DBN
- Every discrete DBN is an HMM
  - HMM state is Cartesian product of DBN state variables



- Sparse dependencies => exponentially fewer parameters in DBN
  - E.g., 20 state variables, 3 parents each;
    DBN has 20 x 2^3 = 160 parameters, HMM has 2^{20} x 2^{20} =~ 10^{12} parameters

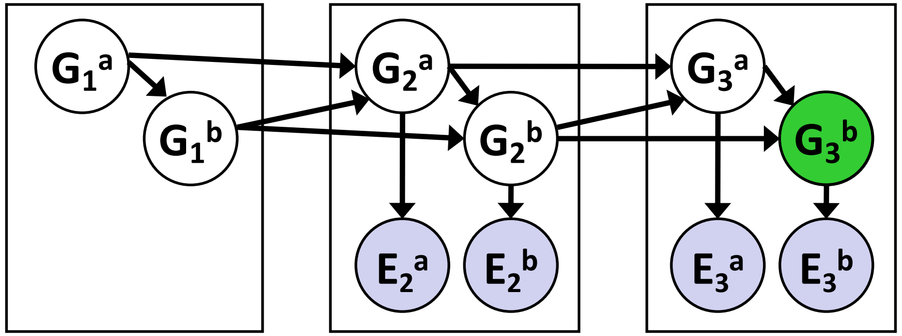
Exact Inference in DBNs

- Variable elimination applies to dynamic Bayes nets
- Offline: "unroll" the network for T time steps, then eliminate variables to find $P(X_T|e_{1:T})$



- Online: eliminate all variables from the previous time step; store factors for current time only
- Problem: largest factor contains all variables for current time (plus a few more)

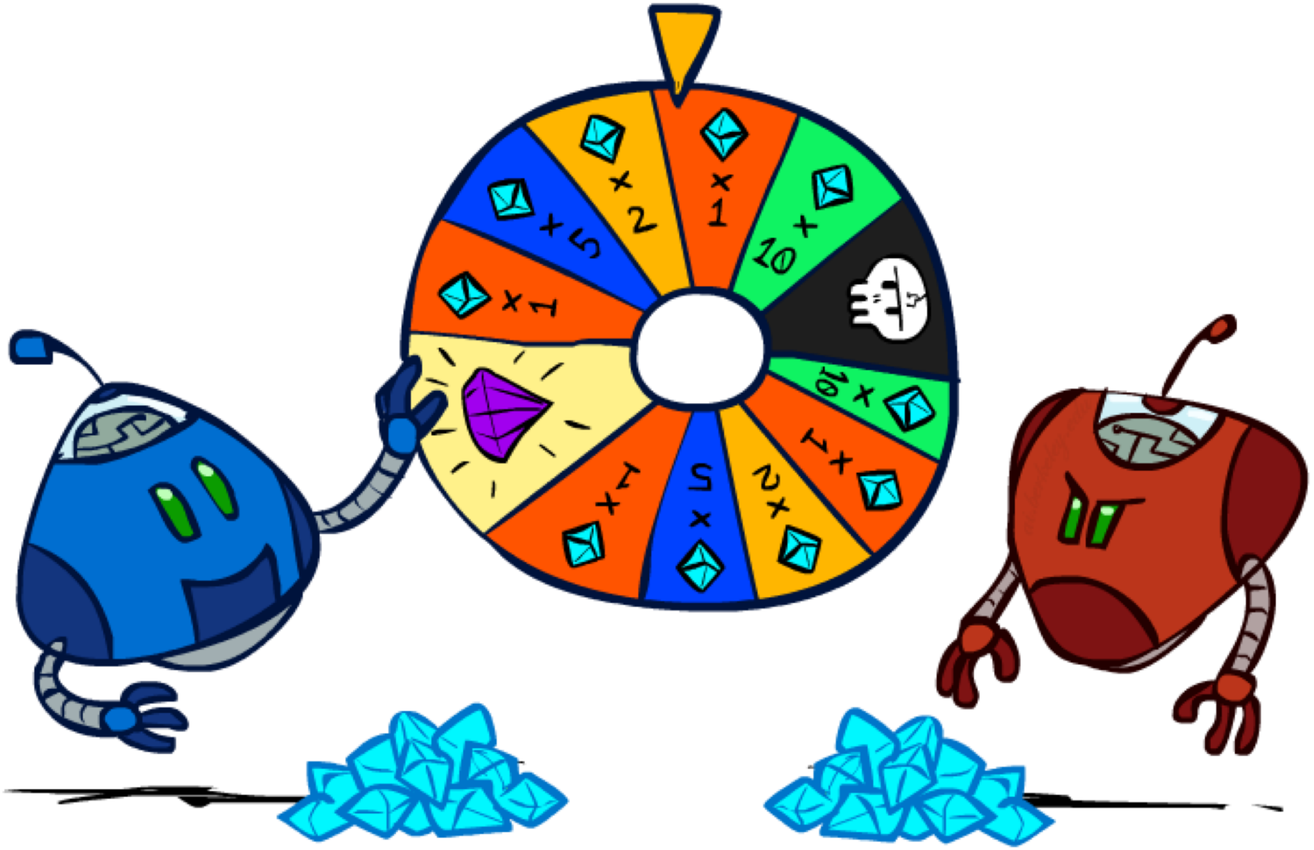
DBN Particle Filters



- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the t=1 Bayes net
  - Example particle: $\mathbf{G_1^a}$ = (3,3) $\mathbf{G_1^b}$ = (5,3)
- **Elapse time**: Sample a successor for each particle
  - Example successor: $\mathbf{G_2^a}$ = (2,3) $\mathbf{G_2^b}$ = (6,3)
- **Observe**: Weight each *entire* sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(\mathbf{E_2^a} \mid \mathbf{G_2^a}) * P(\mathbf{E_2^b} \mid \mathbf{G_2^b})$
- **Resample:** Select prior samples (tuples of values) in proportion to their likelihood

CS 188: Artificial Intelligence

Rational Decisions

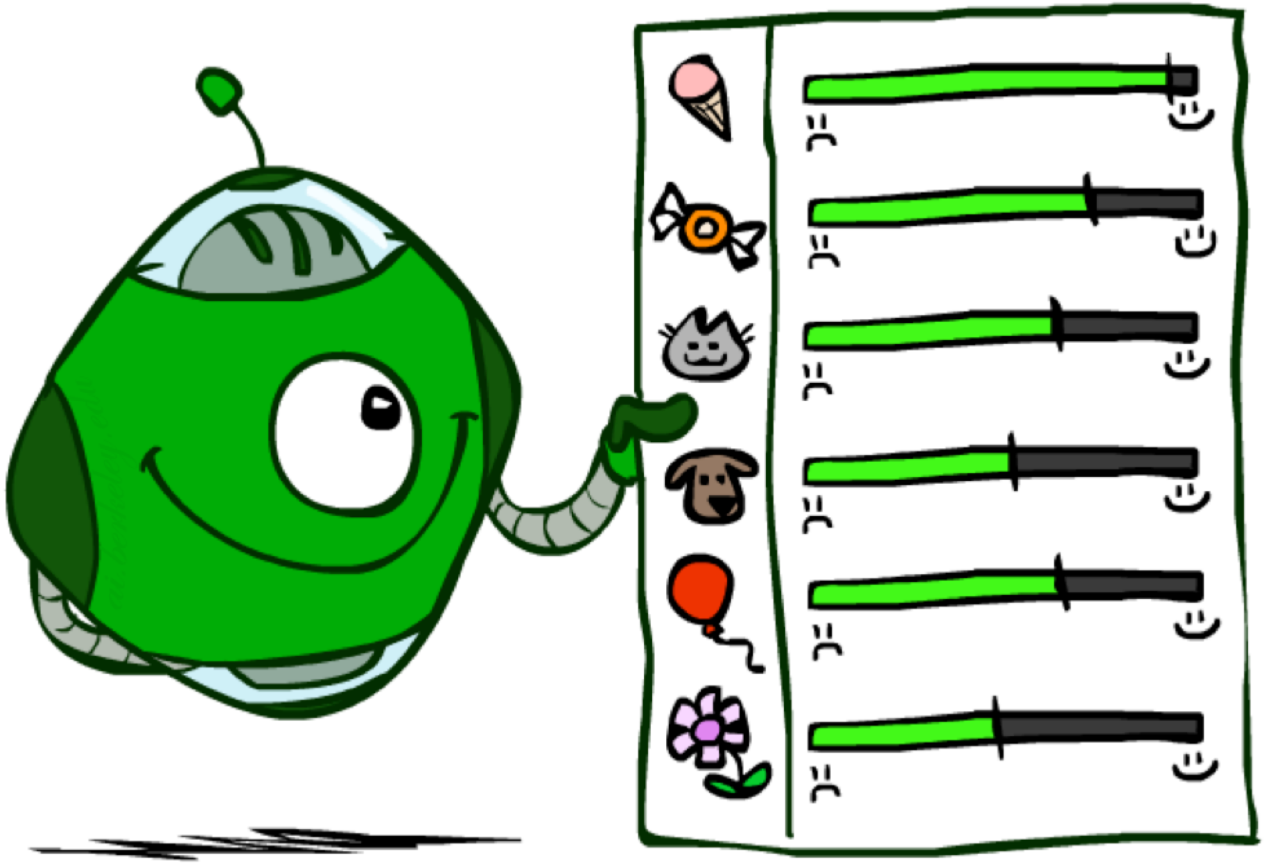


Instructors: Angela Liu and Yanlai Yang

University of California, Berkeley

(Slides adapted Stuart Russell and Dawn Song)

