CS 5/7320
Artificial Intelligence

Making Simple Decisions
AIMA Chapter 16

Introduction slides by Michael Hahsler

Decision network slides by Dan Klein and Pieter Abbeel

Decision theory =

Probability theory (evidence & belief)

+

Utility theory (want)

Decision-theoretic agents

Logical agents

Cannot deal with:

- Uncertainty
- Conflicting goals

Goal-based agents

- Can only assign goal/not goal to states and find goal states.

Decision-theoretic agents

- Assign a utility value to each state.
- A rational agent optimizes the expected utility.
- Utility is related to the external performance measure (see PEAS).

Utility

- A utility function $U(s)$ expresses the desirability of a state.
- Utility functions are derived from rational preferences:

$$U(A) > U(B) \Leftrightarrow A \succ B$$

and

$$U(A) = U(B) \Leftrightarrow A \sim B$$

- Therefore, often it is enough to know the utility ranking of states.

Expected Utility

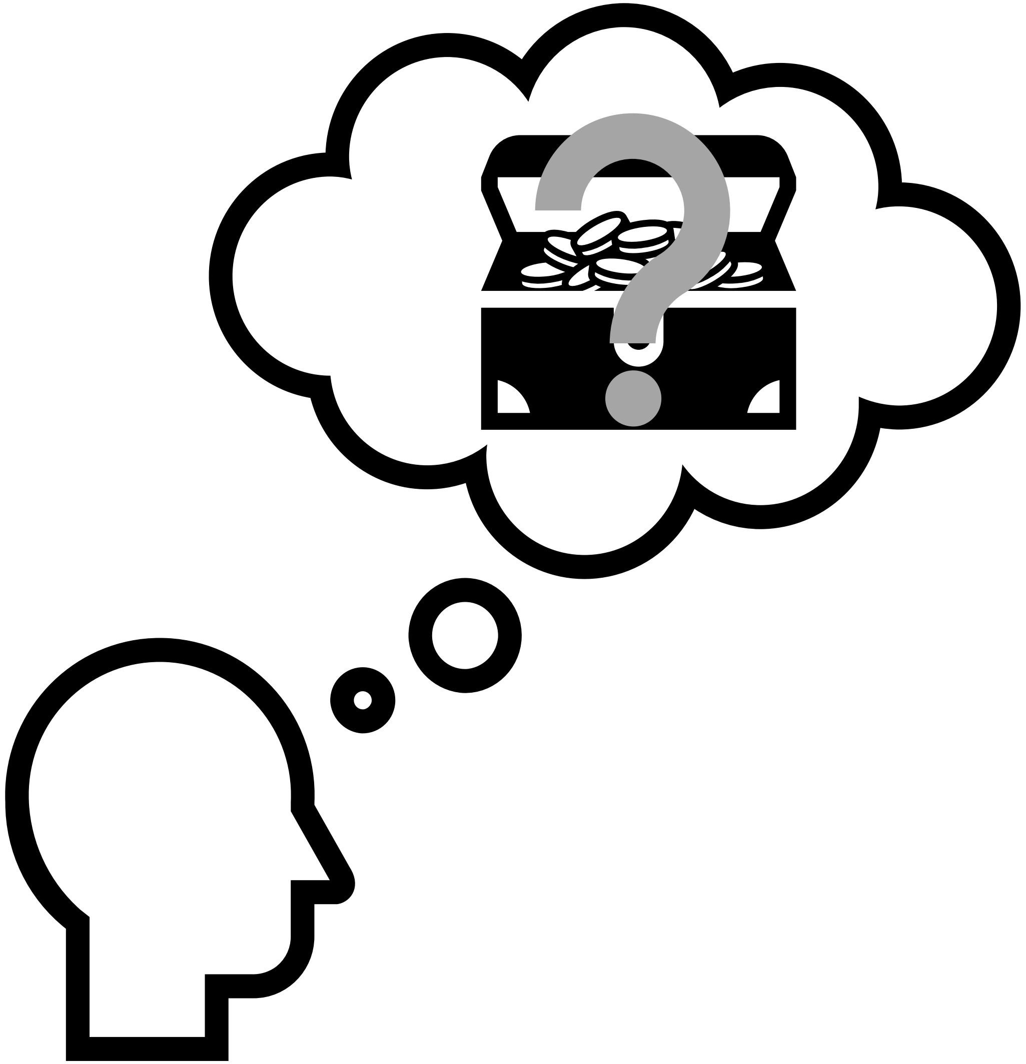
We need:

- A utility function $U(s)$.
- The probability to currently be in every state $P(s)$.
- The probability that an action will get us to different states s'

$$P(Result(a) = s') = \sum_{s} P(s)P(s'|s,a)$$

Expected utility of an action:

$$EU(a) = \sum_{s\prime} P(Result(s) = s')U(s')$$



Principle of Maximum Expected Utility (MEU)

Given the expected utility of an action

$$EU(a) = \sum_{s'} P(Result(s) = s')U(s')$$

choose action that maximizes the expected utility:

$$a^* = \text{argmax}_a \, EU(a)$$

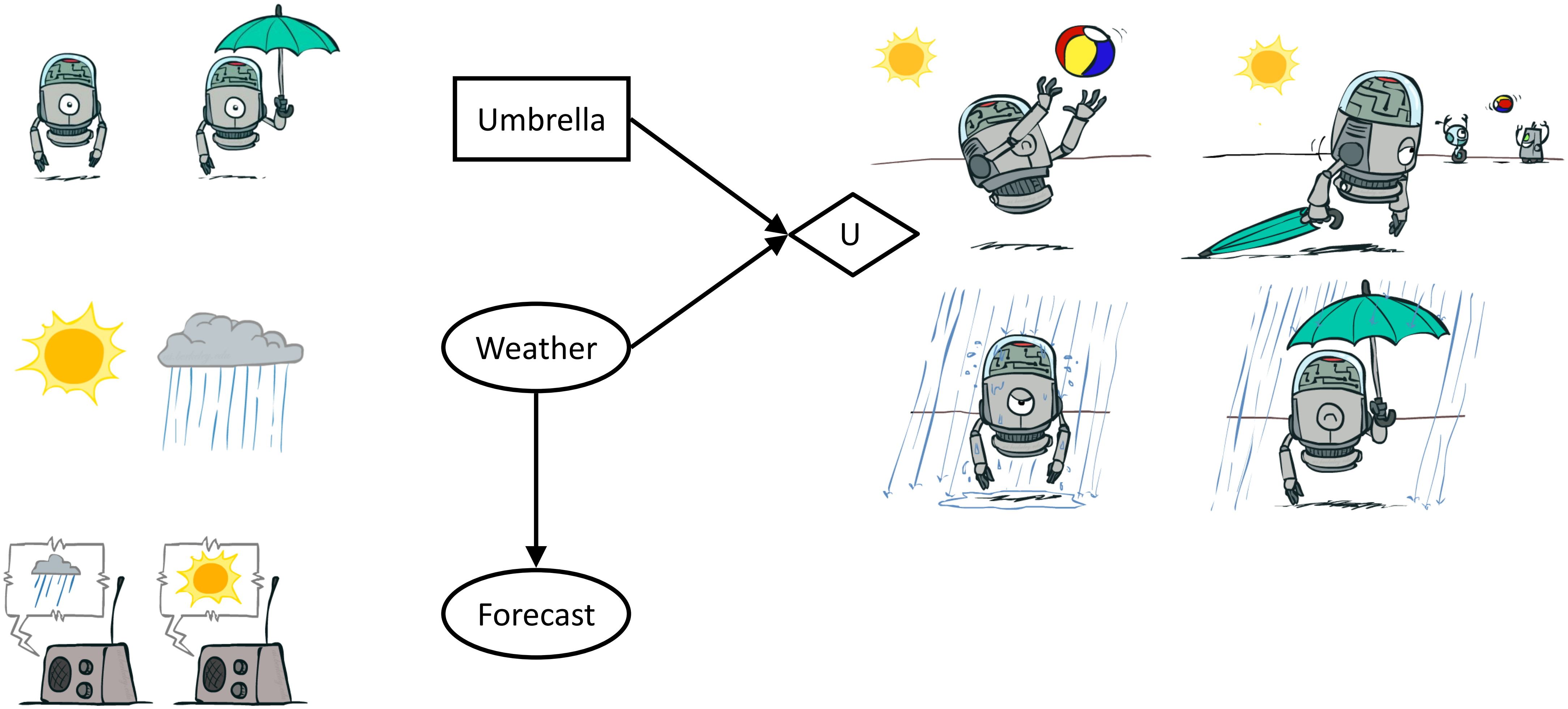
Issues:

- $P(Result(s) = s')$ needs a causal model
- $U(s)$ may be hard to estimate. It may depend on what states we can get to from s.
- MEU leads to the "optimizer's curse" where the estimated expected utility is higher than the actual outcomes with new data.

Decision Networks

These slides were created by Dan Klein, Pieter Abbeel, Sergey Levine, with some materials from A. Farhadi. All CS188 materials are at http://ai.berkeley.edu

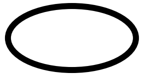
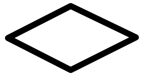
Decision Networks

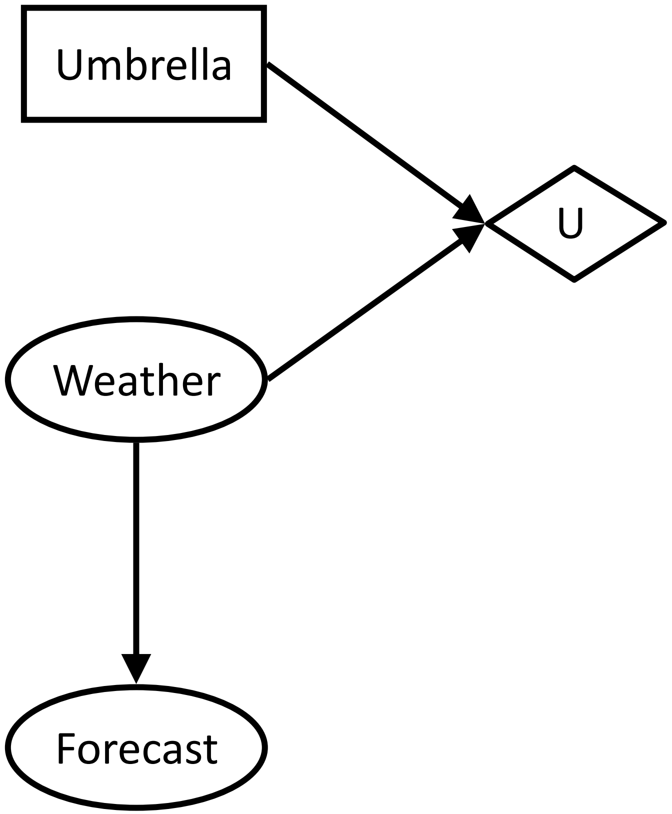


Decision Networks

