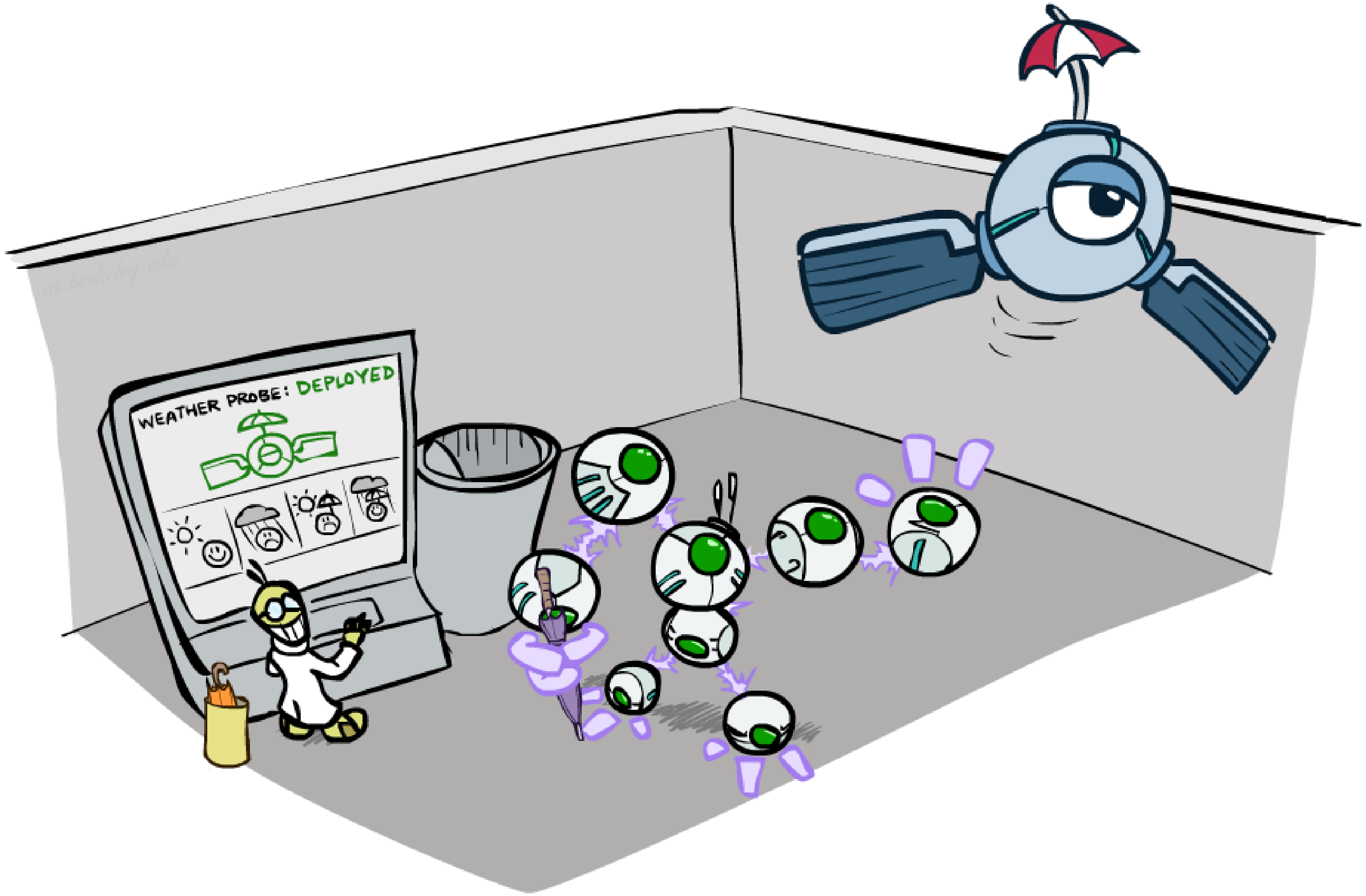
CS 188: Artificial Intelligence

Decision Networks and Value of Perfect Information (VPI)

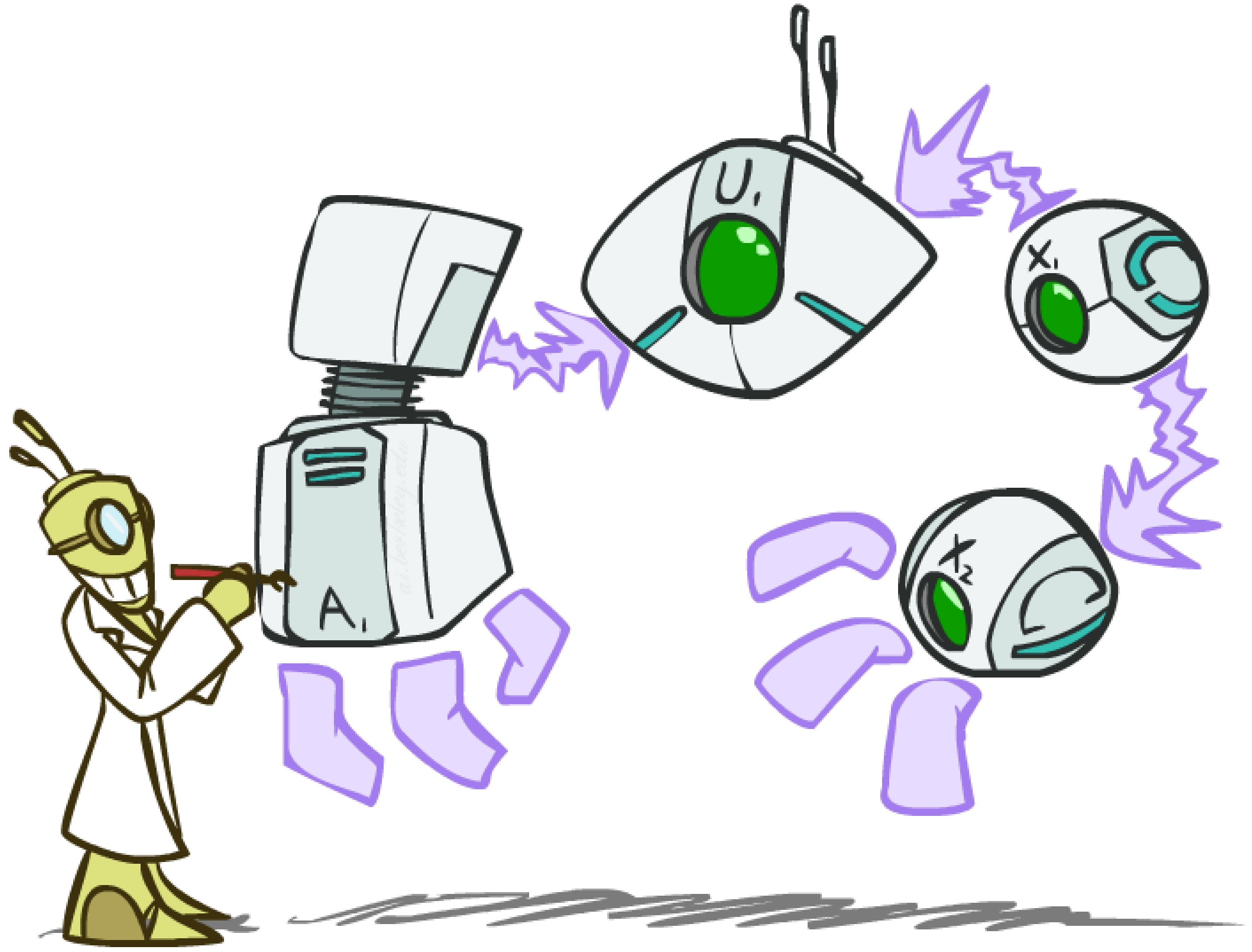


[Many of these slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley]

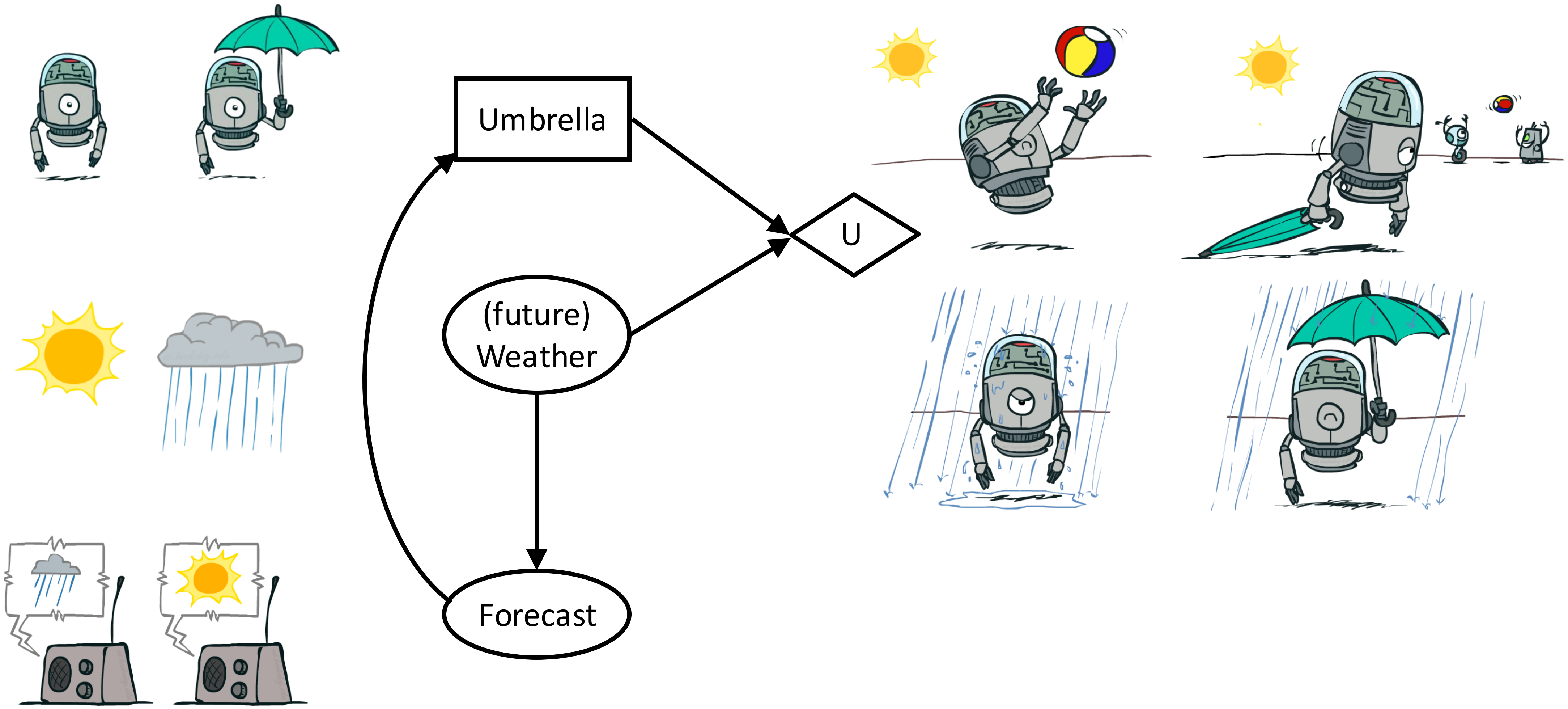
Announcements

- Midterm
  - Wednesday March 19, 7-9pm. You should have received an individual email with the location of your exam by last night. Material up to lecture 14 (Bayes Net independence)
  - Check Ed and Calendar for more midterm logistics/prep sessions, and see exam logistics page near top of course web site for more info.
- HW6 self-assessment
  - Due on Friday 3/21/25 at 11:59 PT

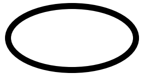
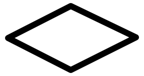
Decision Networks

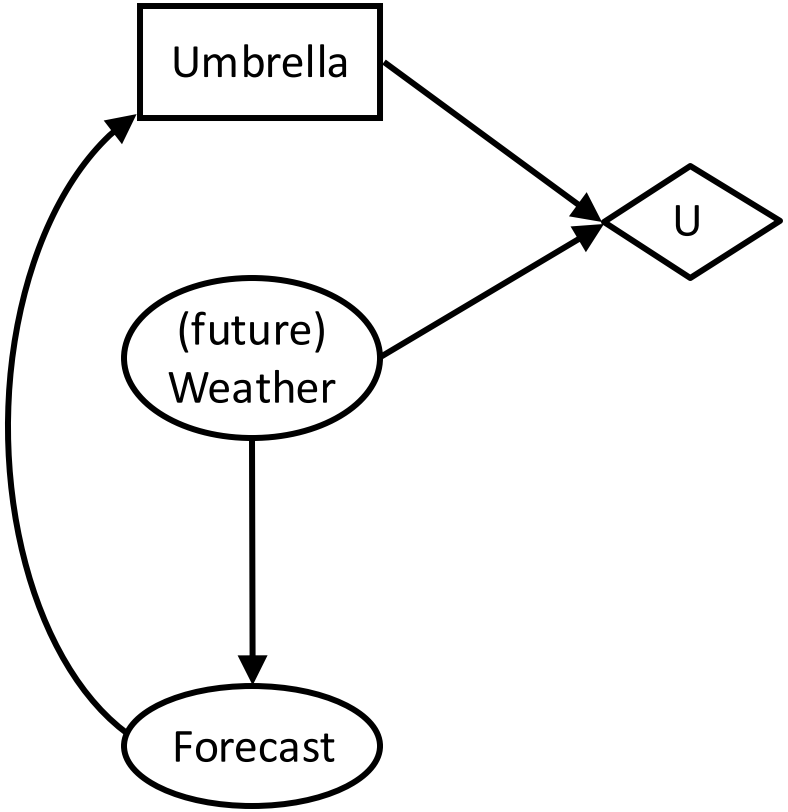


Decision Networks



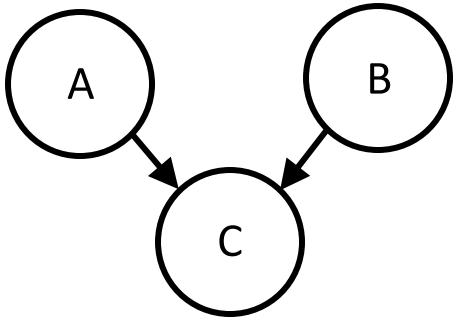
Decision Networks

- New node types:
  - Chance nodes (circular or oval, just like BNs) 
  - Actions (rectangles, like actions in MDPs) 
  - Utility node (diamond, like rewards in MDPs) 