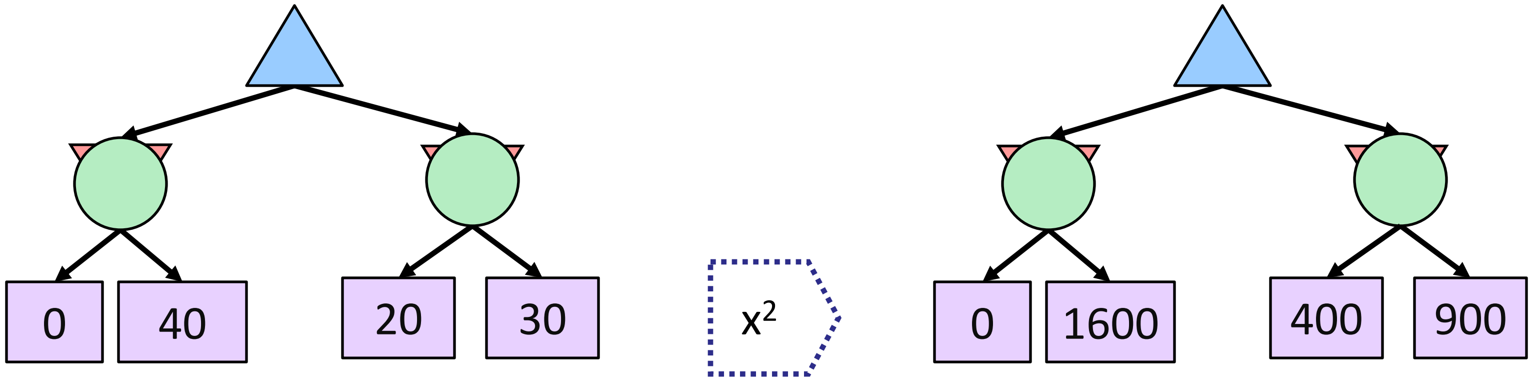
Utilities



Maximum Expected Utility

- Principle of maximum expected utility:
  - A rational agent should choose 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?

The need for numbers

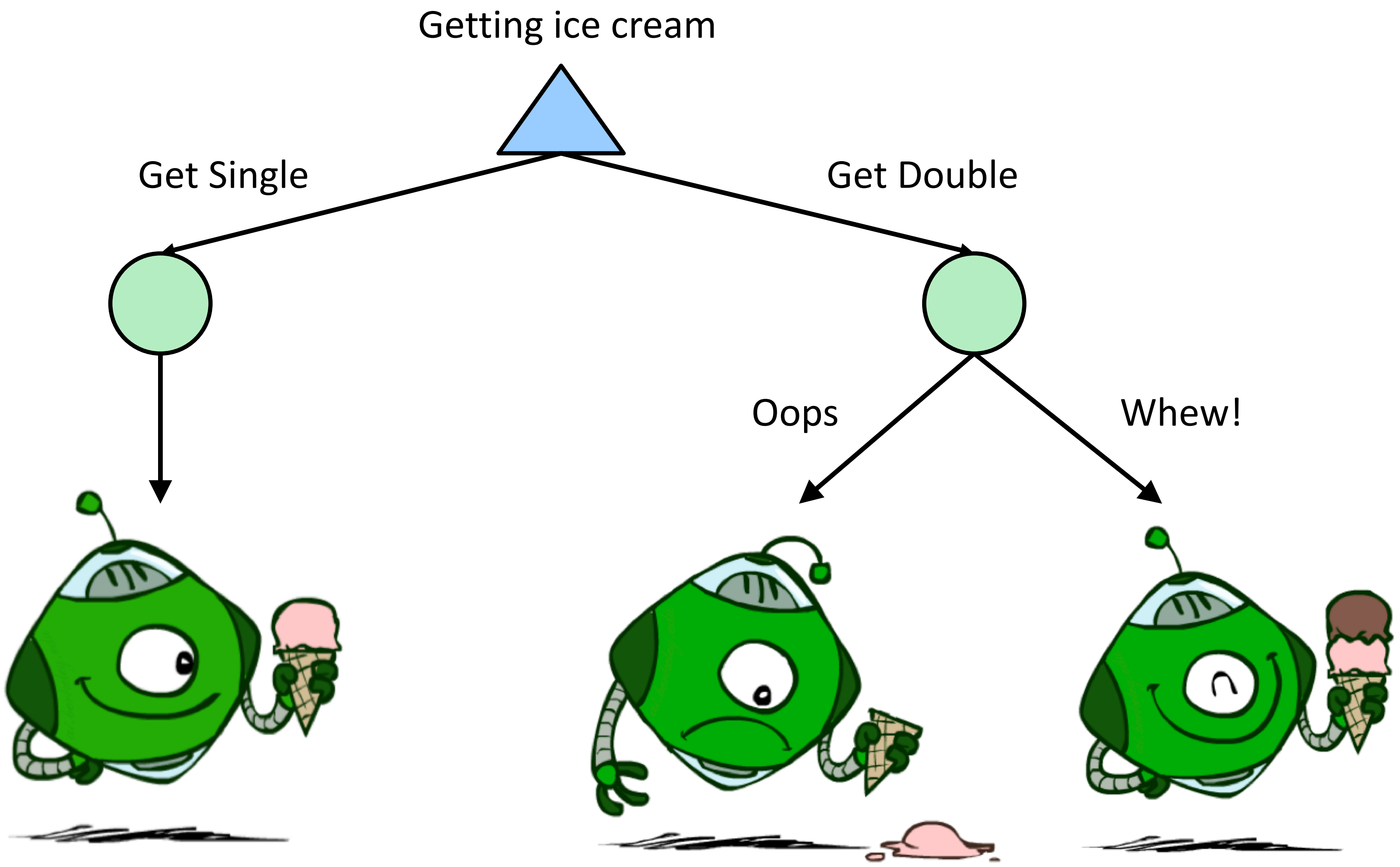


- For worst-case minimax reasoning, terminal value scale doesn't matter
  - We just want better states to have higher evaluations (get the ordering right)
  - The optimal decision is invariant under any ***monotonic transformation***
- For average-case expectimax reasoning, we need ***magnitudes*** to 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



Preferences

- An agent must have preferences among:
  - Prizes: P_1, P_2, etc.
  - Lotteries: situations with uncertain prizes