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

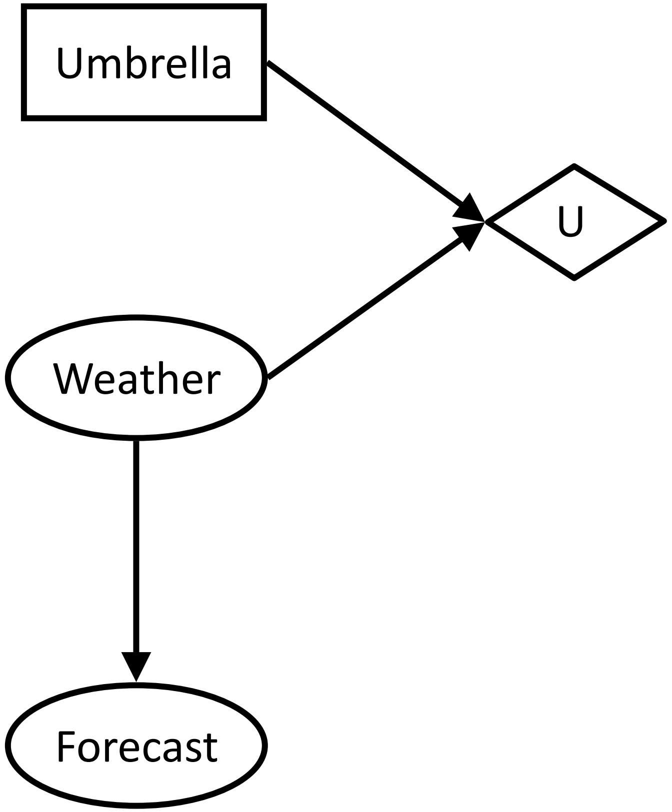
- Can directly operationalize this with decision networks
  - Bayes nets with nodes for utility and actions
  - Let's us calculate the expected utility for each action

- 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

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

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

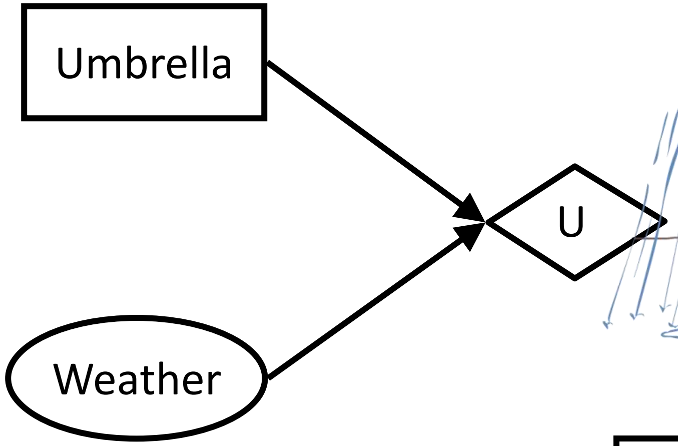
Umbrella = take

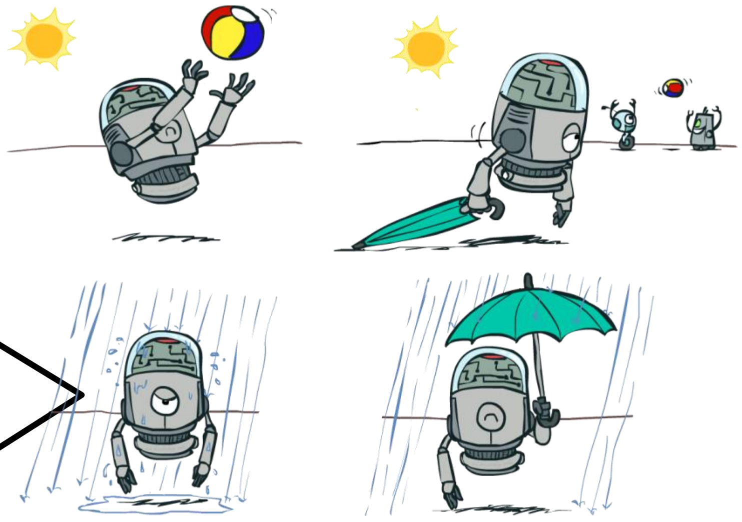
$$\mathrm{EU}(\mathrm{take}) = \sum_{w} P(w)U(\mathrm{take}, w)$$