Decision Networks: Chance Nodes

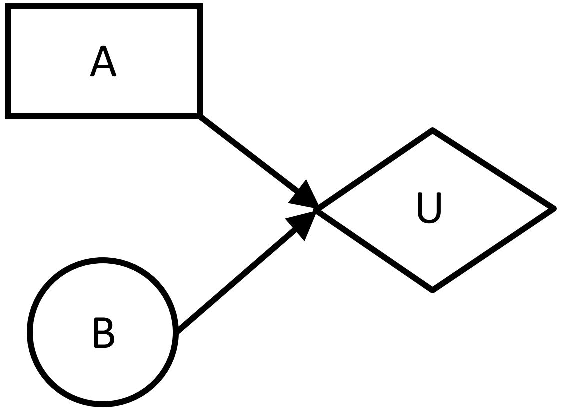
- Chance nodes (just like BNs)
  - Chance nodes represent random variables.
  - Arcs are directed, node probability is conditioned on parents (could be chance or action nodes).
  - Output is determined by a CPT (Conditional Probability Table):
  - Chance nodes can be either hidden or bound to evidence (they can't be queries since the network is built to compute utilities)



A	B	C	P(C|A,B)
+a	+b	+c	0.8
+a	+b	-c	0.2
+a	-b	+c	0.3
+a	-b	-c	0.7
-a	+b	+c	0.1
-a	+b	-c	0.9
-a	-b	+c	0.5
-a	-b	-c	0.5

Decision Networks: Utility Nodes

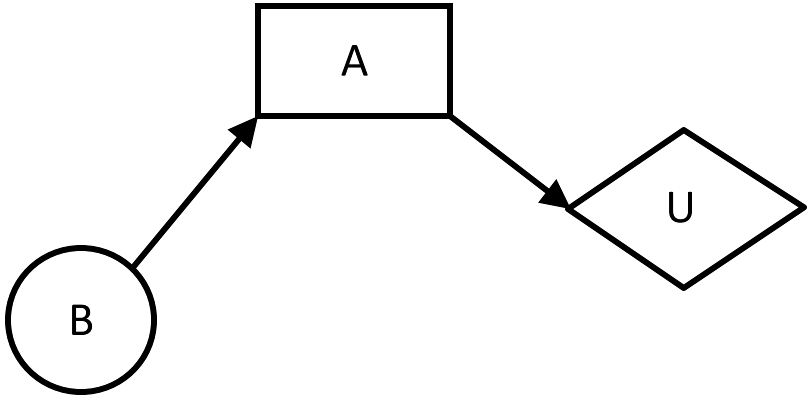
- Utility nodes (like rewards in MDPs)
  - Utility nodes represent the utility (like reward) of the network. Usually a real value.
  - Parents can be actions or chance nodes.
  - We can represent the utility node with a table if parents are discrete variables.
  - The utility function is given as part of the model (unless it has to be learned, but we will assume it is known).



A	B	U(A,B)
+a	+b	5
+a	-b	3.2
-a	+b	-1
-a	-b	-4

Decision Networks: Action Nodes

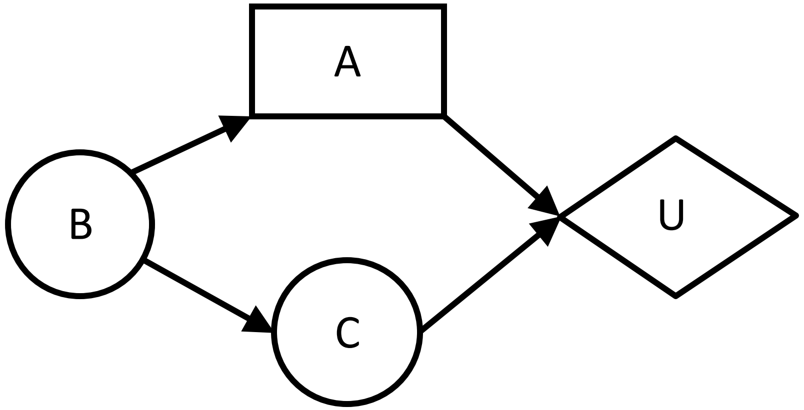
- Action nodes (like actions in MDPs)
  - Action nodes represent the action choices available to an agent.
  - Parents (if any) will be chance nodes.
  - We can represent the agent's action choices with a table if the variables are discrete.
  - The table initially only contains the possible action choices in each state.
  - After we do inference on the network, we can fill in the expected utilities of each state/action pair. This is like a Q-function.
  - We then choose the best action in each state, so the action node represents a policy.



B	A	EU(A,B)
+b	+a	??
+b	-a	??
-b	+a	??
-b	-a	??

Maximum Expected Utility (MEU)

- For each state B and action A, we compute expected utility by averaging over C, conditioned on B.
- Then we take the maximum over A for each B.
- Finally we can average over B to get the MEU.



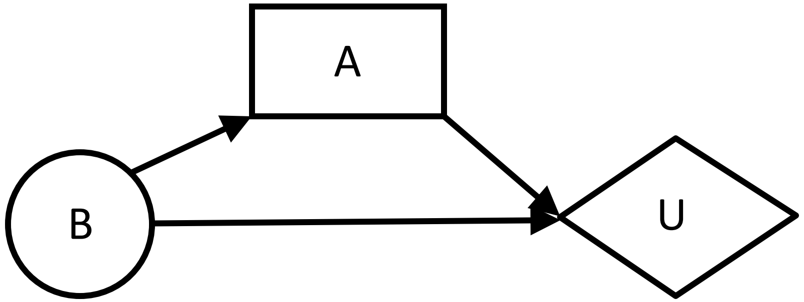
B	A	C	U(A,C)
+b	+a	+c	5
+b	+a	-c	3.2
+b	-a	+c	-1
+b	-a	-c	-4
…	…	…	…

B	A	U(A,B)
+b	+a	4.1
+b	-a	-2.3
-b	+a	0.4
-b	-a	-1.5

Compute expected utility over C | B

Maximum Expected Utility (MEU)

- For each state B and action A, we compute expected utility by averaging over C, conditioned on B.
- Then we take the maximum over A for each B.
- Finally we can average over B to get the MEU.



B	A	C	U(A,C)
+b	+a	+c	5
+b	+a	-c	3.2
+b	-a	+c	-1
+b	-a	-c	-4
…	…	…	…

B	A	U(A,B)
+b	+a	4.1
+b	-a	-2.3
-b	+a	0.4
-b	-a	-1.5

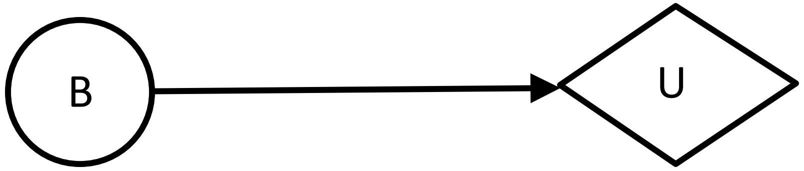
B	MEU(B)
+b	4.1
-b	0.4

Choose best A | B

Compute expected utility over C | B

Maximum Expected Utility (MEU)

- For each state B and action A, we compute expected utility by averaging over C, conditioned on B.
- Then we take the maximum over A for each B.
- Finally we can average over B to get the MEU.



B	A	C	U(A,C)
+b	+a	+c	5
+b	+a	-c	3.2
+b	-a	+c	-1
+b	-a	-c	-4
…	…	…	…

Compute expected utility over C | B

B	A	U(A,B)
+b	+a	4.1
+b	-a	-2.3
-b	+a	0.4
-b	-a	-1.5

Choose best A | B

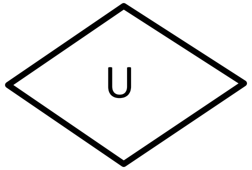
B	MEU(B)
+b	4.1
-b	0.4

Compute expected utility over B

MEU = 3.4

Maximum Expected Utility (MEU)

- For each state B and action A, we compute expected utility by averaging over C, conditioned on B.
- Then we take the maximum over A for each B.
- Finally we can average over B to get the MEU.



B	A	C	U(A,C)
+b	+a	+c	5
+b	+a	-c	3.2
+b	-a	+c	-1
+b	-a	-c	-4
…	…	…	…

Compute expected utility over C | B

B	A	U(A,B)
+b	+a	4.1
+b	-a	-2.3
-b	+a	0.4
-b	-a	-1.5

Choose best A | B

B	MEU(B)
+b	4.1
-b	0.4

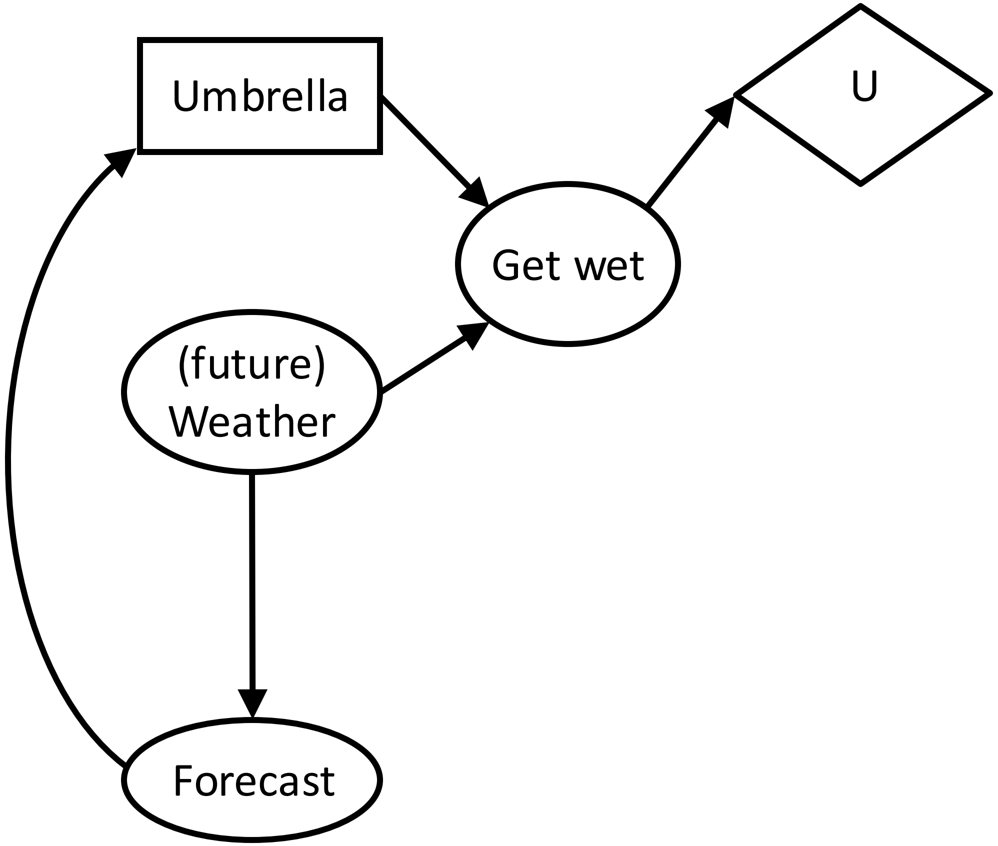
Compute expected utility over B

MEU = 3.4

Decision Networks: Simplification

We'll assume action nodes connect directly to the utility node.

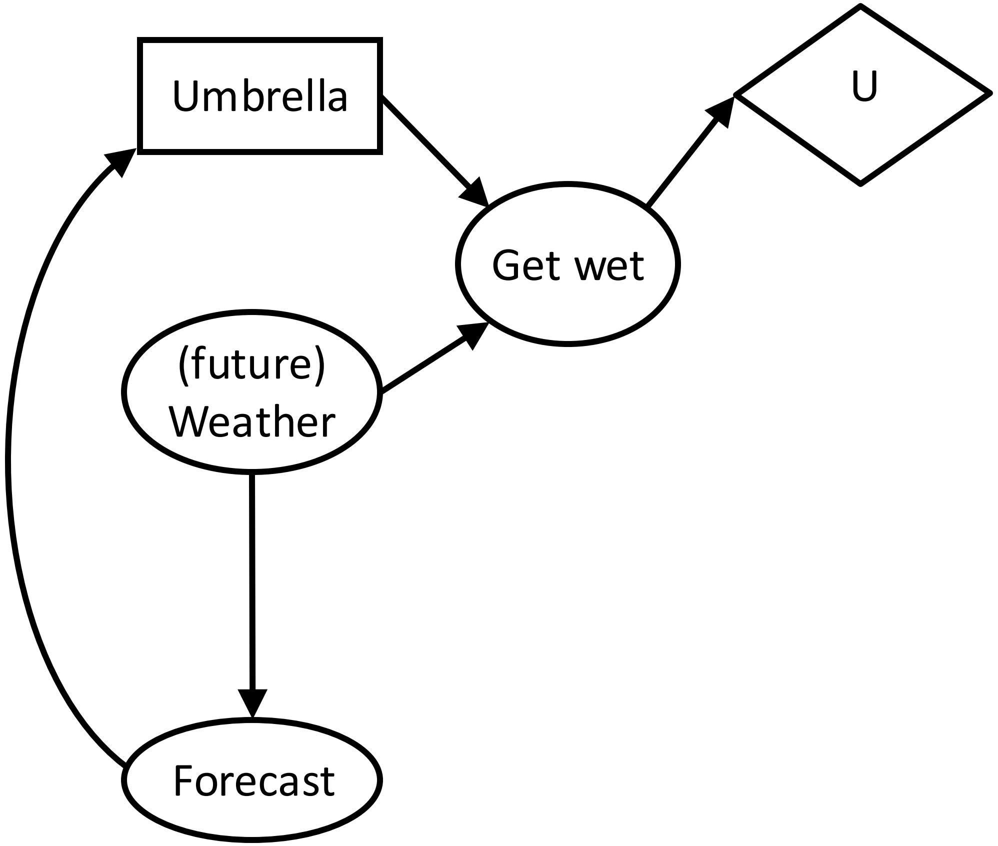
- A more detailed picture could look like this:



Decision Networks: Simplification

We'll assume action nodes connect directly to the utility node.

- A more detailed picture could look like this:
- But we could always average out (eliminate) the intermediate nodes.