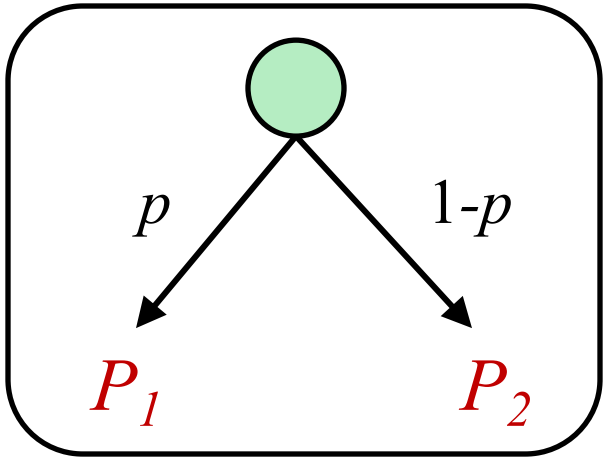
    $L = [p, P_1;\ (1-p), P_2]$

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

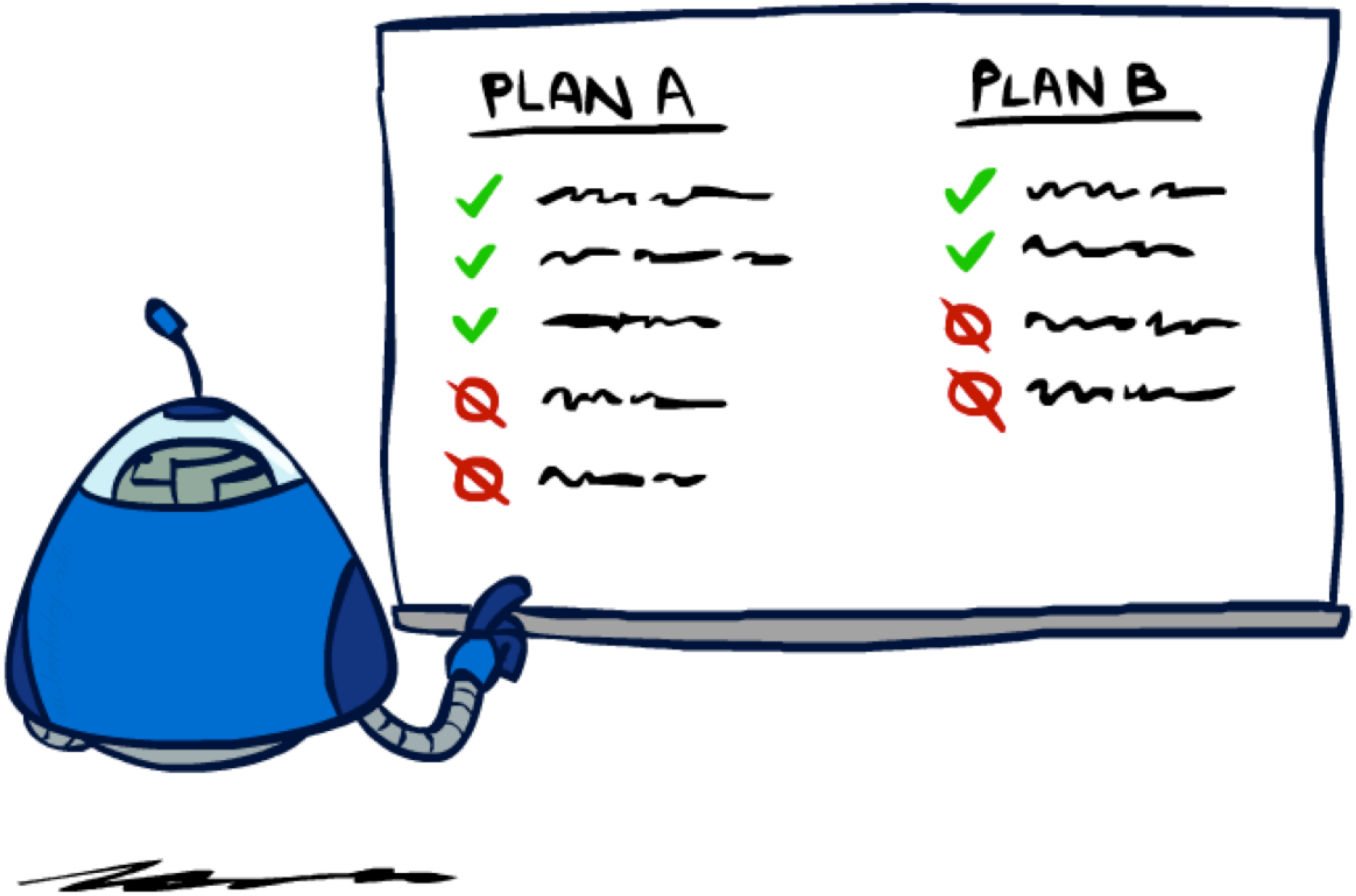
A Prize

$\boxed{P_1}$

A Lottery



Rationality

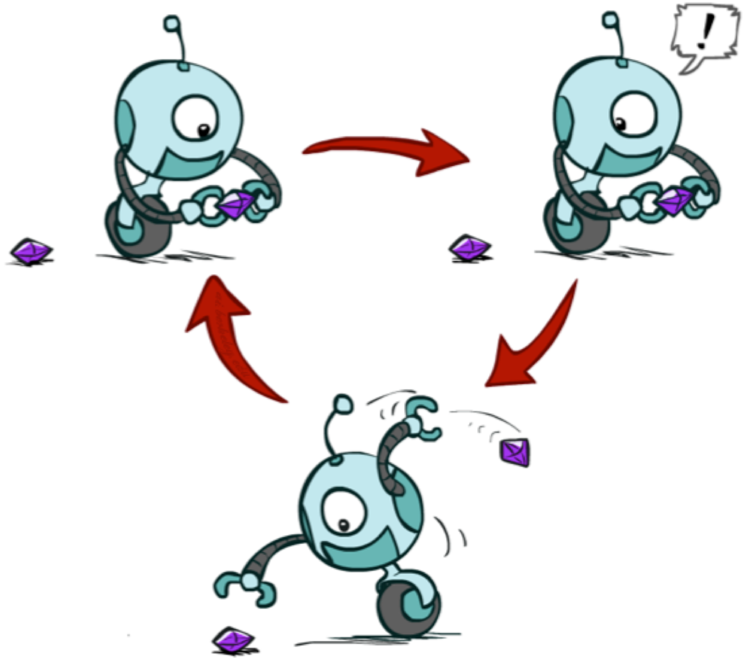


Rational Preferences

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

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:

$$(A \succ B \succ C) \Rightarrow \exists p\ [p, A;\ 1\text{-}p, C] \sim B$$

Substitutability:

$$(A \sim B) \Rightarrow [p, A;\ 1\text{-}p, C] \sim [p, B;\ 1\text{-}p, C]$$

Monotonicity:

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

Theorem: Rational preferences imply behavior describable as maximization of expected utility

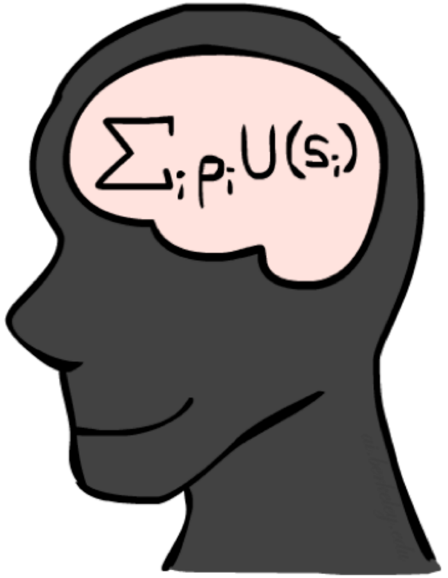
MEU Principle

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

$$U(A) > U(B) \Leftrightarrow A \succ B; \; U(A) = U(B) \Leftrightarrow A \sim B$$

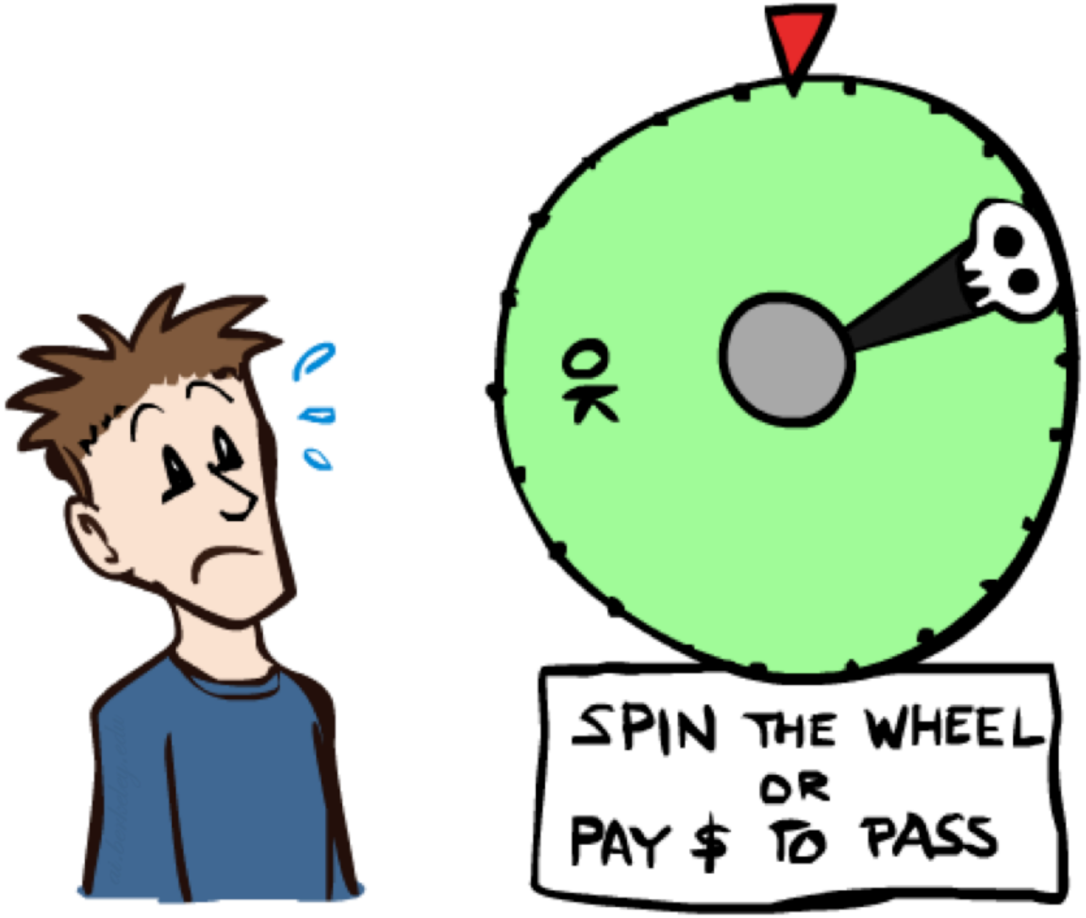
$$U([p_1,S_1; \ldots ; p_n,S_n]) = p_1U(S_1) + \ldots + p_nU(S_n)$$

