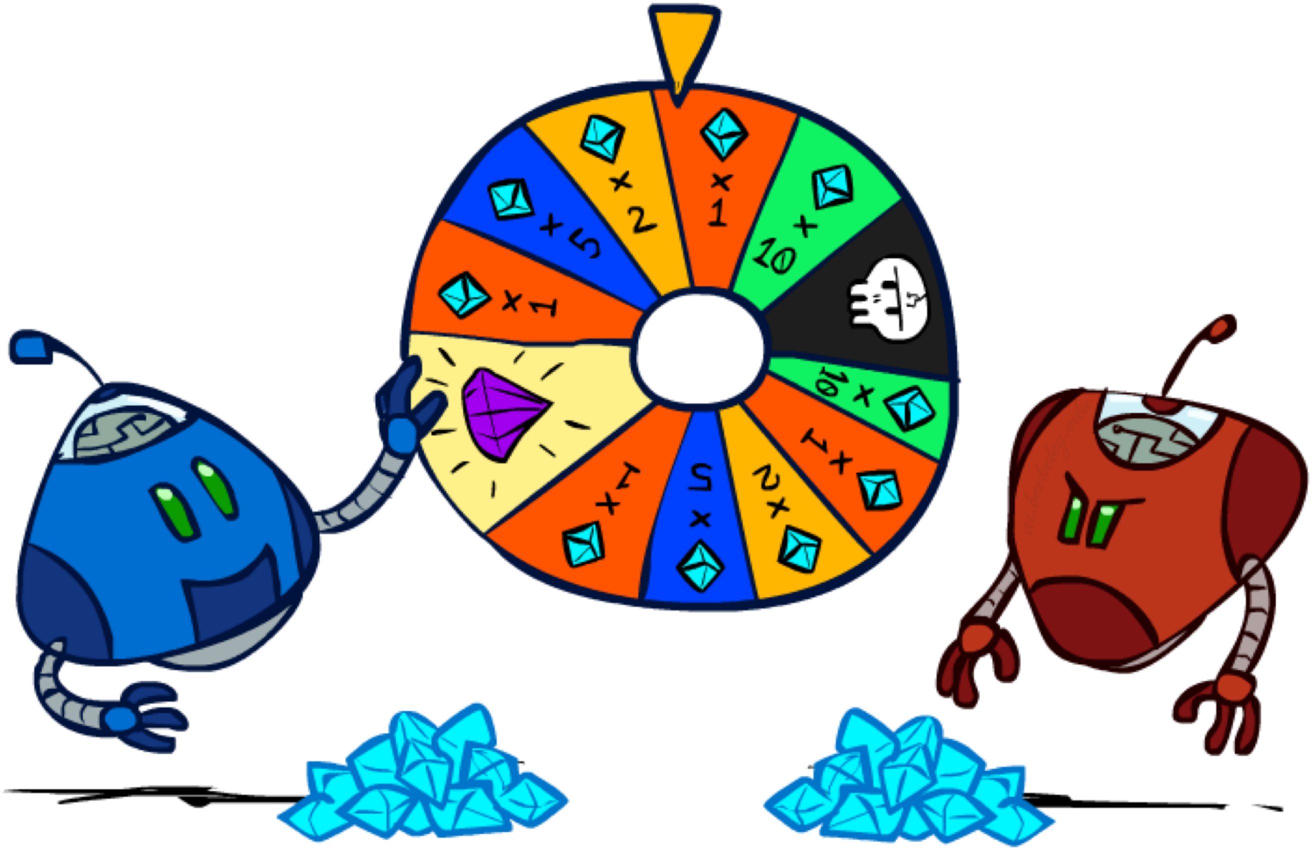
CS 188: Artificial Intelligence

Decision Networks + VPI



Instructors: Angela Liu and Yanlai Yang

University of California, Berkeley

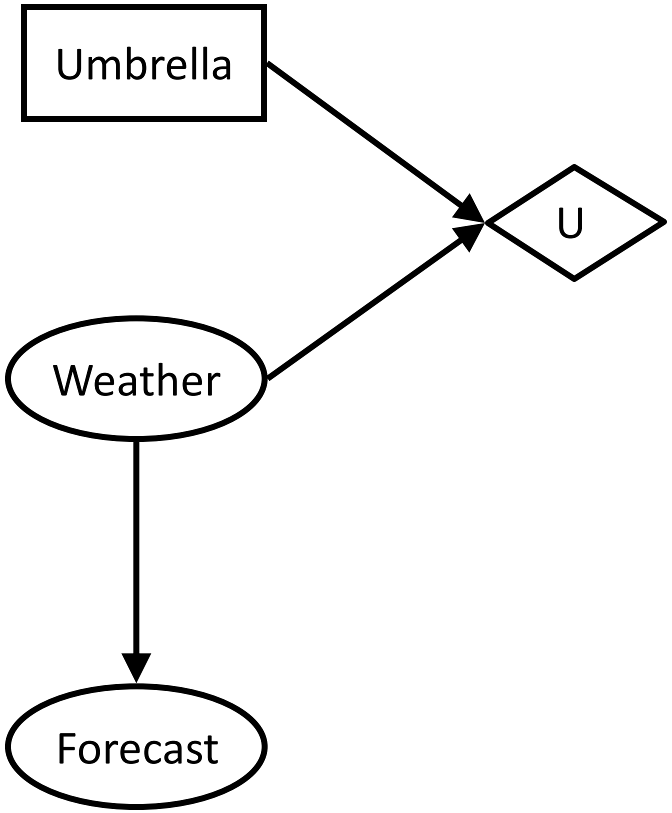
(Slides adapted Stuart Russell and Dawn Song)

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

Bayes net inference!



Maximum Expected Utility

Umbrella = leave

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

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

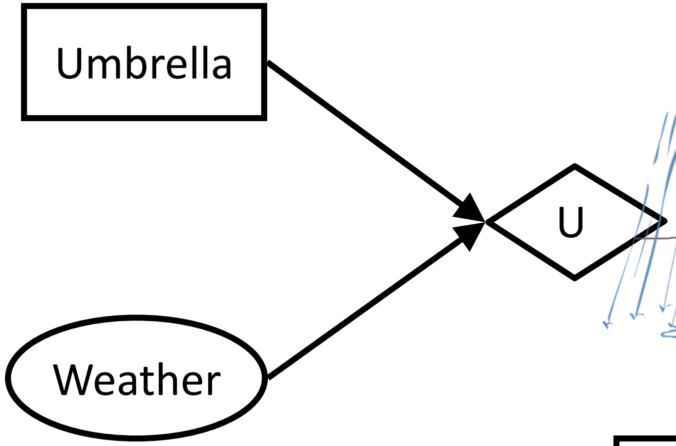
Umbrella = take

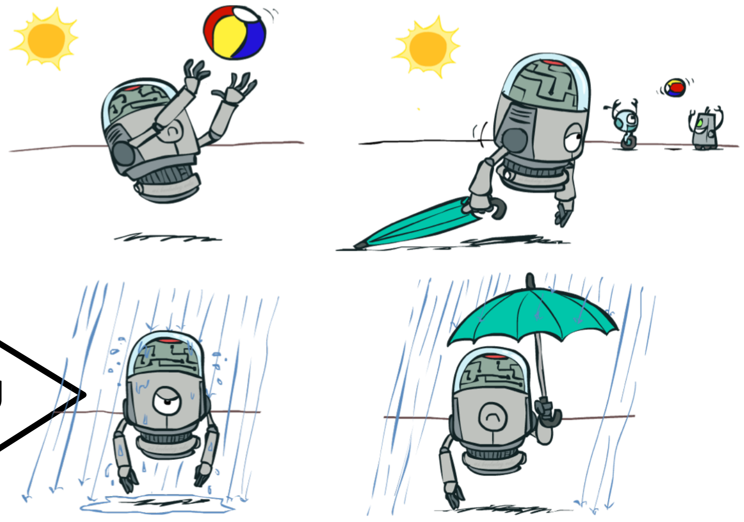
$$\mathrm{EU}(\mathrm{take}) = \sum_w P(w)U(\mathrm{take}, w)$$

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

Optimal decision = leave

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





W	P(W)
sun	0.7
rain	0.3

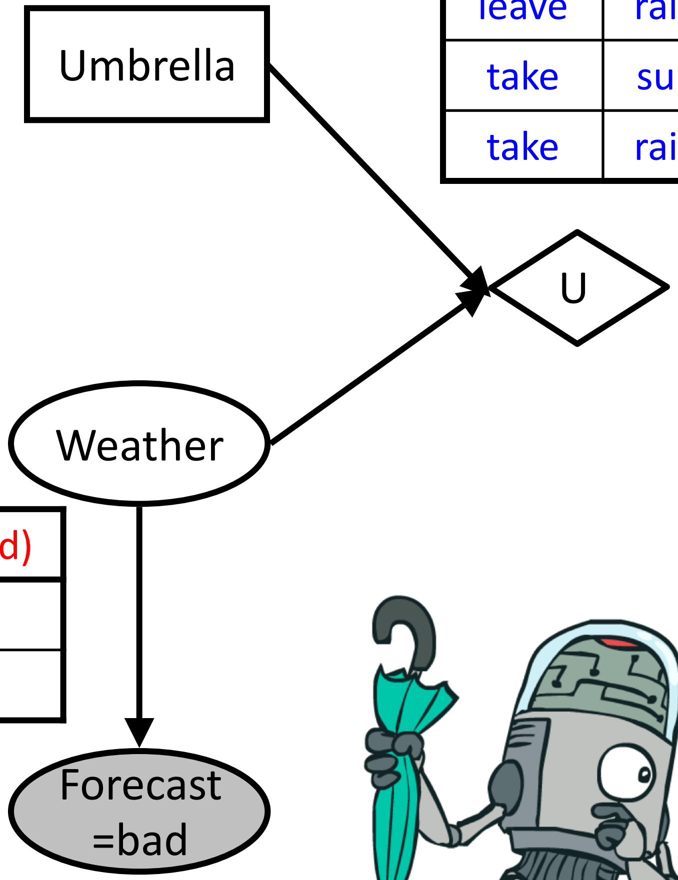
A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

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: $\sum_w P(\boldsymbol{w}|\boldsymbol{e},a)\, U(a,\boldsymbol{w})$
  - 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

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

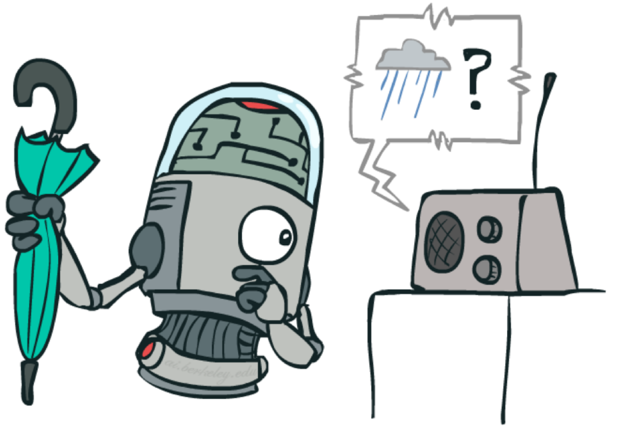
$$= 0.34 \times 100 + 0.66 \times 0 = 34$$

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!

