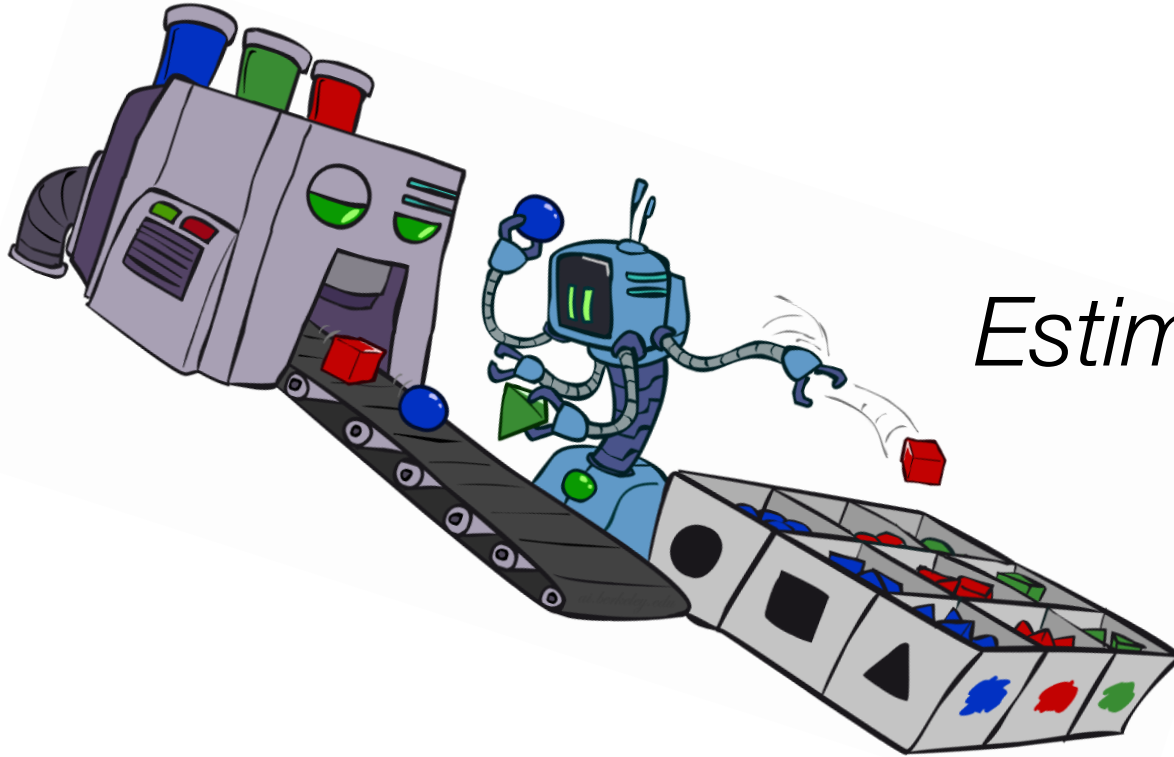


# CS 4100 // artificial intelligence

instructor: [byron wallace](#)



## *Bayes Nets II: Estimation and Inference*

**Attribution:** many of these slides are modified versions of those distributed with the [UC Berkeley CS188](#) materials  
Thanks to [John DeNero](#) and [Dan Klein](#)

# Bayes' net representation

A directed, acyclic graph, one node per random variable

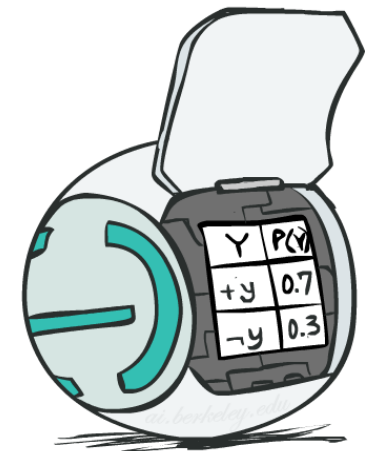
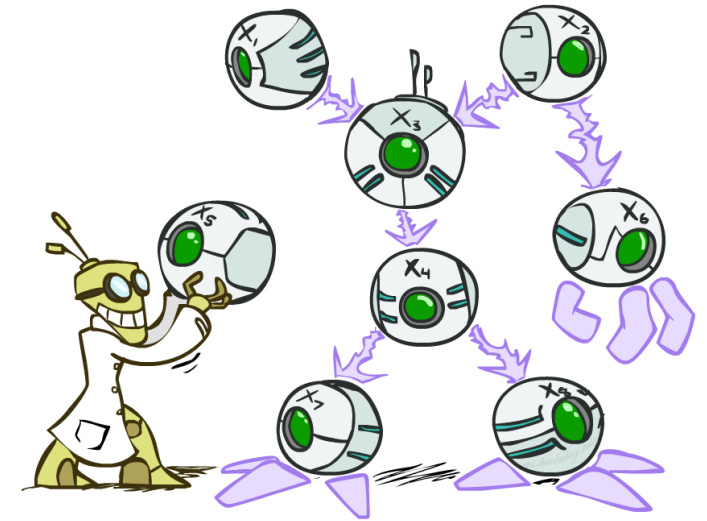
A conditional probability table (CPT) for each node

- A collection of distributions over  $X$ , one for each combination of parents'  $P(X|a_1 \dots a_n)$

Bayes' nets implicitly encode joint distributions

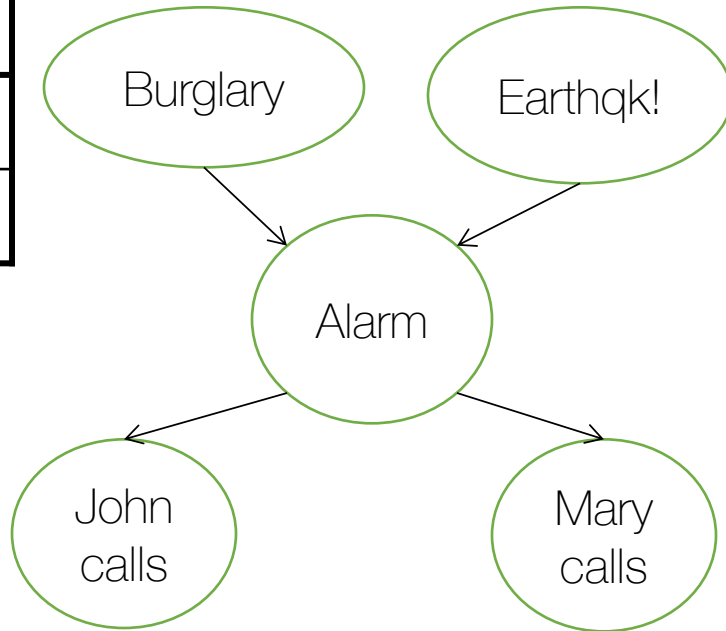
- As a product of local conditional distributions
- To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$



# Example: alarm network

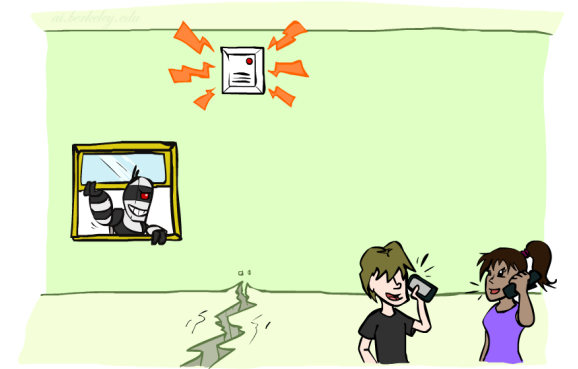
B	P(B)
+b	0.001
-b	0.999



A	J	P(J A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99

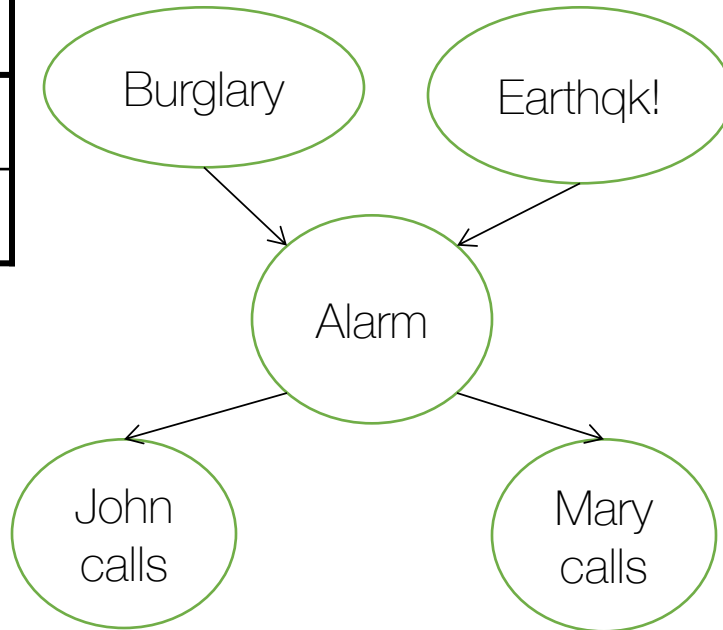
E	P(E)
+e	0.002
-e	0.998



B	E	A	P(A B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

# Example: alarm network

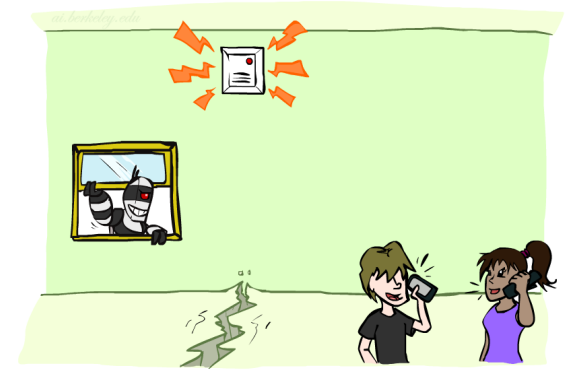
B	P(B)
+b	0.001
-b	0.999



E	P(E)
+e	0.002
-e	0.998

A	J	P(J A)
+a	+j	0.9
+a	-j	0.1
-a	+j	0.05
-a	-j	0.95

A	M	P(M A)
+a	+m	0.7
+a	-m	0.3
-a	+m	0.01
-a	-m	0.99



B	E	A	P(A B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

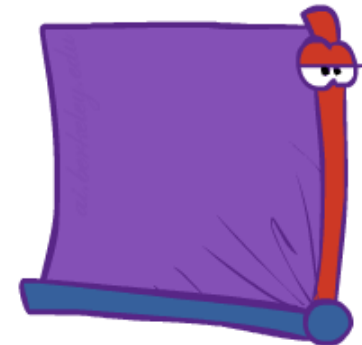
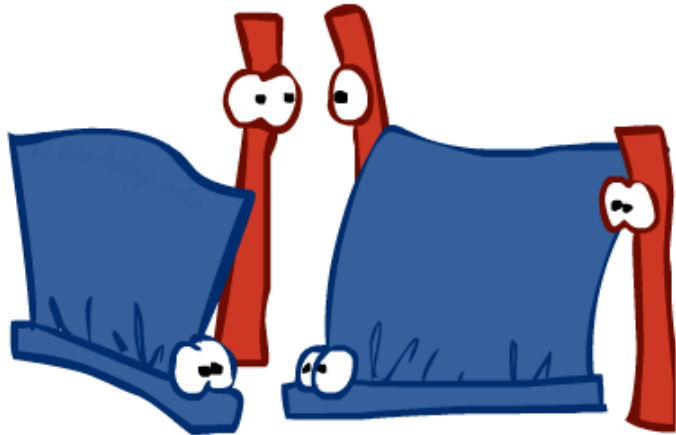
$$\begin{aligned}
 P(+b, -e, +a, -j, +m) &= \\
 P(+b)P(-e)P(+a|+b, -e)P(-j|+a)P(+m|+a) &= \\
 0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7
 \end{aligned}$$

# Inference in Bayes' nets

- **Inference:** calculating some useful quantity from a joint probability distribution

Example: Posterior probability

$$P(Q|E_1 = e_1, \dots, E_k = e_k)$$



# Naïve approach: inference by enumeration

General case:

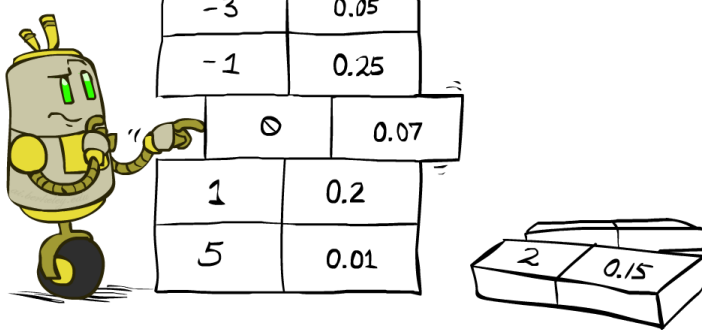
- Evidence variables:  $E_1 \dots E_k = e_1 \dots e_k$
  - Query\* variable:  $Q$
  - Hidden variables:  $H_1 \dots H_r$
- $\left. \begin{array}{l} E_1 \dots E_k = e_1 \dots e_k \\ Q \\ H_1 \dots H_r \end{array} \right\} \begin{array}{l} X_1, X_2, \dots X_n \\ \text{All variables} \end{array}$

We want:

\* Works fine with multiple query variables, too

$$P(Q|e_1 \dots e_k)$$

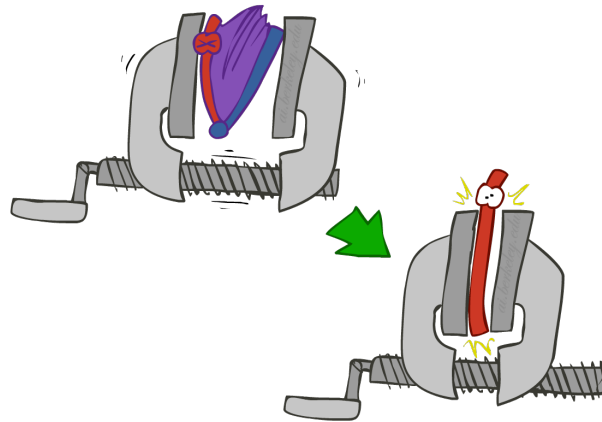
Step 1: Select the entries consistent with the evidence



x	P(x)
-3	0.05
-1	0.25
0	0.07
1	0.2
5	0.01

2 0.15

Step 2: Sum out H to get joint of Query and evidence



Step 3: Normalize

$$\times \frac{1}{Z}$$

$$Z = \sum_q P(Q, e_1 \dots e_k)$$

$$P(Q|e_1 \dots e_k) = \frac{1}{Z} P(Q, e_1 \dots e_k)$$

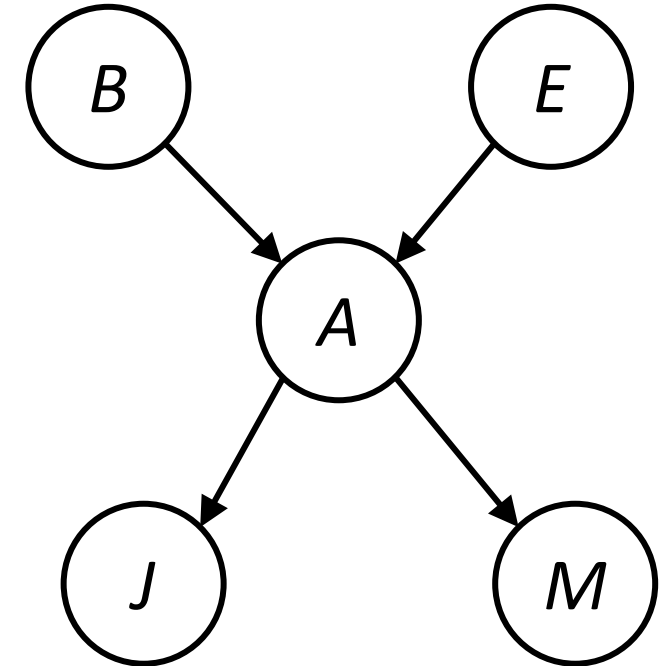
$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} \underbrace{P(Q, h_1 \dots h_r, e_1 \dots e_k)}_{X_1, X_2, \dots X_n}$$

# Inference by enumeration in Bayes' Net

Given unlimited time, inference in BNs is easy

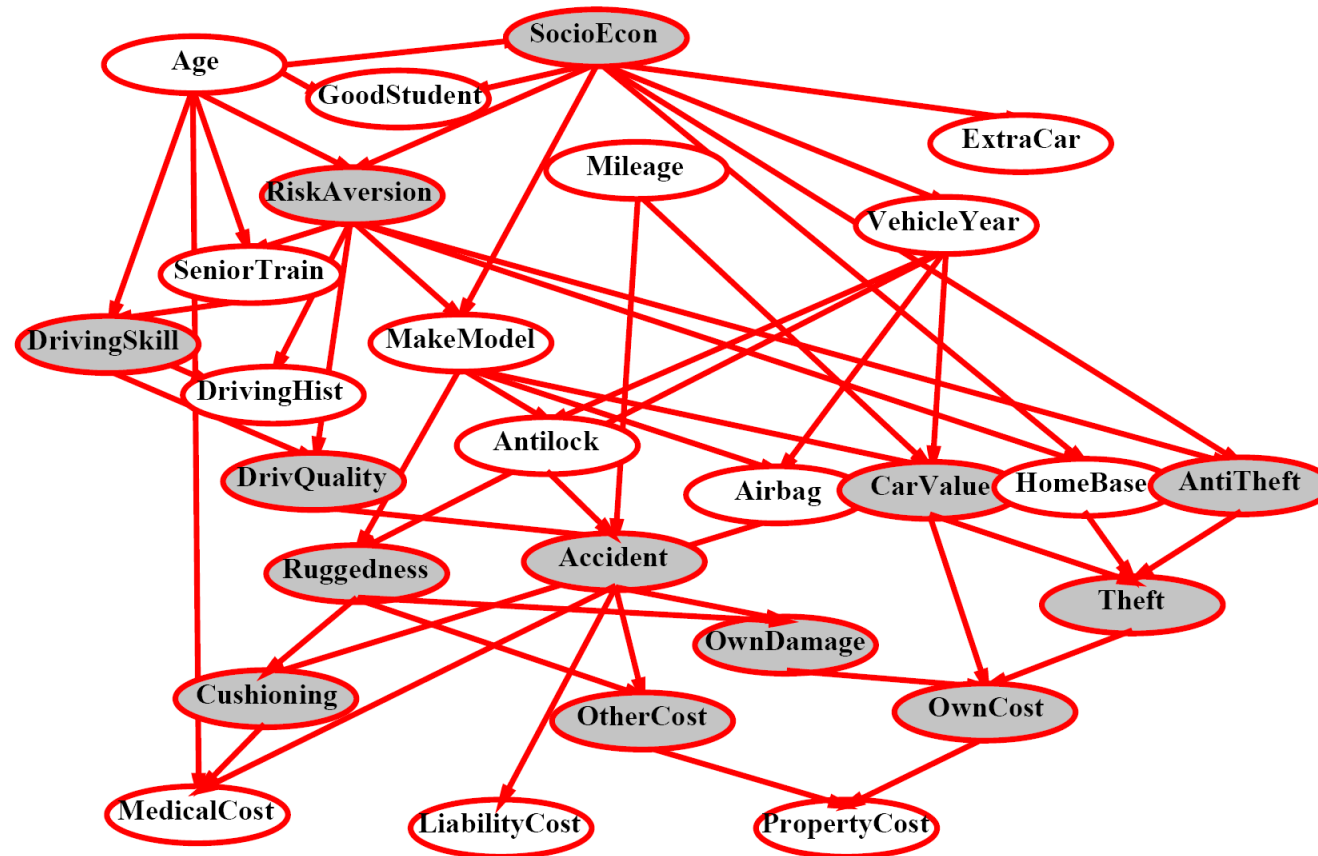
Reminder of inference by enumeration by example:

$$\begin{aligned}P(B \mid +j, +m) &\propto_B P(B, +j, +m) \\&= \sum_{e,a} P(B, e, a, +j, +m) \\&= \sum_{e,a} P(B)P(e)P(a|B, e)P(+j|a)P(+m|a)\end{aligned}$$



$$\begin{aligned}=&P(B)P(+e)P(+a|B, +e)P(+j|+a)P(+m|+a) + P(B)P(+e)P(-a|B, +e)P(+j|-a)P(+m|-a) \\&P(B)P(-e)P(+a|B, -e)P(+j|+a)P(+m|+a) + P(B)P(-e)P(-a|B, -e)P(+j|-a)P(+m|-a)\end{aligned}$$