  - I.e. values assigned by U preserve preferences of both prizes and lotteries!
  - Optimal policy invariant under ***positive affine transformation*** $U' = aU+b$, $a>0$



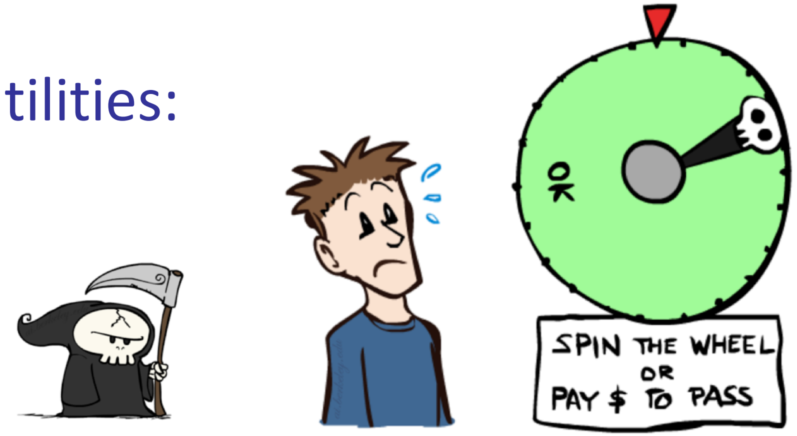
- Maximum expected utility (MEU) principle:
  - Choose the action that maximizes expected utility
  - Note: rationality does ***not*** require representing or manipulating utilities and probabilities
    - E.g., a lookup table for perfect tic-tac-toe

Human Utilities



Human Utilities

- Utilities map states to real numbers. Which numbers?
- Standard approach to assessment (elicitation) of human utilities:
  - Compare a prize A to a ***standard lottery*** L_p between
    - "best possible prize" $u_\top$ with probability p
    - "worst possible catastrophe" $u_\bot$ with probability $1-p$
  - Adjust lottery probability p until indifference: $A \sim L_p$
  - Resulting p is a utility in $[0,1]$



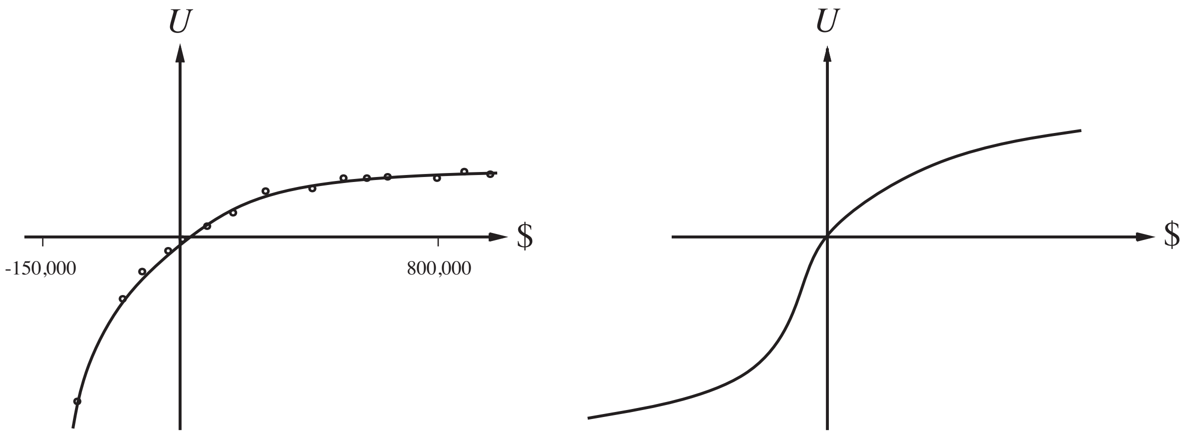
Pay $50 ~

0.999999 → *No change*

0.000001 → *Instant death*

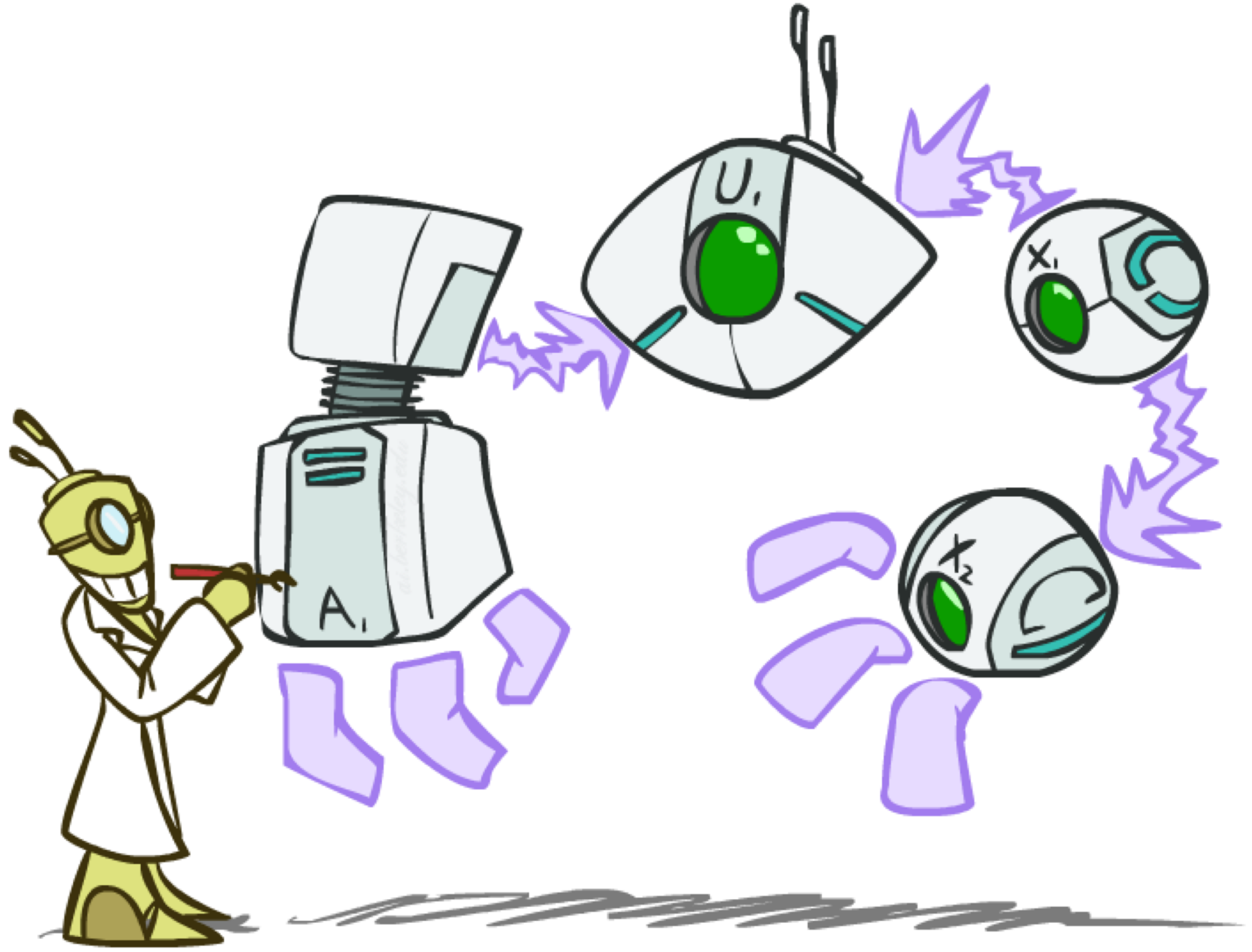
Money

- Money ***does not*** behave as a utility function, but we can talk about the utility of having money (or being in debt)
- Given a lottery $L = [p, \$X; (1\text{-}p), \$Y]$
  - The ***expected monetary value*** $\text{EMV}(L) = pX + (1\text{-}p)Y$
  - The utility is $U(L) = pU(\$X) + (1\text{-}p)U(\$Y)$
  - Typically, $U(L) < U(\text{ EMV}(L))$
  - In this sense, people are ***risk-averse***
  - E.g., how much would you pay for a lottery ticket L=[0.5, \$10,000; 0.5, \$0]?
  - The ***certainty equivalent*** of a lottery $\text{CE}(L)$ is the cash amount such that $\text{CE}(L) \sim L$
  - The ***insurance premium*** is $\text{EMV}(L) - \text{CE}(L)$
  - If people were risk-neutral, this would be zero!