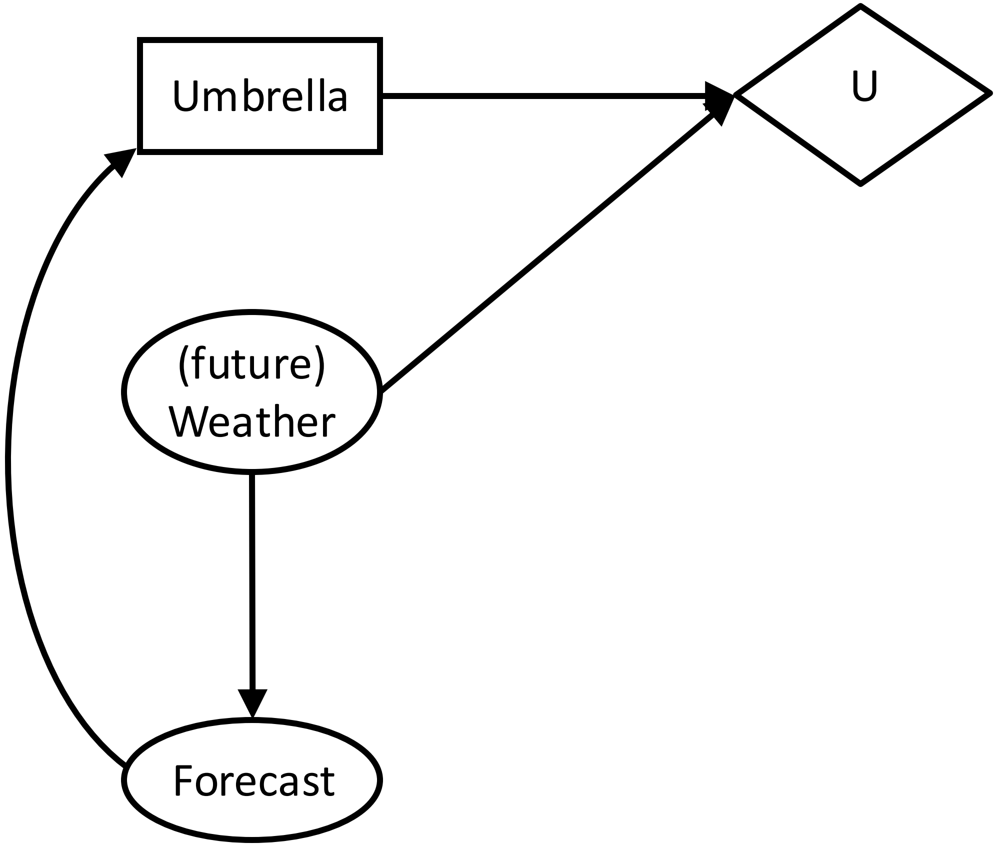


Decision Networks: Simplification

We’ll assume action nodes connect directly to the utility node.

- A more detailed picture could look like this:

- But we could always average out (eliminate) the intermediate nodes.



Decision Networks (no information)

Umbrella = leave

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

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

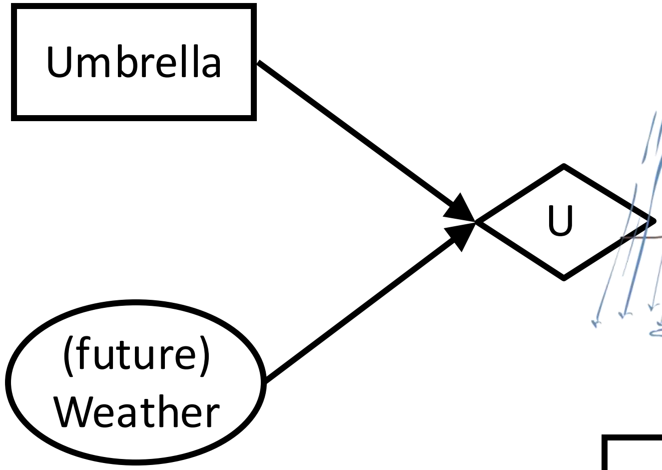
Umbrella = take

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

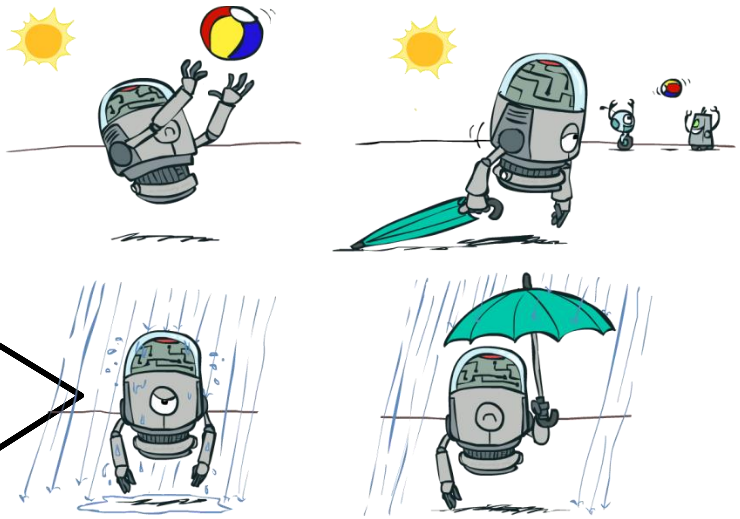
$$= 0.7 \cdot 20 + 0.3 \cdot 70 = 35$$

Optimal decision = leave

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

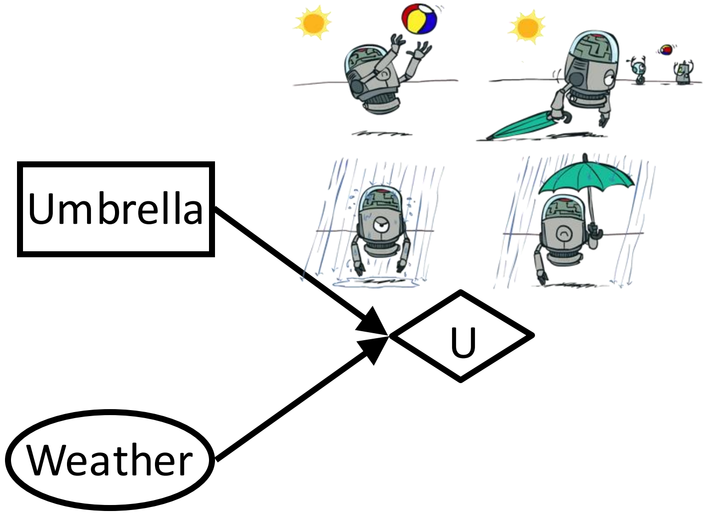


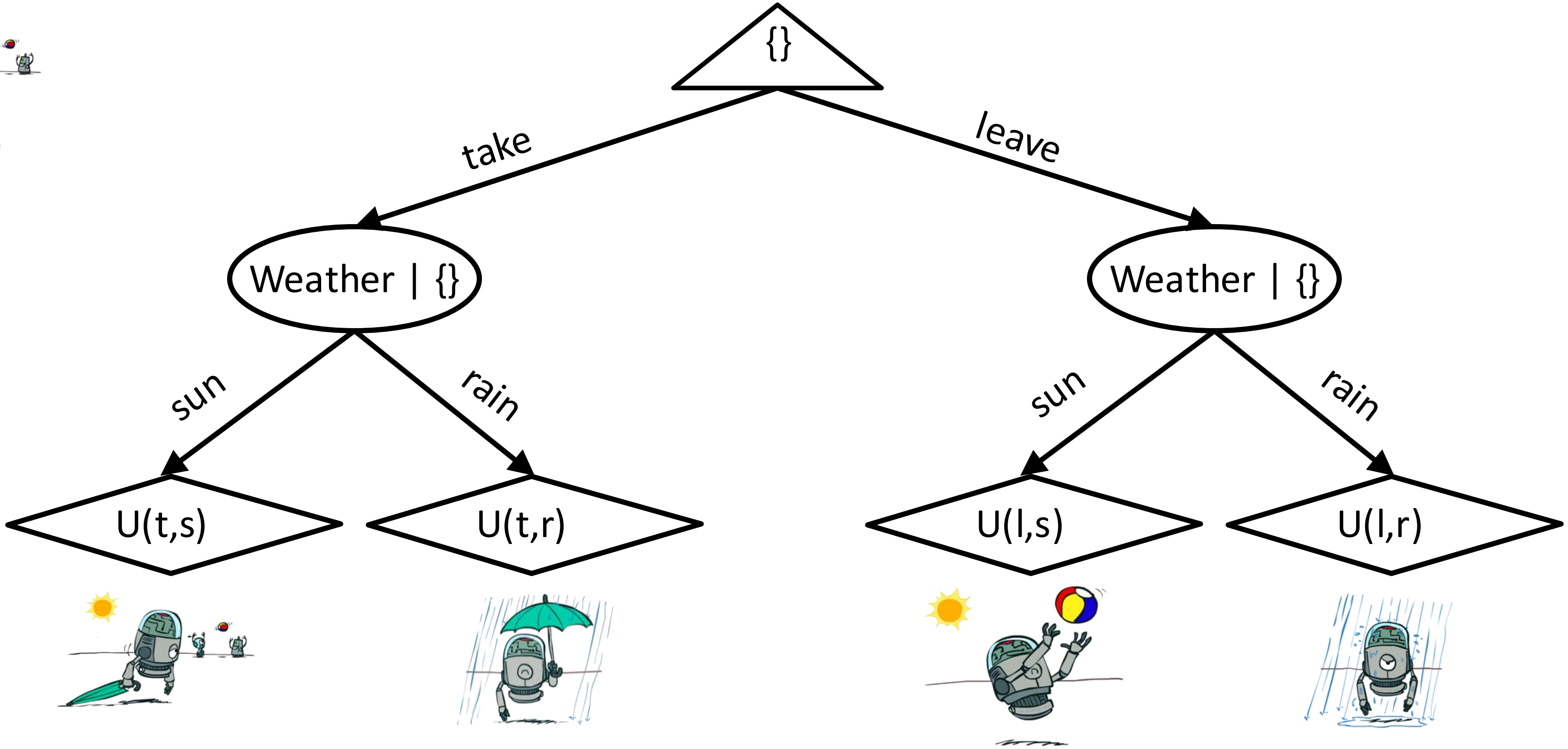
W	P(W)
sun	0.7
rain	0.3

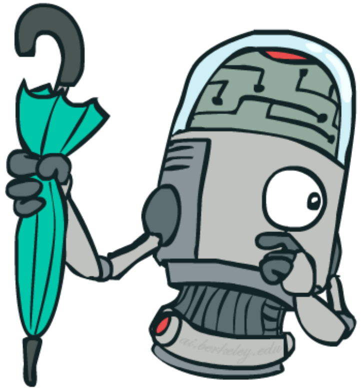


A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

Decisions as Outcome Trees

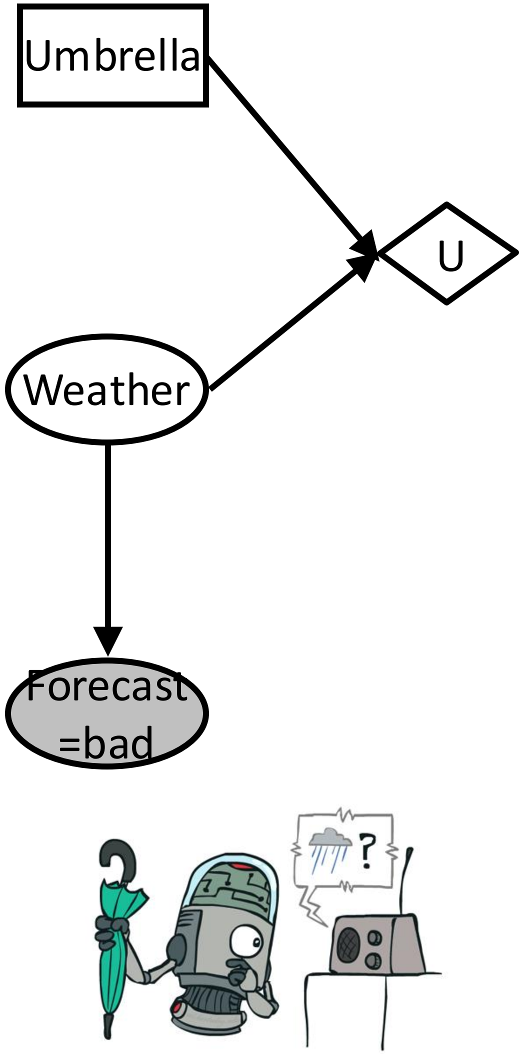


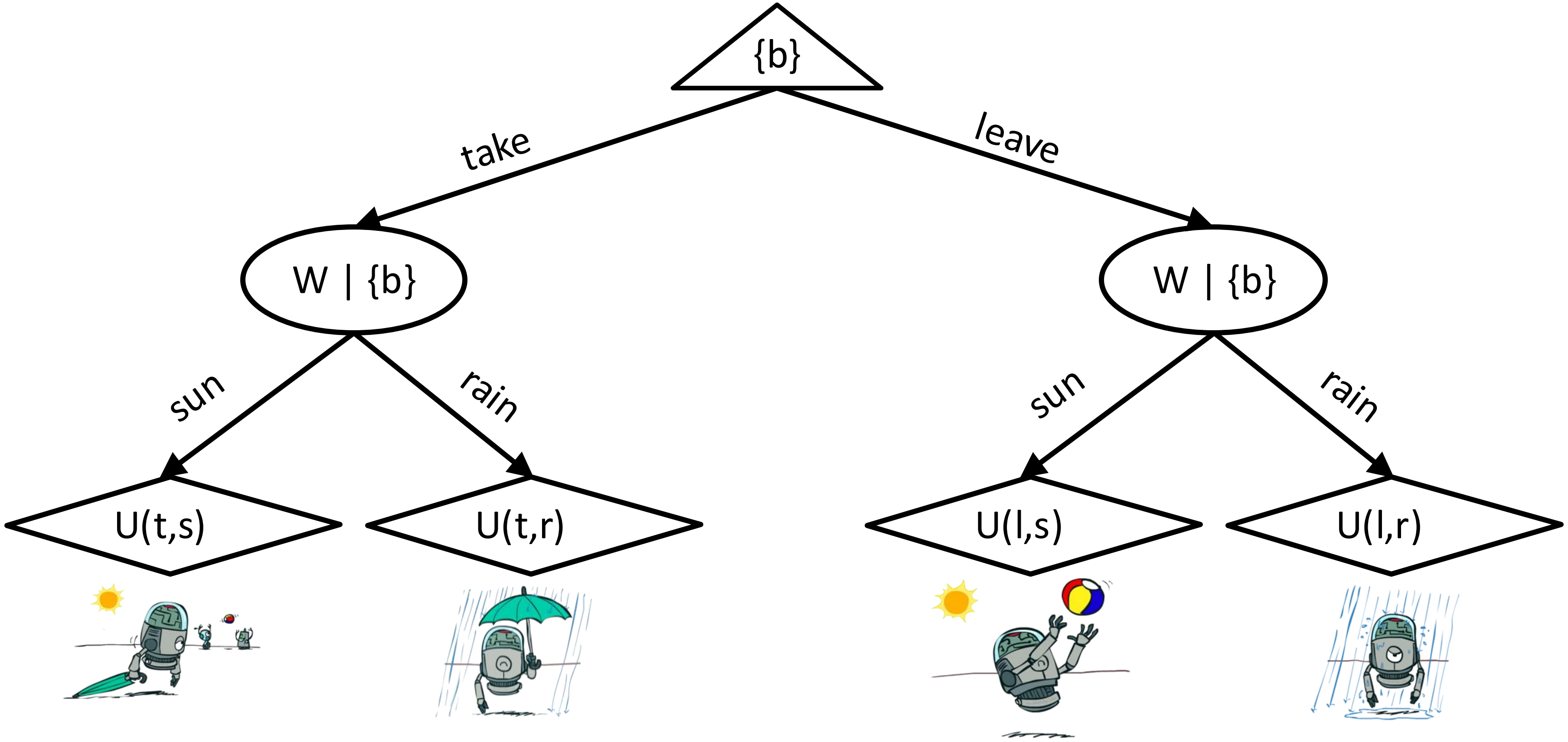




- Almost exactly like expectimax / MDPs
- What's changed?