# Inference by enumeration?



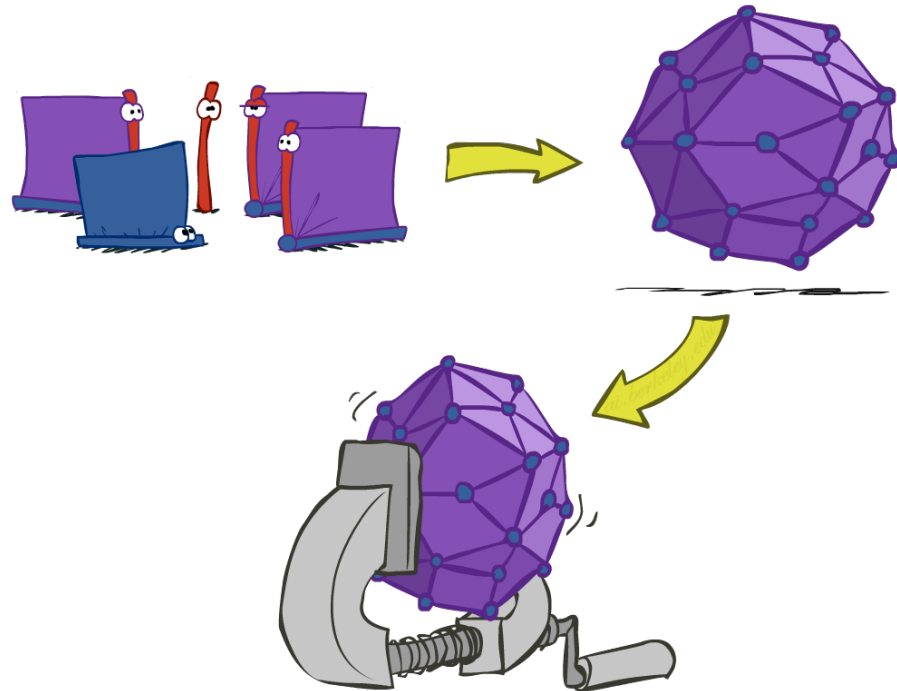
$$P(\textit{Antilock} | \textit{observed variables}) = ?$$



# Inference by enumeration vs. variable elimination

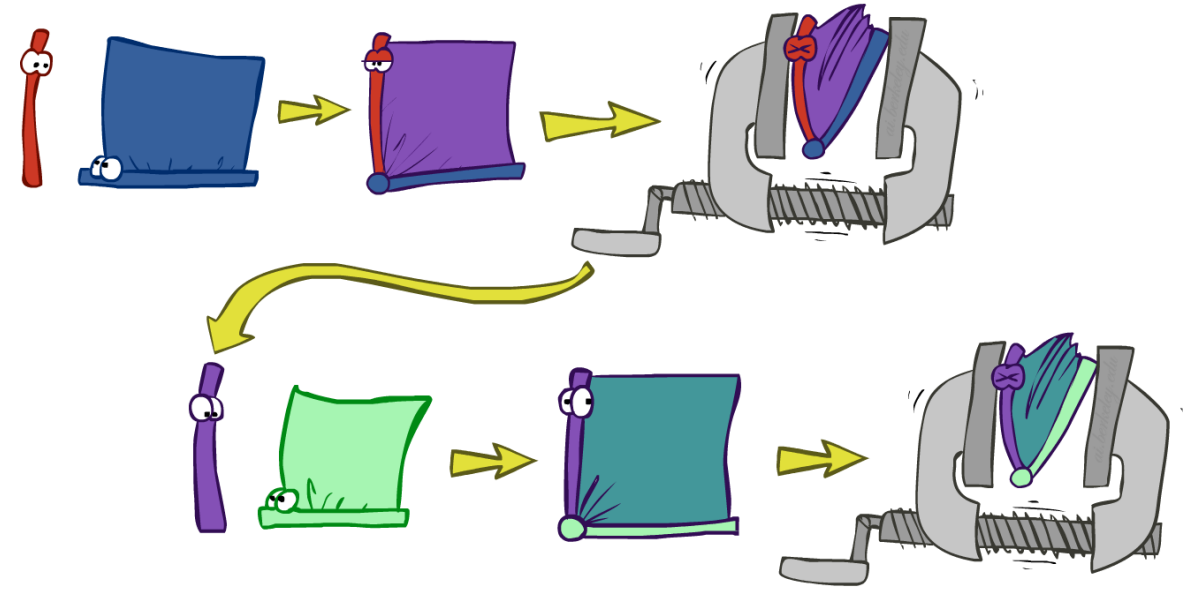
Why is inference by enumeration so slow?

- You join up the whole joint distribution before you sum out the hidden variables



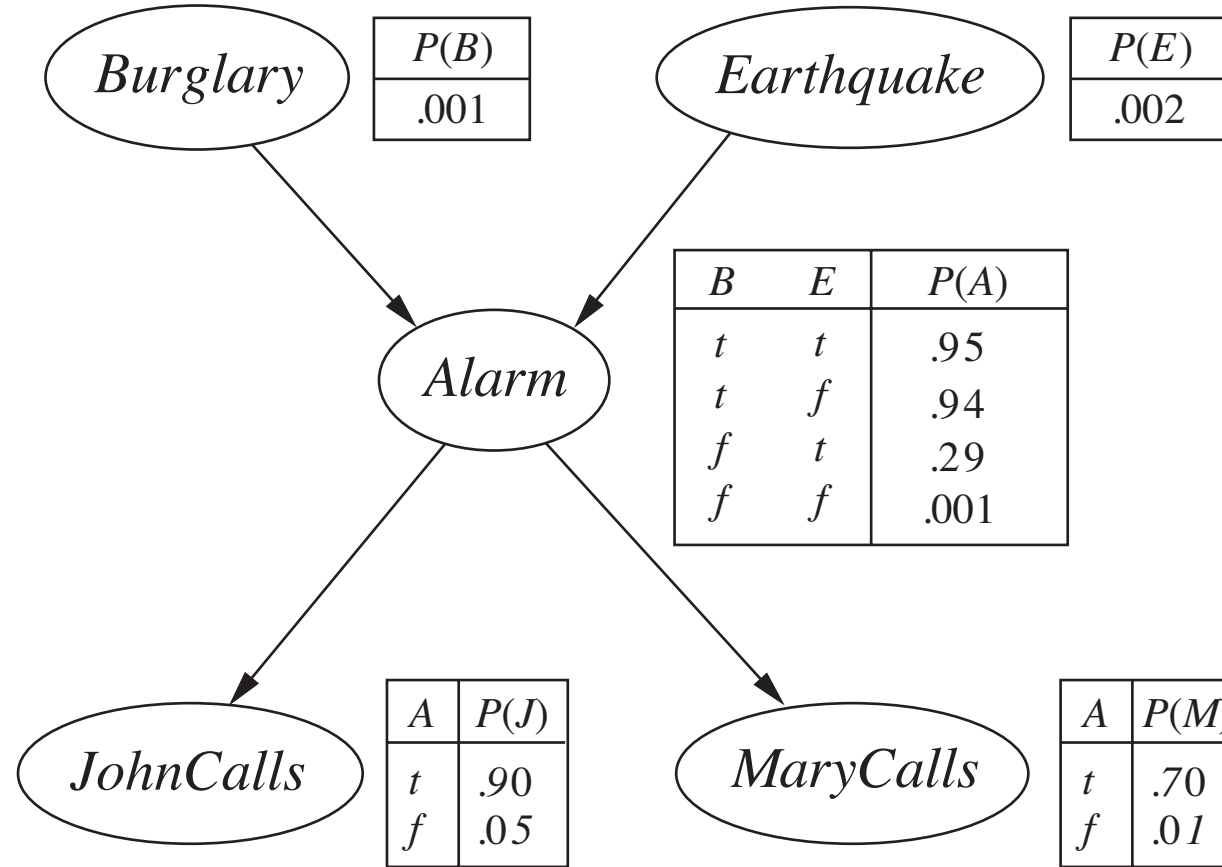
Idea: interleave joining and marginalizing!

- Called “Variable Elimination”
- Still NP-hard, but usually much faster than inference by enumeration