(a) (b)

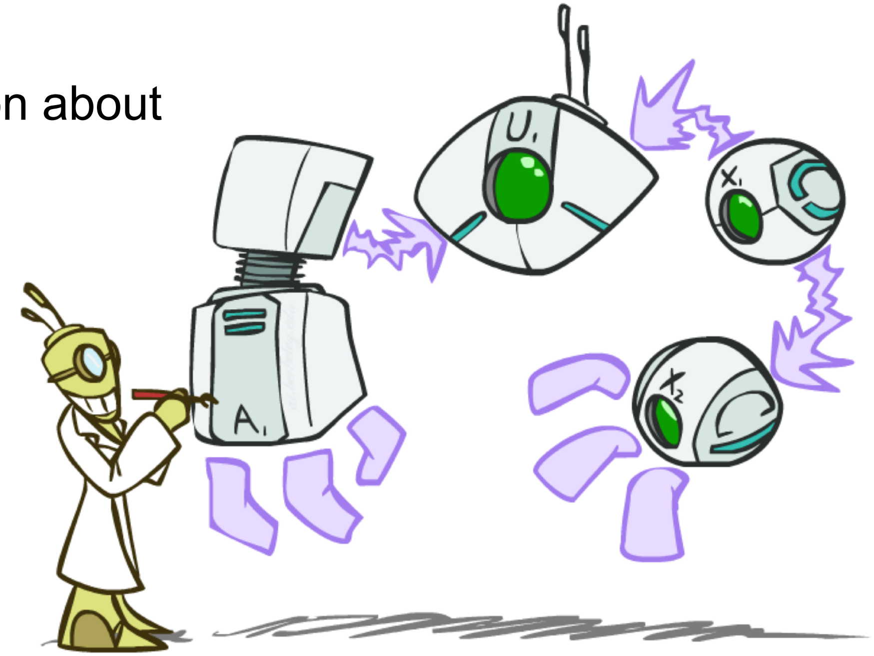
Decision Networks



Decision Networks

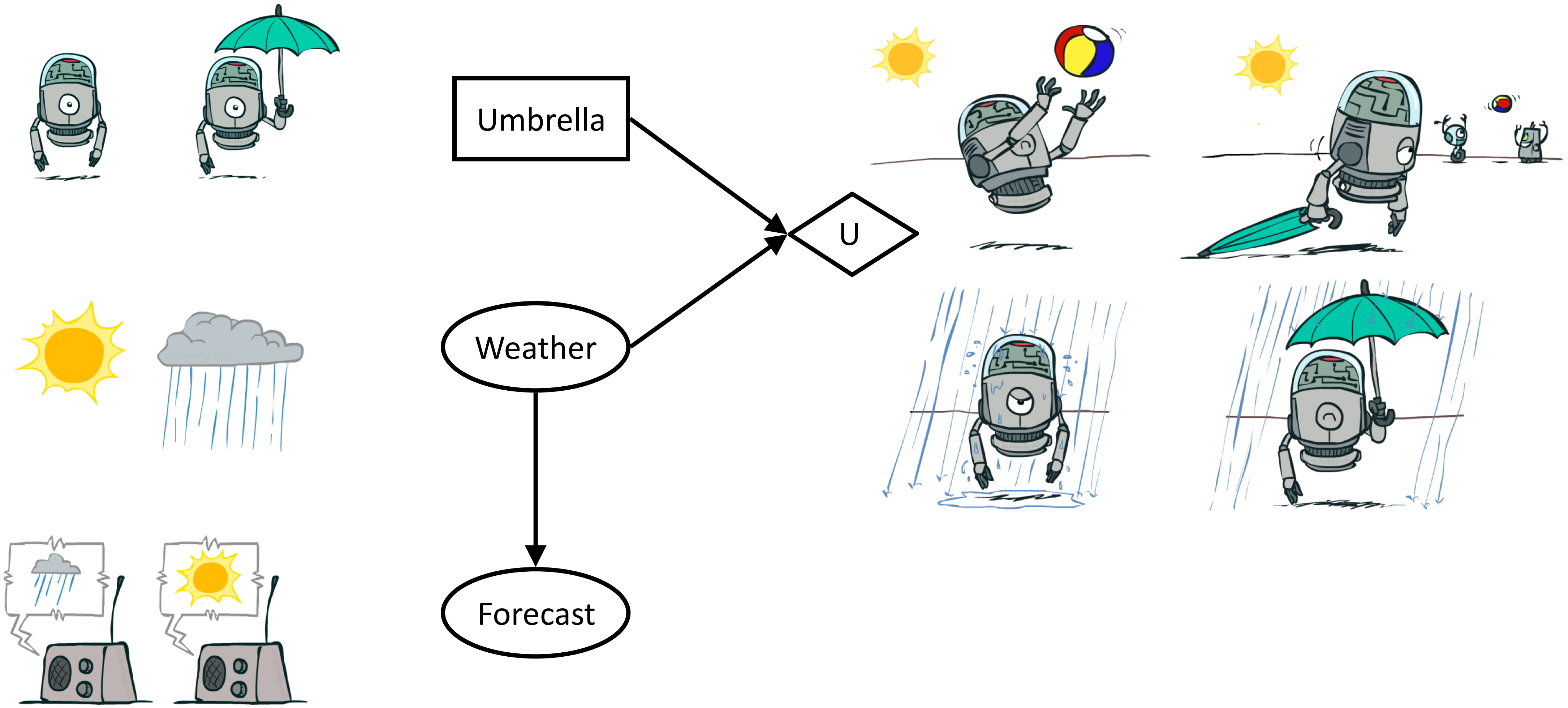
In its most general form, a decision network represents information about

- Its current state
- Its possible actions
- The state that will result from its actions
- The utility of that state



Decision network = Bayes net + Actions + Utilities

Decision Networks

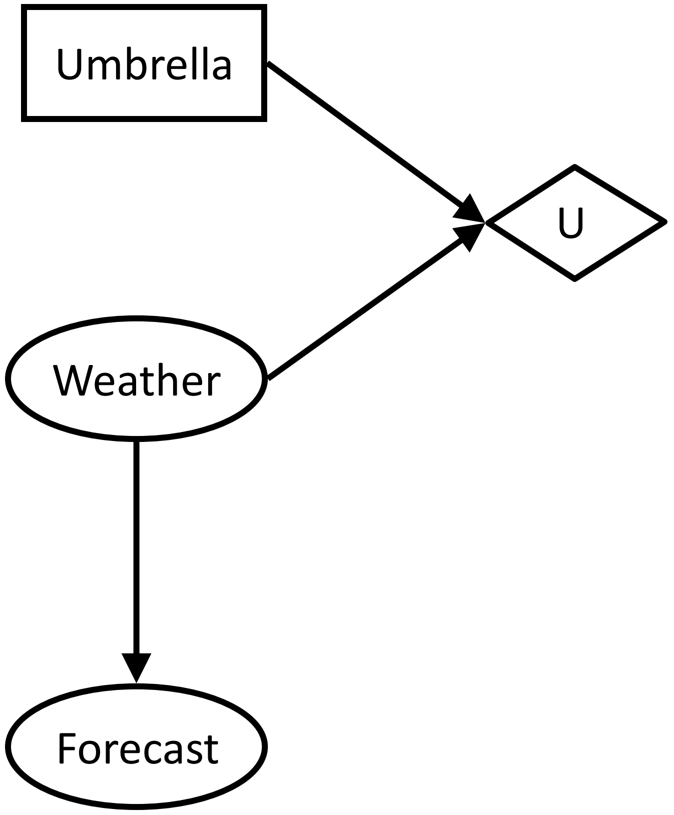


Decision Networks

- Decision network = Bayes net + Actions + Utilities
  - ***Chance nodes*** (just like BNs)
  - ***Action nodes*** (rectangles, cannot have parents, will have value fixed by algorithm)
  - ***Utility nodes*** (diamond, depends on action and chance nodes)

- Decision algorithm:
  - Fix evidence $\boldsymbol{e}$
  - For each possible action a
    - Fix action node to a
    - Compute posterior $P(\boldsymbol{W}|\boldsymbol{e},a)$ for parents $\boldsymbol{W}$ of U
    - Compute expected utility $\sum_{w} P(\mathbf{w}|\mathbf{e},a)\ U(a,\mathbf{w})$
  - Return action with highest expected utility

Bayes net inference!