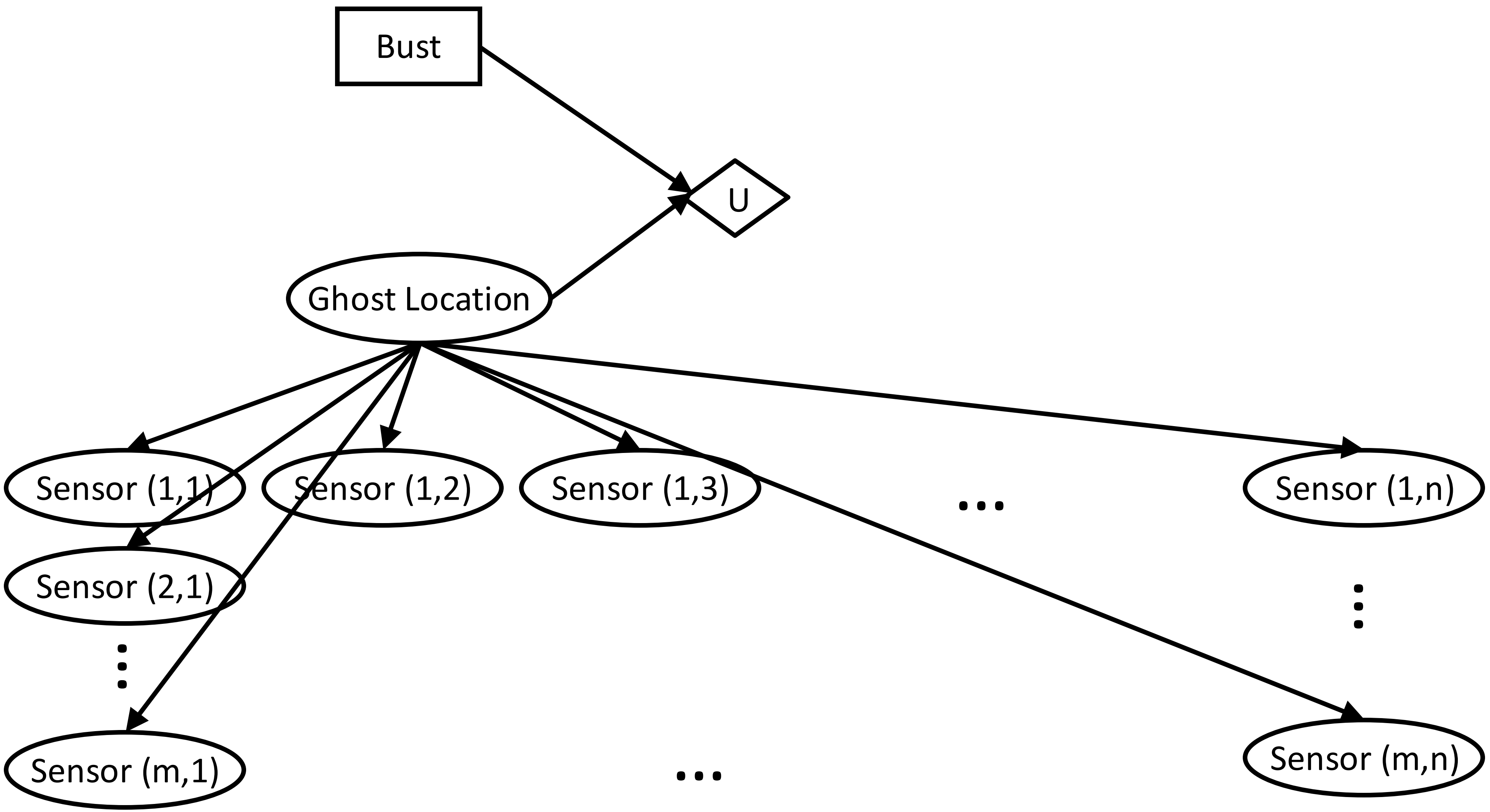
Decisions as Outcome Trees



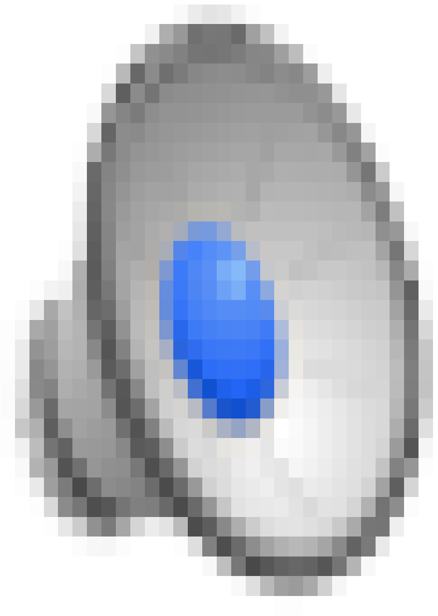


Ghostbusters Decision Network

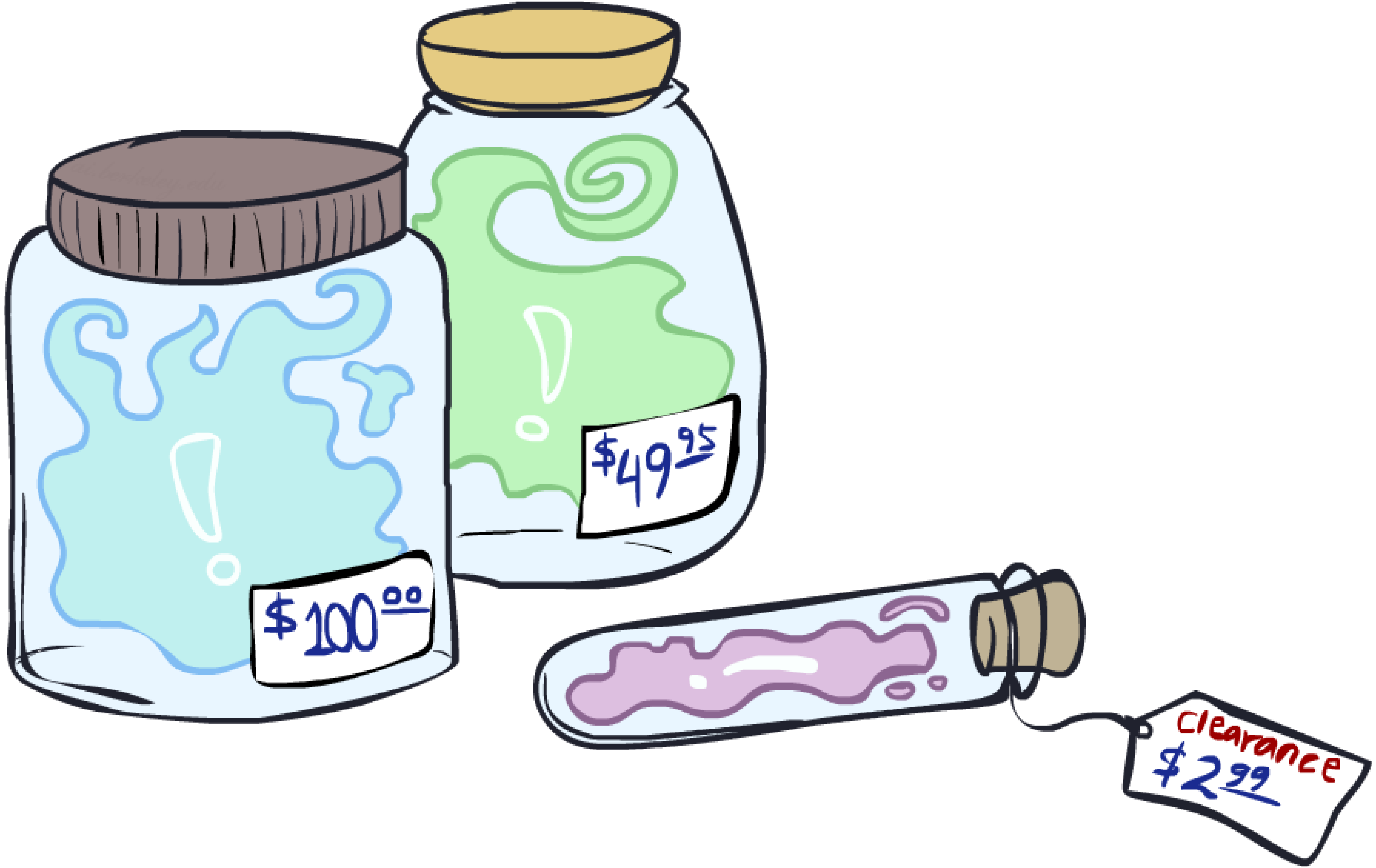
Demo: Ghostbusters with probability



Video of Demo Ghostbusters with Probability

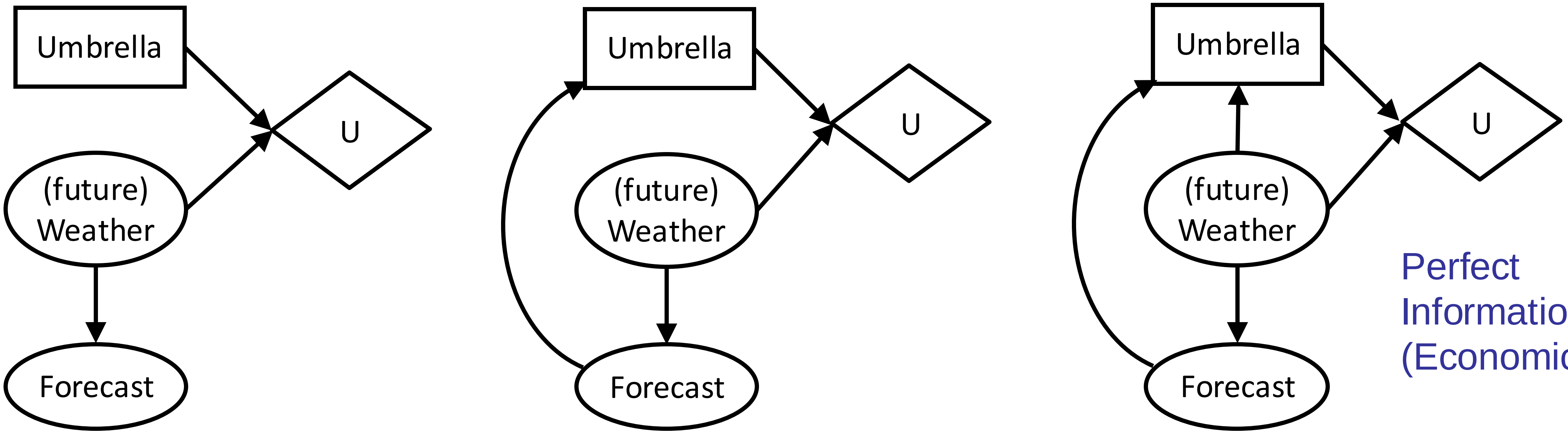


Value of Information



Value of Information

- Value of Information is the difference in MEU between networks with different action conditioning (information).



Perfect Information (Economics)

$MEU(\emptyset)$ $\leq MEU(F)$ $\leq MEU(F,W)$

Difference = $VoI(F)$

Difference = $VoI(W|F)$

Difference = $VoI(F,W)$

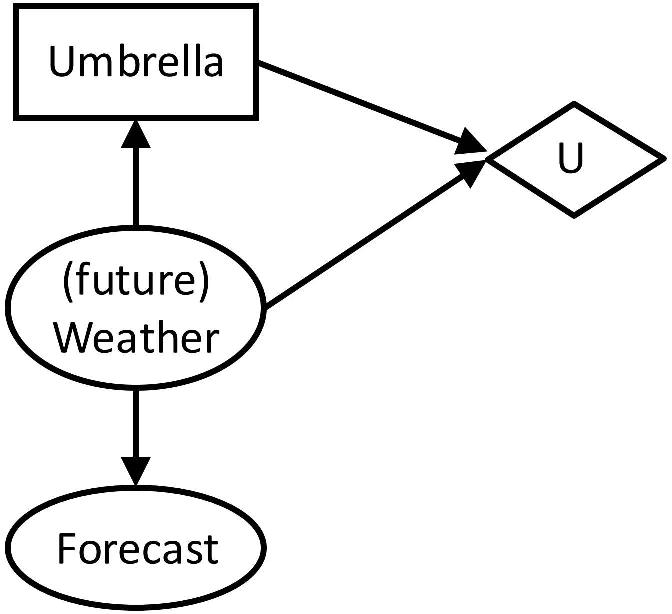
Value of Perfect Information: Weather

MEU with no information (earlier slide)

$$MEU(\emptyset) = 70$$

Note: Info from forecast is not needed

W	P(W)
sun	0.7
rain	0.3



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

W	A	U
sun	leave	100
sun	take	20
rain	leave	0
rain	take	70

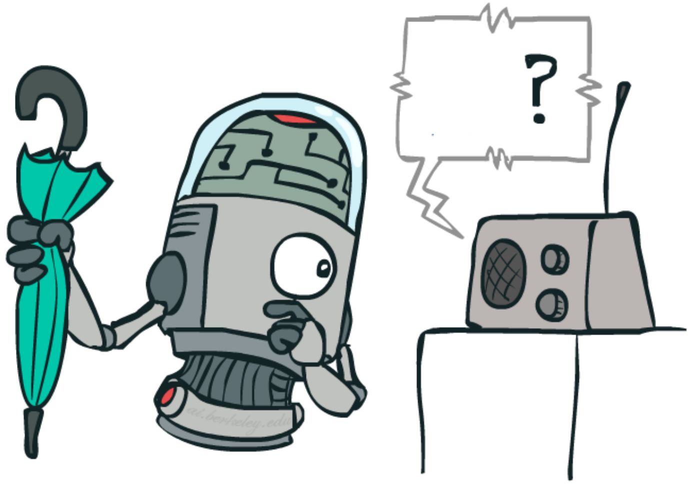
W	A	U	P(W)
sun	leave	100	0.7
rain	take	70	0.3