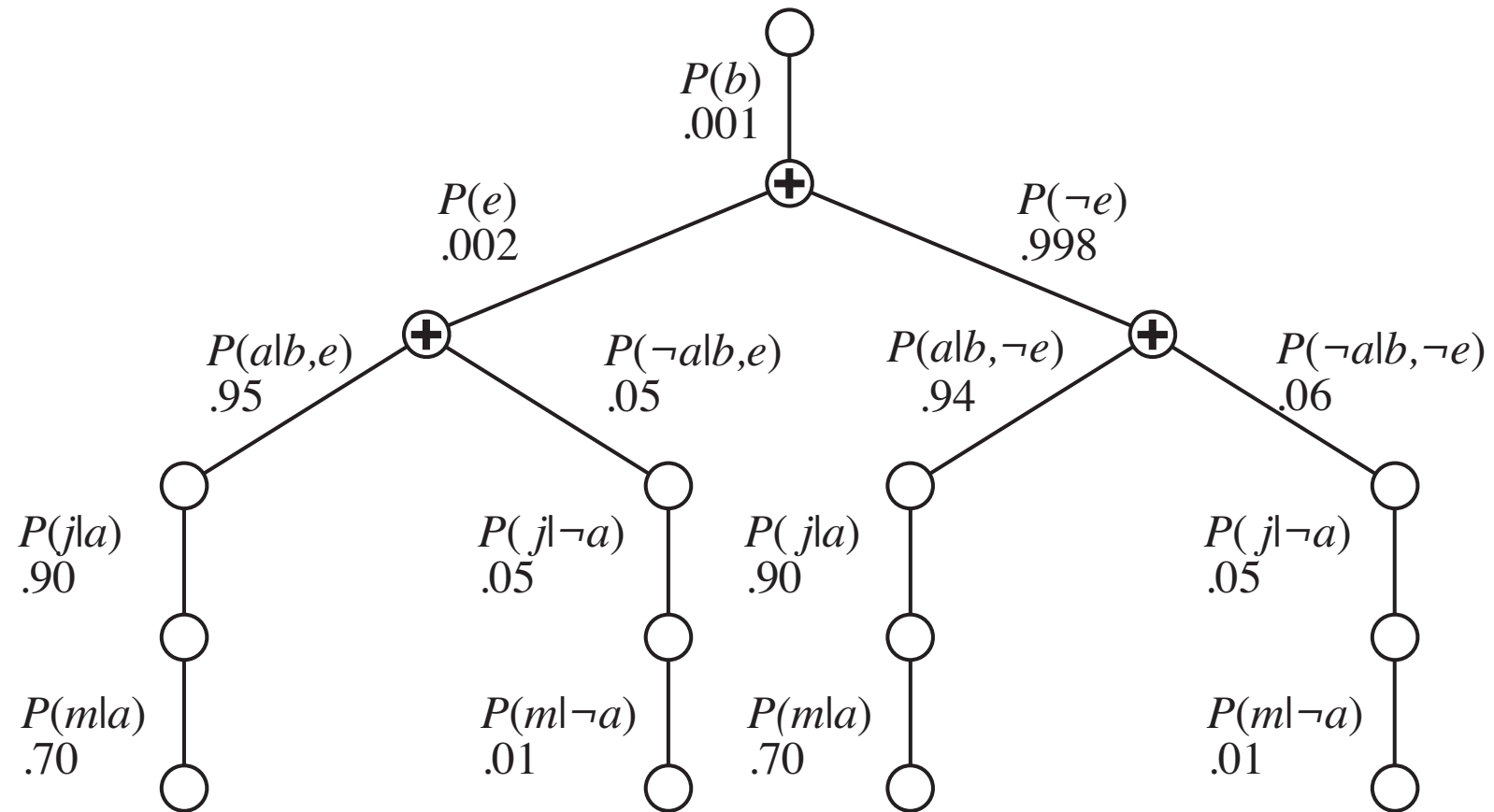


- First we'll need some new notation: factors

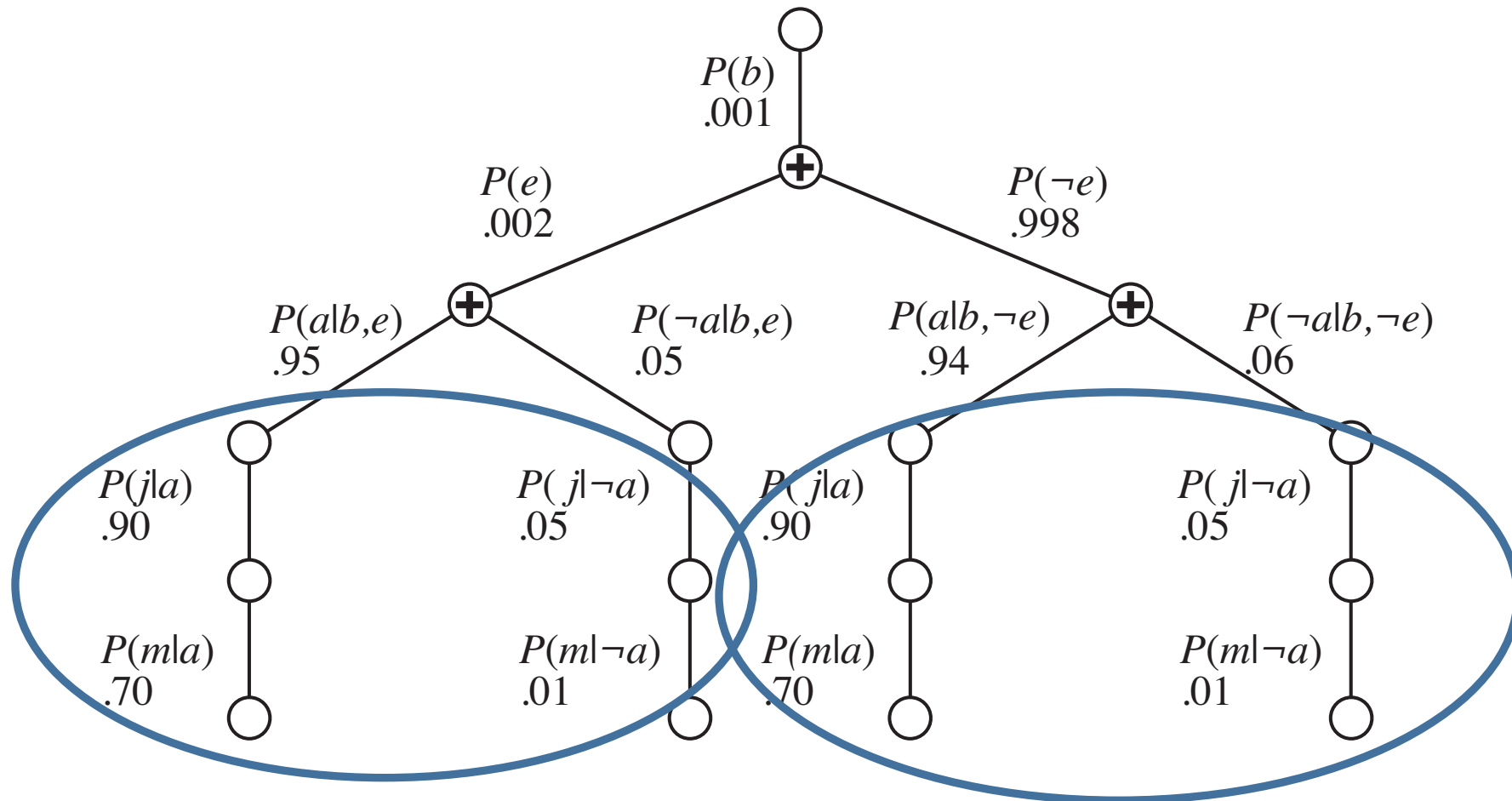
# Factors



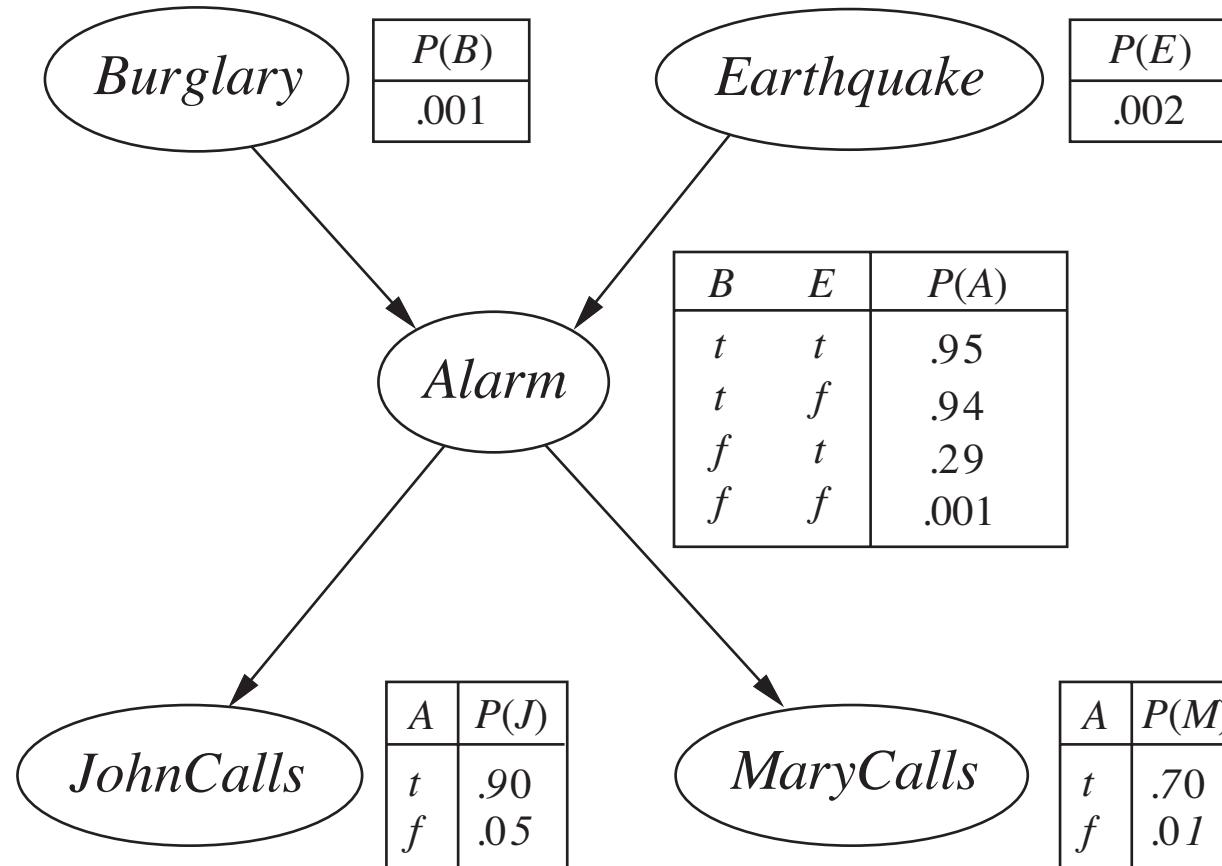
# Factors



# Factors

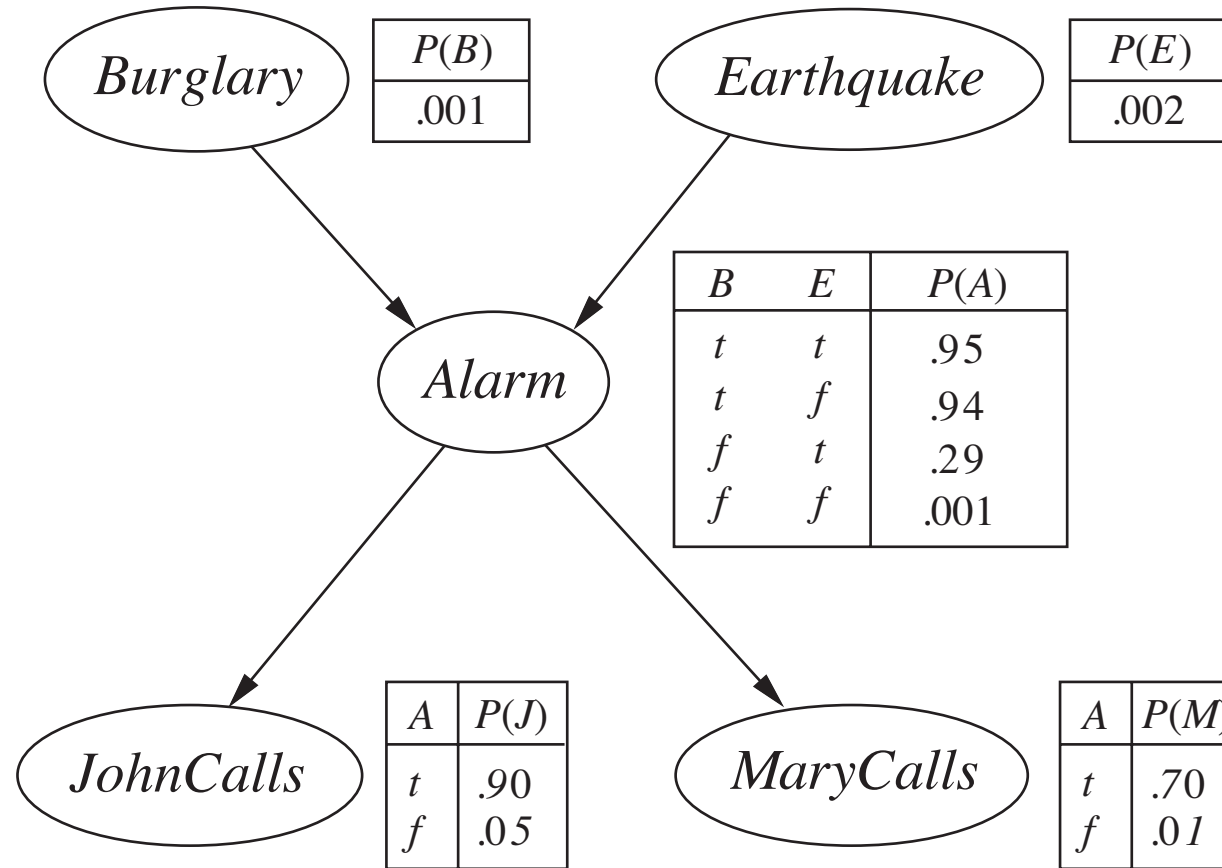


# Factors



$$\mathbf{P}(B \mid j, m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B, e)}_{\mathbf{f}_3(A, B, E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

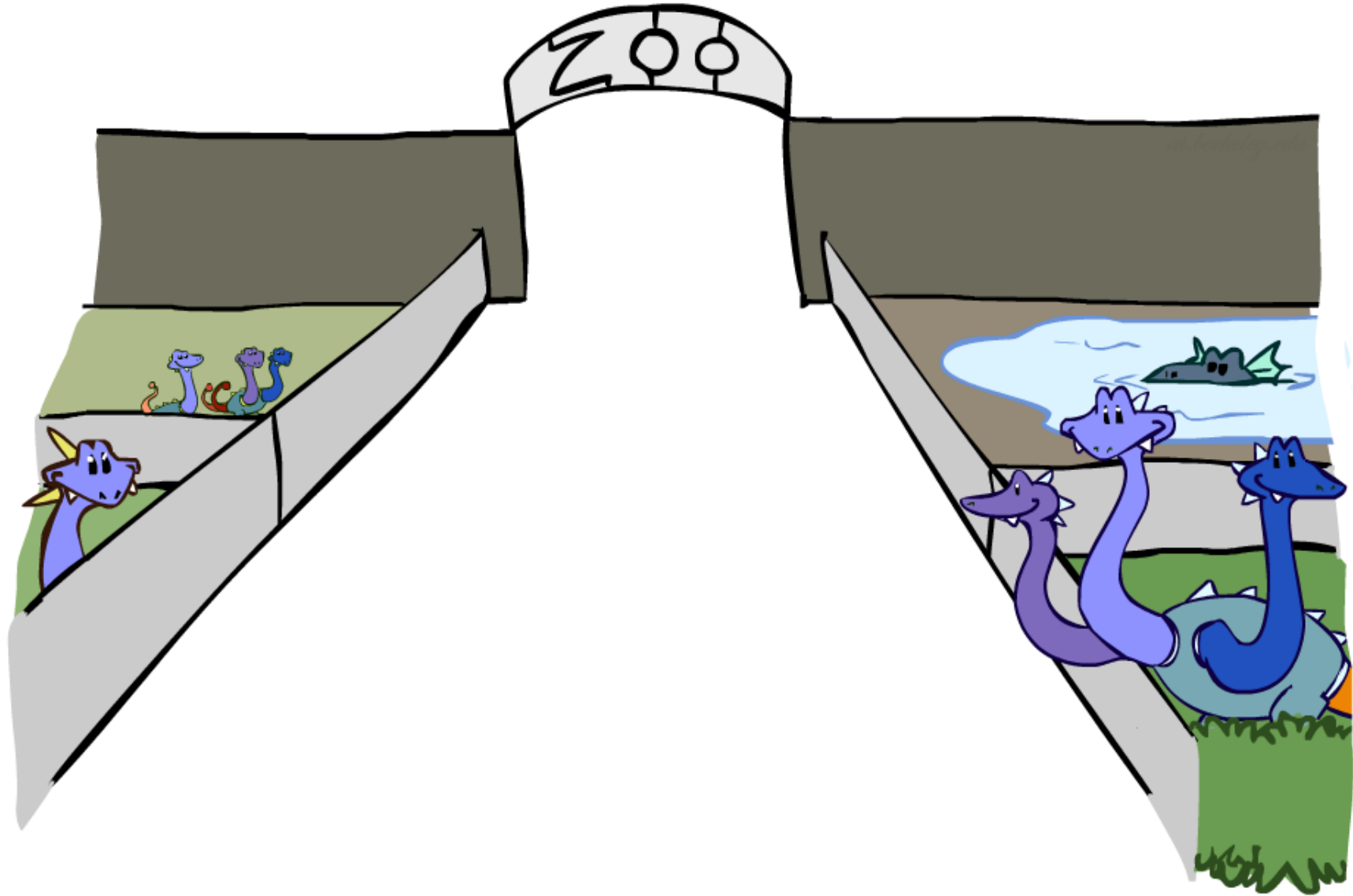
# Factors



$$\mathbf{P}(B \mid j, m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B, e)}_{\mathbf{f}_3(A, B, E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

$$\mathbf{f}_4(A) = \begin{pmatrix} P(j \mid a) \\ P(j \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix}$$

# Factor zoo



# Factor zoo 1

Joint distribution:  $P(X,Y)$

- Entries  $P(x,y)$  for all  $x, y$
- Sums to 1

Selected joint:  $P(x,Y)$

- A slice of the joint distribution
- Entries  $P(x,y)$  for fixed  $x$ , all  $y$
- Sums to  $P(x)$

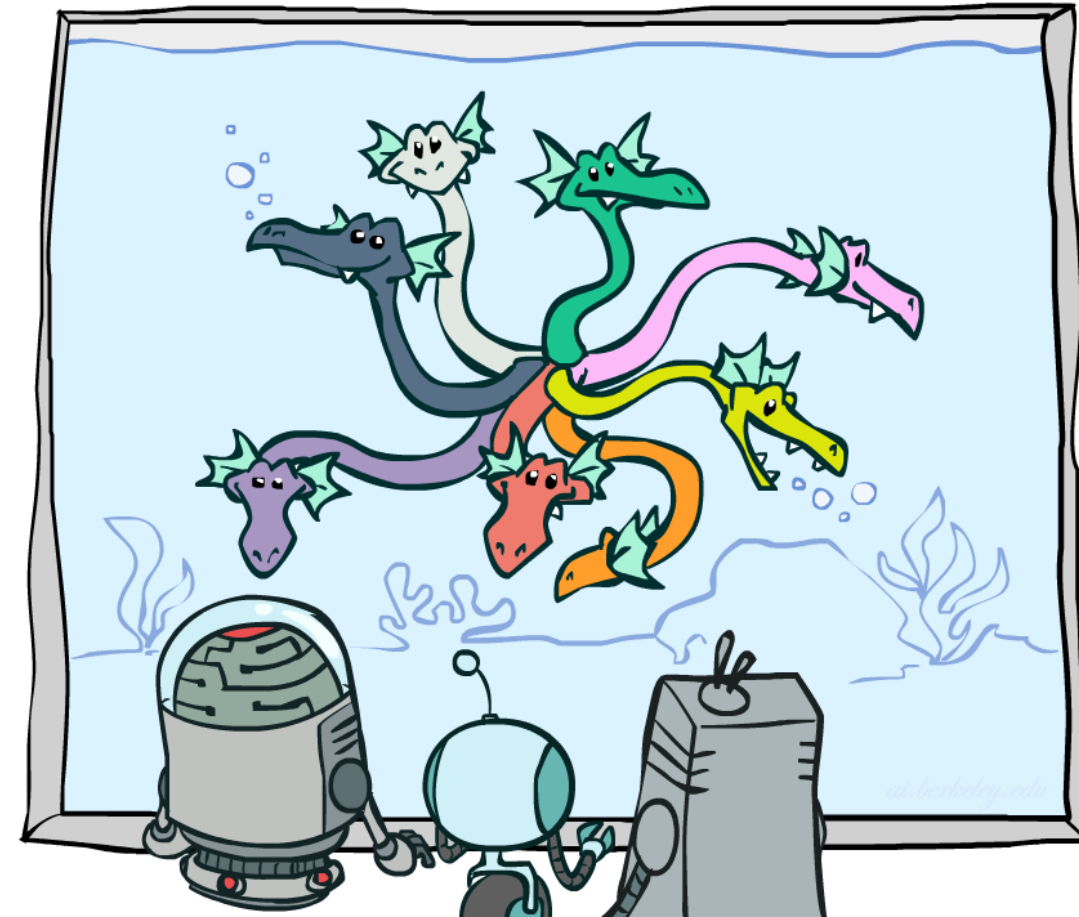
Number of capitals =  
dimensionality of the table

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$P(\text{cold}, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

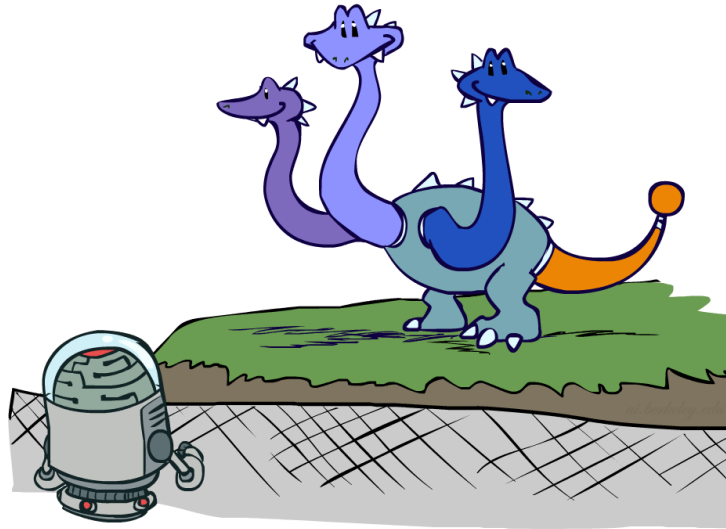




# Factor zoo 2

Single conditional:  $P(Y | x)$

- Entries  $P(y | x)$  for fixed  $x$ , all  $y$
- Sums to 1



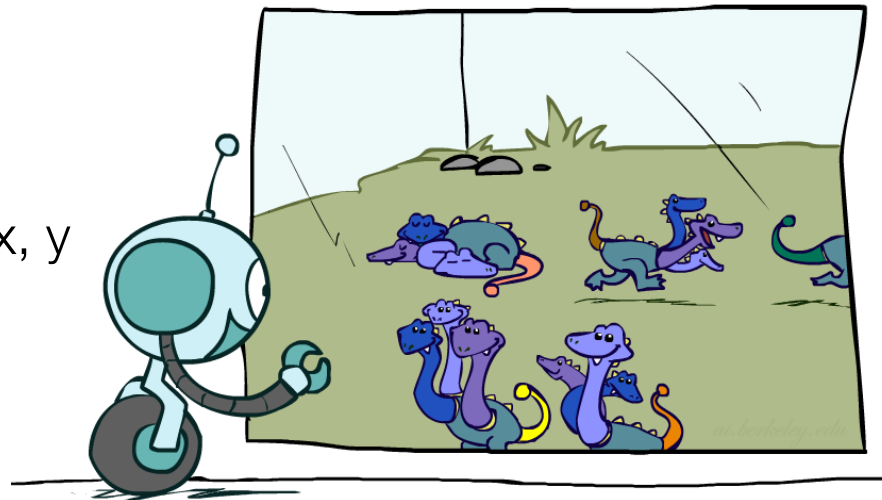
$P(W|cold)$

T	W	P
cold	sun	0.4
cold	rain	0.6

Family of conditionals:

$P(X | Y)$

- Multiple conditionals
- Entries  $P(x | y)$  for all  $x, y$
- Sums to  $|Y|$



$P(W|T)$

T	W	P
hot	sun	0.8
hot	rain	0.2
cold	sun	0.4
cold	rain	0.6

$P(W|hot)$

$P(W|cold)$

# Factor zoo 3

Specified family:  $P(y \mid X)$

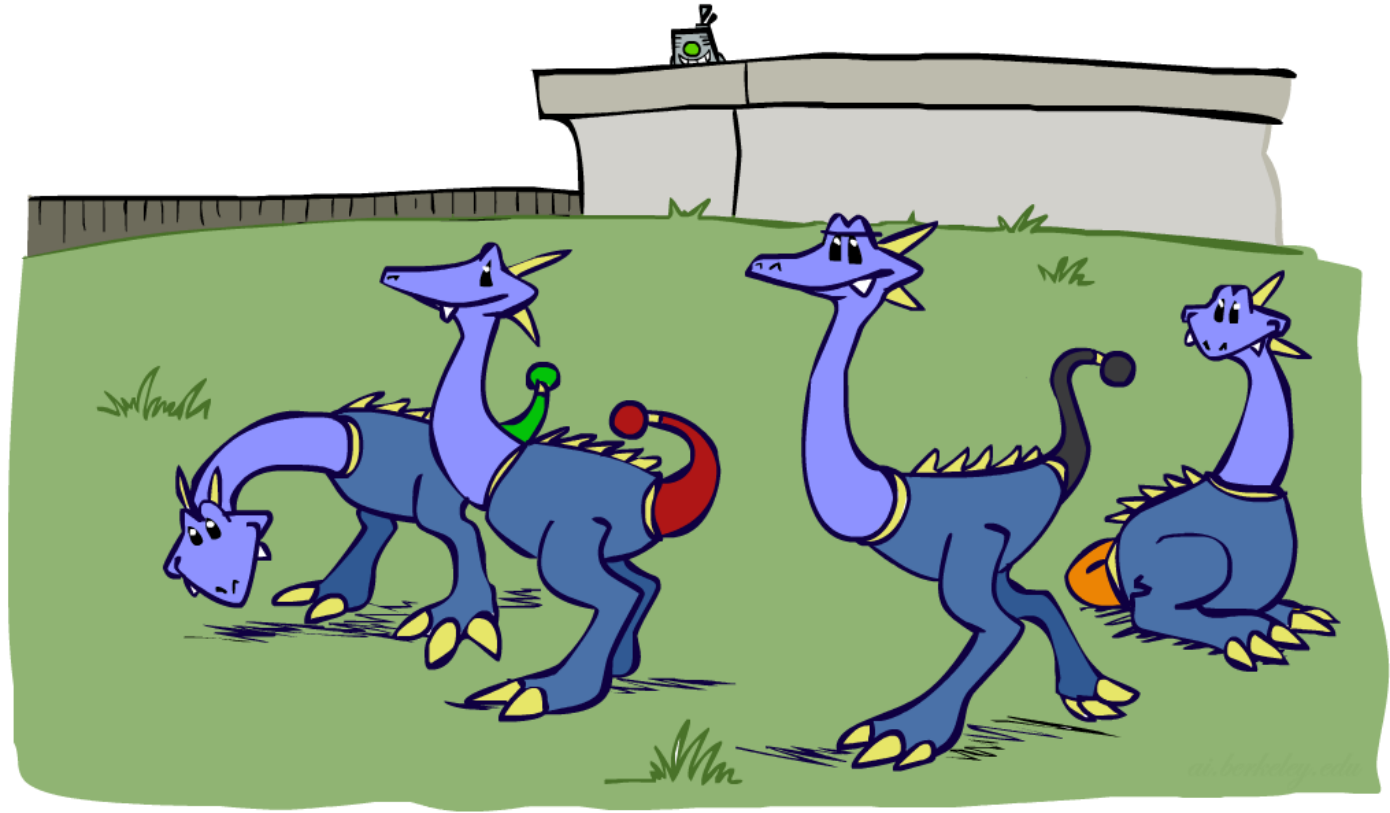
- Entries  $P(y \mid x)$  for fixed  $y$ , but for all  $x$
- Sums to ... who knows!