Maximum Expected Utility

Umbrella = leave

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

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

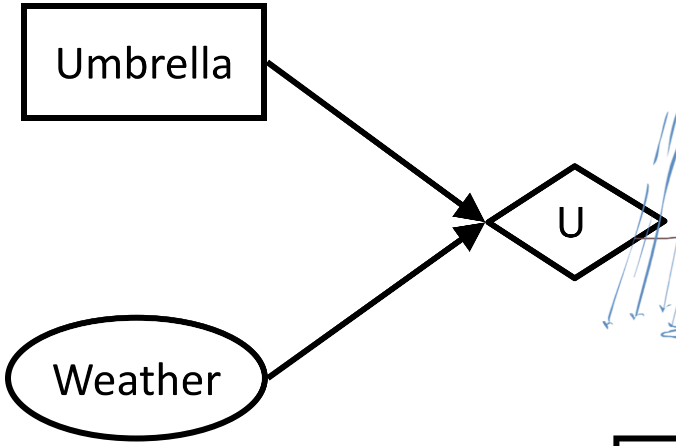
Umbrella = take

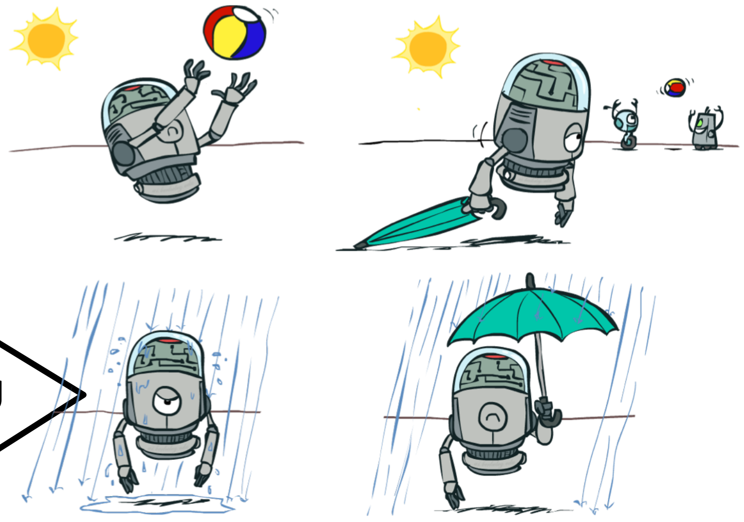
$$\mathrm{EU}(\mathrm{take}) = \sum_w P(w)U(\mathrm{take}, w)$$

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

Optimal decision = leave

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

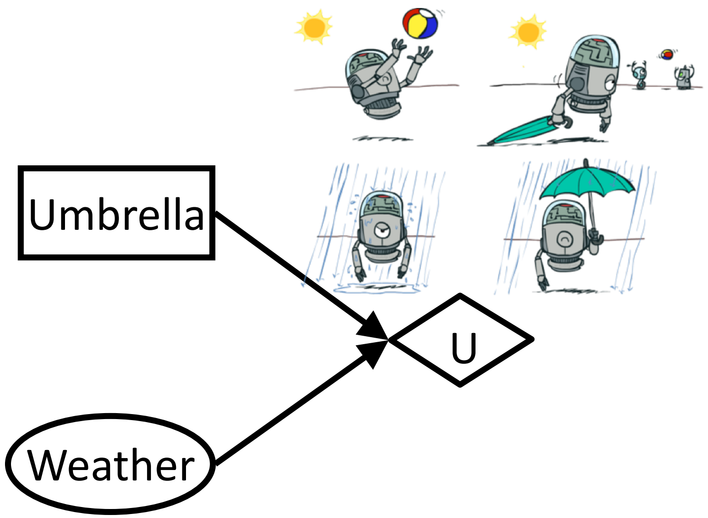


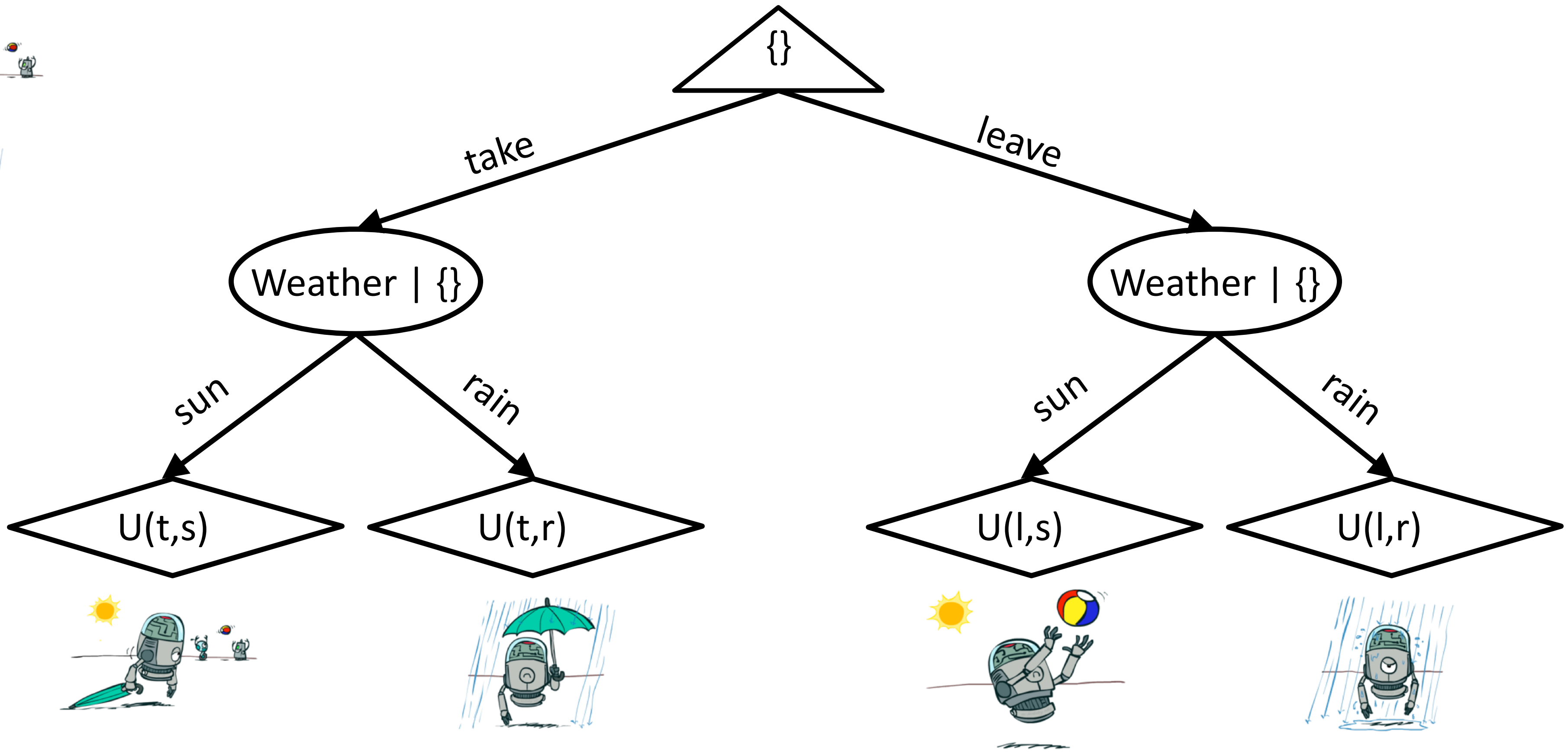


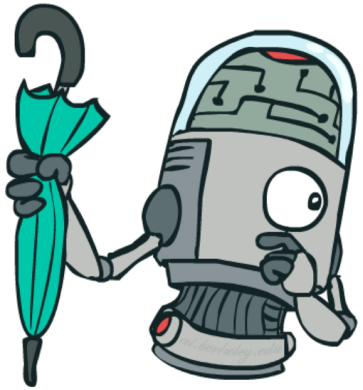
W	P(W)
sun	0.7
rain	0.3

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

Decisions as Outcome Trees



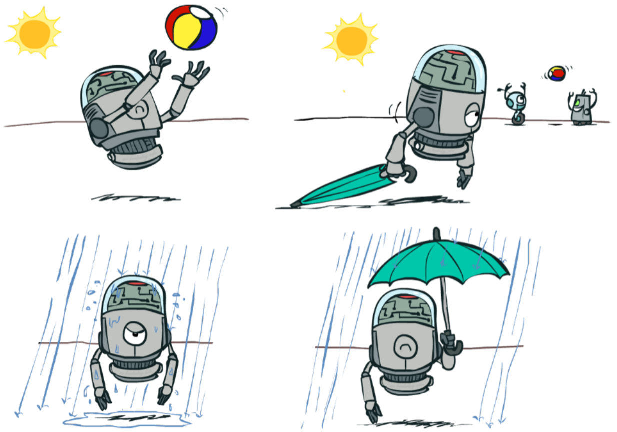


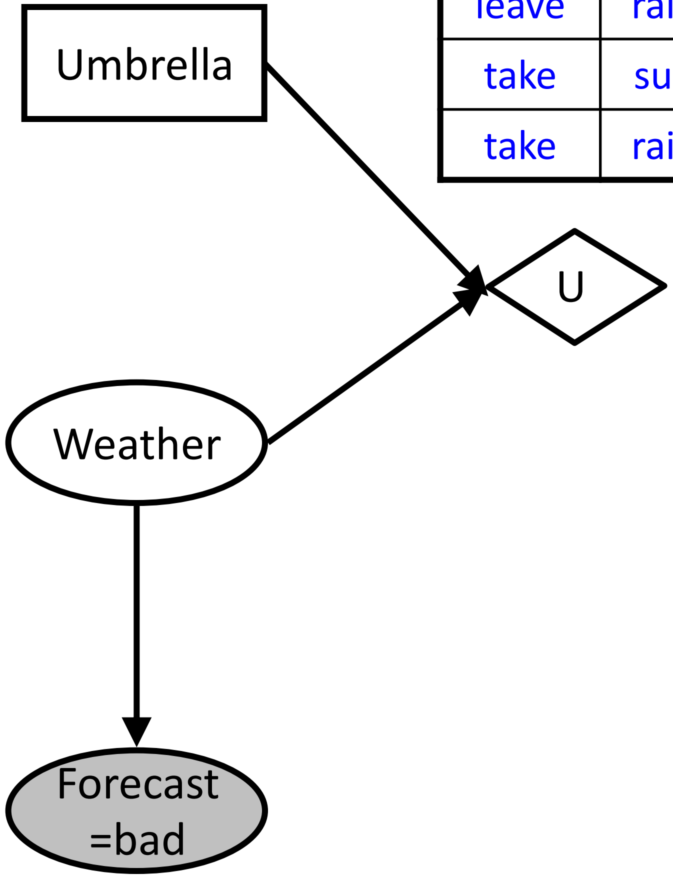


- Almost exactly like expectimax!
- What's changed?