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

Optimal decision = leave

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

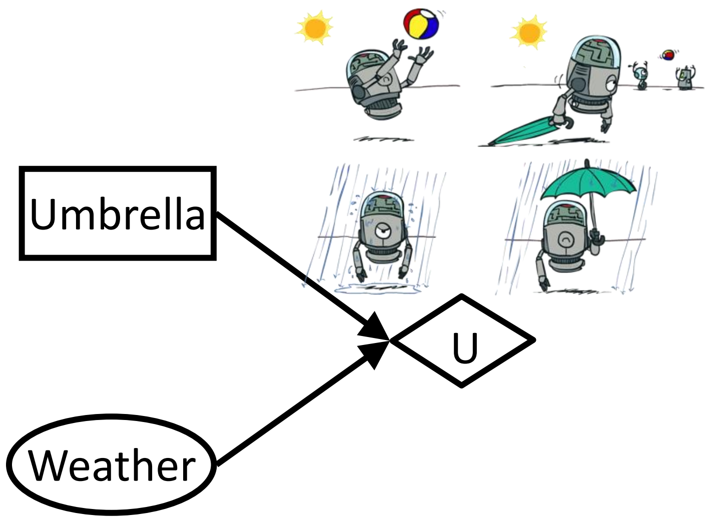


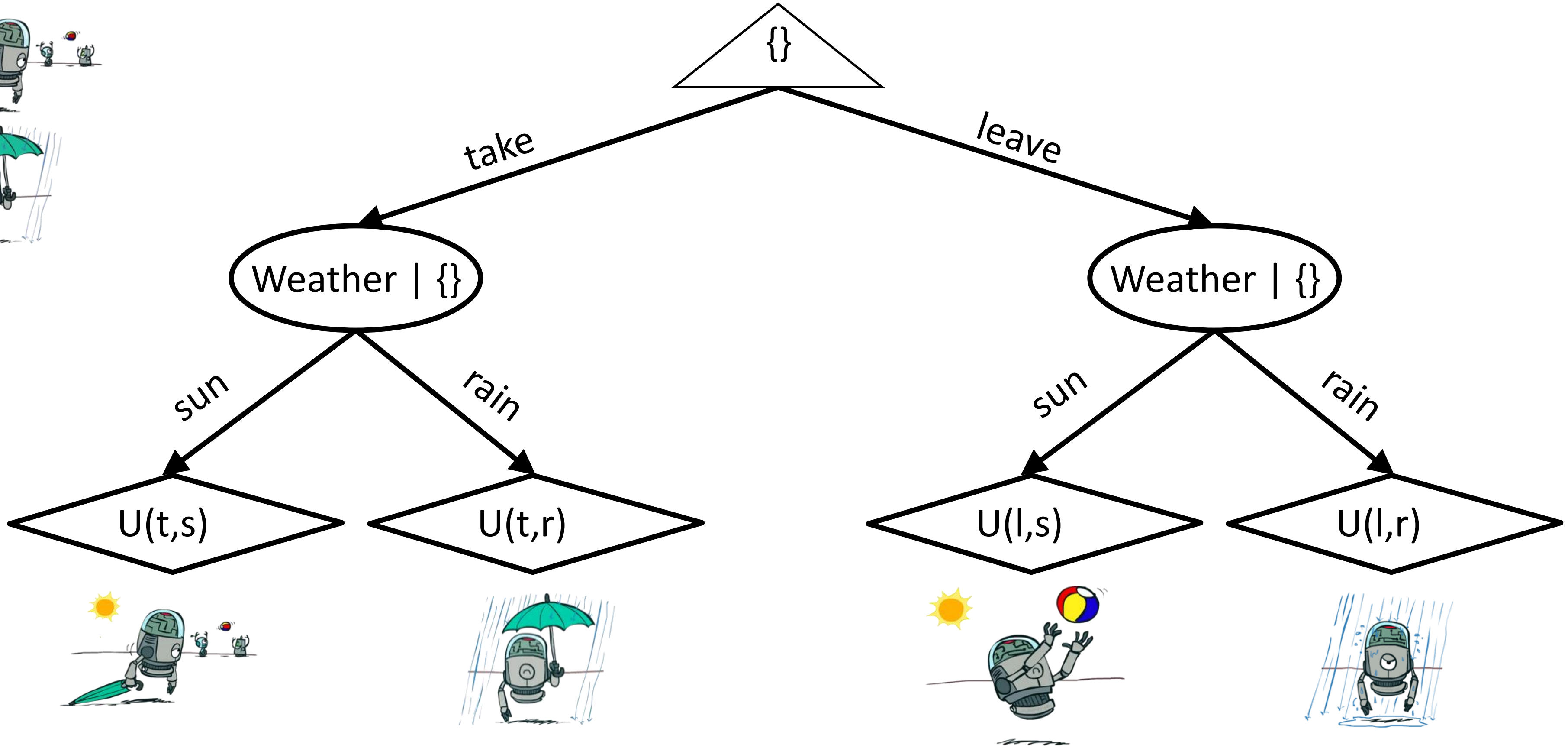


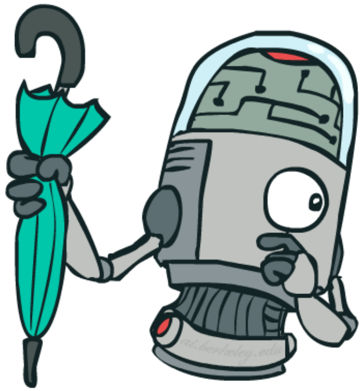
W	P(W)
sun	0.7
rain	0.3

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

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