Choose best action
add P(W) column

Compute Expected U
over W

MEU(W) = 91

$$VPI = MEU(W) - MEU(\emptyset) = 91 - 70 = 21$$



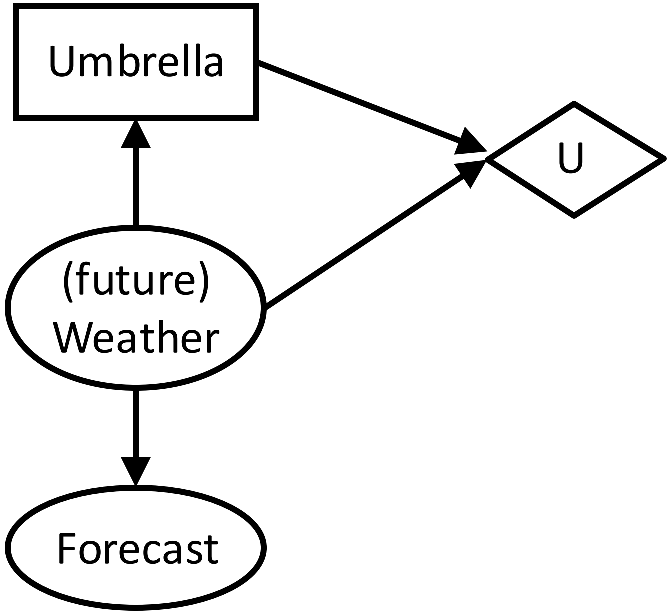
Value of Perfect Information: Weather

MEU with no information (earlier slide)

$$MEU(\emptyset) = 70$$

Note: Info from forecast is not needed

W	P(W)
sun	0.7
rain	0.3



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

W	A	U
sun	leave	100
sun	take	20
rain	leave	0
rain	take	70

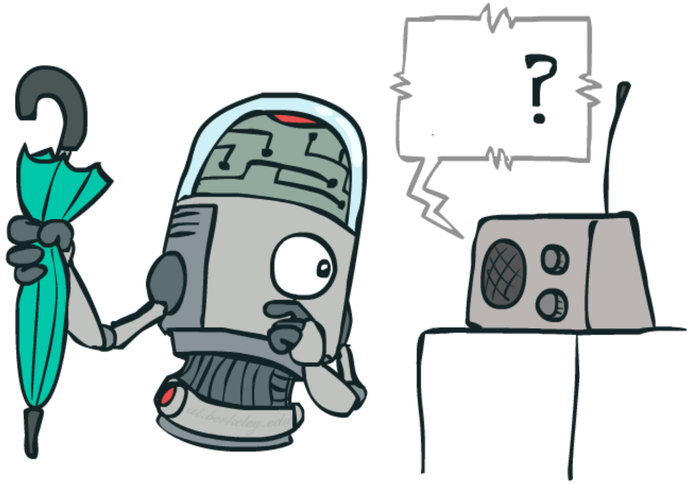
W	A	U	P(W)
sun	leave	100	0.7
rain	take	70	0.3

Choose best action
add P(W) column

Compute Expected U
over W

MEU(W) = 91

$$VPI = MEU(W) - MEU(\emptyset) = 91 - 70 = 21$$



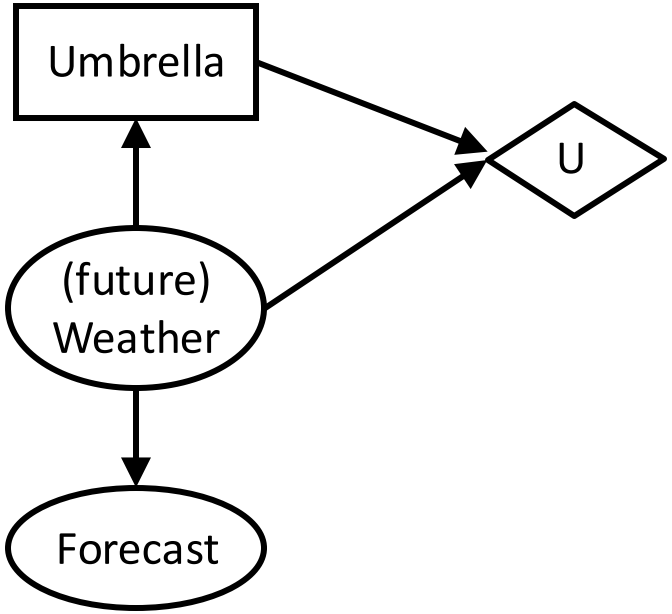
Value of Perfect Information: Weather

MEU with no information (earlier slide)

$$MEU(\emptyset) = 70$$

Note: Info from forecast is not needed

W	P(W)
sun	0.7
rain	0.3



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

W	A	U
sun	leave	100
sun	take	20
rain	leave	0
rain	take	70

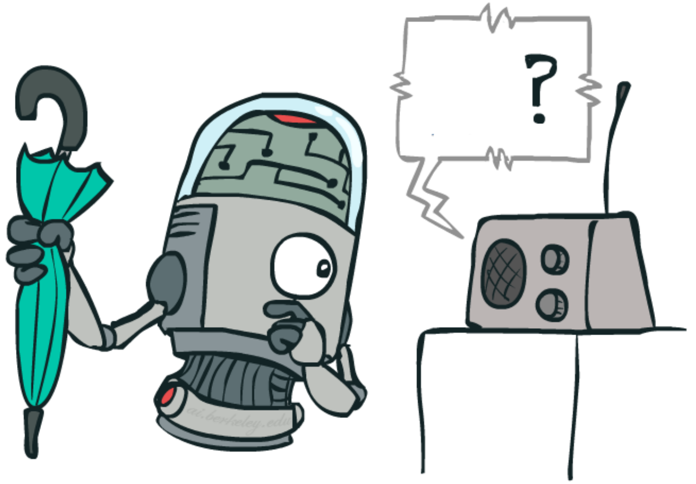
W	A	U	P(W)
sun	leave	100	0.7
rain	take	70	0.3

Choose best action
add P(W) column

Compute Expected U
over W

MEU(W) = 91

$$VPI = MEU(W) - MEU(\emptyset) = 91 - 70 = 21$$



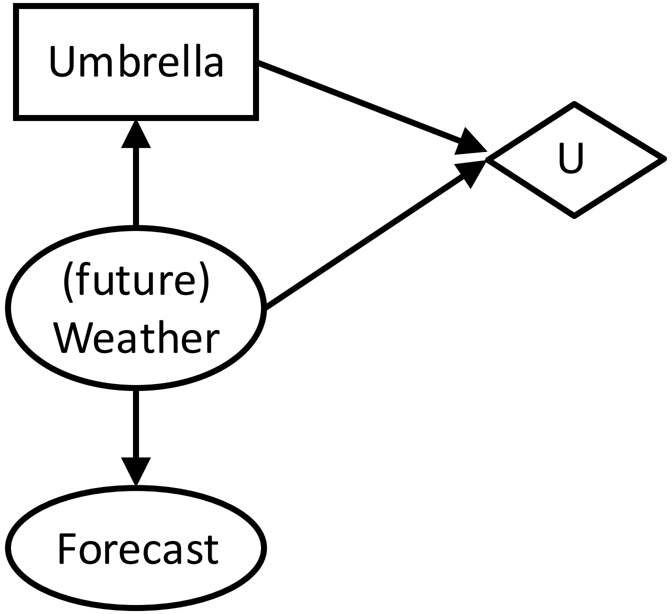
Value of Perfect Information: Weather

MEU with no information (earlier slide)

$$MEU(\emptyset) = 70$$

Note: Info from forecast is not needed if an oracle gives us the future weather.

W	P(W)
sun	0.7
rain	0.3



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

W	A	U
sun	leave	100
sun	take	20
rain	leave	0
rain	take	70

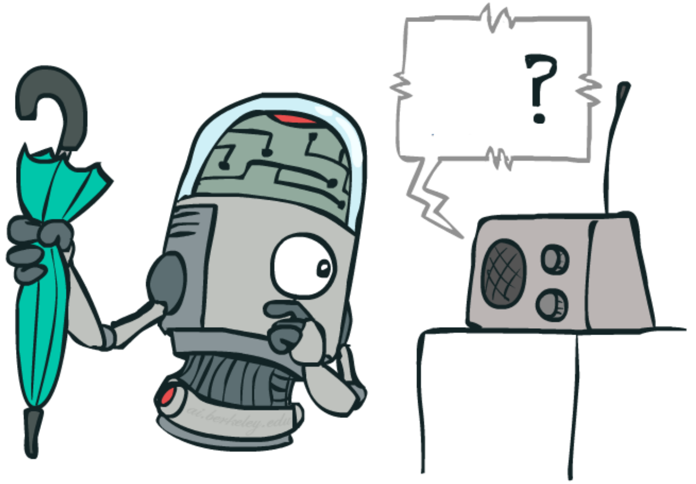
W	A	U	P(W)
sun	leave	100	0.7
rain	take	70	0.3

Choose best action
add P(W) column

Compute Expected U
over W

MEU(W) = 91

$$VPI = MEU(W) - MEU(\emptyset) = 91 - 70 = 21$$



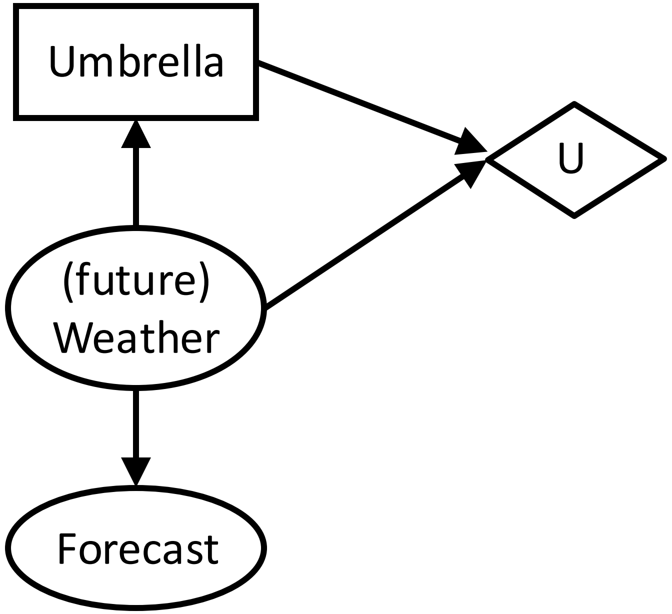
Value of Perfect Information: Weather

MEU with no information (earlier slide)

$$MEU(\emptyset) = 70$$

Note: Info from forecast is not needed

W	P(W)
sun	0.7
rain	0.3



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

W	A	U
sun	leave	100
sun	take	20
rain	leave	0
rain	take	70

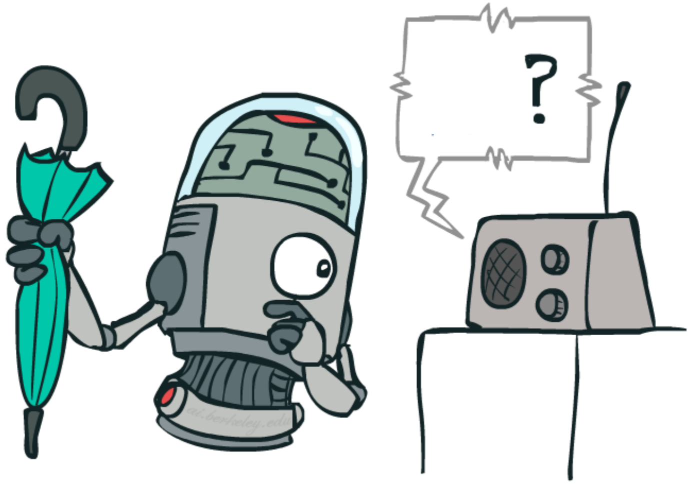
W	A	U	P(W)
sun	leave	100	0.7
rain	take	70	0.3

Choose best action
add P(W) column

Compute Expected U
over W

MEU(W) = 91

$$VPI = MEU(W) - MEU(\emptyset) = 91 - 70 = 21$$



Value of State Information: Weather

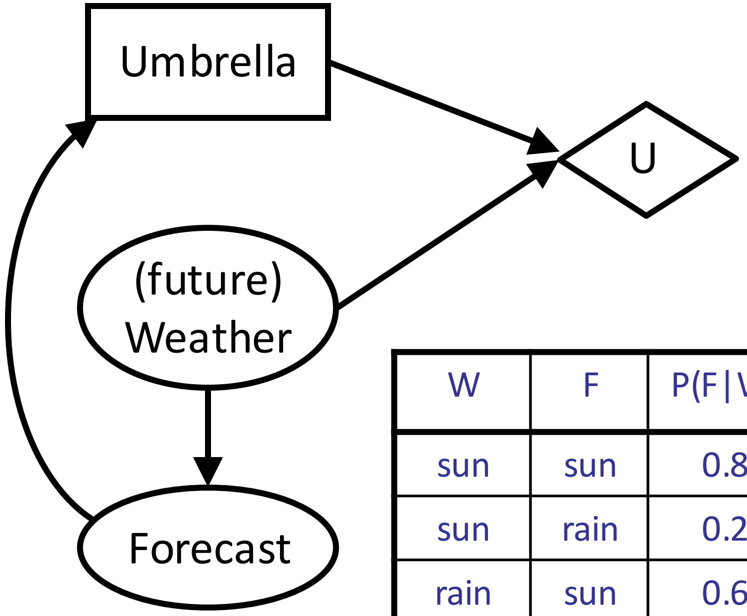
VoI(F) for the Forecast node

Join the U and F tables on W:

W	F	A	U
sun	sun	leave	100
sun	sun	take	20
sun	rain	leave	100
sun	rain	take	20
rain	sun	leave	0
rain	sun	take	70
rain	rain	leave	0
rain	rain	take	70