$$P(\text{rain} \mid T)$$

T	W	P
hot	rain	0.2
cold	rain	0.6

$$\left. \begin{array}{c} \\ \end{array} \right\} P(\text{rain} \mid \text{hot})$$

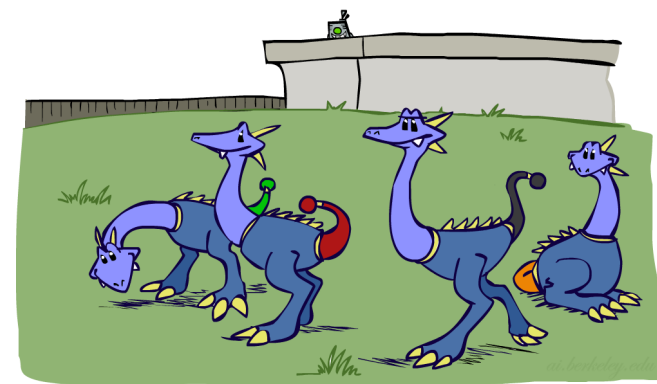
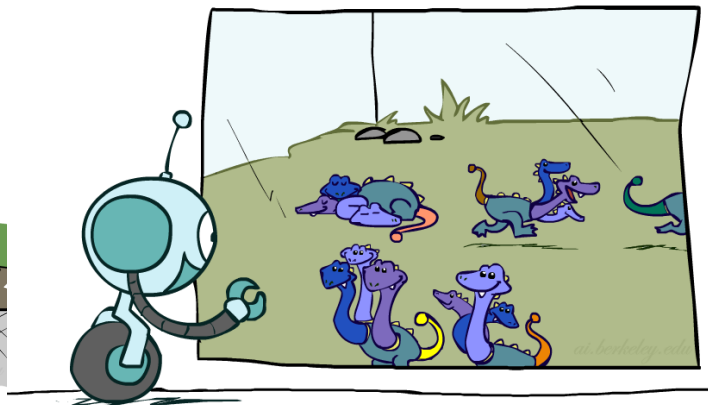
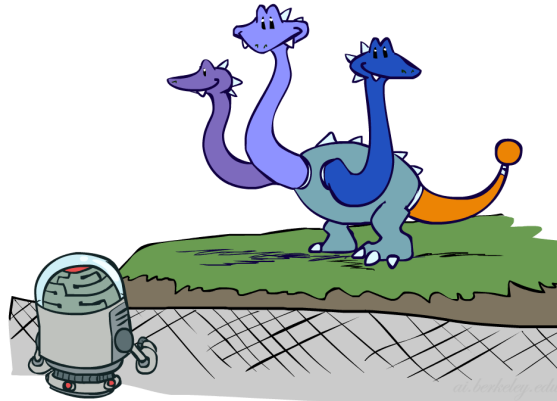
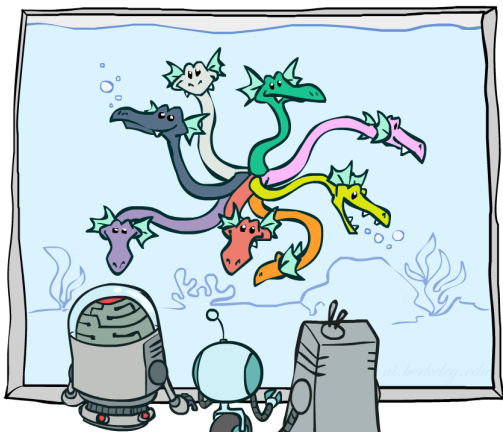
$$\left. \begin{array}{c} \\ \end{array} \right\} P(\text{rain} \mid \text{cold})$$



# Factor zoo summary

In general, when we write  $P(Y_1 \dots Y_N \mid X_1 \dots X_M)$

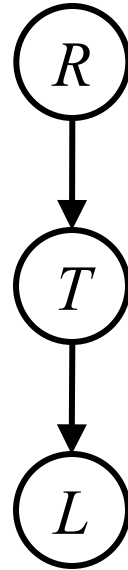
- It is a “factor,” a multi-dimensional array
- Its values are  $P(y_1 \dots y_N \mid x_1 \dots x_M)$
- Any assigned (=lower-case) X or Y is a dimension missing (selected) from the array



# Example: traffic domain

## Random Variables

- R: Raining
- T: Traffic
- L: Late for class!



$$P(L) = ?$$

$$= \sum_{r,t} P(r, t, L)$$

$$= \sum_{r,t} P(r)P(t|r)P(L|t)$$

$$P(R)$$

+r	0.1
-r	0.9

$$P(T|R)$$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$$P(L|T)$$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

# Inference by enumeration: procedural outline

Track objects called **factors**

Initial factors are local CPTs (one per node)

$$P(R)$$

+r	0.1
-r	0.9

$$P(T|R)$$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$$P(L|T)$$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Any known values are selected

E.g. if we know  $L = +\ell$ , the initial factors are

$$P(R)$$

+r	0.1
-r	0.9

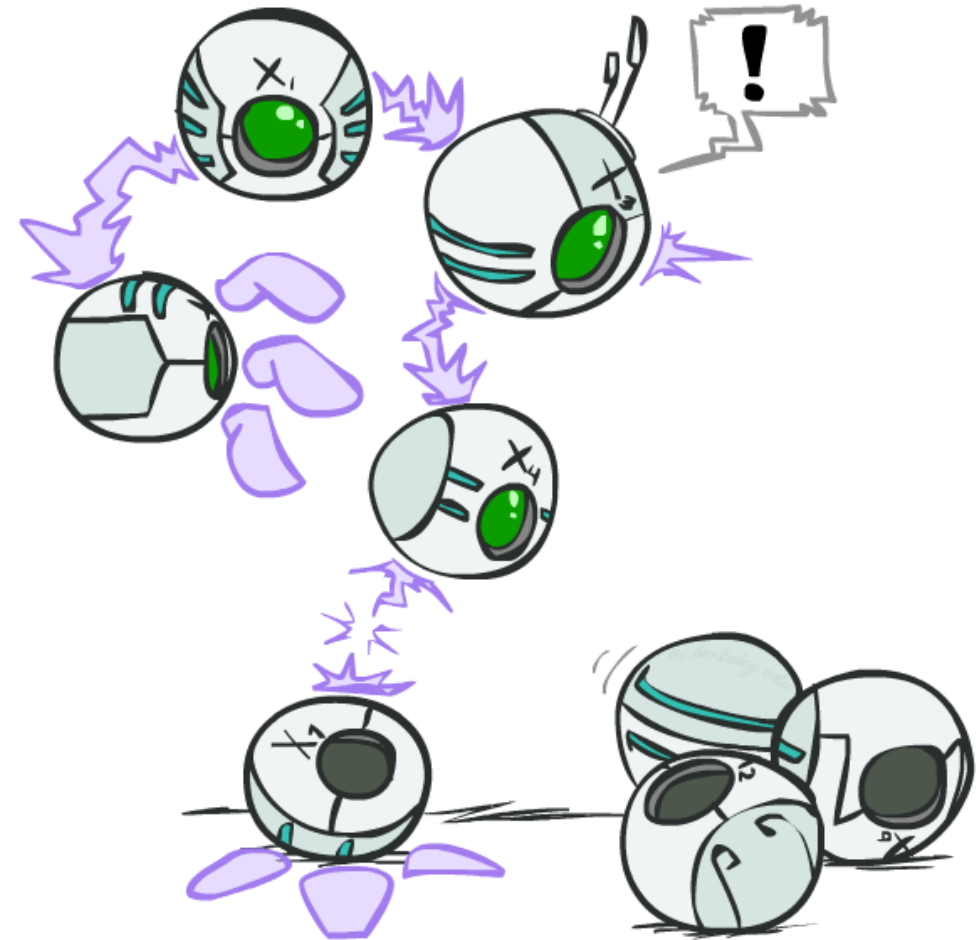
$$P(T|R)$$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$$P(+\ell|T)$$

+t	+l	0.3
-t	+l	0.1

Procedure: Join all factors, then eliminate all hidden variables

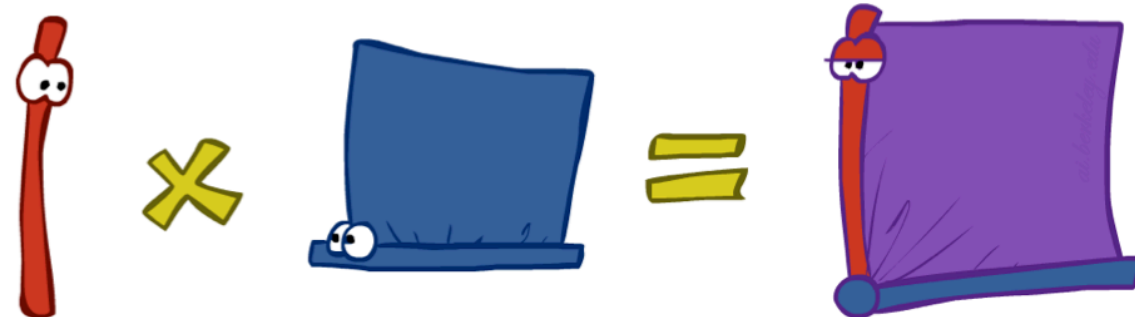


# Operation 1: join factors

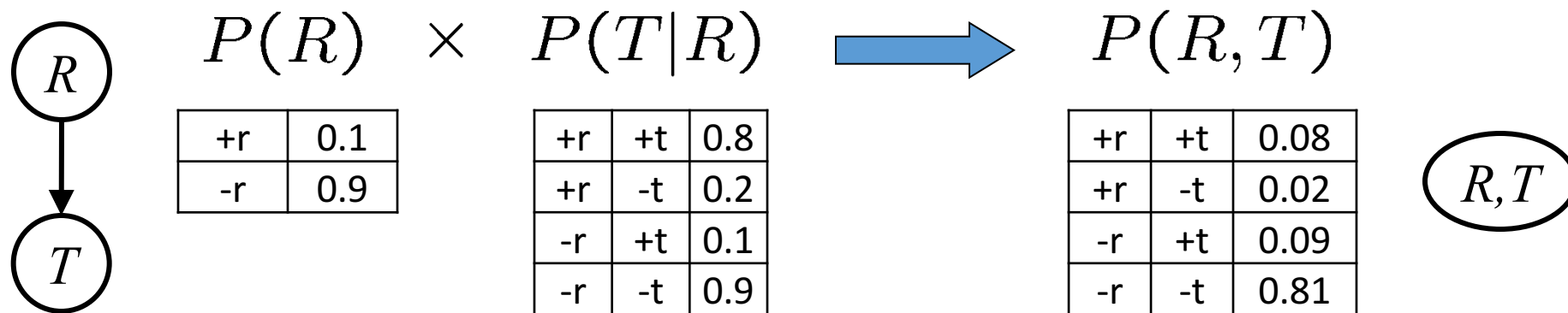
First basic operation: **joining factors**

Combining factors:

- Like a database join
- Get all factors over the joining variable
- Build a new factor over the union of the variables involved

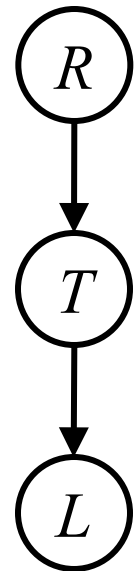


Example: Join on R



- Computation for each entry: pointwise products  $\forall r, t : P(r, t) = P(r) \cdot P(t|r)$

# Example: multiple joins



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

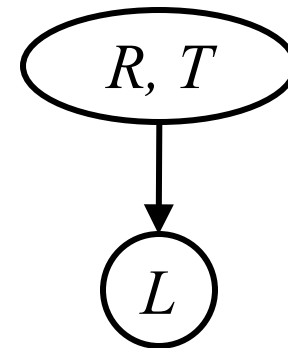


$P(R, T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9



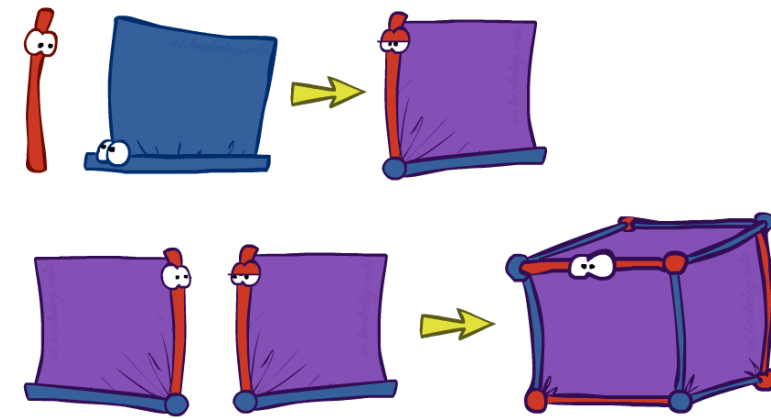
Join T



$R, T, L$

$P(R, T, L)$

+r	+t	+l	0.024
+r	+t	-l	0.056
+r	-t	+l	0.002
+r	-t	-l	0.018
-r	+t	+l	0.027
-r	+t	-l	0.063
-r	-t	+l	0.081
-r	-t	-l	0.729



# Operation 2: eliminate

Second basic operation: **marginalization**

Take a factor and sum out a variable

- Shrinks a factor to a smaller one
- A **projection** operation

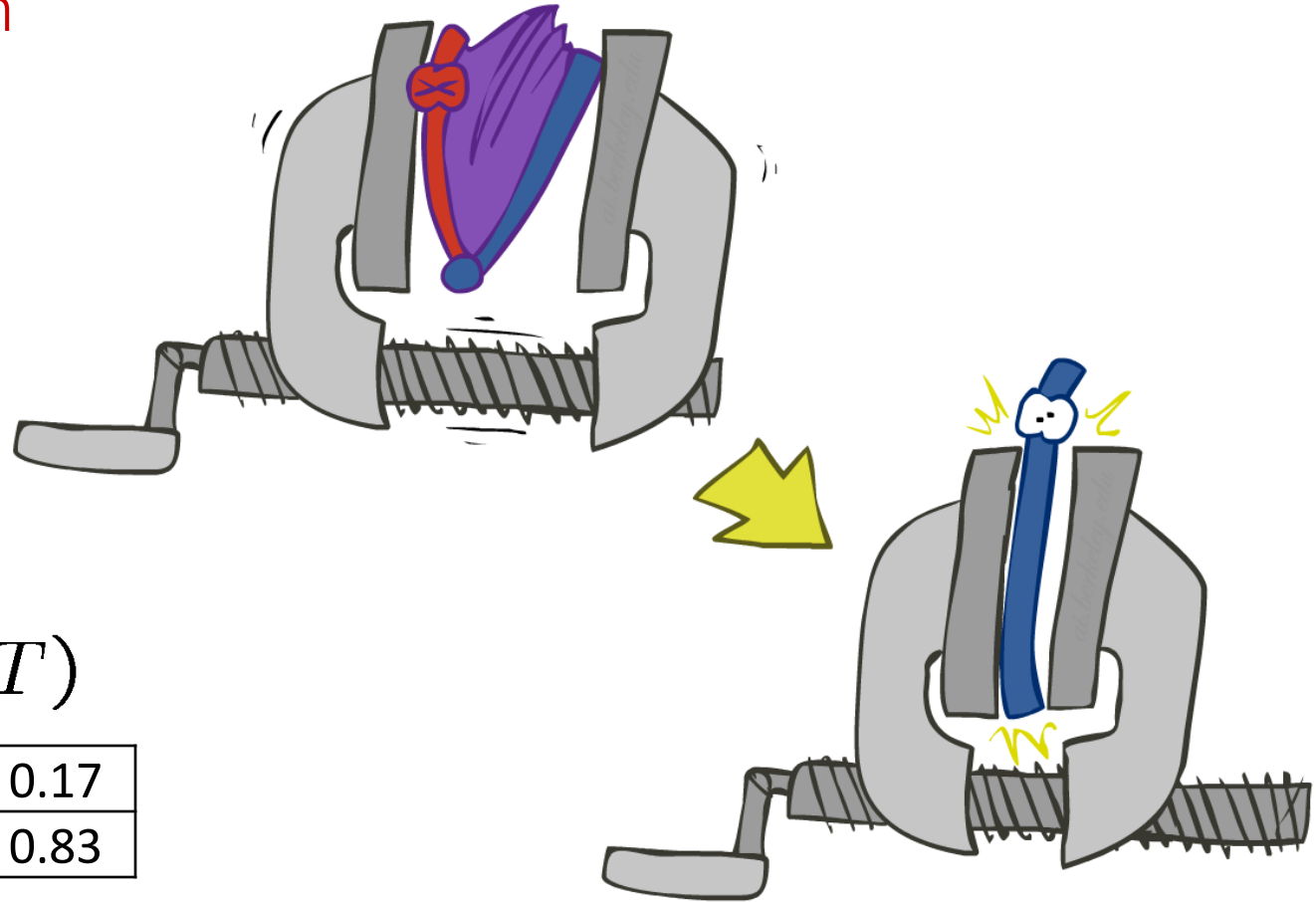
Example:

$P(R, T)$		
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81

sum  $R$



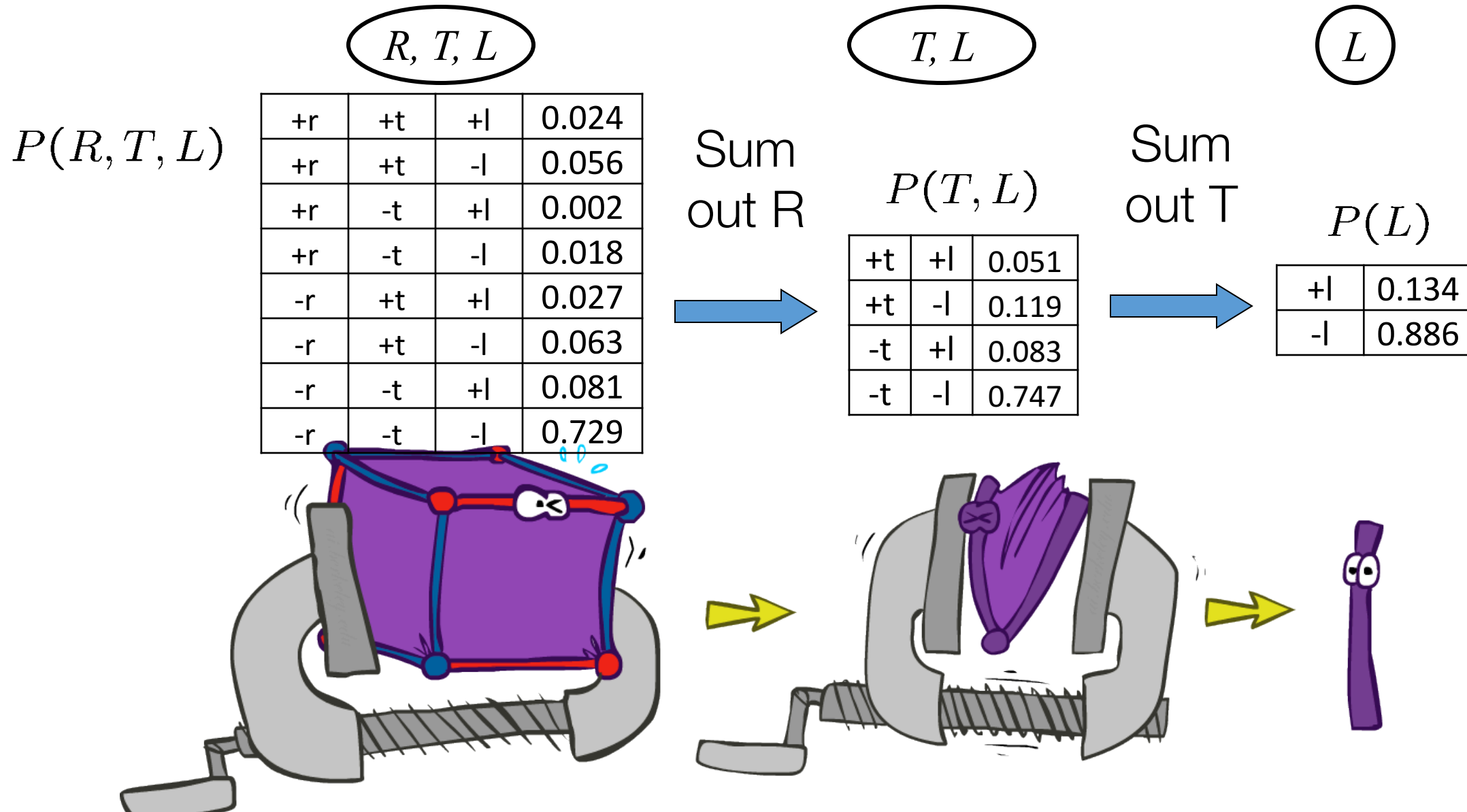
$P(T)$	
+t	0.17
-t	0.83



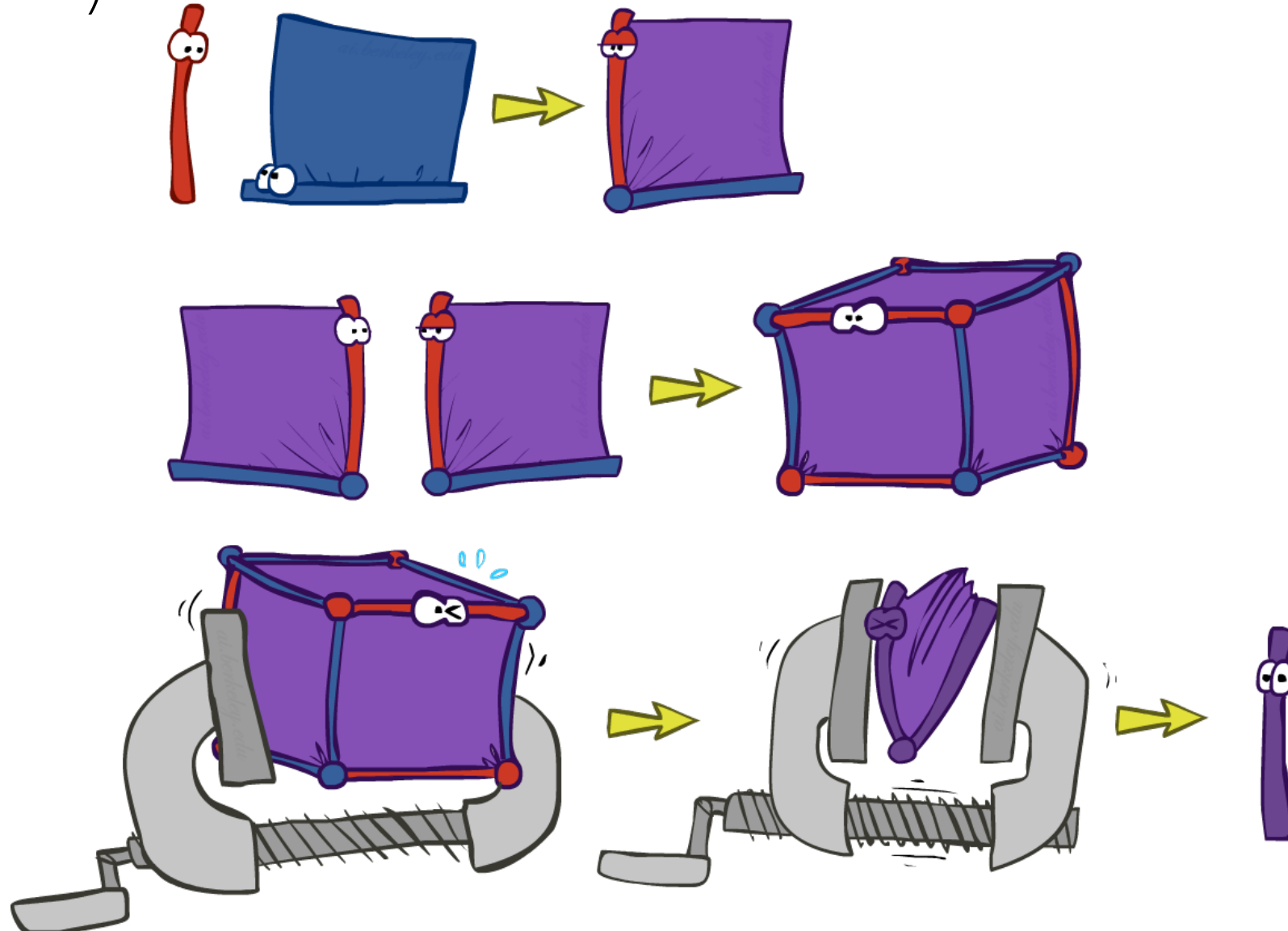


# Multiple elimination

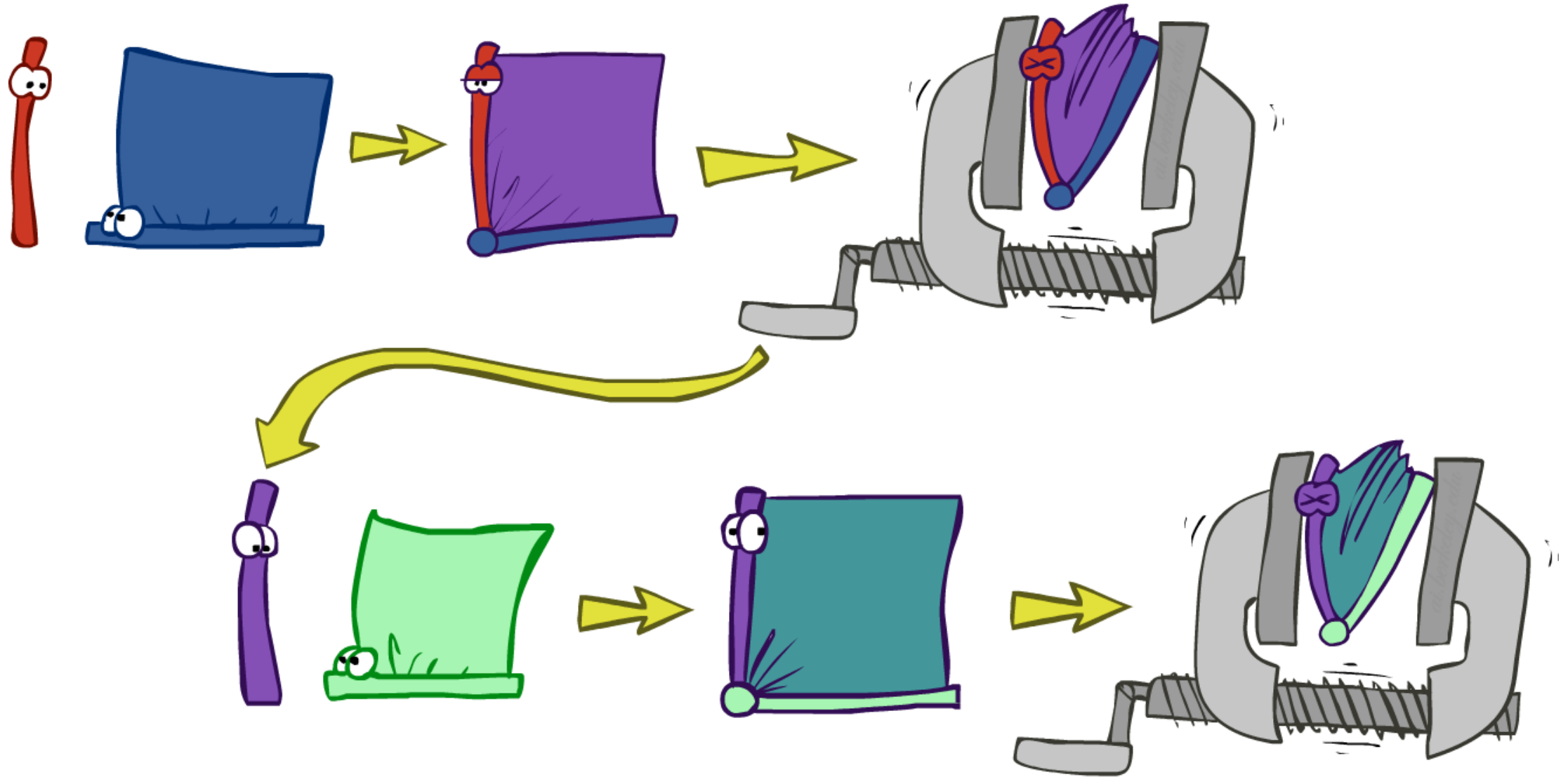
$$P(L) = ?$$



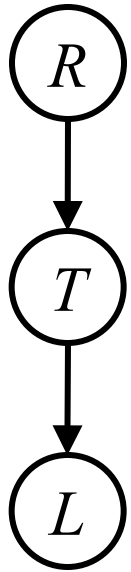
Thus far: multiple join, multiple eliminate (= inference by enumeration)



Marginalizing early (= variable elimination)



# Traffic domain



$$P(L) = ?$$

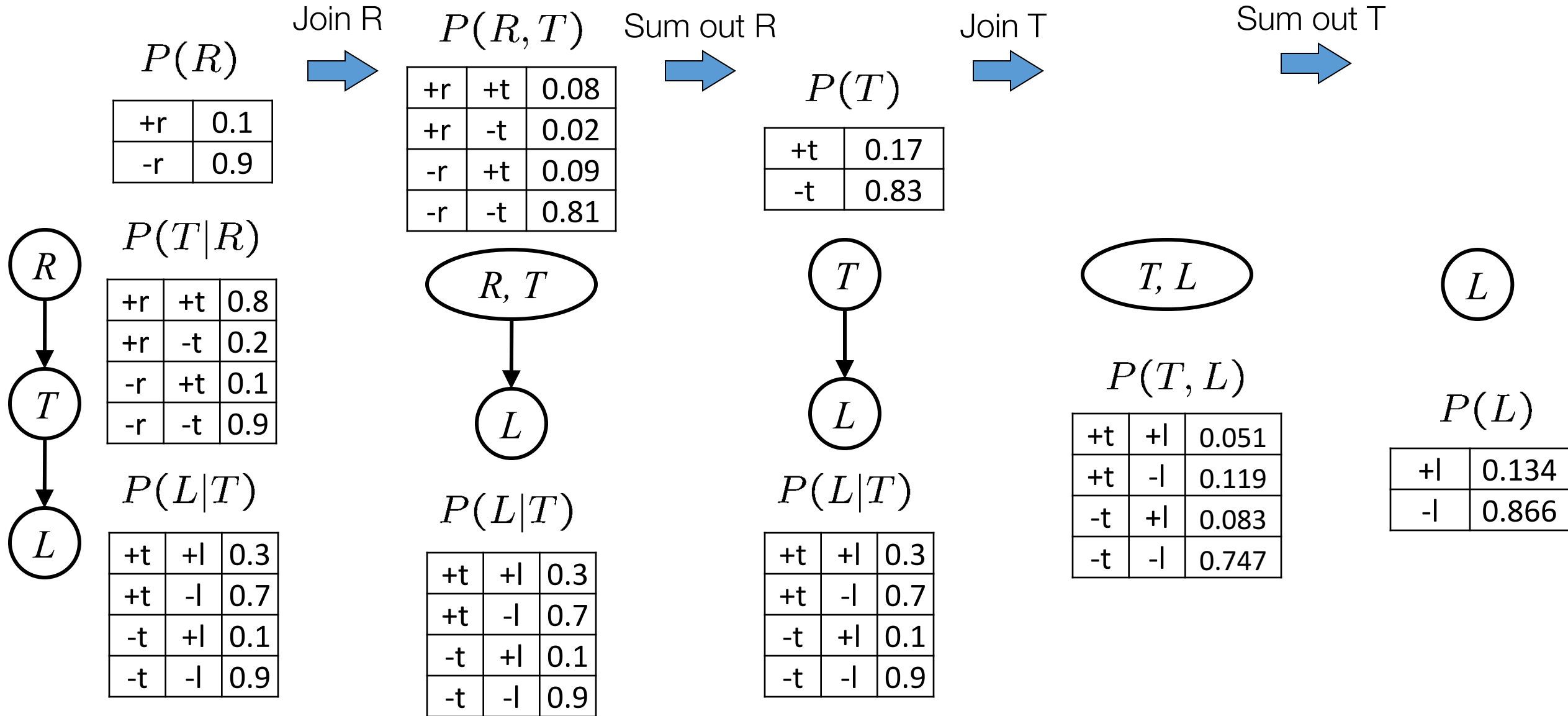
Inference by Enumeration

$$= \sum_t \sum_r \underbrace{P(L|t)P(r)P(t|r)}_{\text{Join on } r} \underbrace{\phantom{P(L|t)P(r)P(t|r)}}_{\text{Join on } t} \underbrace{\phantom{P(L|t)P(r)P(t|r)}}_{\text{Eliminate } r} \underbrace{\phantom{P(L|t)P(r)P(t|r)}}_{\text{Eliminate } t}$$

Variable Elimination

$$= \sum_t P(L|t) \underbrace{\sum_r P(r)P(t|r)}_{\text{Join on } r} \underbrace{\phantom{\sum_r P(r)P(t|r)}}_{\text{Eliminate } r} \underbrace{\phantom{\sum_r P(r)P(t|r)}}_{\text{Join on } t} \underbrace{\phantom{\sum_r P(r)P(t|r)}}_{\text{Eliminate } t}$$