Example: Take an umbrella?

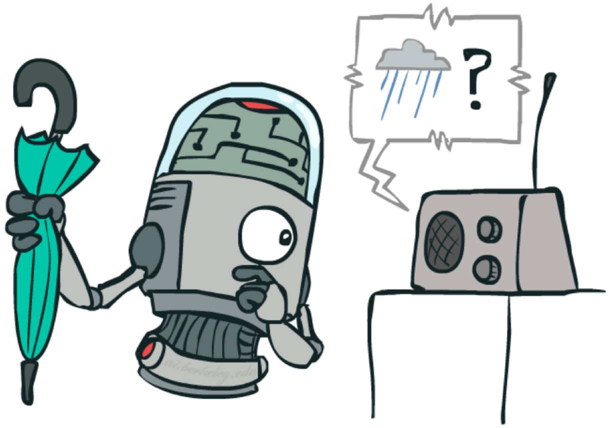
A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70





W	P(W)
sun	0.7

W	P(F=bad|W)
sun	0.17
rain	0.77

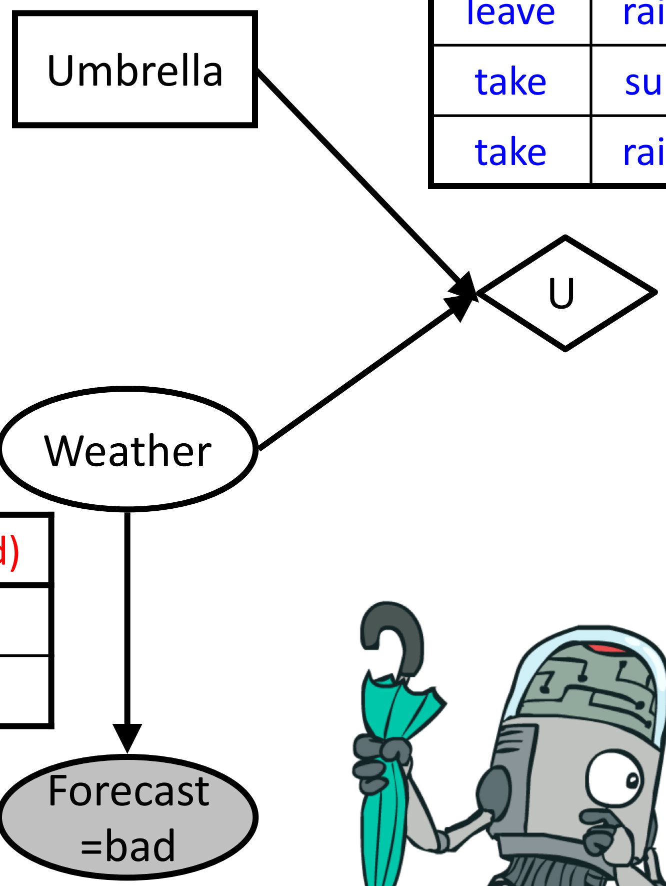


Example: Take an umbrella?

- Decision algorithm:
  - Fix evidence ***e***
  - For each possible action *a*
    - Fix action node to *a*
    - Compute posterior $P(\mathbf{W}|\mathbf{e})$ for parents $\mathbf{W}$ of U
    - Compute expected utility of action a: $\sum_w P(\mathbf{w}|\mathbf{e})\, U(a,\mathbf{w})$
  - Return the action with highest expected utility

Bayes net inference!

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



W	P(W)
sun	0.7

Umbrella = leave

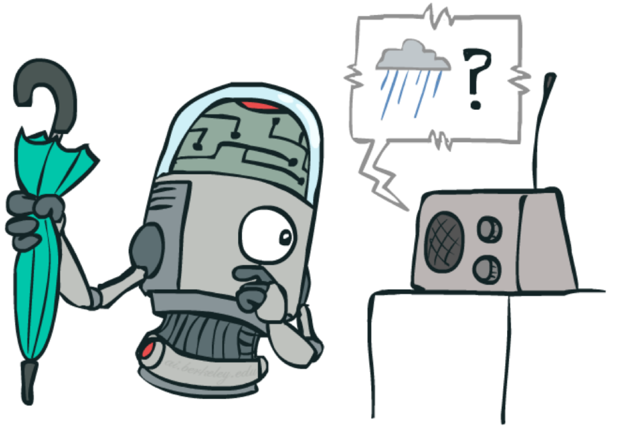
$\text{EU}(\text{leave}|F=\text{bad}) = \sum_w P(w|F=\text{bad})\, U(\text{leave},w)$

We have: $P(W) \quad P(F|W)$

$$P(W|F) = \frac{P(W,F)}{\sum_w P(w,F)} = \frac{P(F|W)P(W)}{\sum_w P(F|w)P(w)}$$

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

W	P(F=bad|W)
sun	0.17
rain	0.77

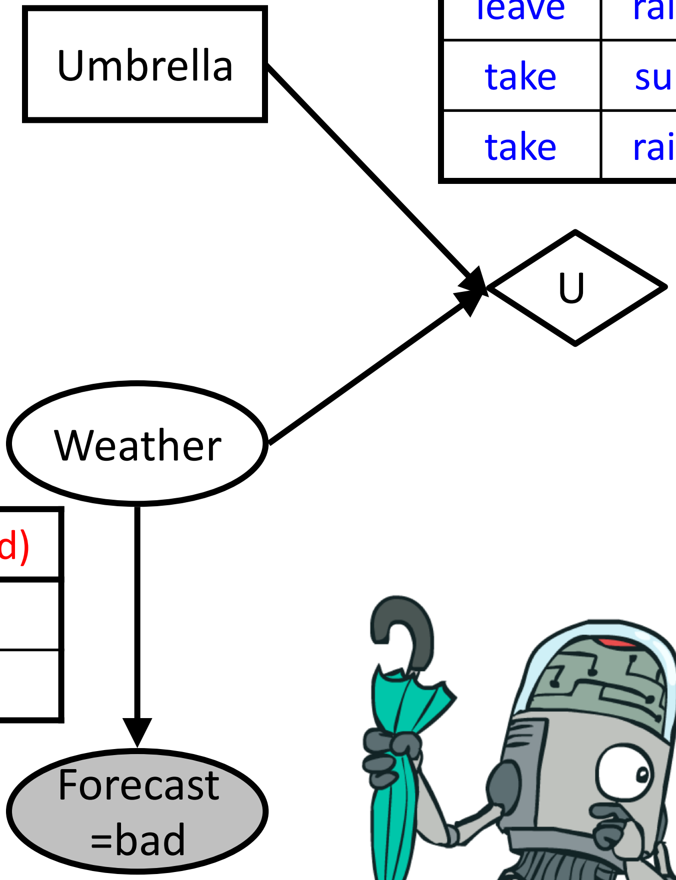


Example: Take an umbrella?

- Decision algorithm:
  - Fix evidence ***e***
  - For each possible action *a*
    - Fix action node to *a*
    - Compute posterior $P(\boldsymbol{W}|\boldsymbol{e},a)$ for parents $\boldsymbol{W}$ of U
    - Compute expected utility of action a: $\sum_{\boldsymbol{w}} P(\boldsymbol{w}|\boldsymbol{e},a)\ U(a,\boldsymbol{w})$
  - Return action with highest expected utility

Bayes net inference!