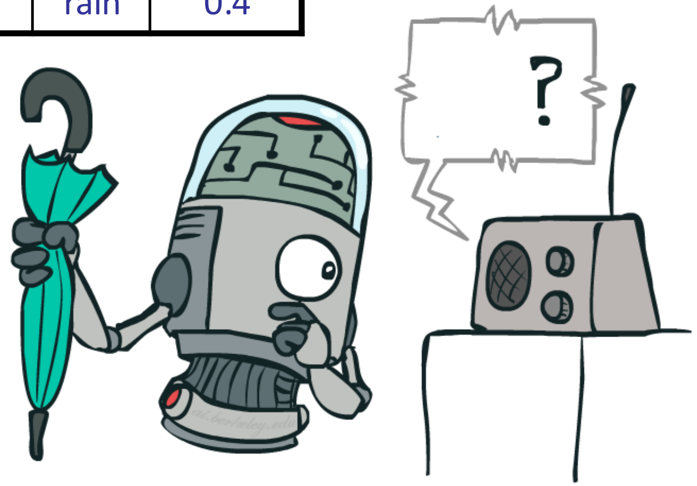
Let’s eliminate W

W	P(W)
sun	0.7
rain	0.3



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

W	F	P(F|W)
sun	sun	0.8
sun	rain	0.2
rain	sun	0.6
rain	rain	0.4



Value of State Information: Weather

Let's eliminate W. We need $P(W|F)$, using Bayes

W	P(W)
sun	0.7
rain	0.3

W	F	P(F|W)
sun	sun	0.8
sun	rain	0.2
rain	sun	0.3
rain	rain	0.7

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

W	F	P(W|F)
sun	sun	0.86
sun	rain	0.4
rain	sun	0.14
rain	rain	0.6

join

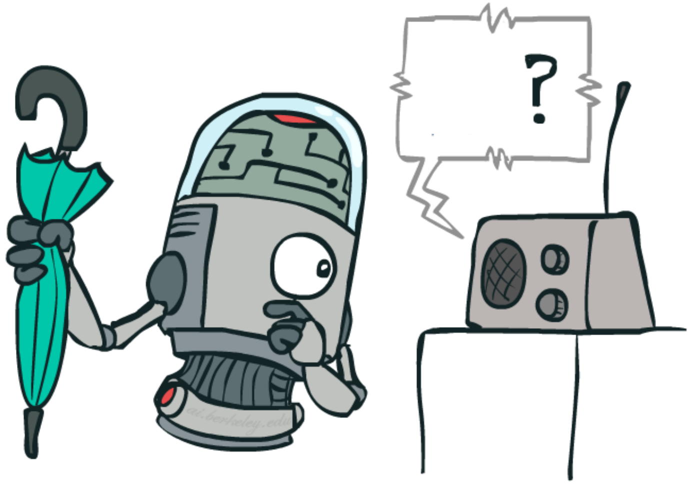
W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

F	A	E(U)
sun	leave	86
rain	take	50

Take the best action

Expected Utility (over W)



Value of State Information: Weather

Let's eliminate W. We need $P(W|F)$, using Bayes

W	P(W)
sun	0.7
rain	0.3

W	F	P(F|W)
sun	sun	0.8
sun	rain	0.2
rain	sun	0.3
rain	rain	0.7

join

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

W	F	P(W|F)
sun	sun	0.86
sun	rain	0.4
rain	sun	0.14
rain	rain	0.6

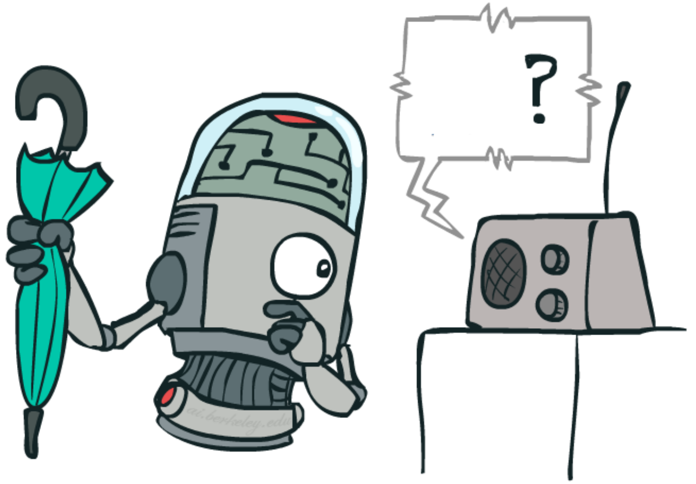
W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)
sun	leave	86
rain	take	50



Value of State Information: Weather

Let's eliminate W. We need $P(W|F)$, using Bayes

W	P(W)
sun	0.7
rain	0.3

W	F	P(F|W)
sun	sun	0.8
sun	rain	0.2
rain	sun	0.3
rain	rain	0.7

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

W	F	P(W|F)
sun	sun	0.86
sun	rain	0.4
rain	sun	0.14
rain	rain	0.6

normalize

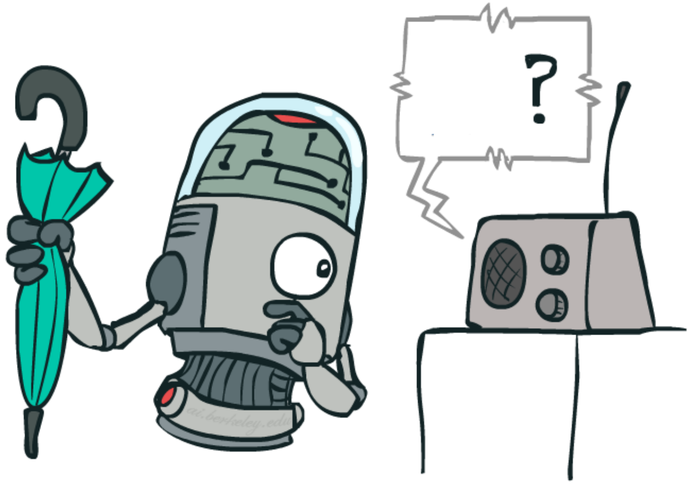
W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

F	A	E(U)
sun	leave	86
rain	take	50

Take the best action

Expected Utility (over W)



Value of State Information: Weather

Let's eliminate W. We need $P(W|F)$, using Bayes

W	P(W)
sun	0.7
rain	0.3

W	F	P(F|W)
sun	sun	0.8
sun	rain	0.2
rain	sun	0.3
rain	rain	0.7

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

W	F	P(W|F)
sun	sun	0.86
sun	rain	0.4
rain	sun	0.14
rain	rain	0.6

normalize

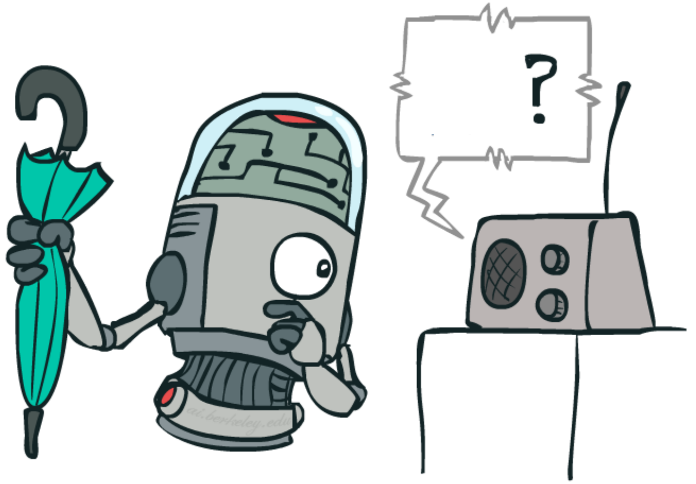
W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)
sun	leave	86
rain	take	50



Value of State Information: Weather

Let's eliminate W. We need $P(W|F)$, using Bayes

W	P(W)
sun	0.7
rain	0.3

W	F	P(F|W)
sun	sun	0.8
sun	rain	0.2
rain	sun	0.3
rain	rain	0.7

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

W	F	P(W|F)
sun	sun	0.86
sun	rain	0.4
rain	sun	0.14
rain	rain	0.6

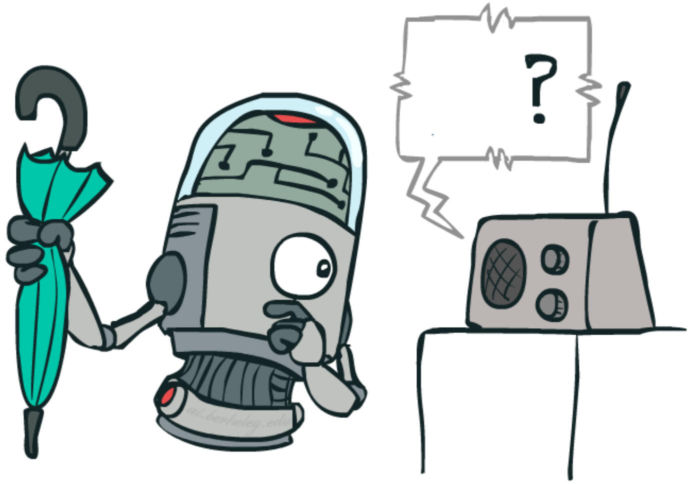
W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)
sun	leave	86
rain	take	50



Value of State Information: Weather

Let's eliminate W. We need $P(W|F)$, using Bayes

W	P(W)
sun	0.7
rain	0.3

W	F	P(F|W)
sun	sun	0.8
sun	rain	0.2
rain	sun	0.3
rain	rain	0.7

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

W	F	P(W|F)
sun	sun	0.86
sun	rain	0.4
rain	sun	0.14
rain	rain	0.6

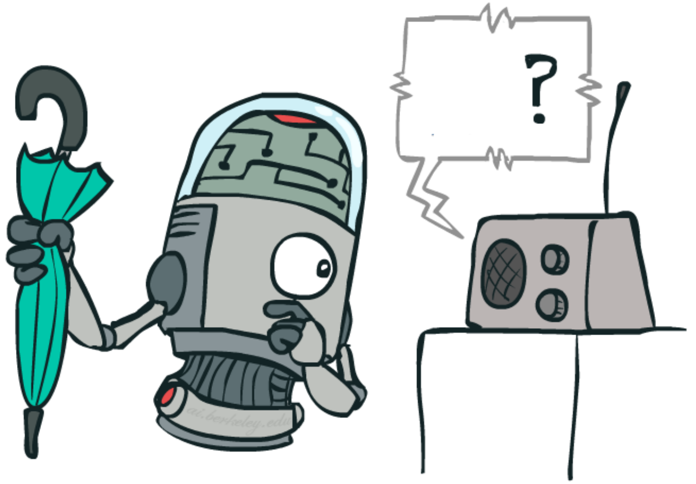
W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)
sun	leave	86
rain	take	50



Value of State Information: Weather

Let's eliminate W. We need $P(W|F)$, using Bayes

W	P(W)
sun	0.7
rain	0.3

W	F	P(F|W)
sun	sun	0.8
sun	rain	0.2
rain	sun	0.3
rain	rain	0.7

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

W	F	P(W|F)
sun	sun	0.86
sun	rain	0.4
rain	sun	0.14
rain	rain	0.6

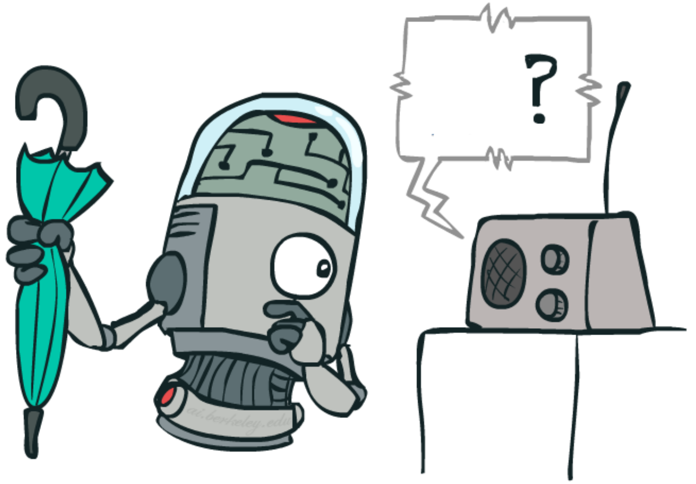
W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)
sun	leave	86
rain	take	50



Value of State Information: Weather

W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

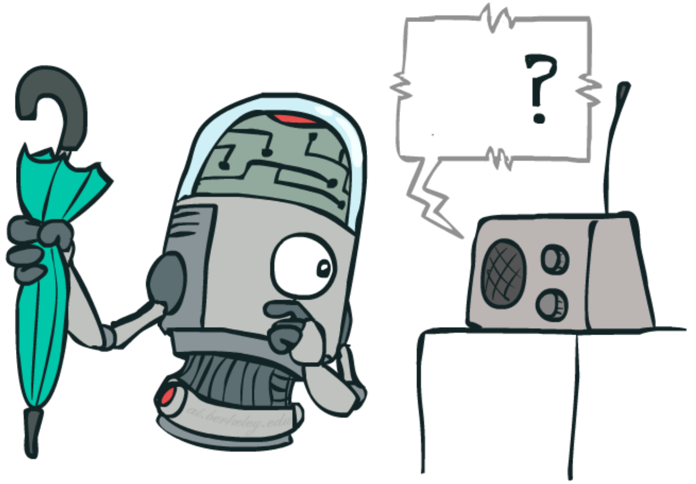
Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)	P(F)
sun	leave	86	0.65
rain	take	50	0.35

= 73.4



Value of State Information: Weather

W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

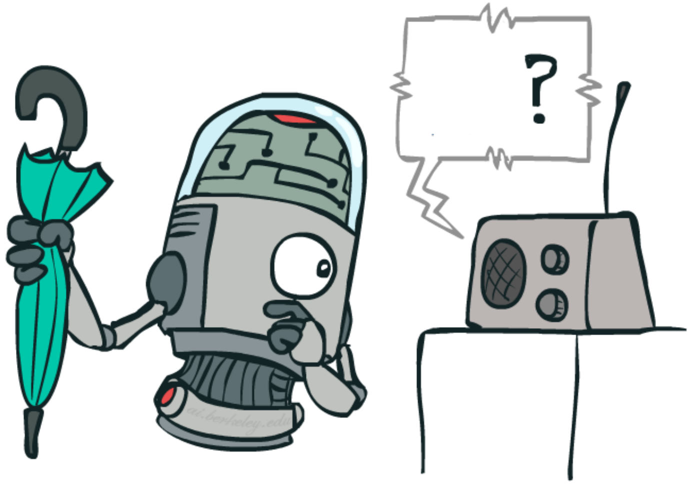
Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)	P(F)
sun	leave	86	0.65
rain	take	50	0.35