# Marginalizing early! (aka VE)



# Evidence

If evidence, start with factors that select that evidence

- *No evidence* uses these initial factors:

$$P(R)$$

+r	0.1
-r	0.9

$$P(T|R)$$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$$P(L|T)$$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- With evidence, +r, the initial factors become:

$$P(+r)$$

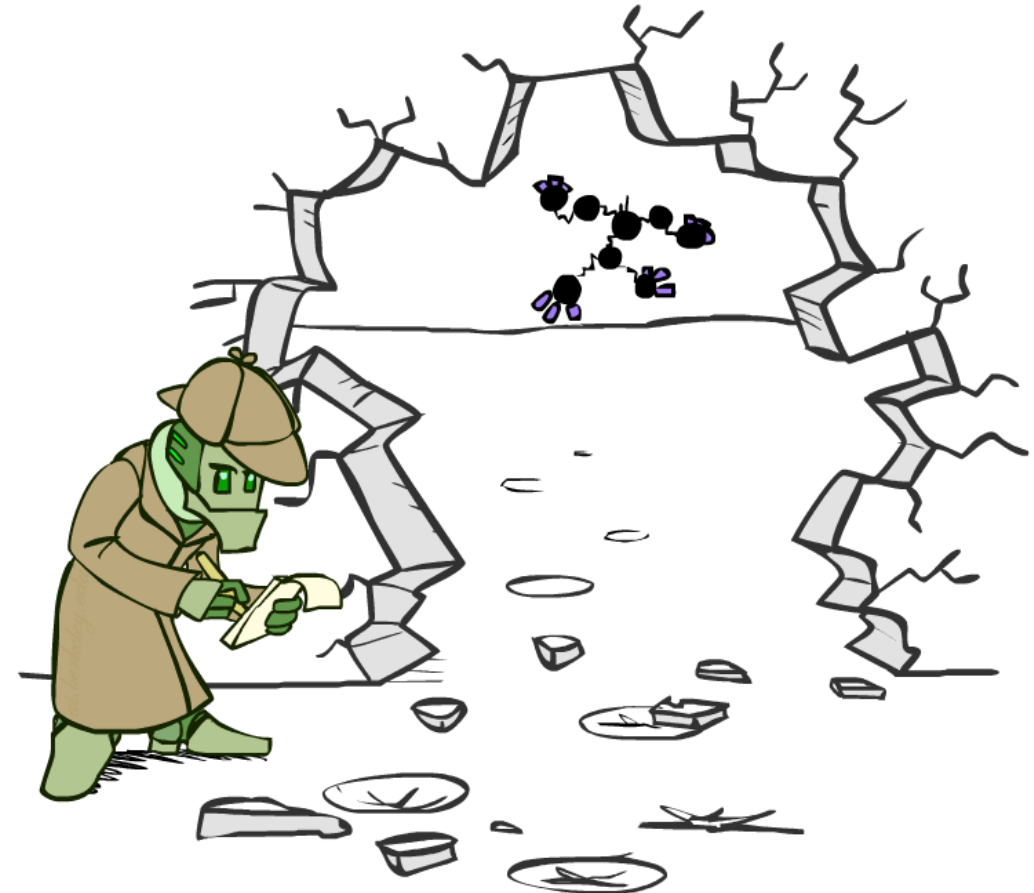
+r	0.1
----	-----

$$P(T|+r)$$

+r	+t	0.8
+r	-t	0.2

$$P(L|T)$$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9



We eliminate all vars other than query + evidence

# Evidence

Result will be a selected joint of query and evidence

- E.g. for  $P(L \mid +r)$ , we would end up with:

$$P(+r, L)$$

+r	+l	0.026
+r	-l	0.074

Normalize

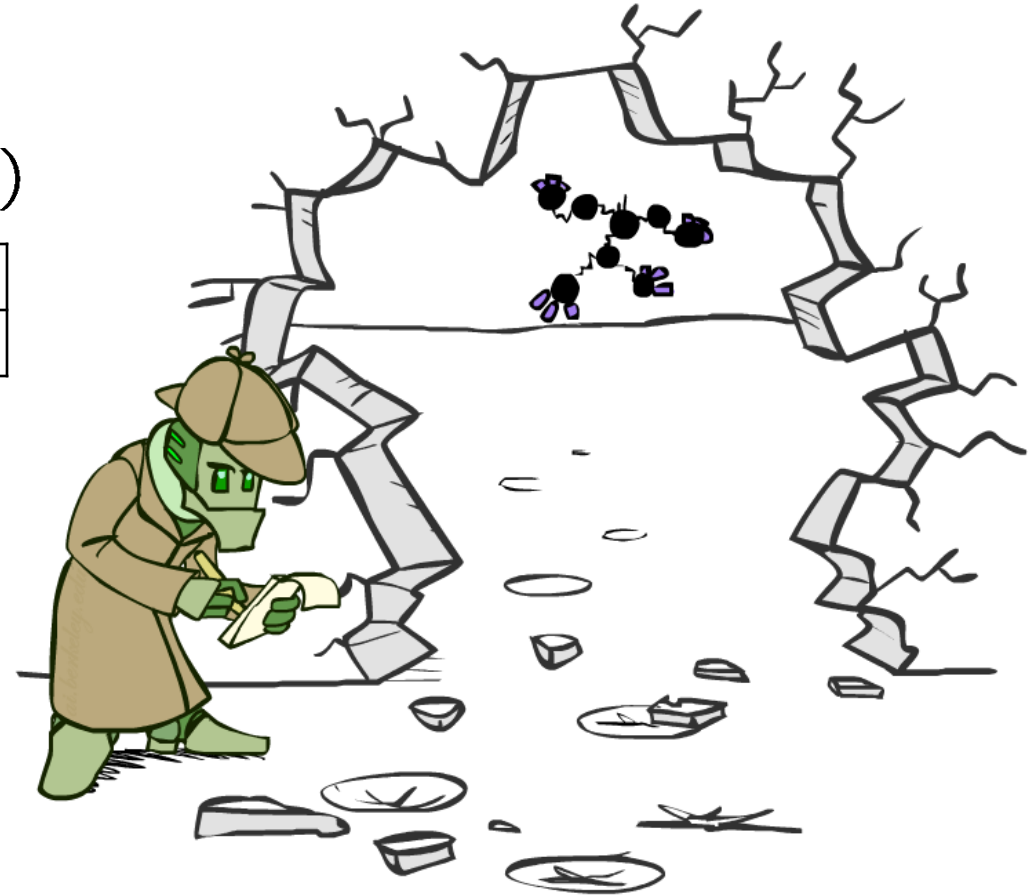


$$P(L \mid +r)$$

+l	0.26
-l	0.74

To get our answer, just normalize this!

That 's it!

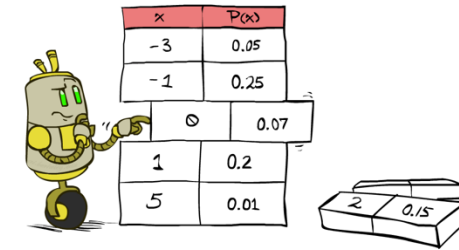


# General variable elimination

Query:  $P(Q|E_1 = e_1, \dots, E_k = e_k)$

Start with initial factors:

- Local CPTs (but instantiated by evidence)

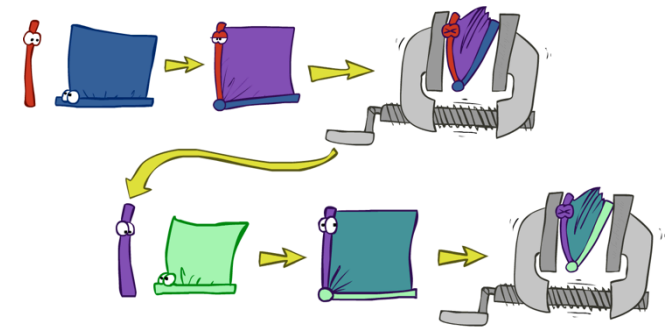


x	P(x)
-3	0.05
-1	0.25
0	0.07
1	0.2
5	0.01

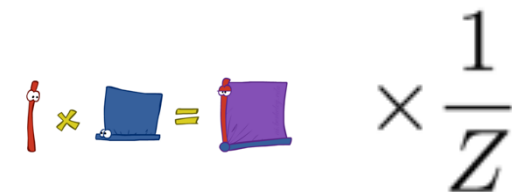
2 0.15

While there are still hidden variables (not Q or evidence):

- Pick a hidden variable H
- Join all factors mentioning H
- Eliminate (sum out) H



Join all remaining factors and normalize


$$\text{red stick} \times \text{blue square} = \text{purple square} \times \frac{1}{Z}$$

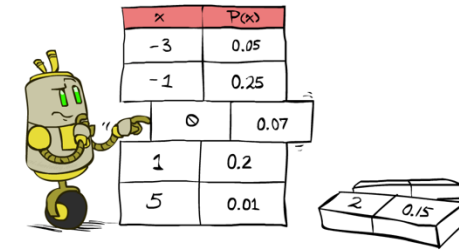


# General variable elimination

Query:  $P(Q|E_1 = e_1, \dots, E_k = e_k)$

Start with initial factors:

- Local CPTs (but instantiated by evidence)

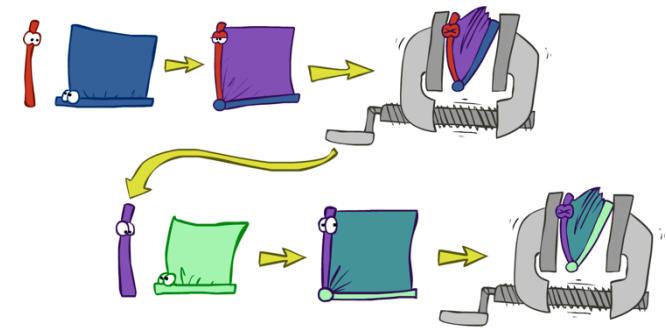


x	P(x)
-3	0.05
-1	0.25
0	0.07
1	0.2
5	0.01

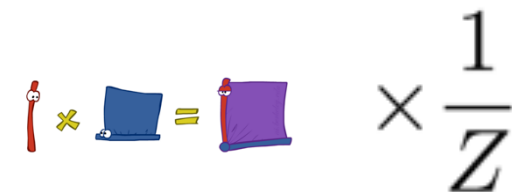
2 0.15

While there are still hidden variables (not Q or evidence):

- **Pick a hidden variable H**
- Join all factors mentioning H
- Eliminate (sum out) H



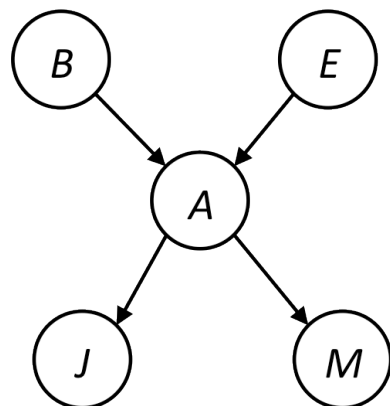
Join all remaining factors and normalize


$$\text{red stick figure} \times \text{blue square} = \text{purple square} \times \frac{1}{Z}$$

Example (on board first)

$$P(B|j, m) \propto P(B, j, m)$$

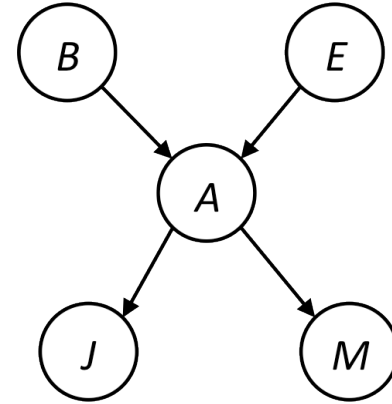
$P(B)$	$P(E)$	$P(A B, E)$	$P(j A)$	$P(m A)$
--------	--------	-------------	----------	----------



# Example

$$P(B|j, m) \propto P(B, j, m)$$

$P(B)$	$P(E)$	$P(A B, E)$	$P(j A)$	$P(m A)$
--------	--------	-------------	----------	----------

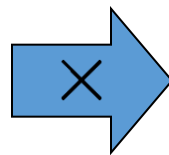


Choose A

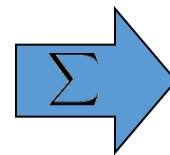
$$P(A|B, E)$$

$$P(j|A)$$

$$P(m|A)$$



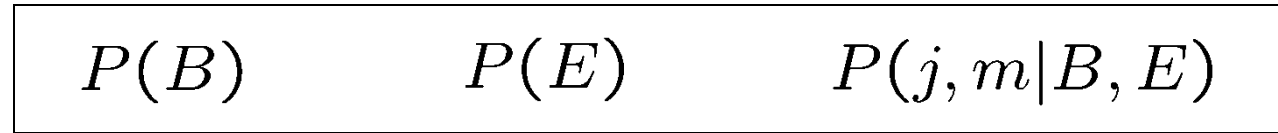
$$P(j, m, A|B, E)$$



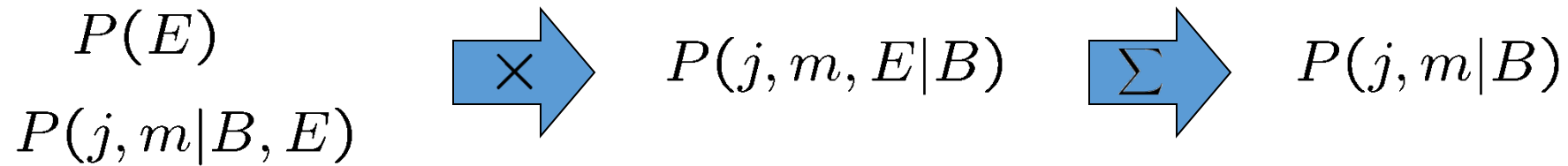
$$P(j, m|B, E)$$

$P(B)$	$P(E)$	$P(j, m B, E)$
--------	--------	----------------

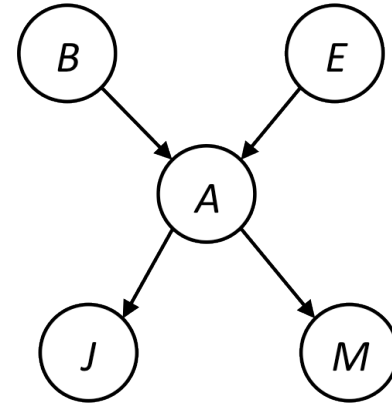
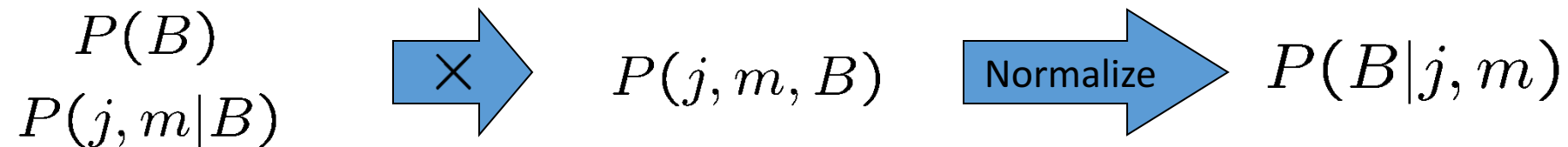
# Example



Next choose E



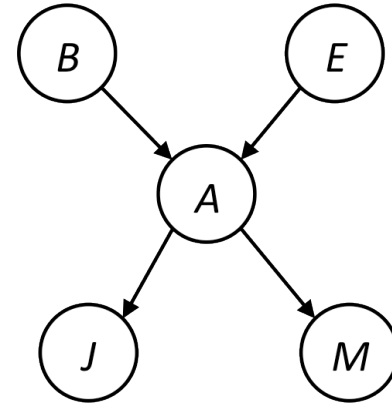
Finish with B



# Same example in equations

$$P(B|j, m) \propto P(B, j, m)$$

$P(B)$	$P(E)$	$P(A B, E)$	$P(j A)$	$P(m A)$
--------	--------	-------------	----------	----------



$$\begin{aligned} P(B|j, m) &\propto P(B, j, m) \\ &= \sum_{e, a} P(B, j, m, e, a) \\ &= \sum_{e, a} P(B)P(e)P(a|B, e)P(j|a)P(m|a) \\ &= \sum_e P(B)P(e) \sum_a P(a|B, e)P(j|a)P(m|a) \\ &= \sum_e P(B)P(e)f_1(B, e, j, m) \\ &= P(B) \sum_e P(e)f_1(B, e, j, m) \\ &= P(B)f_2(B, j, m) \end{aligned}$$

marginal can be obtained from joint by summing out

use Bayes' net joint distribution expression

use  $x^*(y+z) = xy + xz$  (to sum out over  $a$ !)

joining on  $a$ , and then summing out gives  $f_1$

use  $x^*(y+z) = xy + xz$

joining on  $e$ , and then summing out gives  $f_2$

# Another variable elimination example

Query:  $P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$

Start by inserting evidence, which gives the following initial factors:

$$p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)$$

Eliminate  $X_1$ , this introduces the factor  $f_1(Z, y_1) = \sum_{x_1} p(x_1|Z)p(y_1|x_1)$ , and we are left with:

$$p(Z)f_1(Z, y_1)p(X_2|Z)p(X_3|Z)p(y_2|X_2)p(y_3|X_3)$$

Eliminate  $X_2$ , this introduces the factor  $f_2(Z, y_2) = \sum_{x_2} p(x_2|Z)p(y_2|x_2)$ , and we are left with:

$$p(Z)f_1(Z, y_1)f_2(Z, y_2)p(X_3|Z)p(y_3|X_3)$$

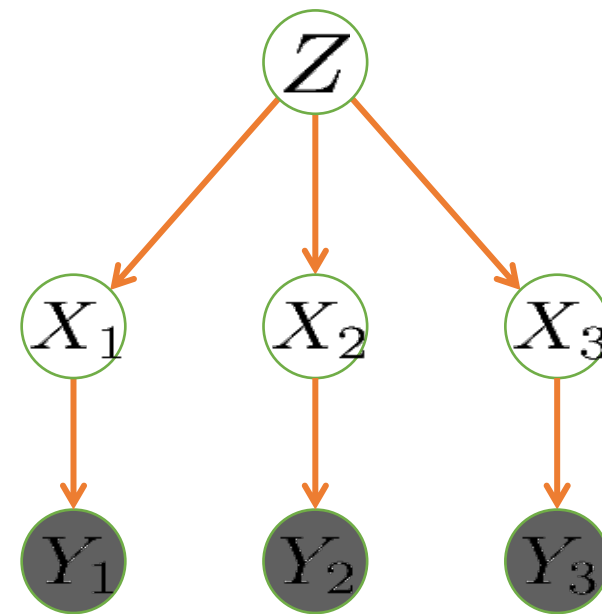
Eliminate  $Z$ , this introduces the factor  $f_3(y_1, y_2, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z)$ , and we are left:

$$p(y_3|X_3), f_3(y_1, y_2, X_3)$$

No hidden variables left. Join the remaining factors to get:

$$f_4(y_1, y_2, y_3, X_3) = P(y_3|X_3)f_3(y_1, y_2, X_3).$$

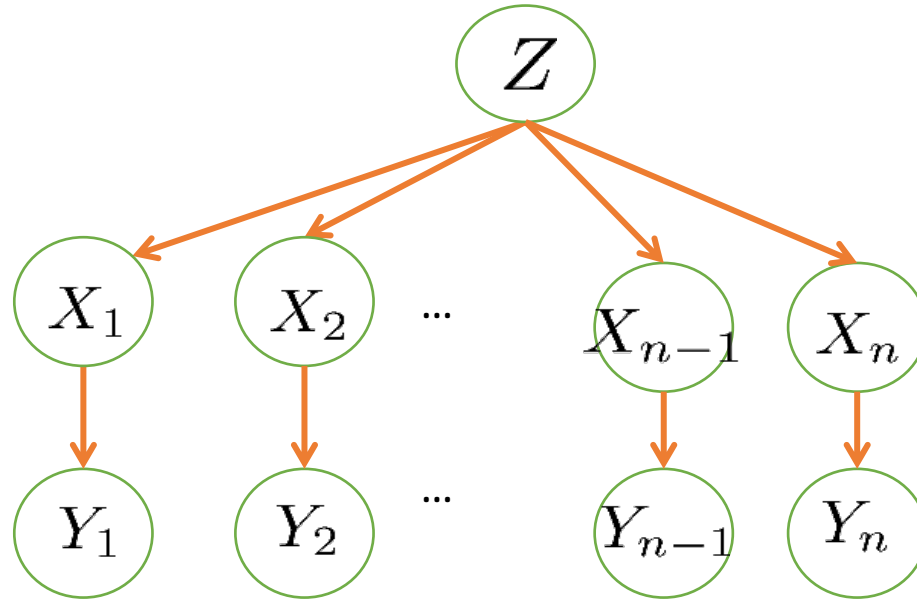
Normalizing over  $X_3$  gives  $P(X_3|y_1, y_2, y_3)$ .



Computational complexity critically depends on the largest factor being generated in this process. Size of factor = number of entries in table. In example above (assuming binary) all factors generated are of size 2 --- as they all only have one variable ( $Z$ ,  $Z$ , and  $X_3$  respectively).