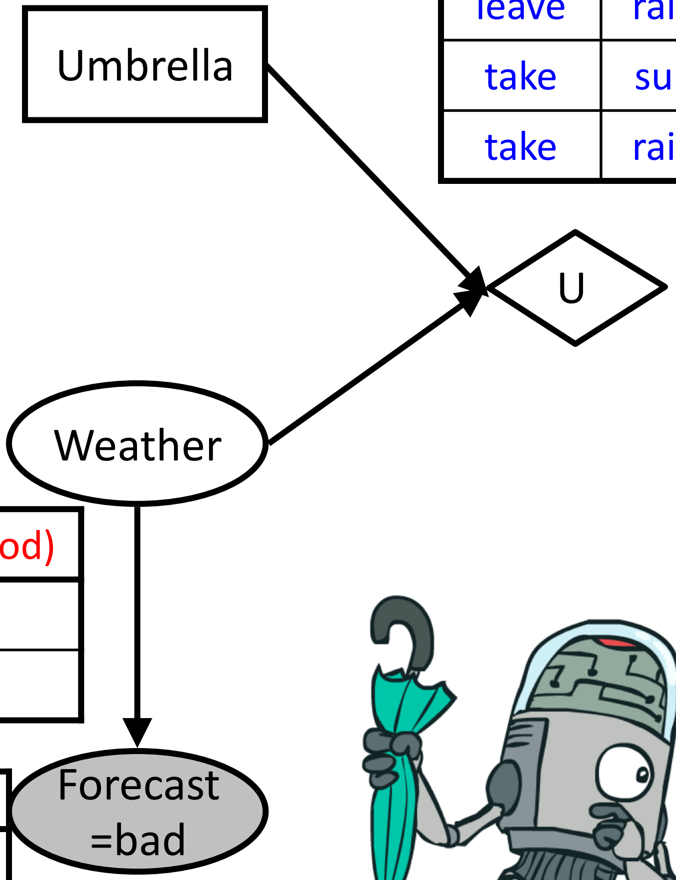


Example: Take an umbrella?

- Decision algorithm:
  - Fix evidence ***e***
  - For each possible action *a*
    - Fix action node to *a*
    - Compute posterior $P(\mathbf{W}|\mathbf{e},a)$ for parents $\mathbf{W}$ of U
    - Compute expected utility of action a: $\sum_{\mathbf{w}} P(\mathbf{w}|\mathbf{e},a)\, U(a,\mathbf{w})$
  - 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=good)
sun	0.89
rain	0.11

W	P(F=good|W)
sun	0.83
rain	0.23

Umbrella = leave

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

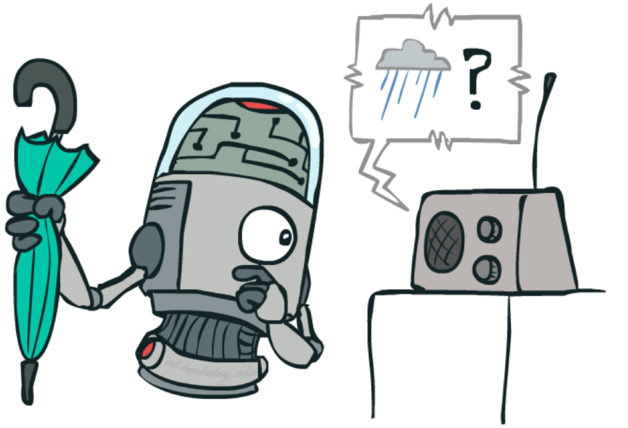
$$= 0.89\text{x}100 + 0.11\text{x}0 = 89$$

Umbrella = take

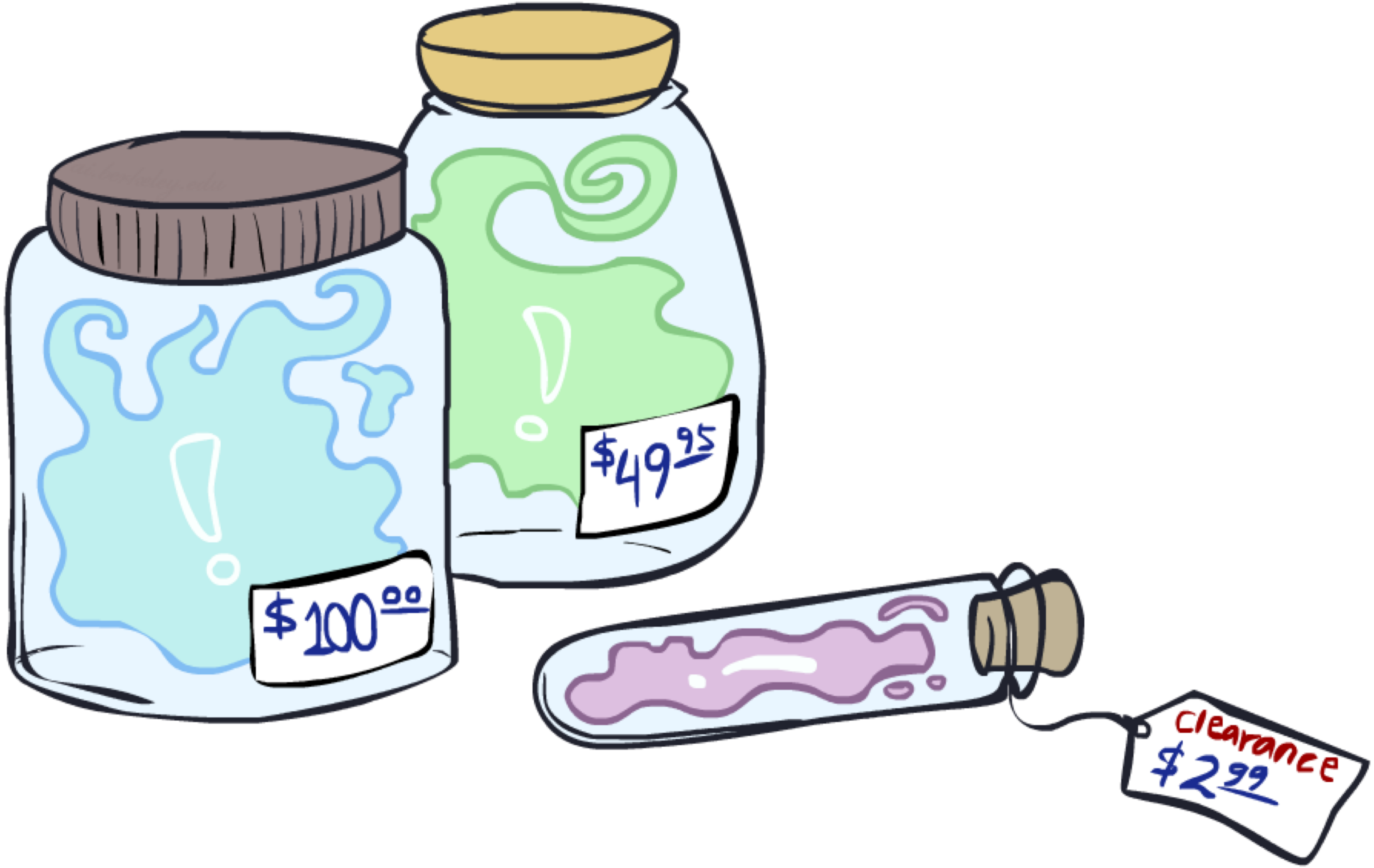
$$EU(\text{take}|F=\text{good}) = \sum_w P(w|F=\text{good})\, U(\text{take},w)$$

$$= 0.89\text{x}20 + 0.11\text{x}70 = 26$$

Optimal decision = leave!



Value of Information



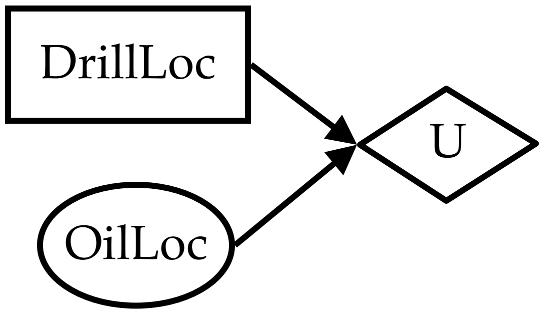
A question to motivate VPI

How do you tell if you want to take a specific class next semester?

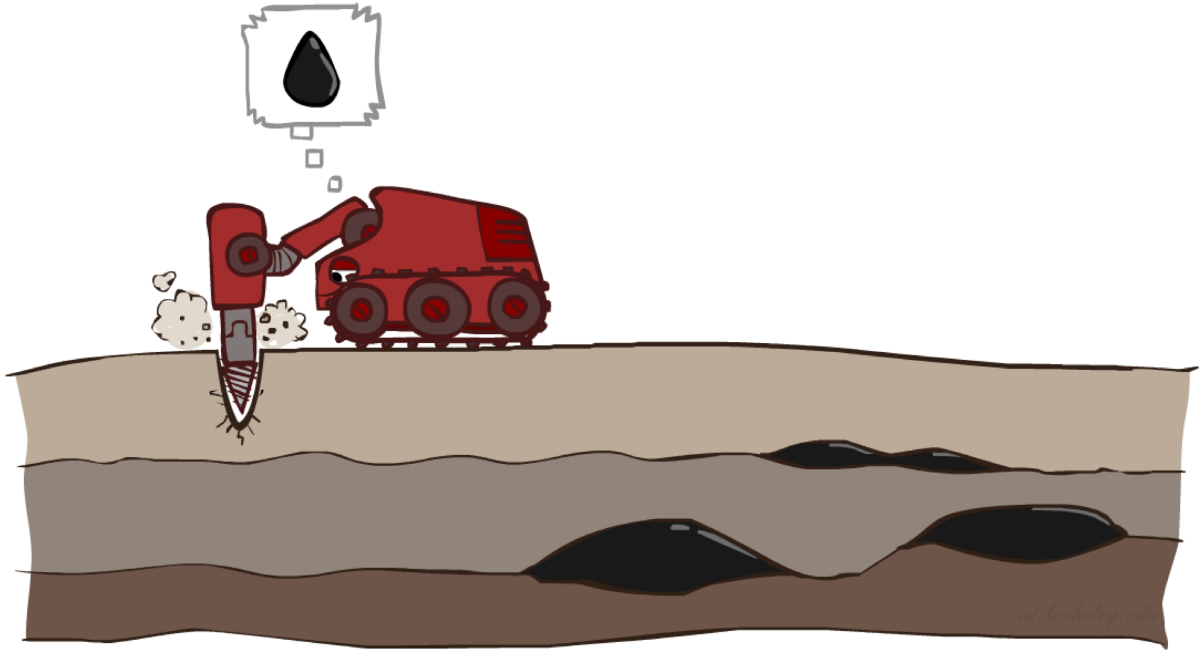
Value of Perfect Information

- 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
  - If we know OilLoc, MEU is k (either way)
  - Gain in MEU from knowing OilLoc?
  - VPI(OilLoc) = k – k/2 = 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