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

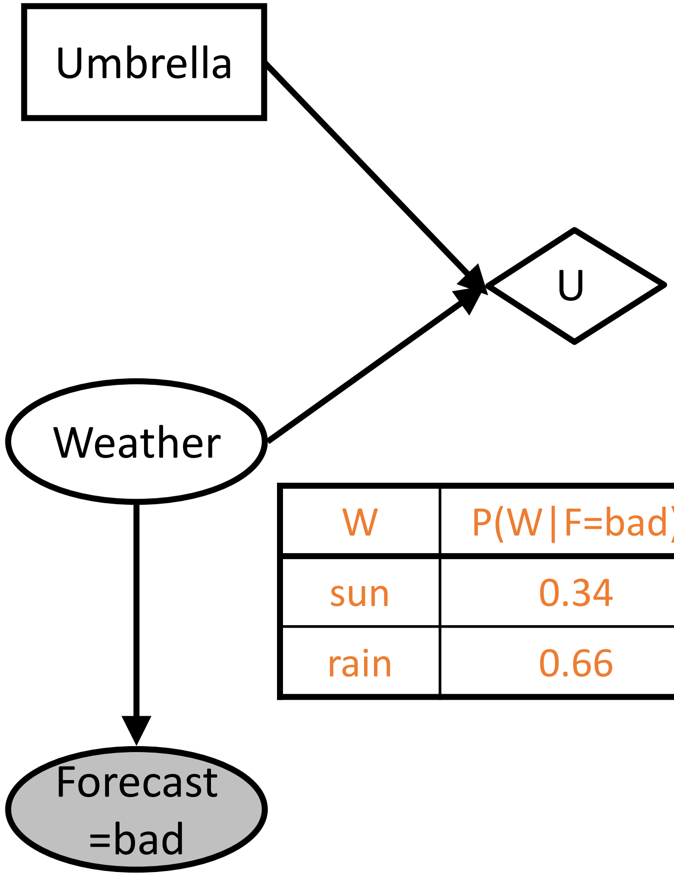
Umbrella = take

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

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

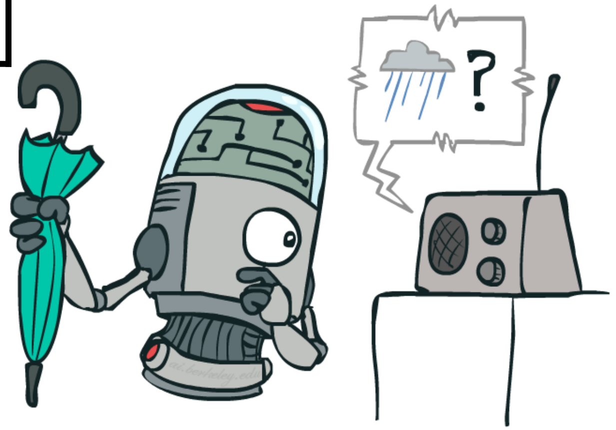
Optimal decision = take

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



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