# Variable elimination ordering

- For the query  $P(X_n | y_1, \dots, y_n)$  work through the following two different orderings as done in previous slide:  $Z, X_1, \dots, X_{n-1}$  and  $X_1, \dots, X_{n-1}, Z$ . What is the size of the maximum factor generated for each of the orderings?



- Answer:  $2^{n+1}$  versus  $2^2$  (assuming binary)
- In general: the ordering can greatly affect efficiency.

# VE: computational and space complexity

The computational and space complexity of variable elimination is determined by the largest factor

The elimination ordering can greatly affect the size of the largest factor.

- E.g., previous slide's example  $2^n$  vs. 2

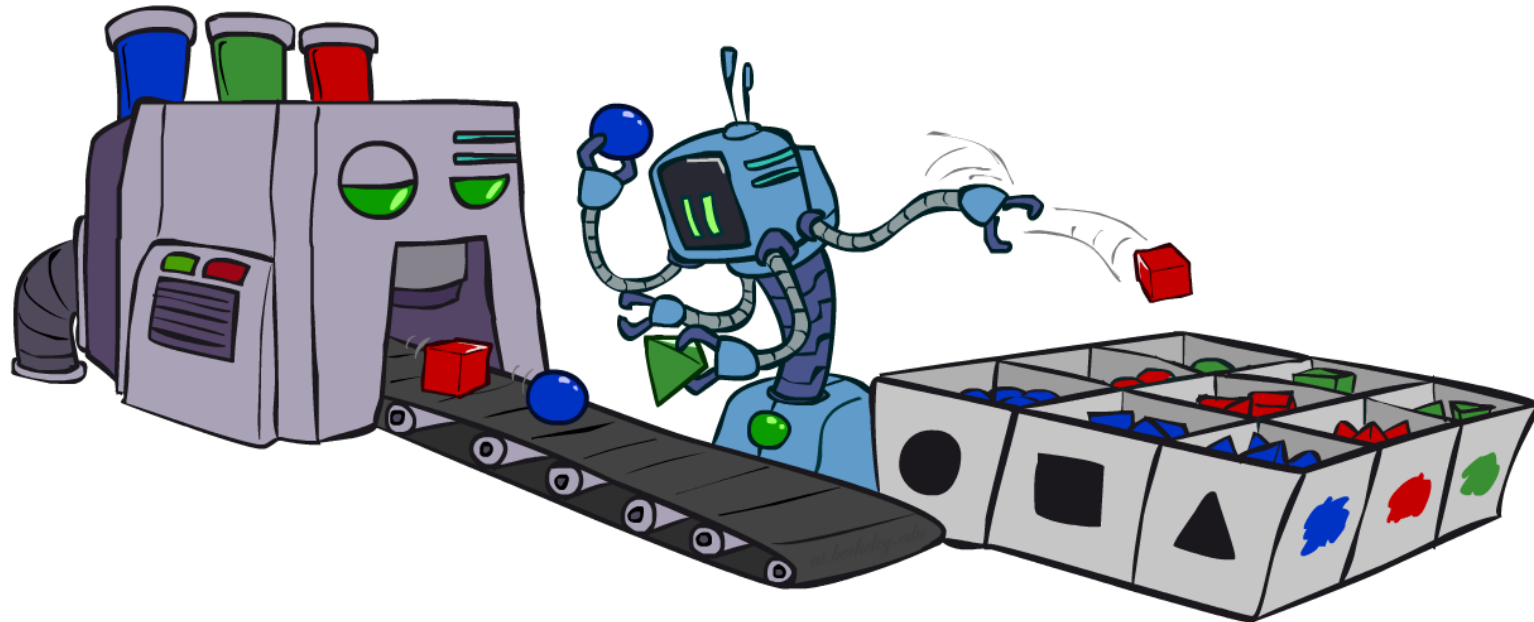
Does there always exist an ordering that only results in small factors?

- **No!**



# Bayes nets: so far

- Last time: representation and semantics
- Thus far today: inference. Remaining: approximate inference (and learning) via **sampling**.



# Sampling

Sampling is a lot like repeated simulation

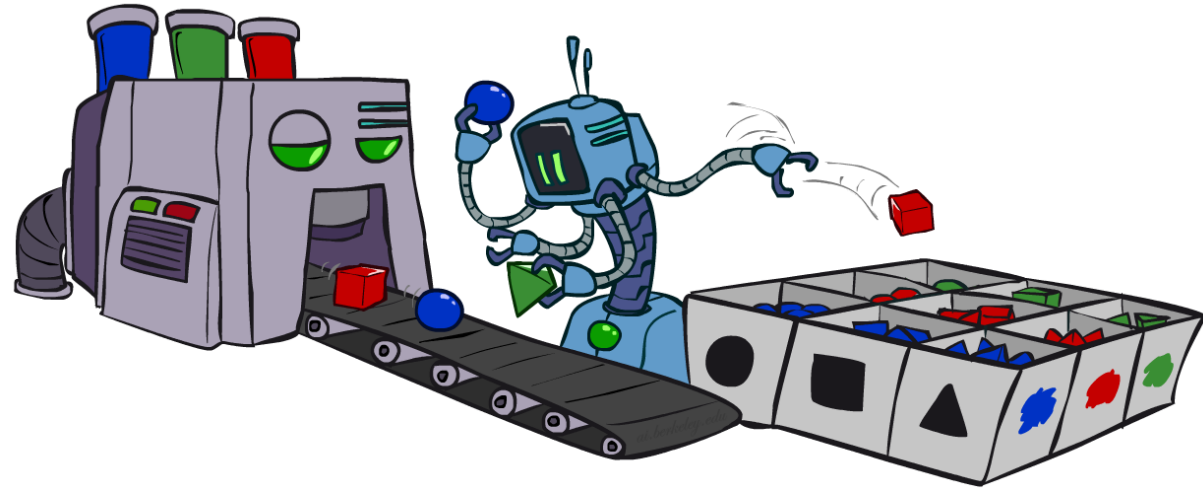
- Predicting the weather, basketball games, ...

Basic idea

- Draw  $N$  samples from a sampling distribution  $S$
- Compute an approximate posterior probability
- Show this converges to the true probability  $P$

Why sample?

- Learning: get samples from a distribution you don't know
- Inference: getting a sample is faster than computing the right answer (e.g. with variable elimination)



# Sampling

## Sampling from given distribution

- Step 1: Get sample  $u$  from uniform distribution over  $[0, 1)$ 
  - E.g. `random()` in python
- Step 2: Convert this sample  $u$  into an outcome for the given distribution by having each outcome associated with a sub-interval of  $[0, 1)$  with sub-interval size equal to probability of the outcome

## Example

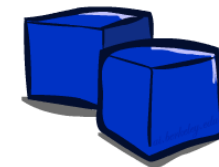
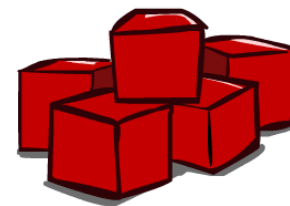
C	P(C)
red	0.6
green	0.1
blue	0.3

$$0 \leq u < 0.6, \rightarrow C = \text{red}$$

$$0.6 \leq u < 0.7, \rightarrow C = \text{green}$$

$$0.7 \leq u < 1, \rightarrow C = \text{blue}$$

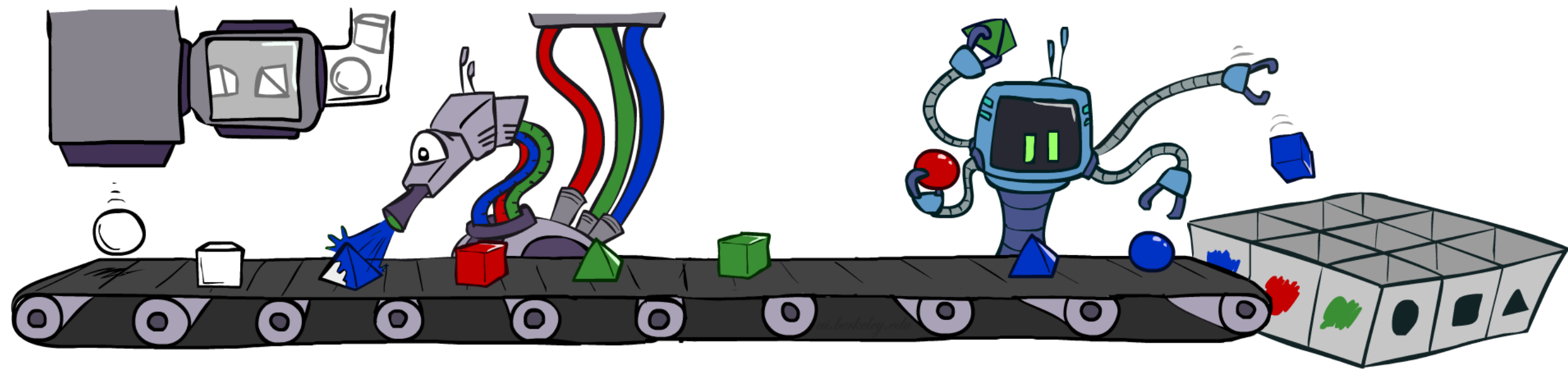
- If `random()` returns  $u = 0.83$ , then our sample is  $C = \text{blue}$
- E.g, after sampling 8 times:



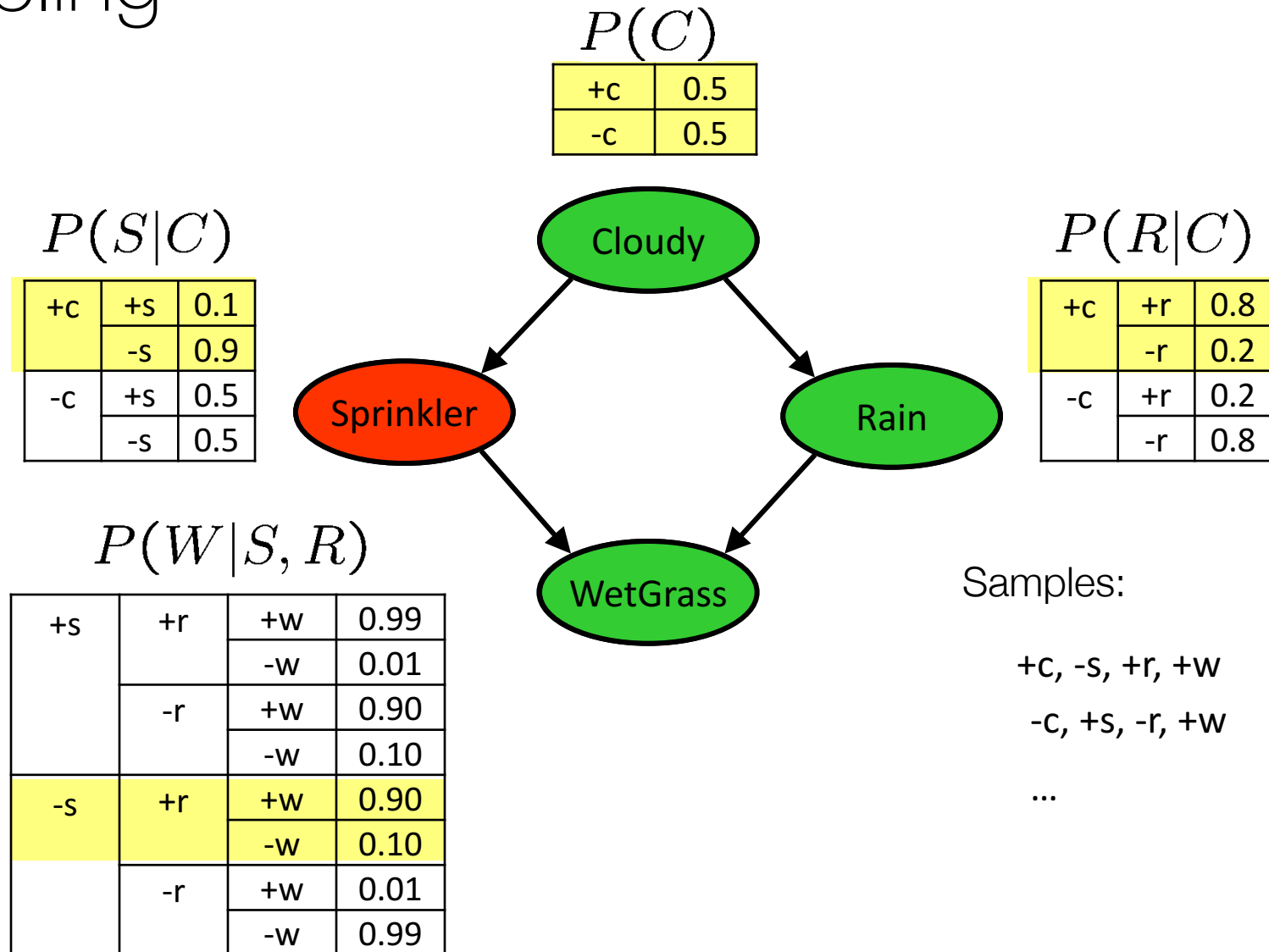
# Sampling in Bayes' nets

- Prior Sampling
- Rejection Sampling
- Likelihood Weighting
- Gibbs Sampling

# Prior sampling



# Prior sampling

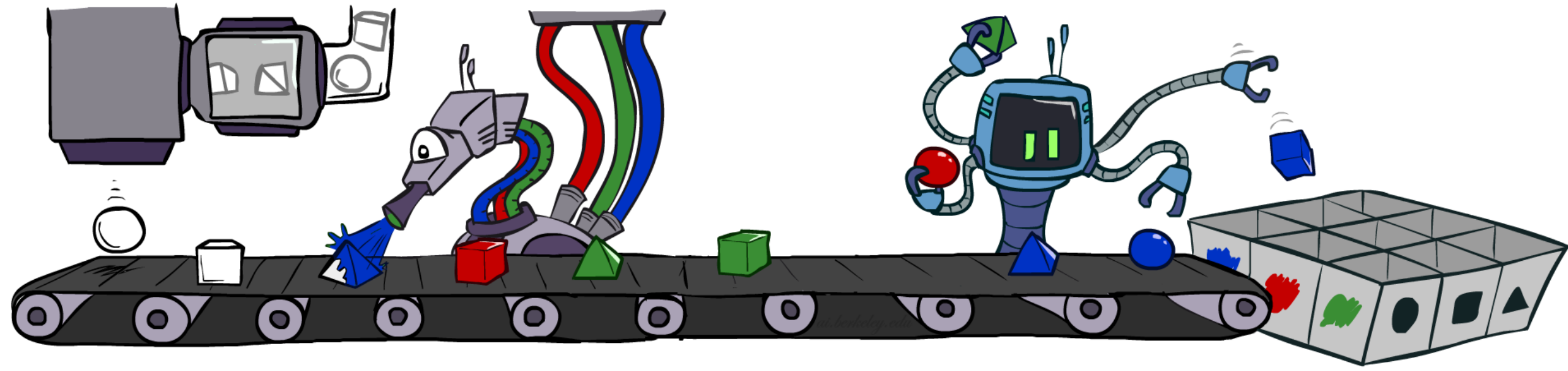


# Prior sampling

For  $i=1, 2, \dots, n$

Sample  $x_i$  from  $P(X_i \mid \text{Parents}(X_i))$

Return  $(x_1, x_2, \dots, x_n)$



# Prior sampling

This process generates samples with probability:

$$S_{PS}(x_1 \dots x_n) = \prod_{i=1}^n P(x_i | \text{Parents}(X_i)) = P(x_1 \dots x_n)$$

...i.e. the BN' s joint probability

Let the number of samples of an event be  $N_{PS}(x_1 \dots x_n)$

$$\begin{aligned} \text{Then } \lim_{N \rightarrow \infty} \hat{P}(x_1, \dots, x_n) &= \lim_{N \rightarrow \infty} N_{PS}(x_1, \dots, x_n) / N \\ &= S_{PS}(x_1, \dots, x_n) \\ &= P(x_1 \dots x_n) \end{aligned}$$

i.e., the sampling procedure is **consistent**



# Example

We'll get a bunch of samples from the BN:

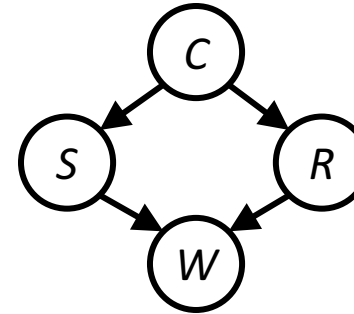
+C, -S, +r, +W

+C, +S, +r, +W

-C, +S, +r, -W

+C, -S, +r, +W

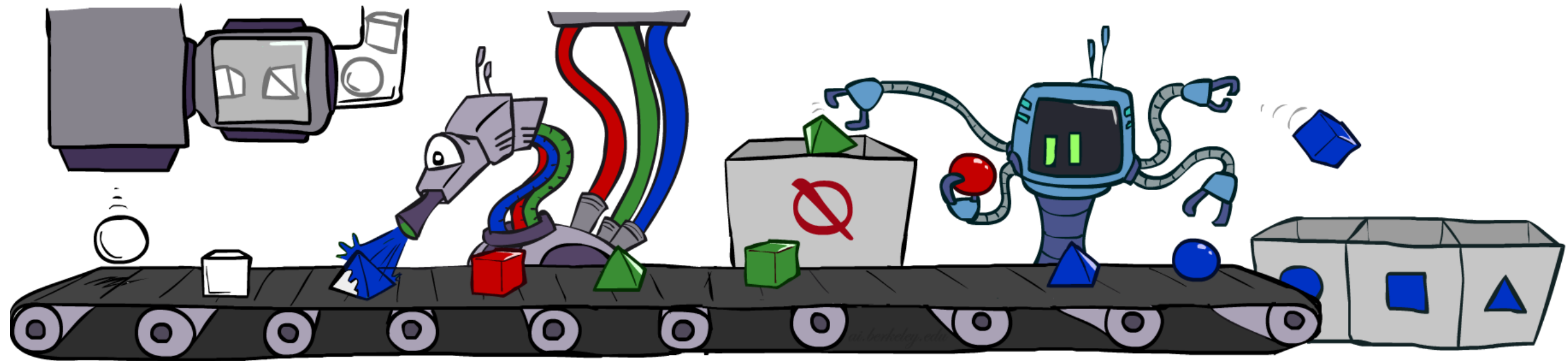
-C, -S, -r, +W



If we want to know  $P(W)$

- We have counts  $\langle +w:4, -w:1 \rangle$
- Normalize to get  $P(W) = \langle +w:0.8, -w:0.2 \rangle$
- This will get closer to the true distribution with more samples
- Can estimate anything else, too
- What about  $P(C \mid +w)$ ?  $P(C \mid +r, +w)$ ?  $P(C \mid -r, -w)$ ?
- Fast: can use fewer samples if less time (what's the drawback?)