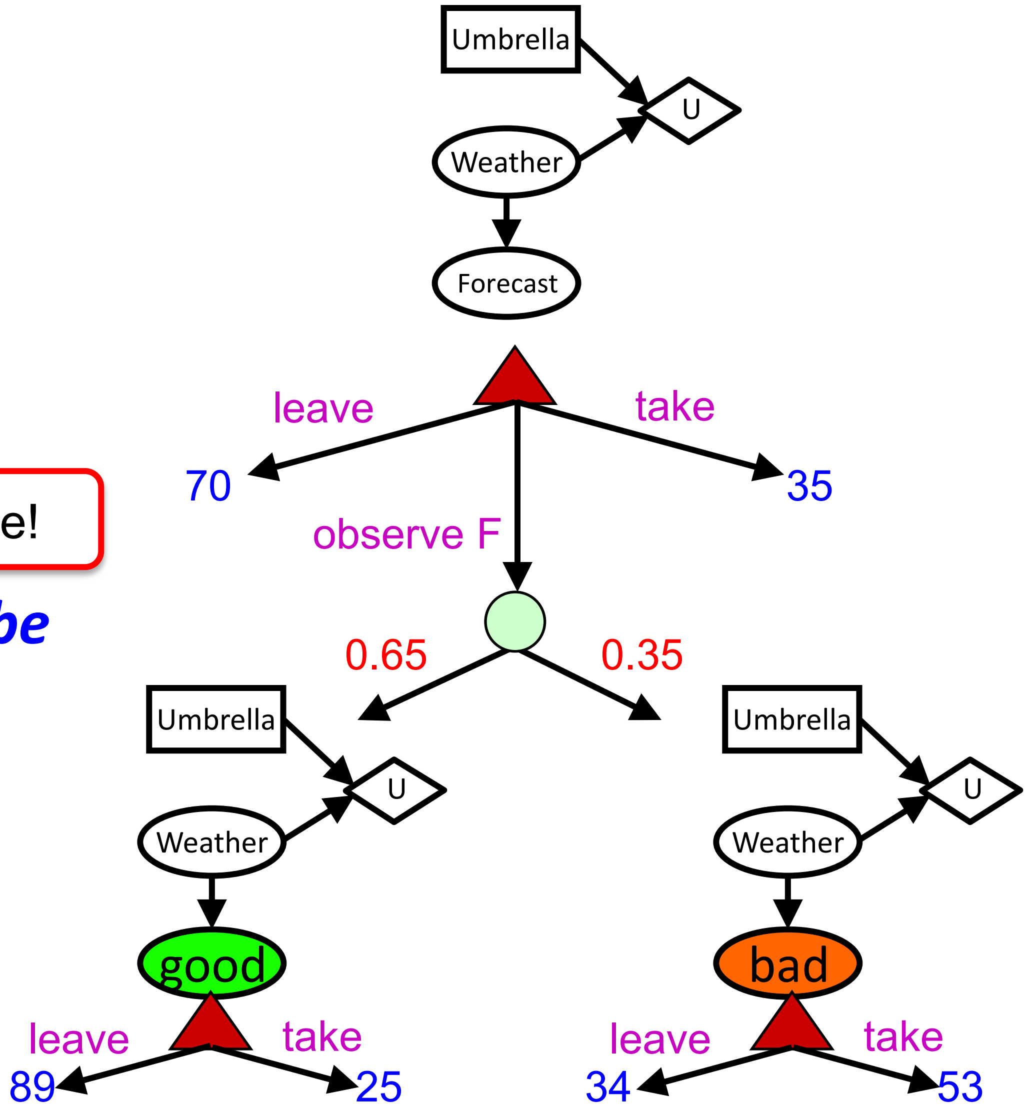


Value of information

- Before you see the forecast (no evidence)
  - MEU(ø) = $\max_a$EU(a) = 70
- ***What if you look at the forecast?***
- If Forecast=bad
  - MEU(F=bad) = $\max_a$EU(a | F=bad) = 53
- If Forecast=good
  - MEU(F=good) = $\max_a$EU(a |
- ***But, we don't know what the ecast will be ahead of time!***
- So we need a distribution of P(F)
- Expected utility given forecast
  - = 0.35 x 53 + 0.65 x 89 = 76.4
- ***Value of information*** = 76.4-70 = 6.4

Bayes net inference!

F	P(F)
good	0.65
bad	0.35



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:

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

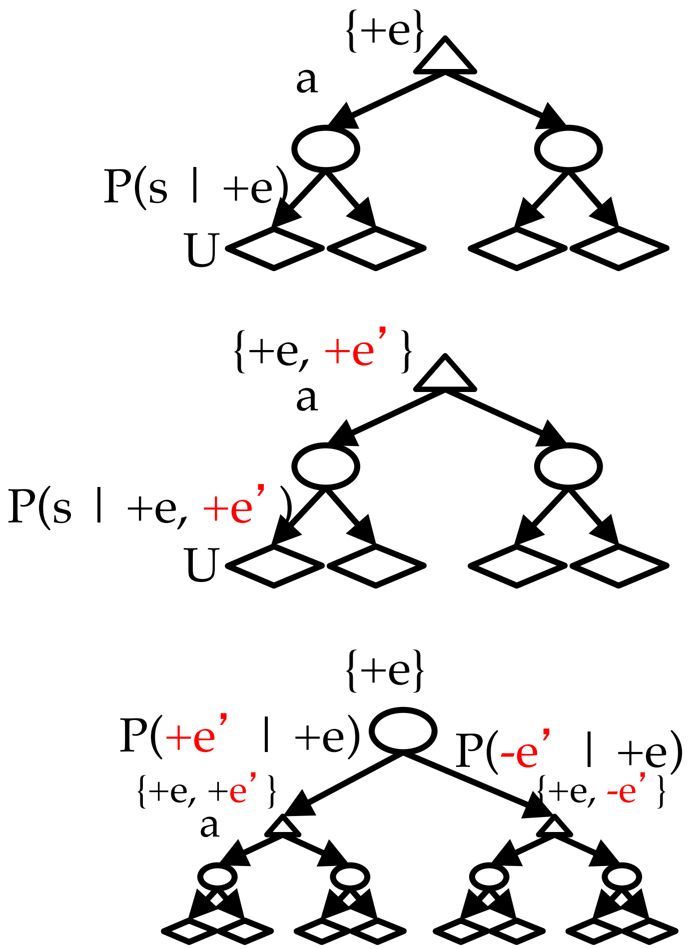
- BUT E' is a random variable whose value is unknown, so we don't know what e' will be

- 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:

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



Value of Information

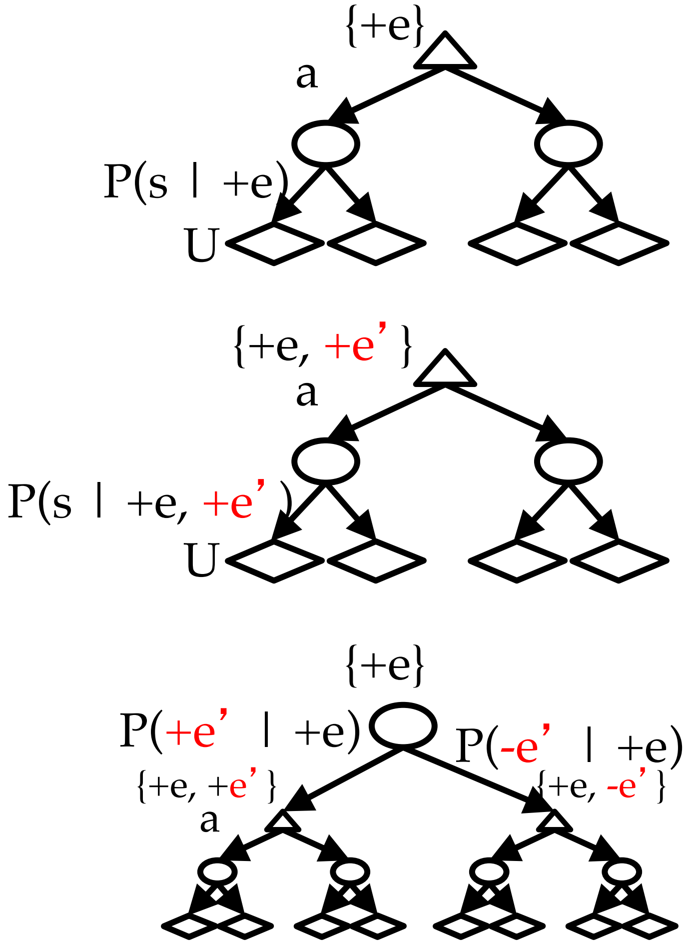
$$\text{MEU}(e, E') = \sum_{e'} P(e'|e)\text{MEU}(e, e')$$

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

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

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

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



VPI Properties

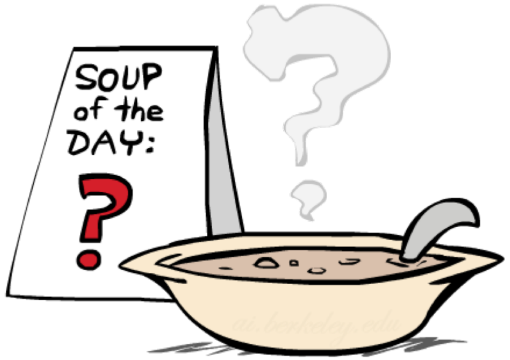
VPI is non-negative! $\mathrm{VPI}(E_i \mid e) \geq 0$

VPI is not (usually) additive: $\mathrm{VPI}(E_i, E_j \mid e) \neq \mathrm{VPI}(E_i \mid e) + \mathrm{VPI}(E_j \mid e)$

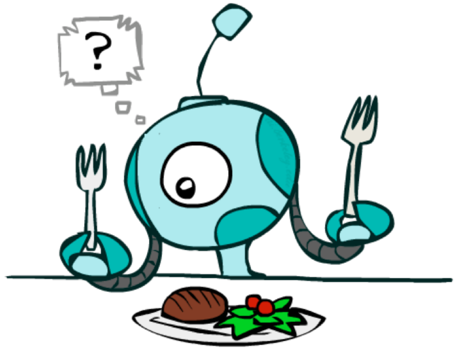
VPI is order-independent: $\mathrm{VPI}(E_i, E_j \mid e) = \mathrm{VPI}(E_j, E_i \mid e)$

Quick VPI Questions

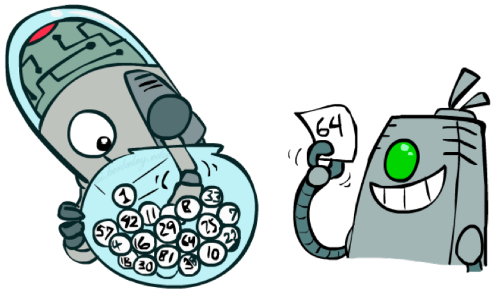
- The soup of the day is either clam chowder or split pea, but you wouldn't order either one. What's the value of knowing which it is?



- 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?