= 73.4



Value of State Information: Weather

W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

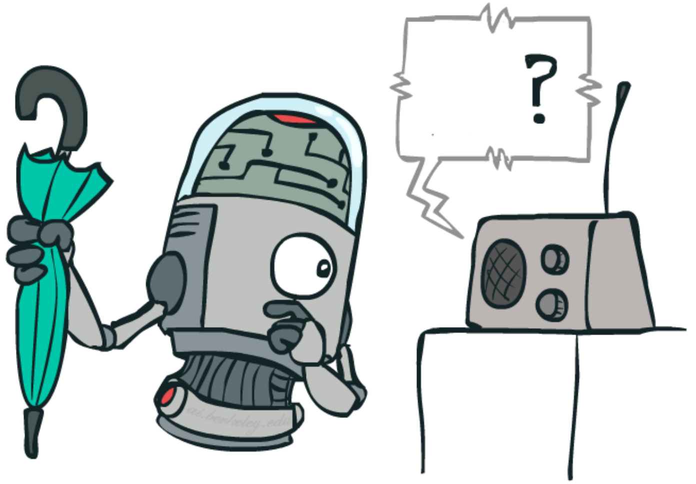
Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)	P(F)
sun	leave	86	0.65
rain	take	50	0.35

= 73.4



Value of State Information: Weather

We just need the marginal P(F) to finish

And we can use the joint table from Bayes slide:

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

F	P(F)
sun	0.65
rain	0.35

W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

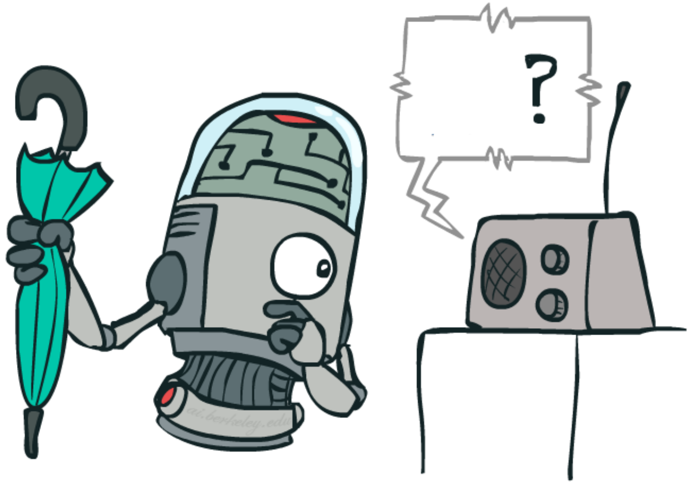
F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

F	A	E(U)	P(F)
sun	leave	86	0.65
rain	take	50	0.35

= 73.4

Take the best action

Expected Utility (over W)



Value of State Information: Weather

We just need the marginal P(F) to finish

And we can use the joint table from Bayes slide:

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

F	P(F)
sun	0.65
rain	0.35

W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

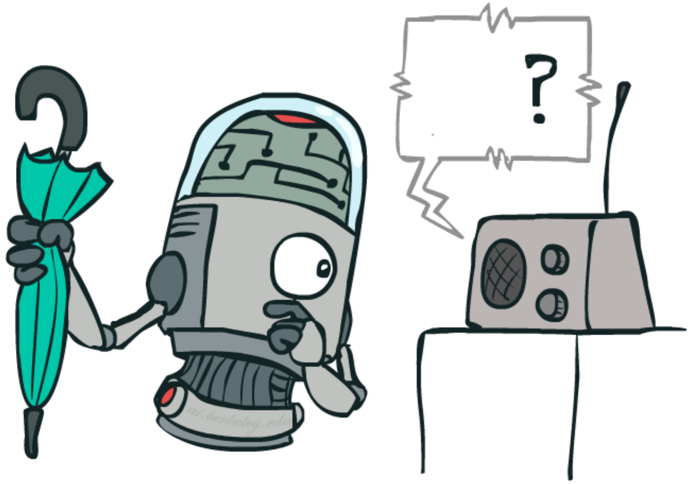
Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)	P(F)
sun	leave	86	0.65
rain	take	50	0.35

= 73.4



Value of State Information: Weather

We just need the marginal P(F) to finish

And we can use the joint table from Bayes slide:

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

F	P(F)
sun	0.65
rain	0.35

W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

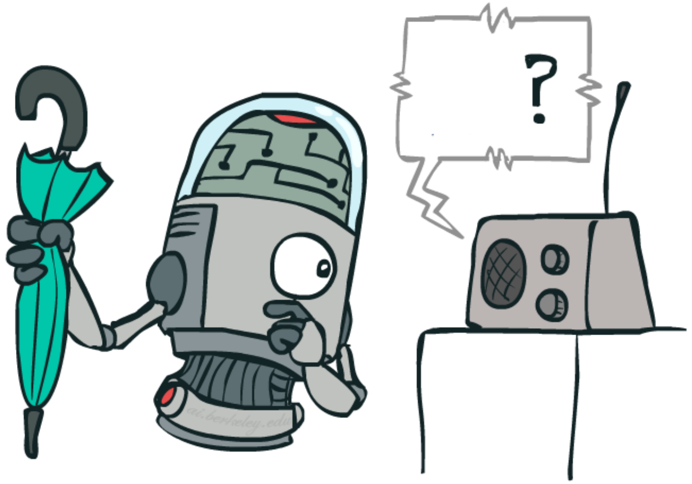
Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

F	A	E(U)	P(F)
sun	leave	86	0.65
rain	take	50	0.35

= 73.4



Value of State Information: Weather

We just need the marginal P(F) to finish

And we can use the joint table from Bayes slide:

W	F	P(F, W)
sun	sun	0.56
sun	rain	0.14
rain	sun	0.09
rain	rain	0.21

F	P(F)
sun	0.65
rain	0.35

W	F	A	U	P(W|F)
sun	sun	leave	100	0.86
sun	sun	take	20	0.86
sun	rain	leave	100	0.4
sun	rain	take	20	0.4
rain	sun	leave	0	0.14
rain	sun	take	70	0.14
rain	rain	leave	0	0.6
rain	rain	take	70	0.6

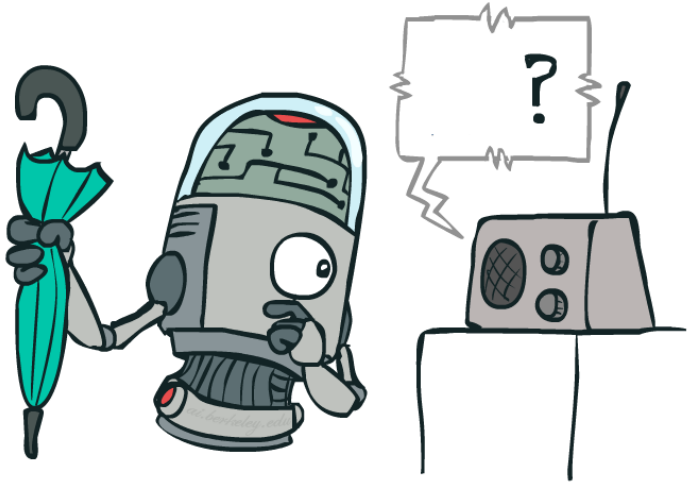
Expected Utility (over W)

F	A	E(U)
sun	leave	86
sun	take	27
rain	leave	40
rain	take	50

Take the best action

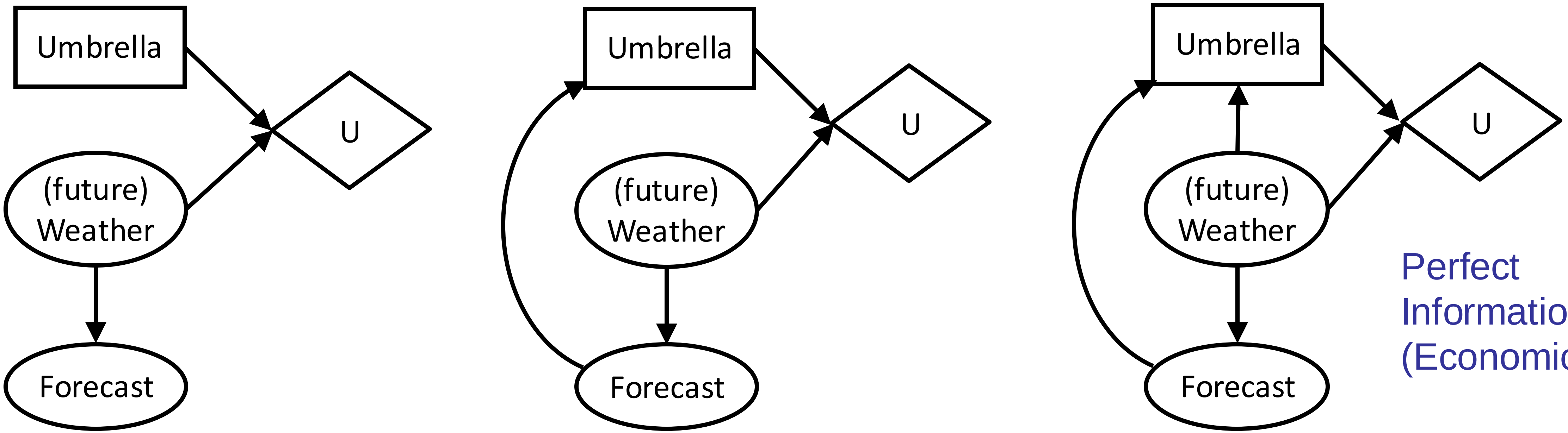
F	A	E(U)	P(F)
sun	leave	86	0.65
rain	take	50	0.35

= 73.4



Value of Information

- Value of Information is the difference in MEU between networks with different action conditioning (information).



Perfect Information (Economics)

$MEU(\emptyset) = 70$ $\leq MEU(F) = 73.4$ $\leq MEU(F, W) = 91$

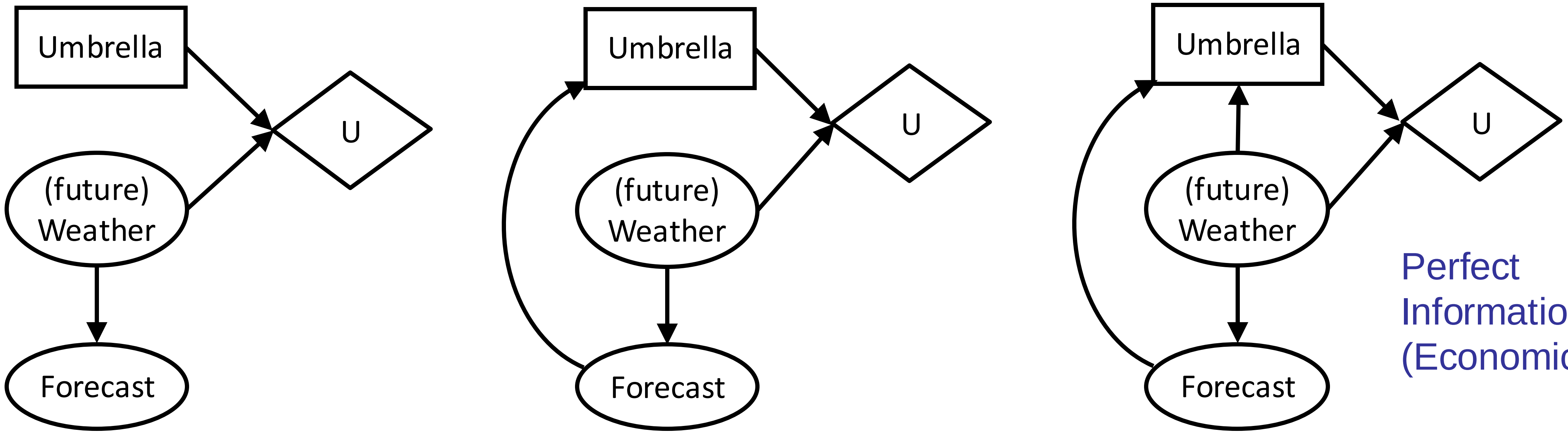
Difference = $VoI(F) = 3.4$

Difference = $VoI(W|F) = 17.6$

Difference = $VoI(F, W) = 21$

Value of Information

- The textbook uses the term "Value of Perfect Information" (VPI) for any information known perfectly. We'll follow that from now on.



Perfect Information (Economics)

$MEU(\emptyset) = 70$ $\leq MEU(F) = 73.4$ $\leq MEU(F, W) = 91$

Difference = $VPI(F) = 3.4$

Difference = $VPI(W|F) = 17.6$

Difference = $VPI(F, W) = 21$

VPI Properties

$VPI(E'|e)$ is value of knowing E' given evidence e.

- Nonnegative:

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

- Nonadditive (think of observing E_j twice)

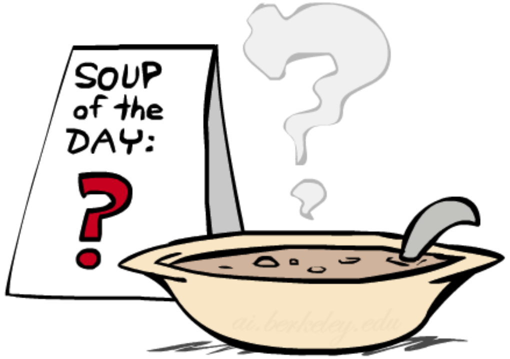
$$\mathrm{VPI}(E_j, E_k|e) \neq \mathrm{VPI}(E_j|e) + \mathrm{VPI}(E_k|e)$$

- Order-independent

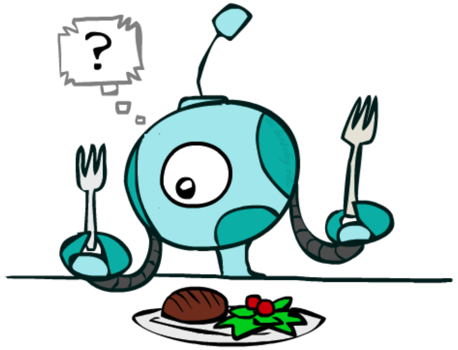
$$\begin{aligned}\mathrm{VPI}(E_j, E_k|e) &= \mathrm{VPI}(E_j|e) + \mathrm{VPI}(E_k|e, E_j) \\ &= \mathrm{VPI}(E_k|e) + \mathrm{VPI}(E_j|e, E_k)\end{aligned}$$

Quick VPI Questions

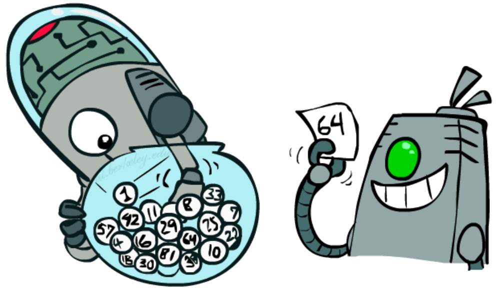
- The soup of the day is either clam chowder or split pea, but you wouldn't order either one. What's the value of knowing which it is?



- There are two kinds of plastic forks in a basket 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?