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



W	P(W)
sun	0.7

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

W	P(F=bad|W)
sun	0.17
rain	0.77

Umbrella = leave

$$EU(\text{leave}|F=\text{bad}) = \sum_w P(w|F=\text{bad})\ U(\text{leave},w)$$

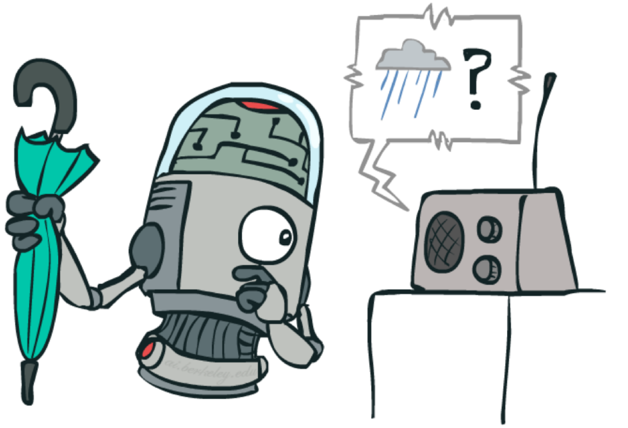
$$= 0.34 \times 100 + 0.66 \times 0 = 34$$

Umbrella = take

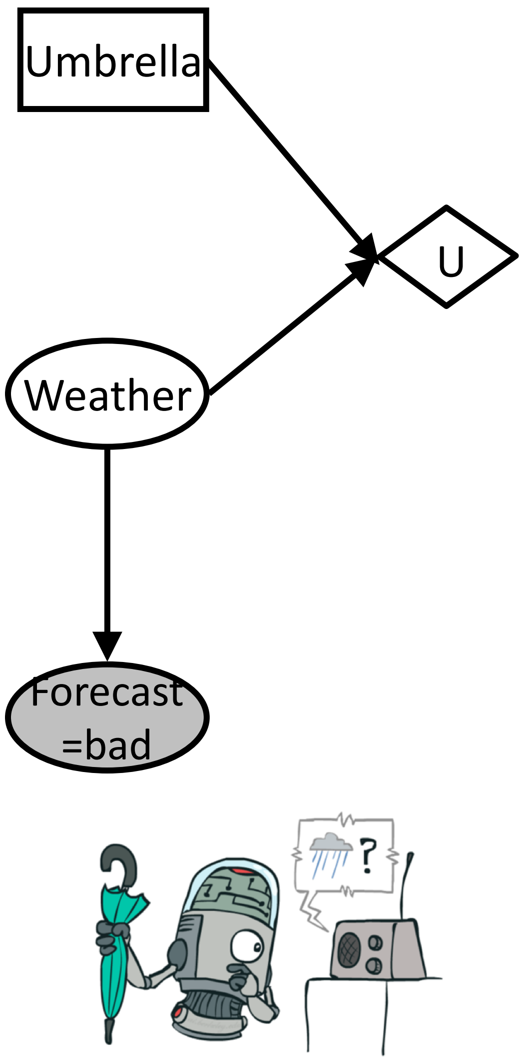
$$EU(\text{take}|F=\text{bad}) = \sum_w P(w|F=\text{bad})\ U(\text{take},w)$$

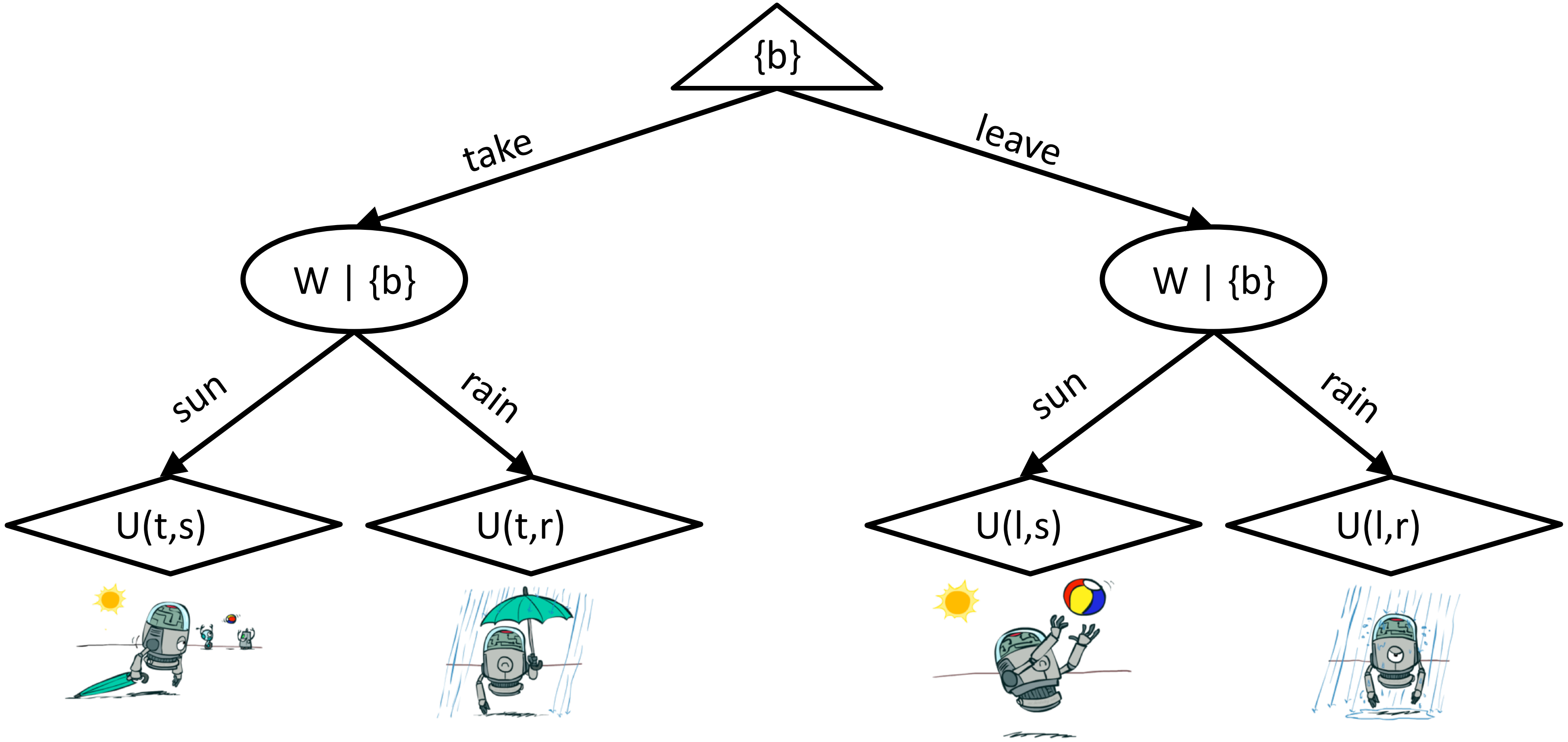
$$= 0.34 \times 20 + 0.66 \times 70 = 53$$

Optimal decision = take!

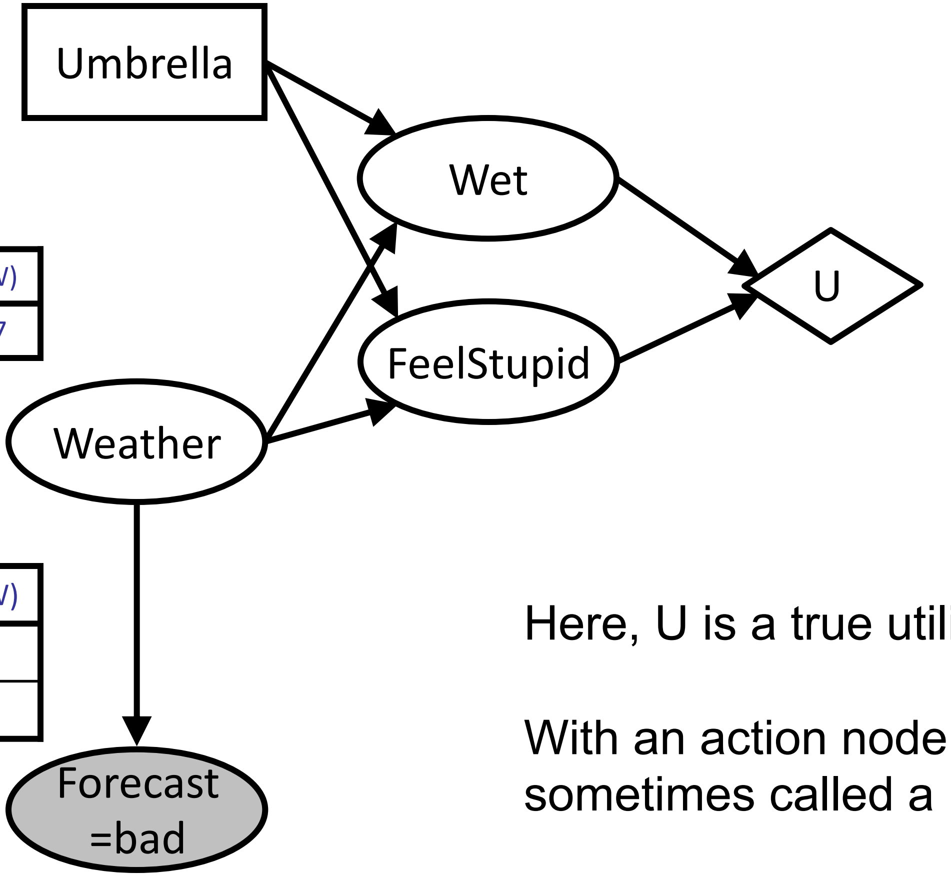


Decisions as Outcome Trees





Decision network with utilities on outcome states



W	P(W)
sun	0.7

W	P(F=bad|W)
sun	0.17
rain	0.77

Wet	FeelStupid	U
dry	false	100
wet	true	0
dry	true	20
damp	false	70

Here, U is a true utility.

With an action node as parent, it is sometimes called a ***Q-value***