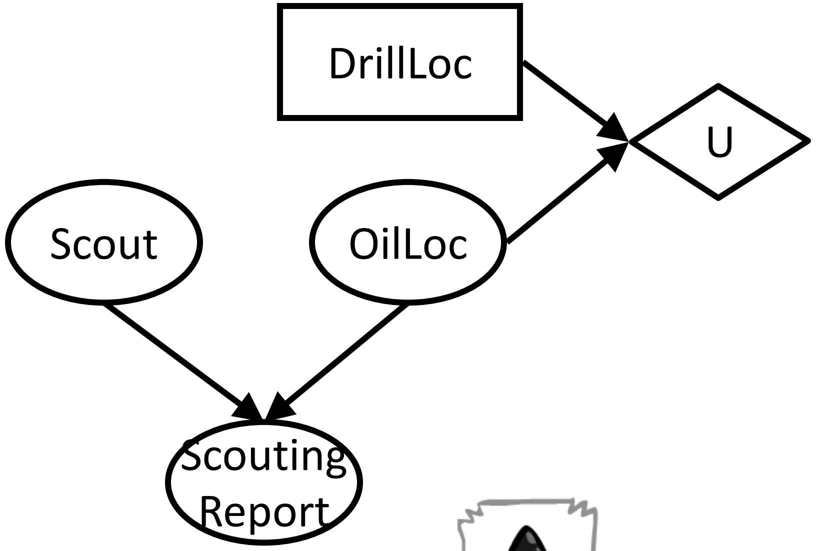


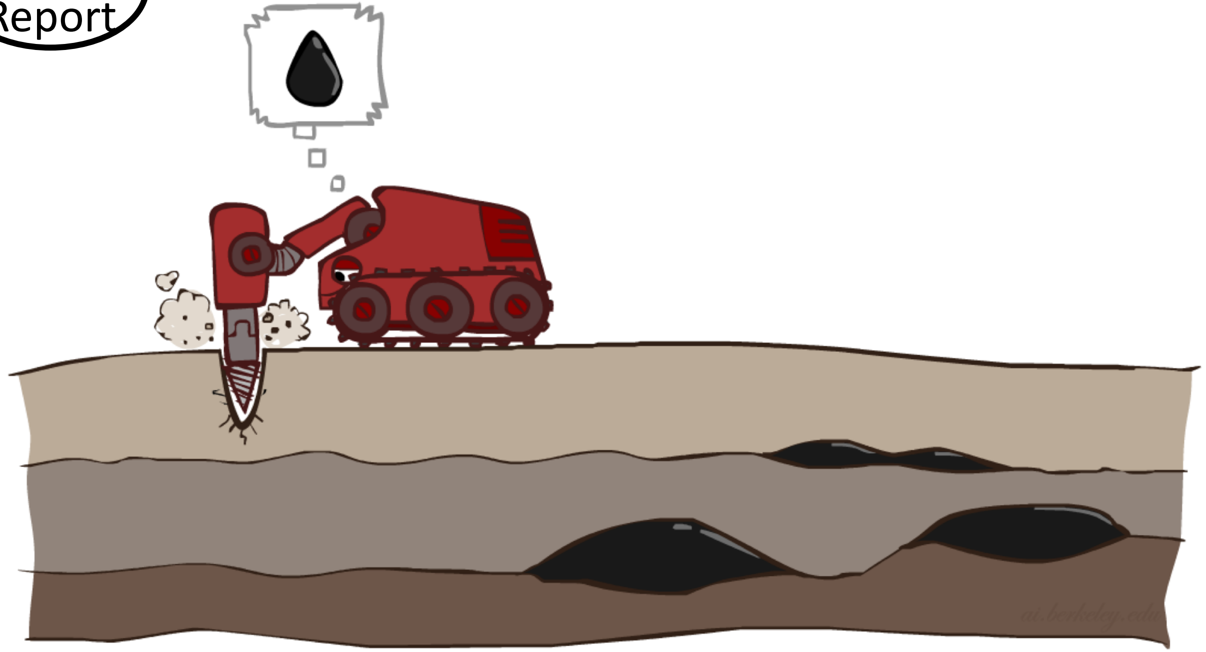
Value of Imperfect Information?

- 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 Parents(U) $\perp\!\!\!\perp$ Z | CurrentEvidence)

  Then VPI(Z | CurrentEvidence) = 0

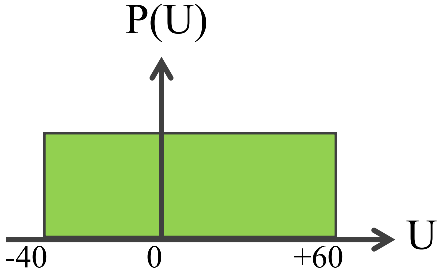
Decisions with unknown preferences

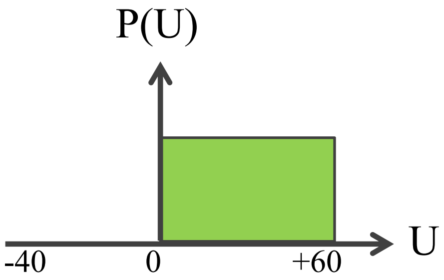
- In reality the assumption that we can write down our exact preferences for the machine to optimize is false
- A machine optimizing the wrong preferences causes problems

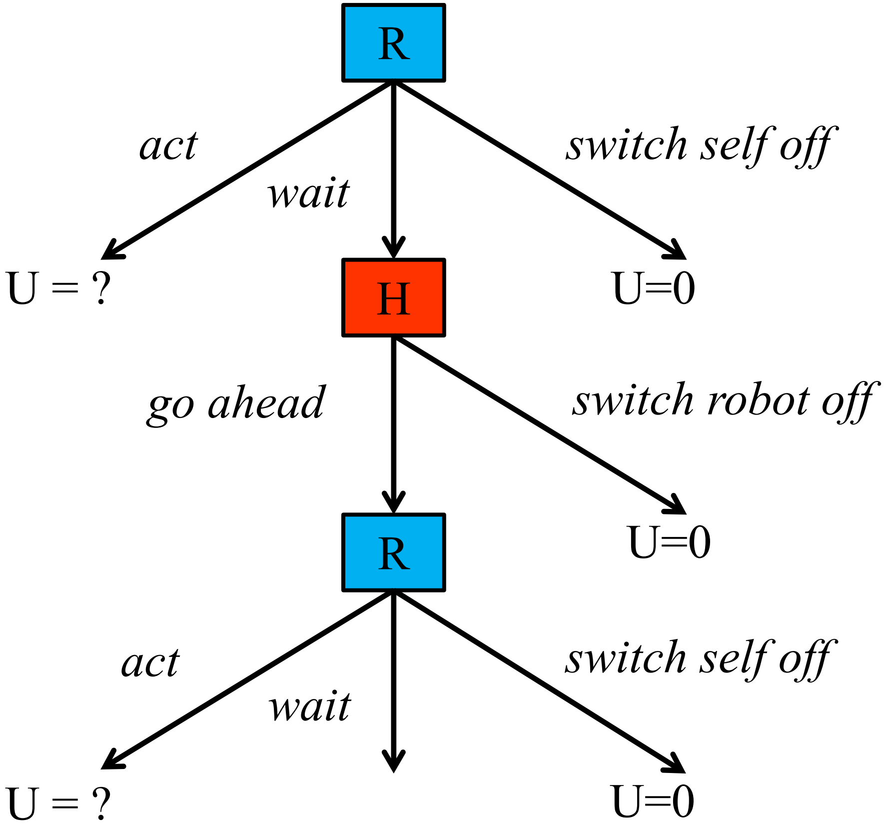
Decisions with unknown preferences

- In reality the assumption that we can write down our exact preferences for the machine to optimize is false
- A machine optimizing the wrong preferences causes problems
- A machine that is explicitly uncertain about the human's preferences will defer to the human (e.g., allow itself to be switched off)

Off-switch problem (example)







EU(act) = +10

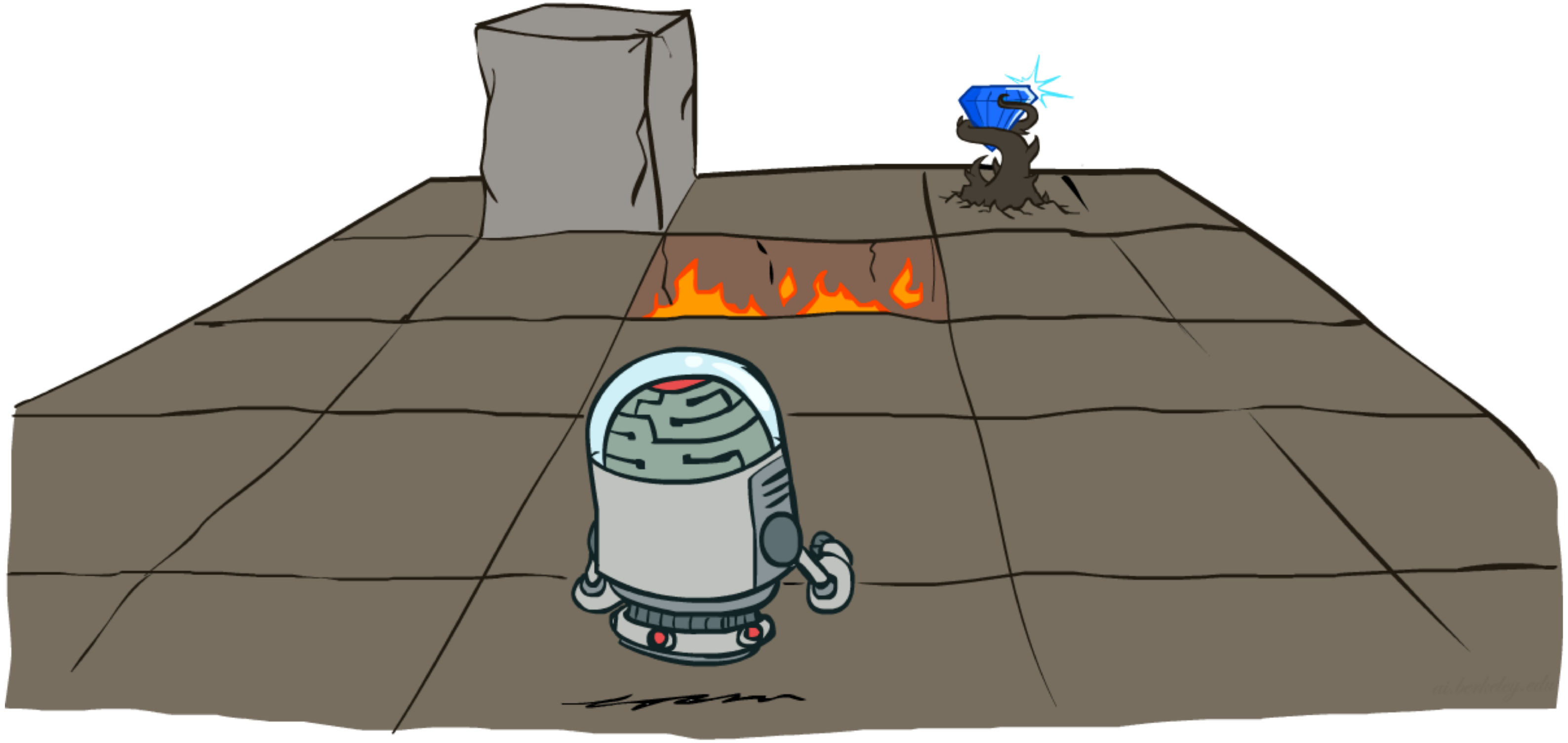
*EU(wait) = (0.4 * 0) + (0.6 * 30)*
= +18

Off-switch problem (general proof)

- $EU(act) = \int_{-\infty}^{+\infty} P(u) \cdot u \, du = \int_{-\infty}^{0} P(u) \cdot u \, du + \int_{0}^{+\infty} P(u) \cdot u \, du$
- $EU(wait) = \int_{-\infty}^{0} P(u) \cdot 0 \, du + \int_{0}^{+\infty} P(u) \cdot u \, du$
- Obviously $\int_{-\infty}^{0} P(u) \cdot u \, du \leq \int_{-\infty}^{0} P(u) \cdot 0 \, du$
- Hence $EU(act) \leq EU(wait)$
  - "If H doesn't switch me off, then the action must be good for H, and I'll get to do it, so that's good; if H does switch me off, then it's because the action must be bad for H, so it's good that I won't be allowed to do it."