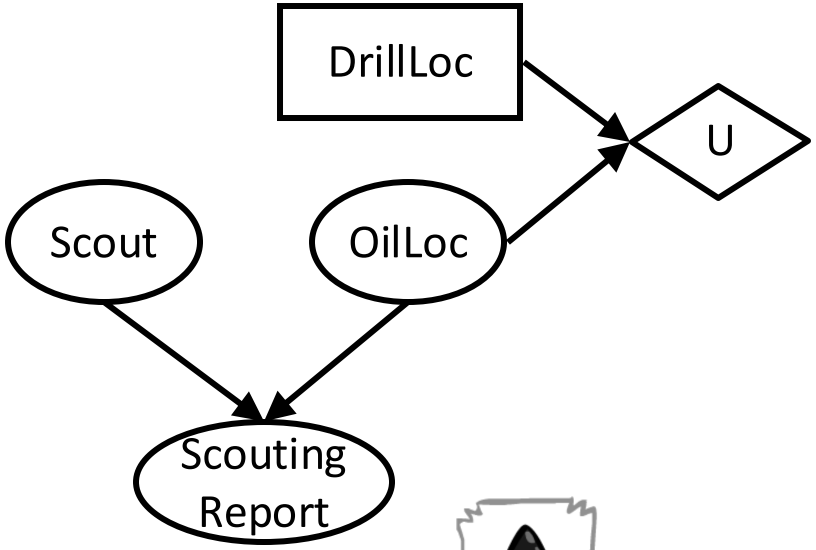


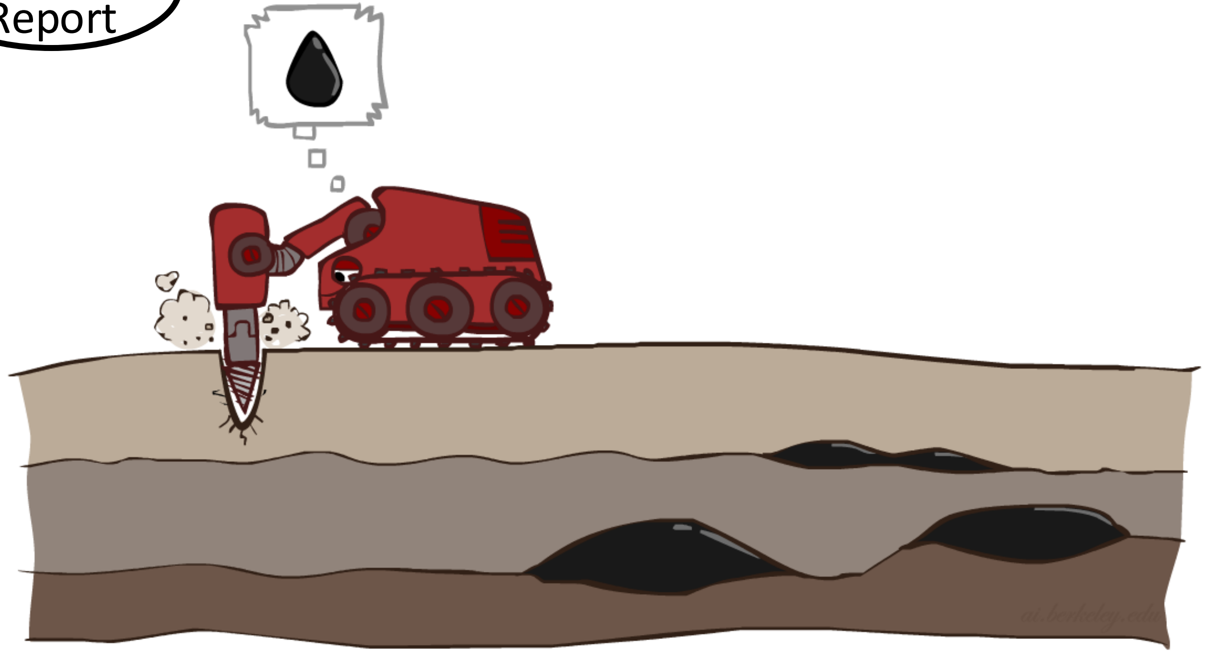
VPI Question

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

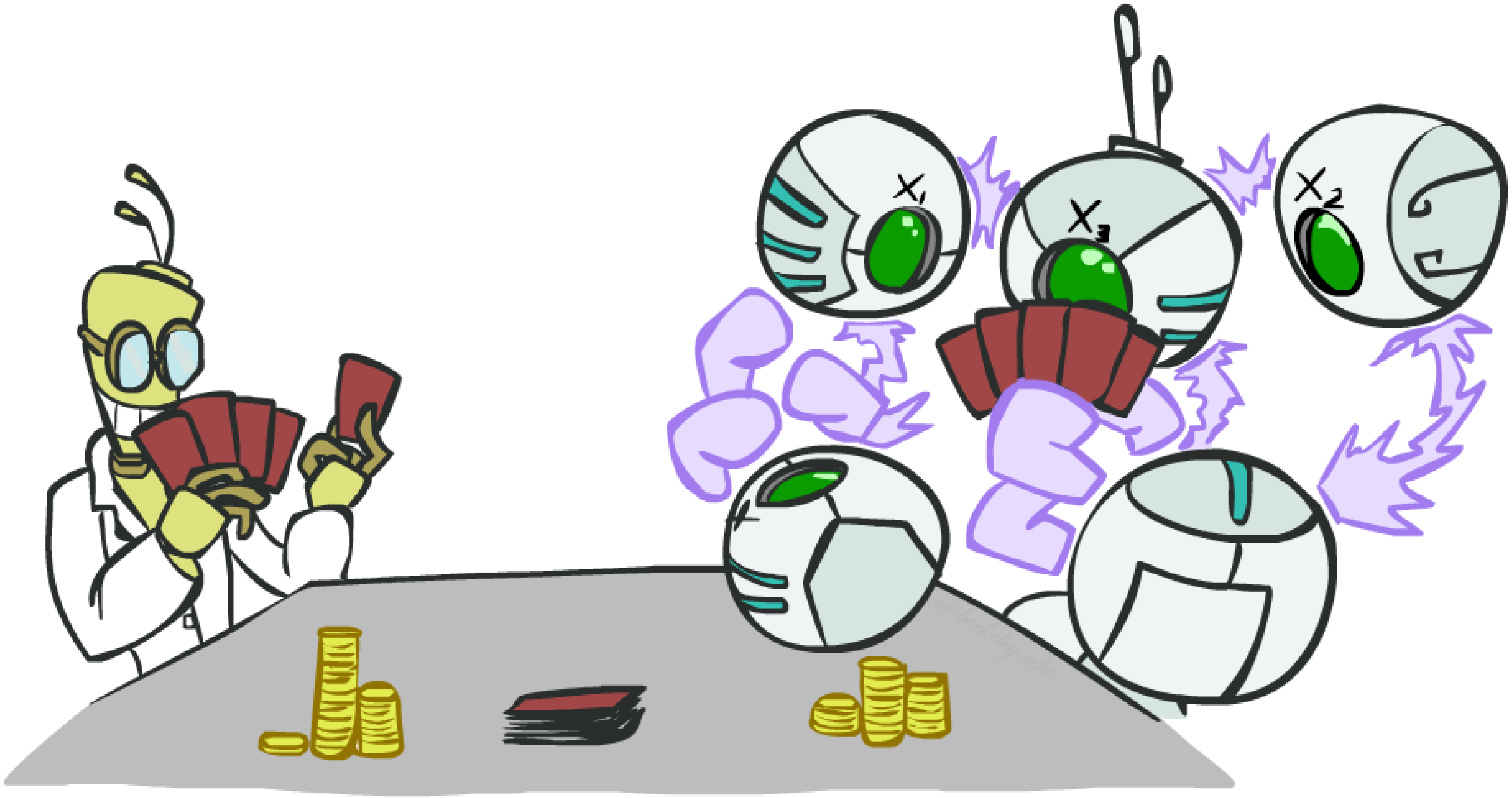
  If Parents(U) $\perp\!\!\!\perp$ Z | CurrentEvidence

  Then VPI(Z | CurrentEvidence) = 0



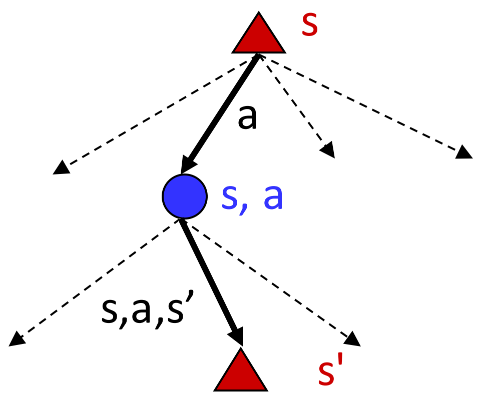


POMDPs

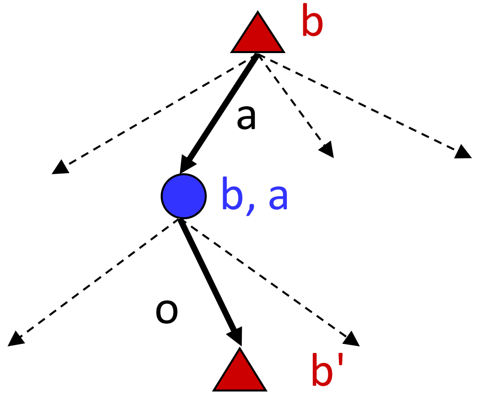


POMDPs

- 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))



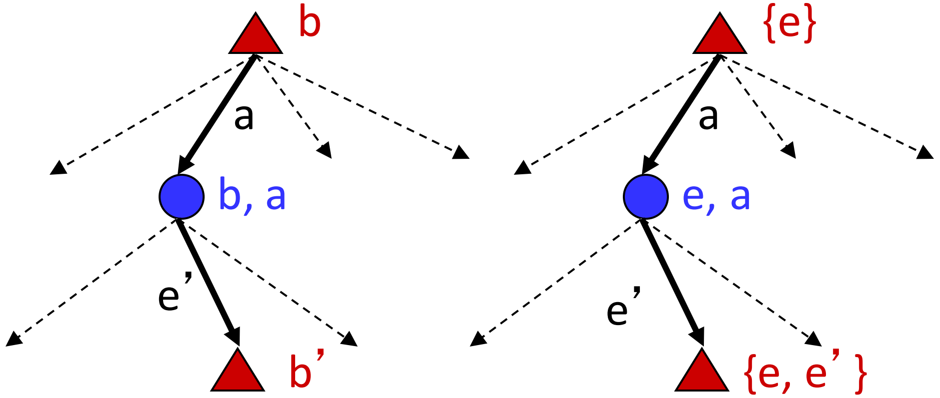
- POMDPs are MDPs over belief states b (distributions over S)

- We'll be able to say more in a few lectures

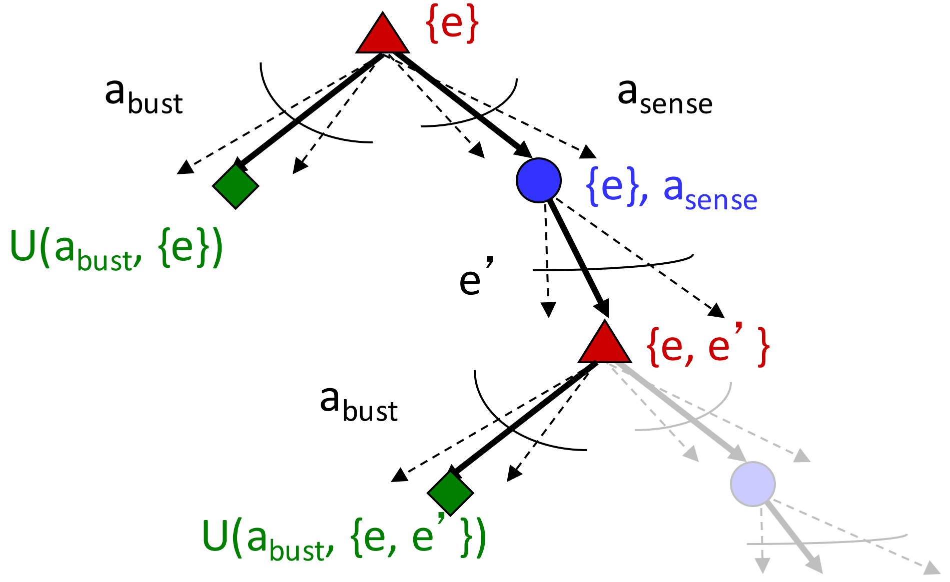
Demo: Ghostbusters with VPI

Example: Ghostbusters

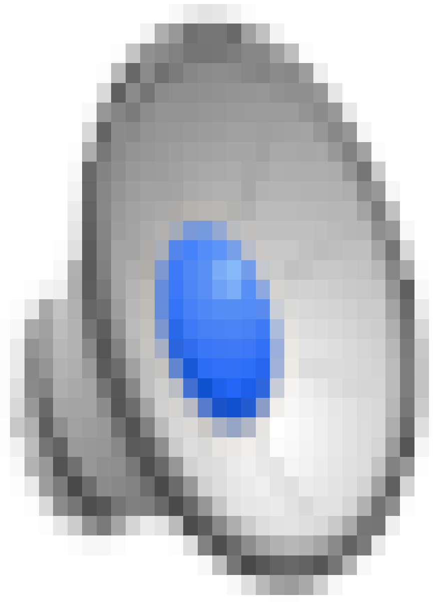
- In (static) Ghostbusters:
  - Belief state determined by evidence to date {e}
  - Tree really over evidence sets
  - Probabilistic reasoning needed to predict new evidence given past evidence



- 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!

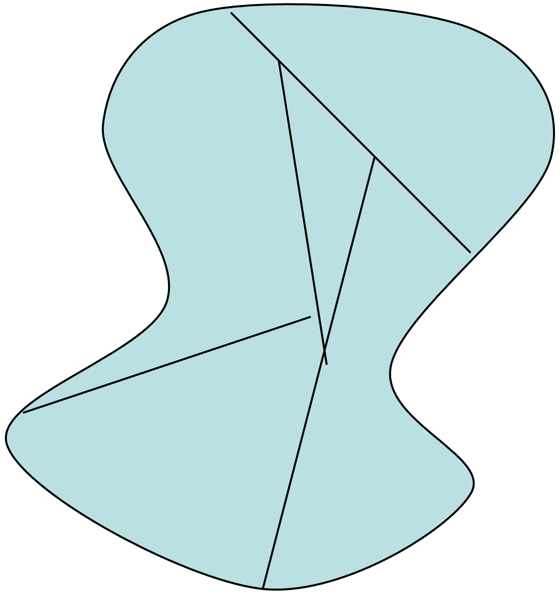


Video of Demo Ghostbusters with VPI



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, and we can rarely solve then in their full generality

Next Time: Dynamic Models (HMMs)