CS 188: Artificial Intelligence

Markov Decision Processes



Instructor: Angela Liu and Yanlai Yang

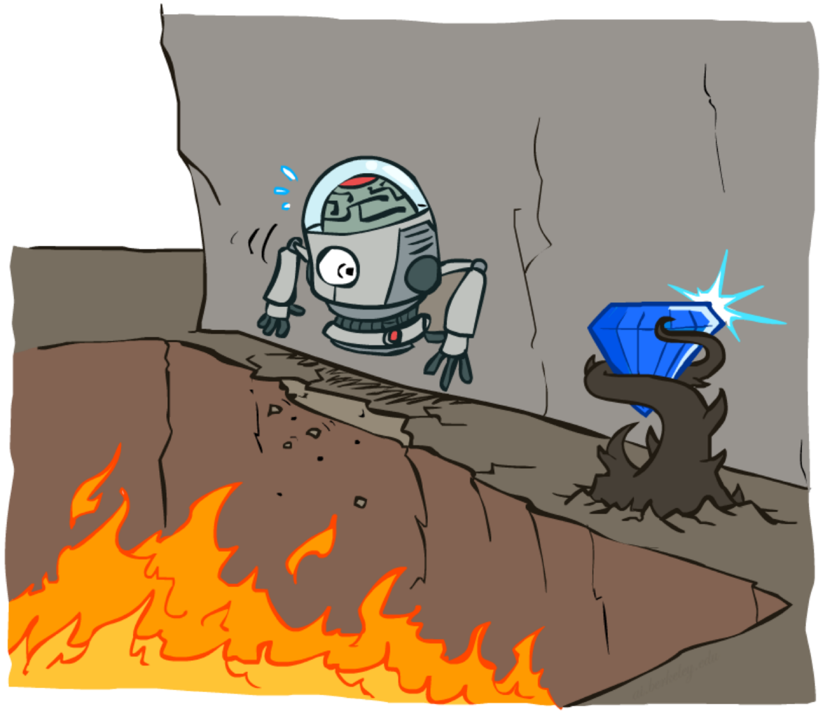
University of California, Berkeley

[These slides adapted from Dan Klein and Pieter Abbeel]

Sequential decisions under uncertainty

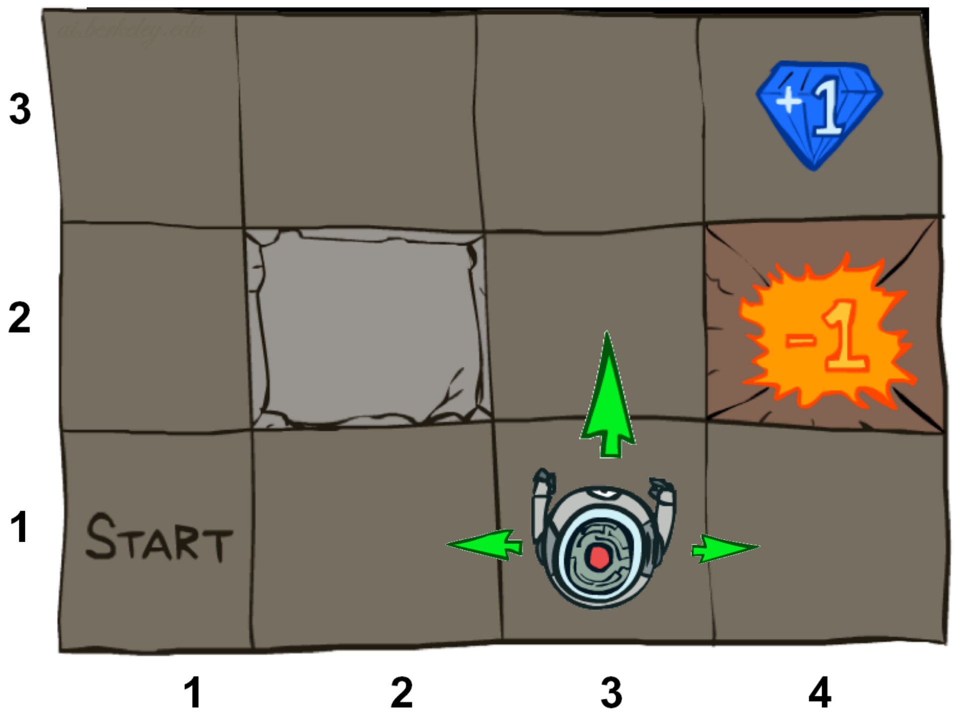
So far, decision problem is one-shot --- concerning only one action

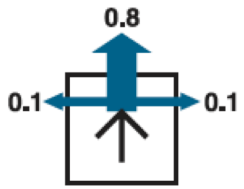
Sequential decision problem: agent's utility depends on a sequence of actions



Example: Grid World

- A maze-like problem
  - The agent lives in a grid
  - Walls block the agent's path
- Noisy movement: actions do not always go as planned
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent would have been taken, the agent stays put
- The agent receives rewards each time step
  - Small "living" reward r each step (can be negative)
  - Big rewards come at the end (good or bad)
- Goal: maximize sum of rewards





Markov Decision Process (MDP)

- Environment history: [s_0, a_0, s_1, a_1, …, s_t]
- "Markov" generally means that given the present state, the future and the past are independent

- For Markov decision processes, "Markov" means action outcomes depend only on the current state

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1}, \ldots S_0 = s_0)$$

$$=$$

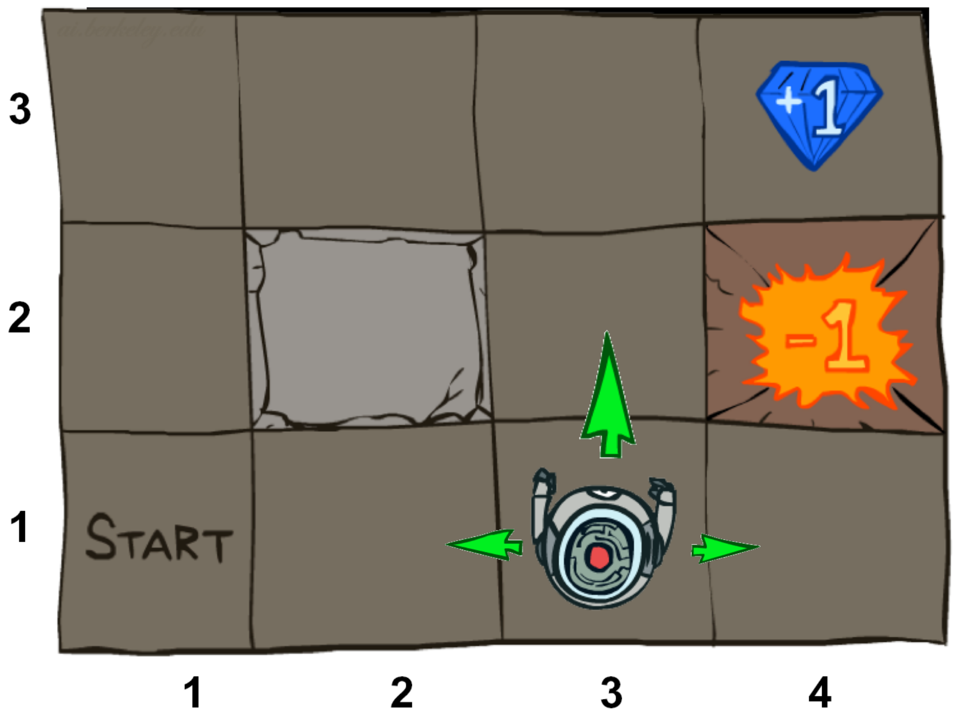
$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

- This is just like search, where the successor function could only depend on the current state (not the history)

Andrey Markov
(1856-1922)

Markov Decision Process (MDP)

- An MDP is defined by:
  - A set of states $s \in S$
  - A set of actions $a \in A$
  - A transition model $T(s, a, s')$
    - Probability that a from s leads to s', i.e., $P(s' | s, a)$
  - A reward function $R(s, a, s')$ for each transition
  - A start state
  - Possibly a terminal state (or absorbing state)
  - Utility function which is additive (discounted) rewards

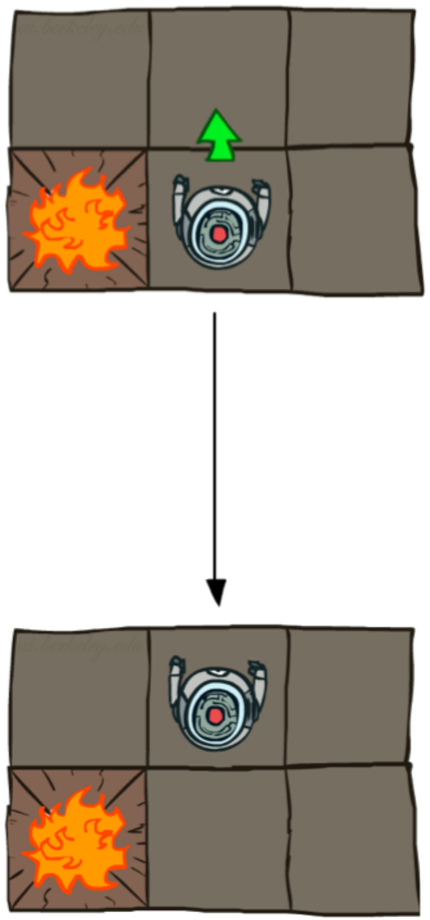


- MDPs are fully observable but probabilistic search problems

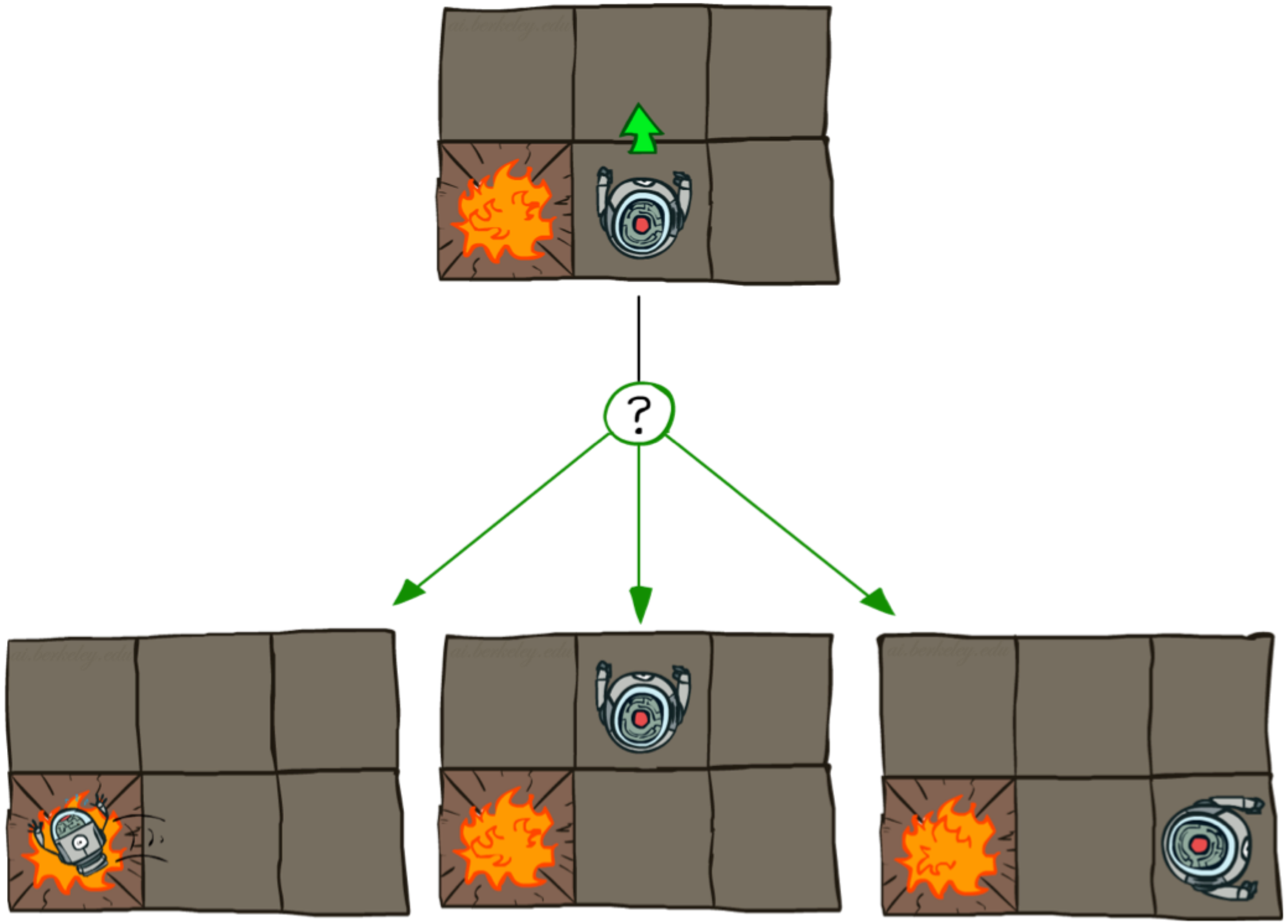
[Demo – gridworld manual intro (L8D1)]

Grid World Actions

Deterministic Grid World



Stochastic Grid World

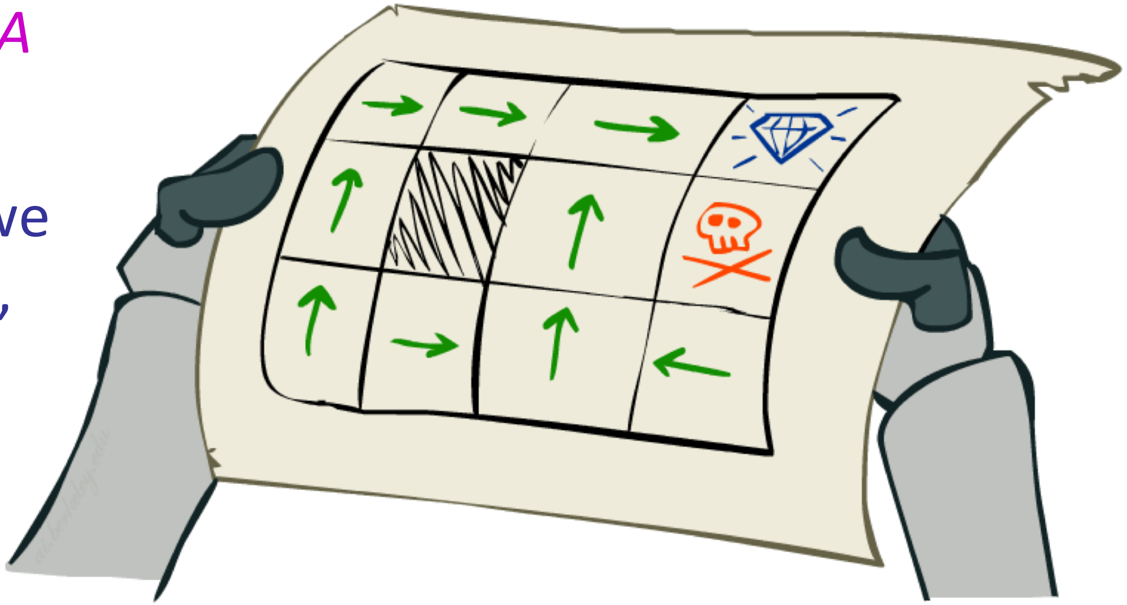


Policies

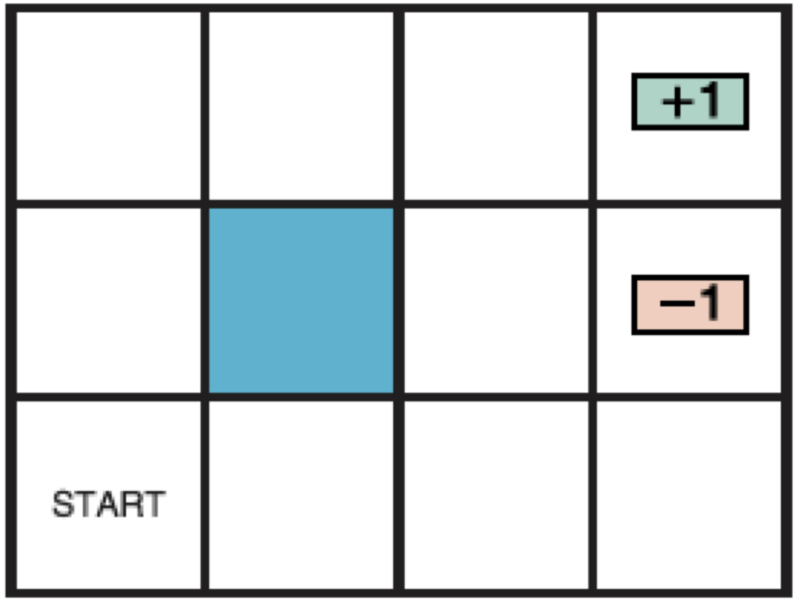
- A policy π gives an action for each state, π: *S* → *A*

- In deterministic single-agent search problems, we wanted an optimal ***plan***, or sequence of actions, from start to a goal

- For MDPs, we want an optimal ***policy*** π^*: *S* → *A*
  - An optimal policy maximizes expected utility
  - An explicit policy defines a reflex agent

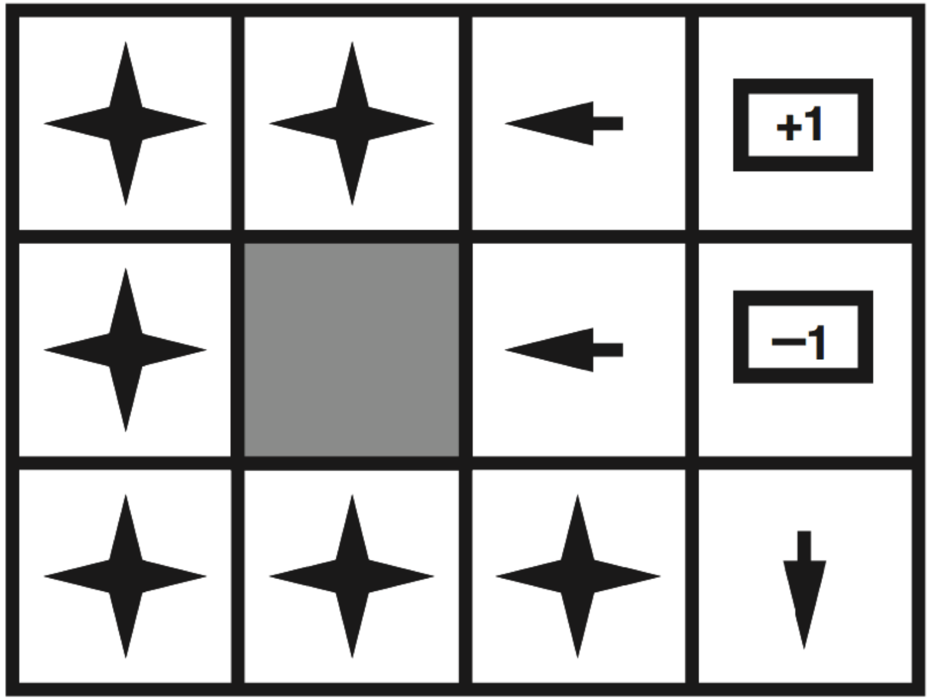


Optimal policy for r>0



r > 0

Optimal policy for r>0



r > 0

Sample Optimal Policies

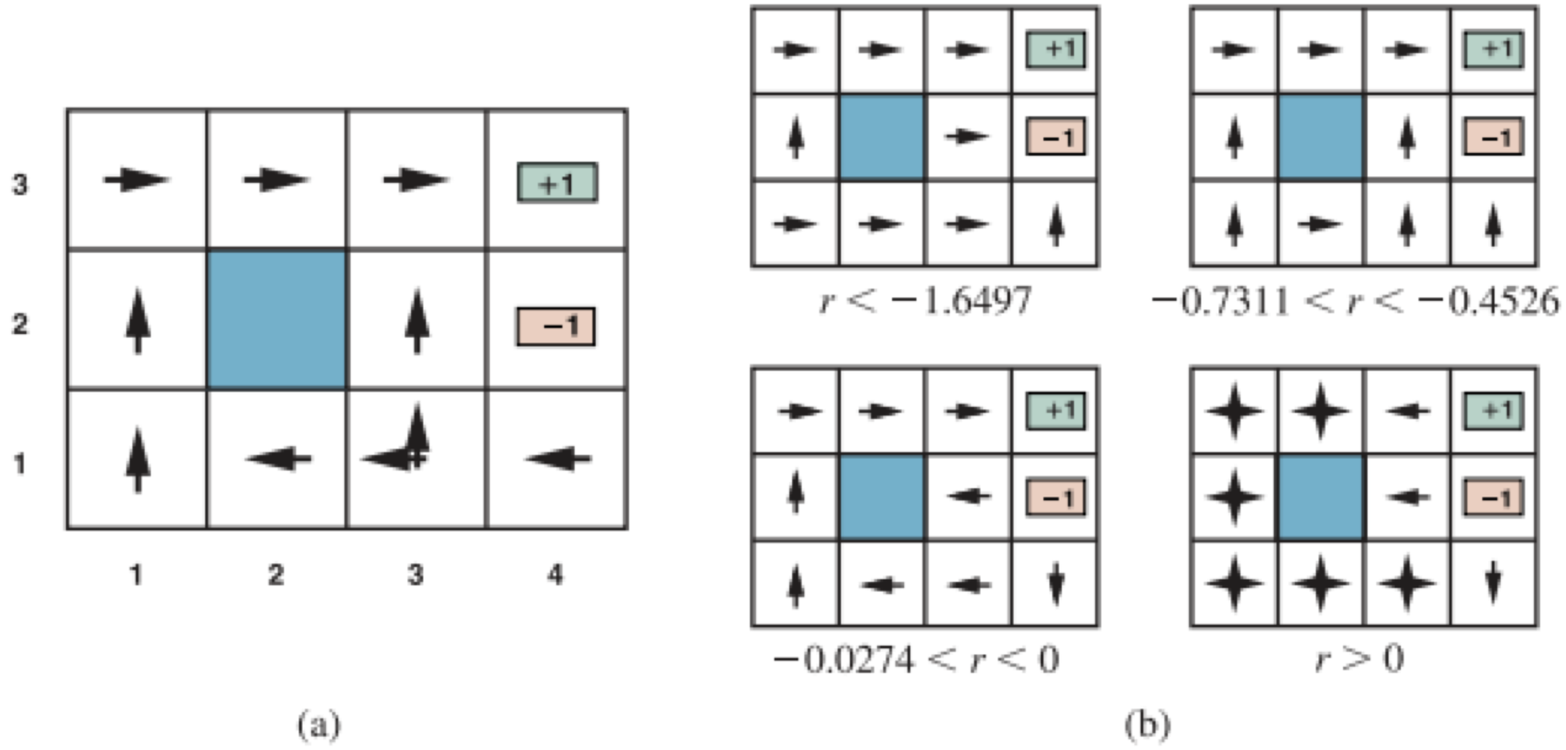
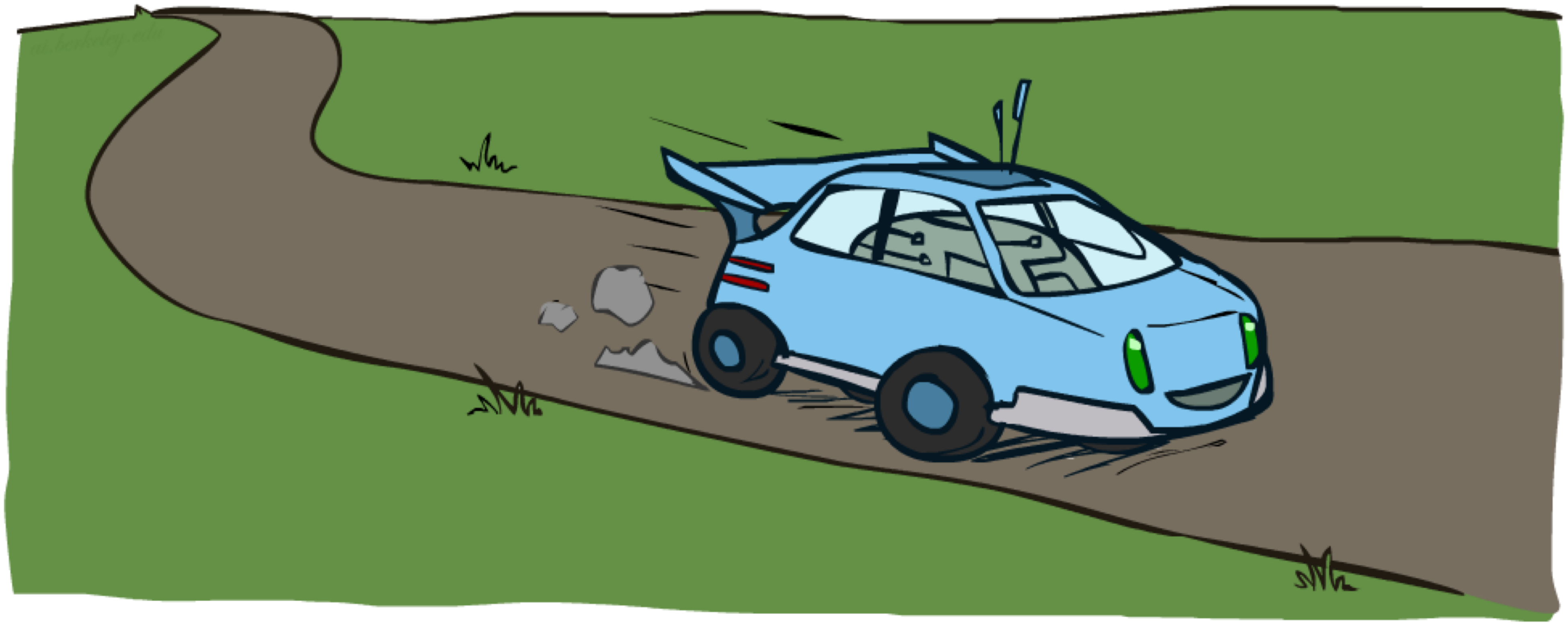


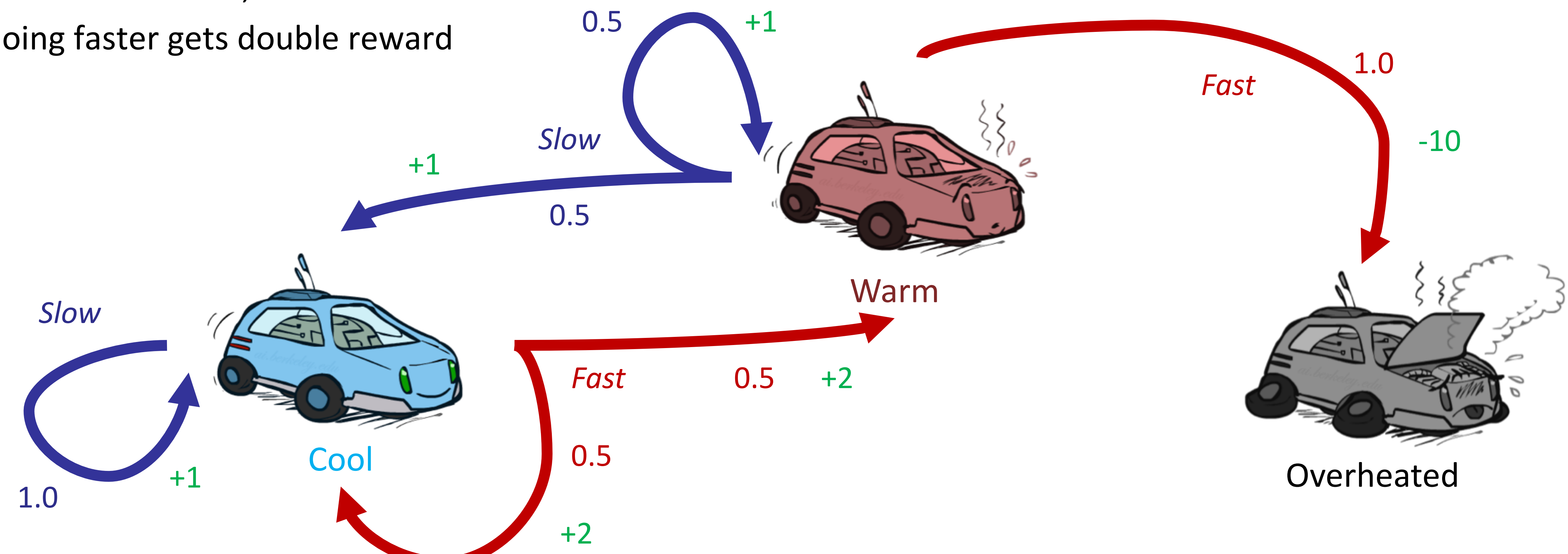
Figure 17.2 (a) The optimal policies for the stochastic environment with $r = -0.04$ for transitions between nonterminal states. There are two policies because in state (3,1) both *Left* and *Up* are optimal. (b) Optimal policies for four different ranges of r.

Example: Racing

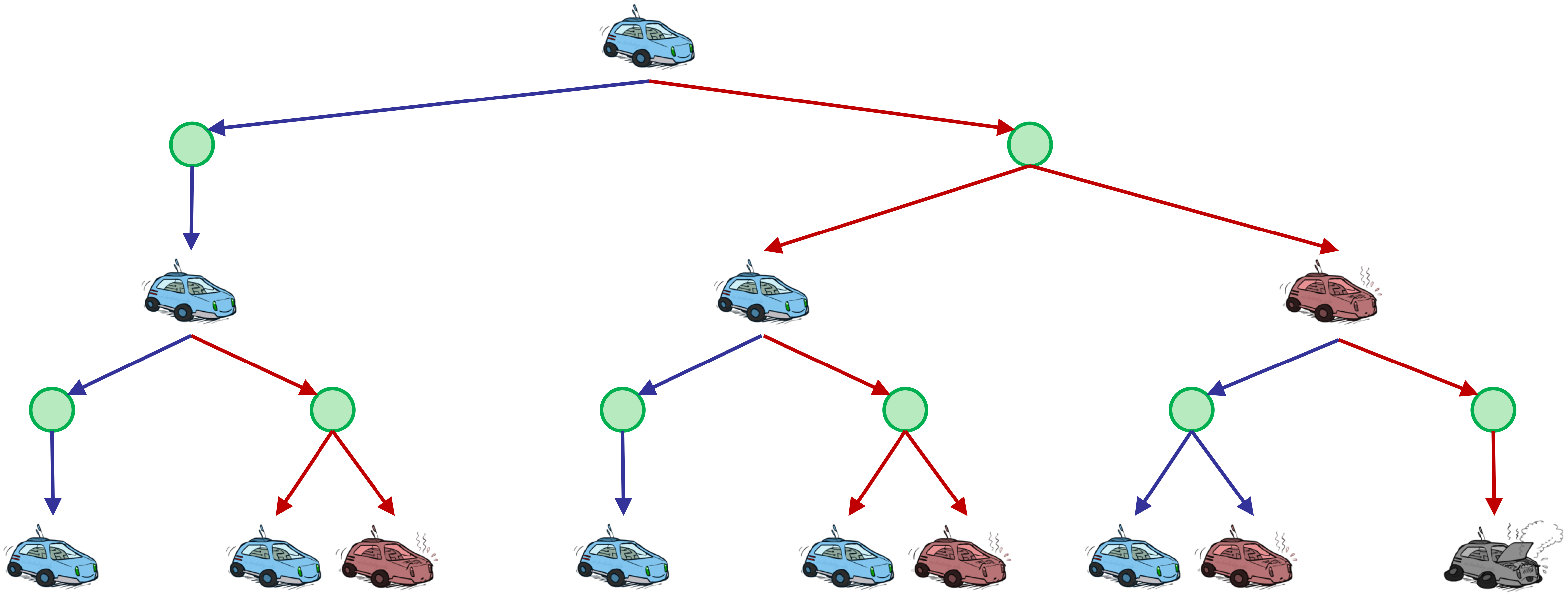


Example: Racing

- A robot car wants to travel far, quickly
- Three states: Cool, Warm, Overheated
- Two actions: *Slow*, *Fast*
- Going faster gets double reward

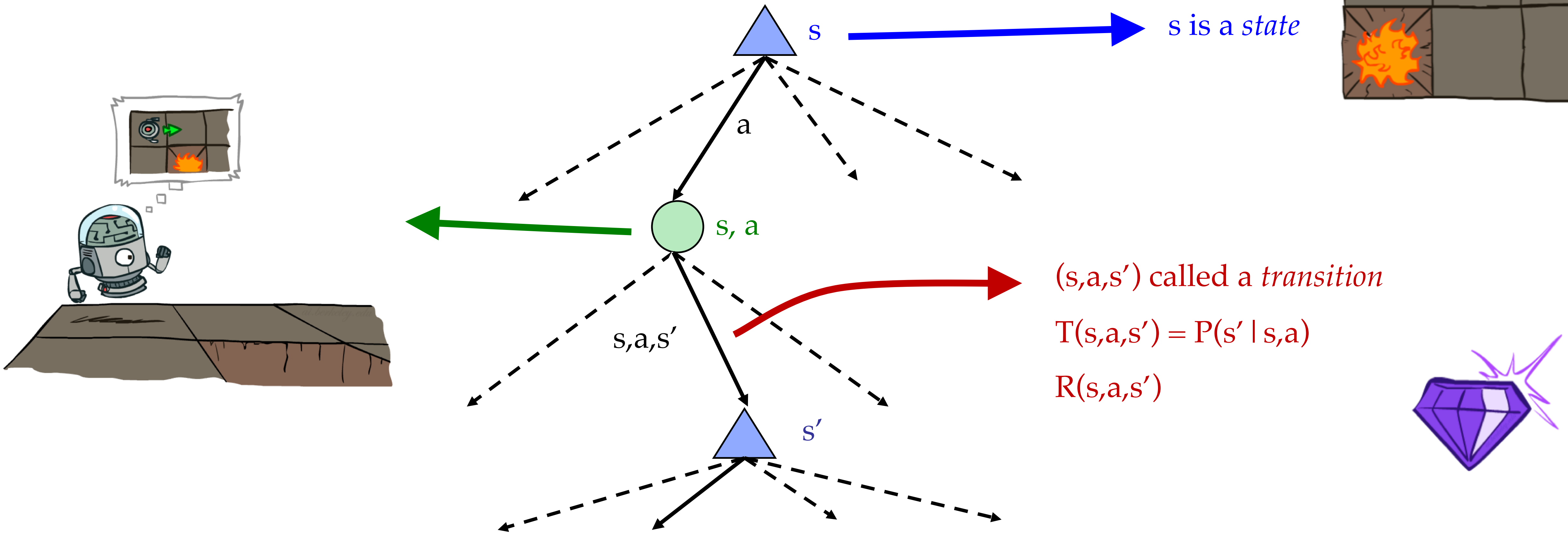


Racing Search Tree

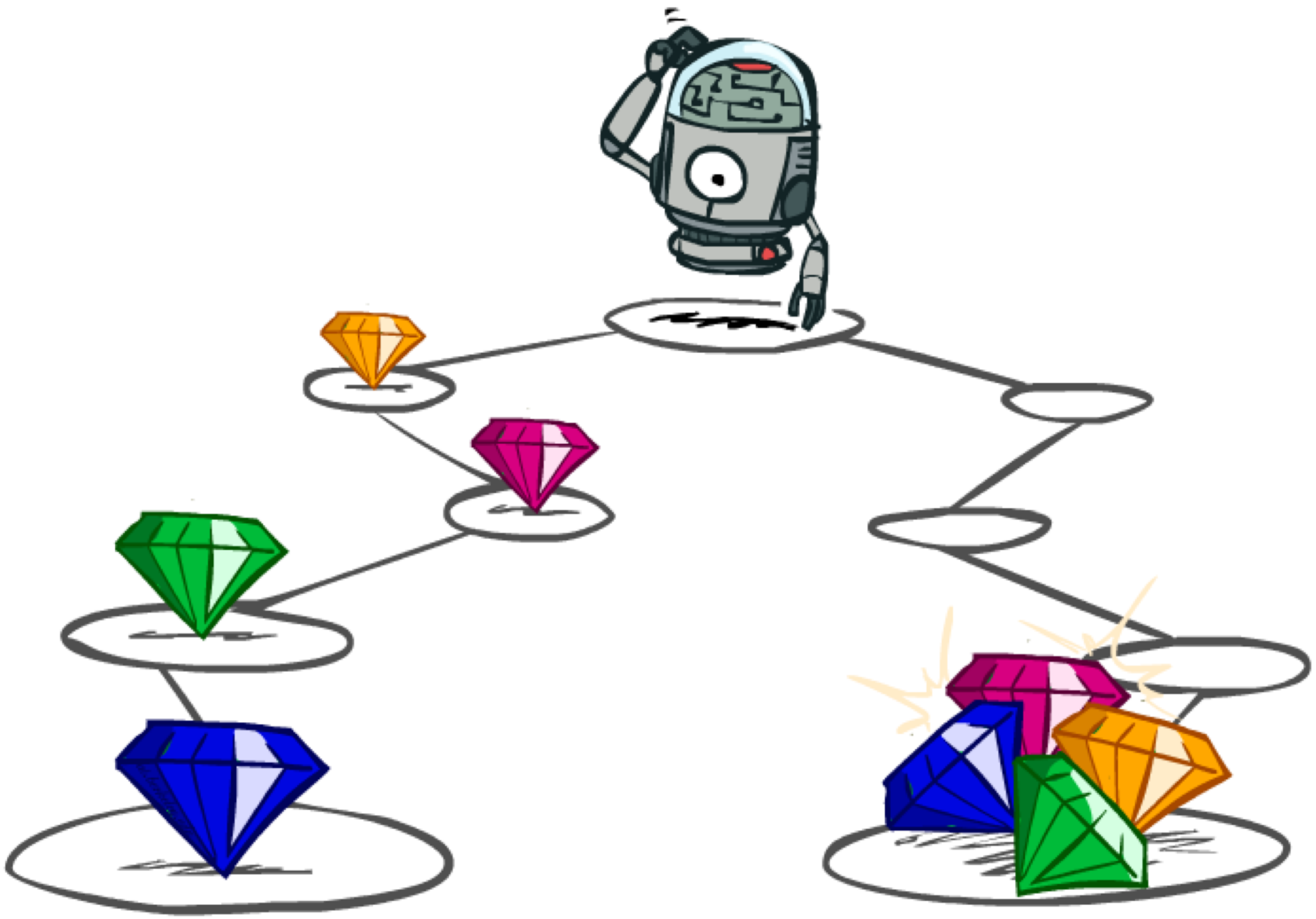


MDP Search Trees

- Each MDP state projects an expectimax-like search tree



Utilities of Sequences

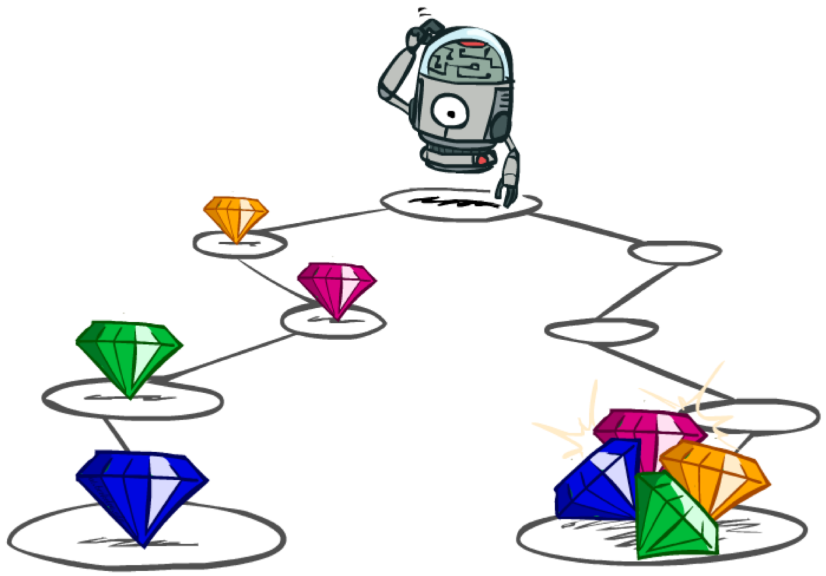


Utilities of Sequences

- What preferences should an agent have over reward sequences?

- More or less? [1, 2, 2] or [2, 3, 4]

- Now or later? [0, 0, 1] or [1, 0, 0]



Discounting

- It's reasonable to maximize the sum of rewards
- It's also reasonable to prefer rewards now to rewards later
- One solution: values of rewards decay exponentially

1

Worth Now

γ

Worth Next Step

γ^2

Worth In Two Steps

Discounting

Worth r now

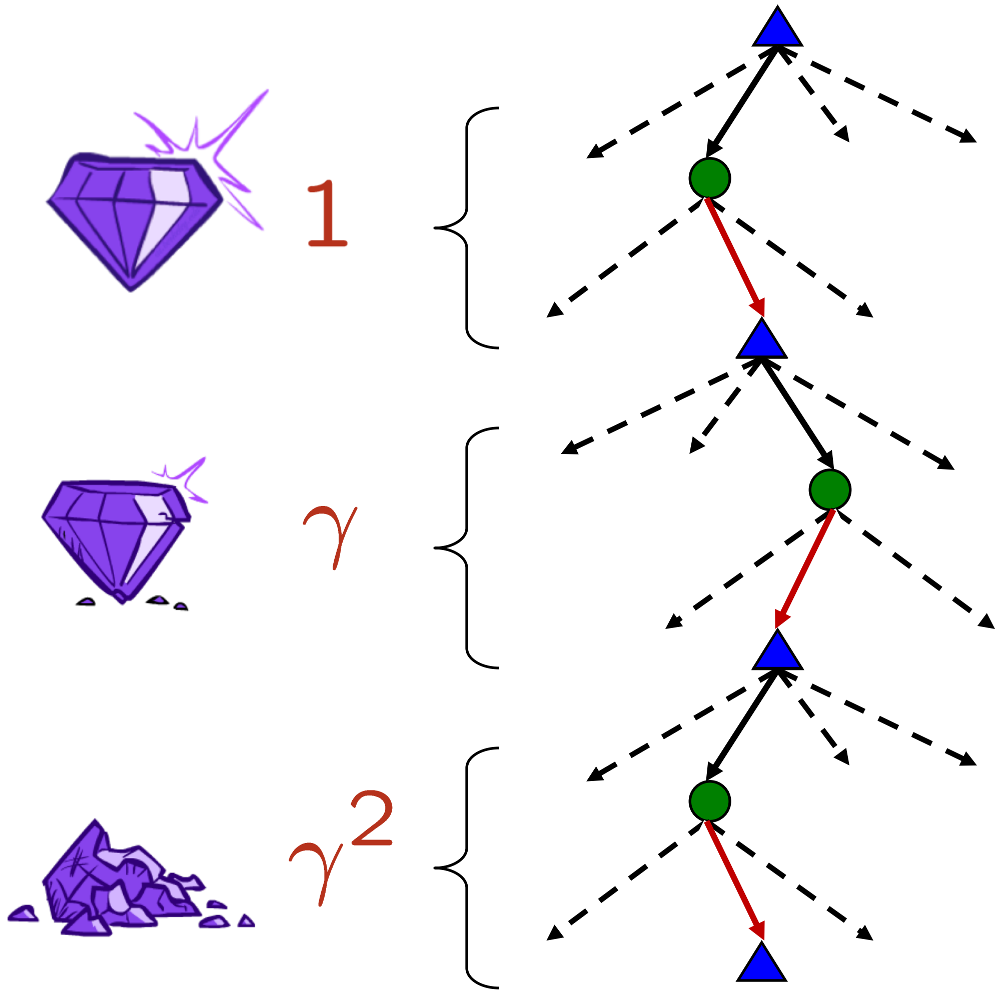
Worth γr next step

Worth $\gamma^2 r$ in two steps

- Discounting with γ conveniently solves the problem of infinite reward streams!
  - Geometric series: $1 + \gamma + \gamma^2 + \ldots = 1/(1 - \gamma)$
  - Assume rewards bounded by $\pm R_{max}$
  - Then $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$ is bounded by $\pm R_{max}/(1 - \gamma)$
- (Another solution: environment contains a ***terminal state***; ***and*** agent reaches it with probability 1)

Discounting

- How to discount?
  - Each time we descend a level, we multiply in the discount once
- Why discount?
  - Reward now is better than later
  - Can also think of it as a 1-gamma chance of ending the process at every step
  - Also helps our algorithms converge
- Example: discount of 0.5
  - U([1,2,3]) = 1*1 + 0.5*2 + 0.25*3
  - U([1,2,3]) < U([3,2,1])



Quiz: Discounting

- Given:

10				1
a	b	c	d	e

  - Actions: East, West, and Exit (only available in exit states a, e)
  - Transitions: deterministic

- Quiz 1: For $\gamma = 1$, what is the optimal policy?

10	<-	<-	<-	1

- Quiz 2: For $\gamma = 0.1$, what is the optimal policy?

10	<-	<-	->	1

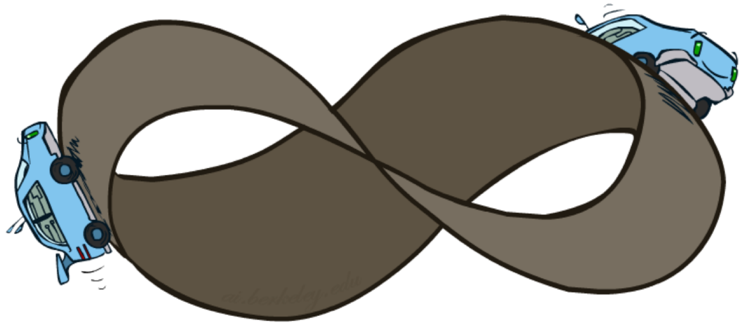
- Quiz 3: For which γ are West and East equally good when in state d?

$1\gamma=10\ \gamma^3$

Infinite Utilities?!

- Problem: What if the game lasts forever? Do we get infinite rewards?

- Solutions:
  - Finite horizon: (similar to depth-limited search)
    - Terminate episodes after a fixed T steps (e.g. life)
    - Gives nonstationary policies (π depends on time left)



- Discounting with γ solves the problem of infinite reward streams!
  - Geometric series: $1 + \gamma + \gamma^2 + \ldots = 1/(1 - \gamma)$
  - Assume rewards bounded by $\pm R_{max}$
  - Then $r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$ is bounded by $\pm R_{max}/(1 - \gamma)$

- Absorbing state: guarantee that for every policy, a terminal state will eventually be reached (like "overheated" for racing)