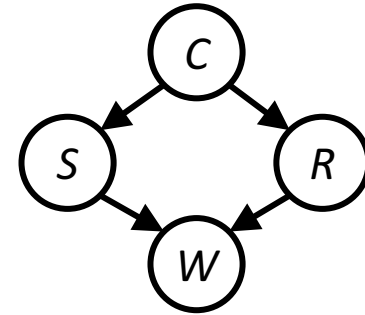
# Rejection sampling



# Rejection sampling

Let's say we want  $P(C)$

- No point keeping all samples around
- Just tally counts of  $C$  as we go



Let's say we want  $P(C \mid +s)$

- Same thing: tally  $C$  outcomes, but ignore (reject) samples which don't have  $S=+s$
- This is called rejection sampling
- It is also consistent for conditional probabilities (i.e., correct in the limit)

+c, -s, +r, +w  
+c, +s, +r, +w  
-c, +s, +r, -w  
+c, -s, +r, +w  
-c, -s, -r, +w

# Rejection sampling

IN: evidence instantiation

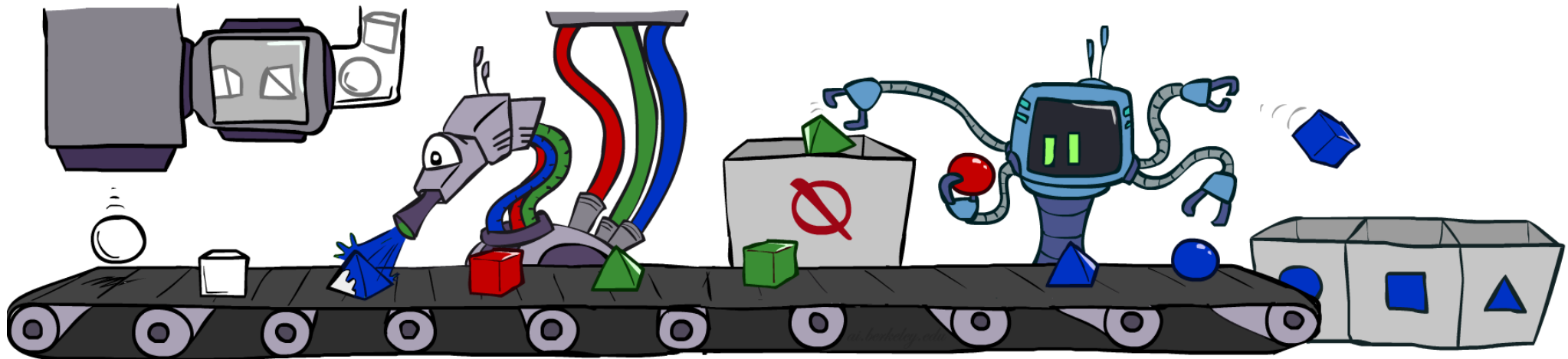
For  $i=1, 2, \dots, n$

Sample  $x_i$  from  $P(X_i \mid \text{Parents}(X_i))$

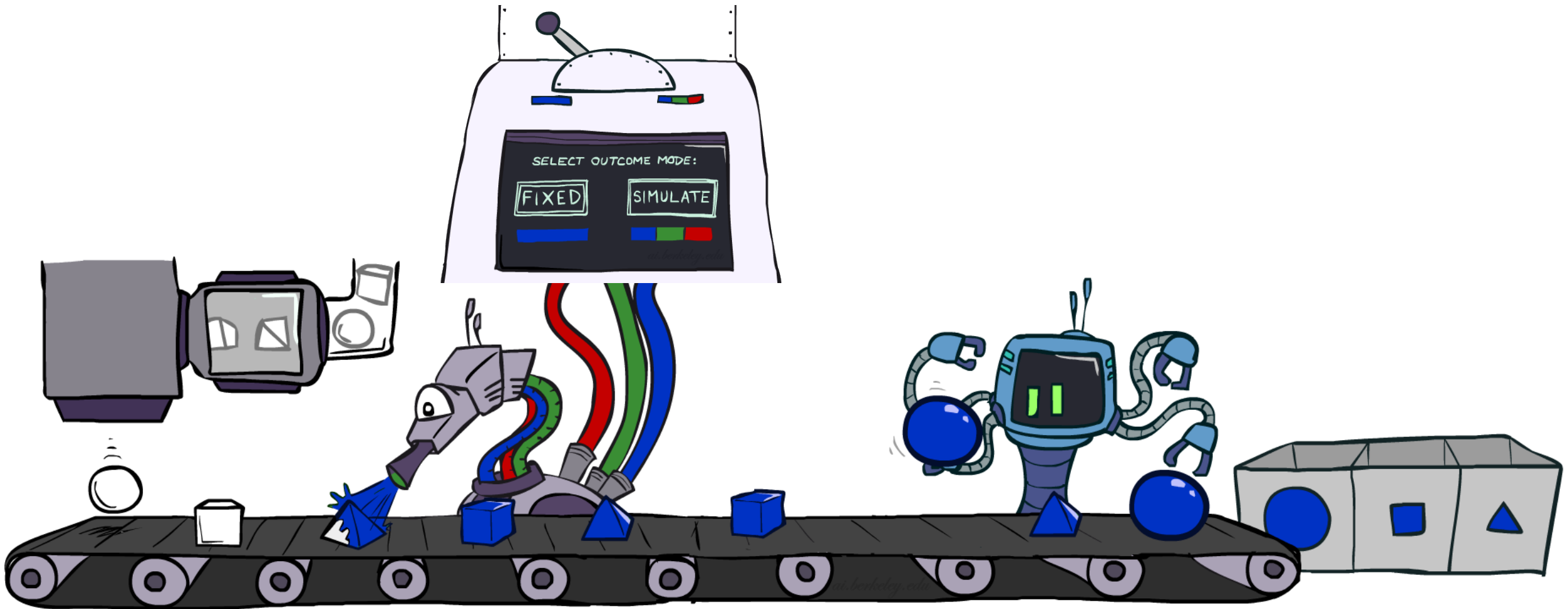
If  $x_i$  not consistent with evidence

Reject: Return, and no sample is generated in this cycle

Return  $(x_1, x_2, \dots, x_n)$



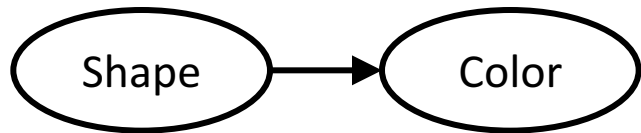
# Likelihood weighting



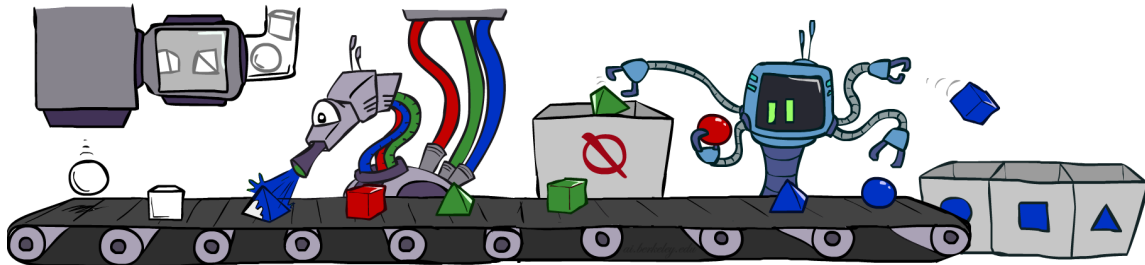
# Likelihood weighting

Problem with rejection sampling:

- If evidence is unlikely, rejects lots of samples
- Evidence not exploited as you sample
- Consider  $P(\text{Shape}|\text{blue})$

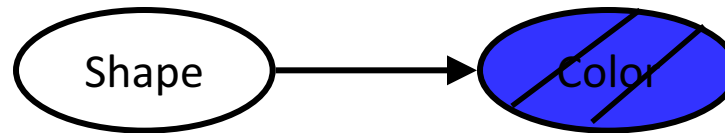


~~pyramid, green~~  
~~pyramid, red~~  
sphere, blue  
~~cube, red~~  
~~sphere, green~~

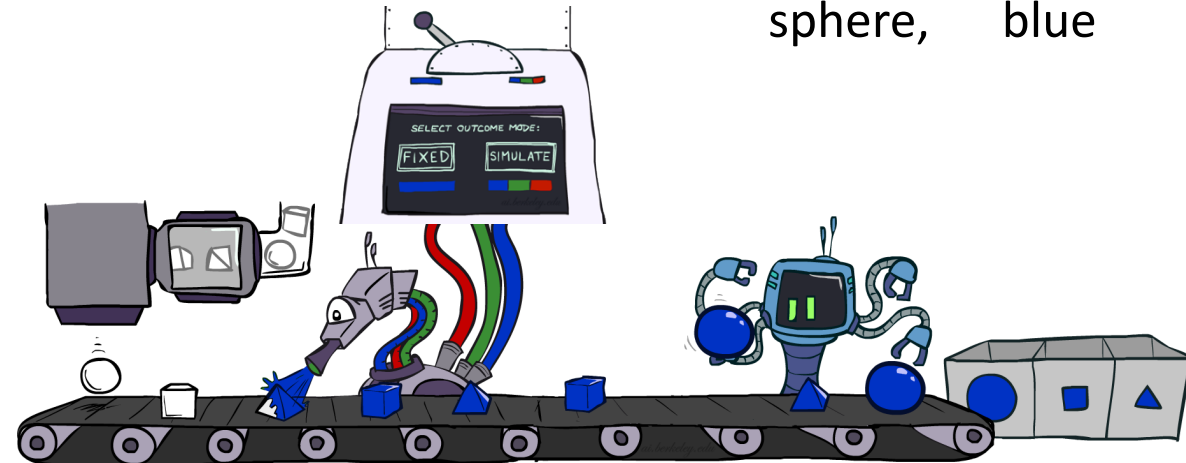


Idea: fix evidence variables and sample the rest

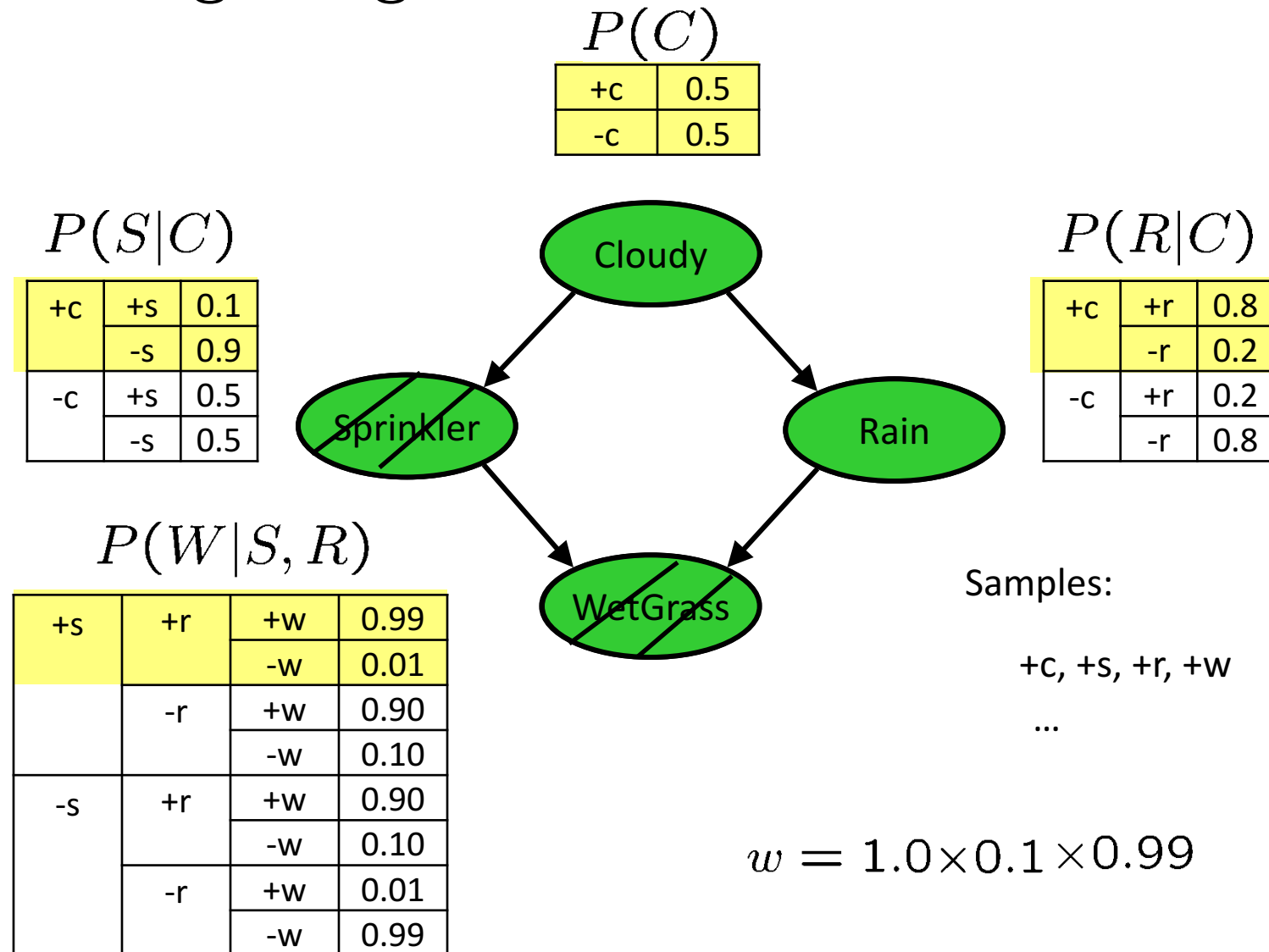
- Problem: sample distribution not consistent!
- Solution: weight by probability of evidence given parents



pyramid, blue  
pyramid, blue  
sphere, blue  
cube, blue  
sphere, blue



# Likelihood weighting



# Likelihood weighting

```
IN: evidence instantiation
w = 1.0
for i=1, 2, ..., n
    if  $X_i$  is an evidence variable
         $X_i = \text{observation } x_i \text{ for } X_i$ 
        Set  $w = w * P(x_i \mid \text{Parents}(X_i))$ 
    else
        Sample  $x_i$  from  $P(X_i \mid \text{Parents}(X_i))$ 
return  $(x_1, x_2, \dots, x_n), w$ 
```



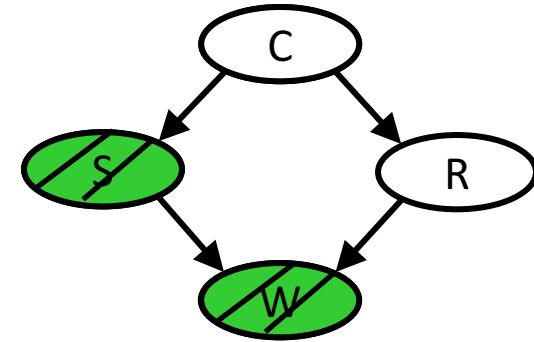
# Likelihood weighting

Sampling distribution if  $z$  sampled and  $e$  fixed evidence

$$S_{WS}(z, e) = \prod_{i=1}^l P(z_i | \text{Parents}(Z_i))$$

Now, samples have weights

$$w(z, e) = \prod_{i=1}^m P(e_i | \text{Parents}(E_i))$$



Together, weighted sampling distribution is consistent

$$\begin{aligned} S_{WS}(z, e) \cdot w(z, e) &= \prod_{i=1}^l P(z_i | \text{Parents}(z_i)) \prod_{i=1}^m P(e_i | \text{Parents}(e_i)) \\ &= P(z, e) \end{aligned}$$

# Likelihood weighting

Likelihood weighting is good

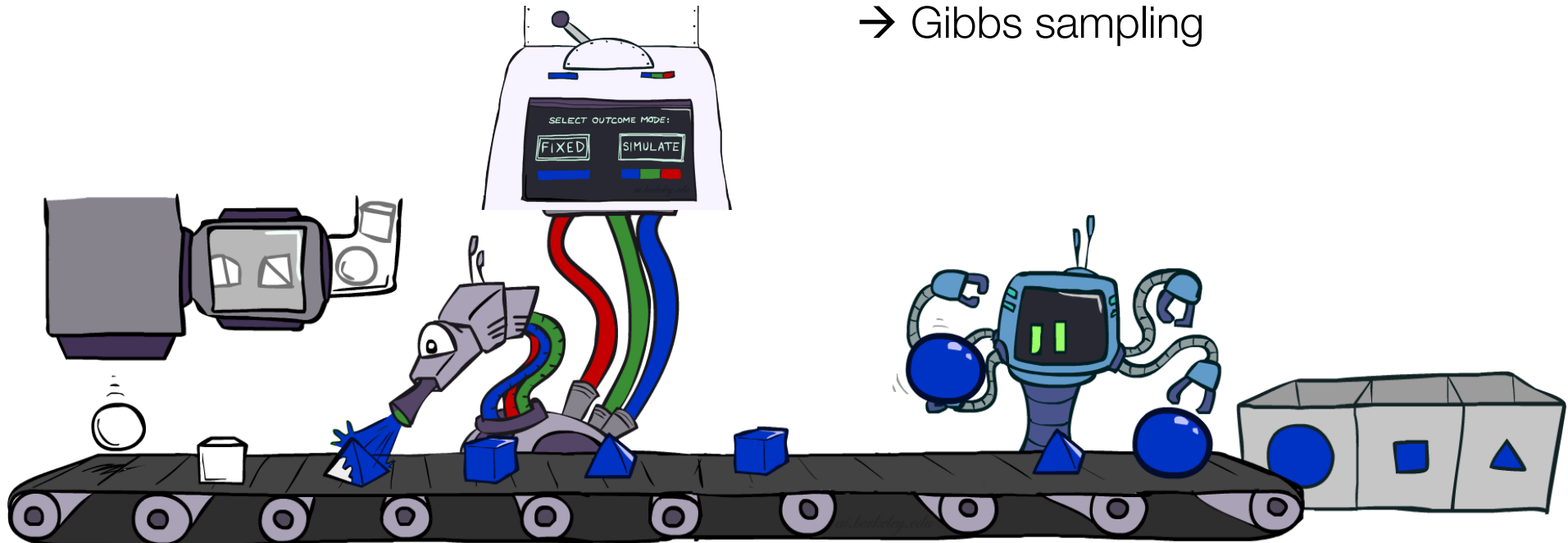
- We have taken evidence into account as we generate the sample
- E.g. here,  $W$ 's value will get picked based on the evidence values of  $S, R$
- More of our samples will reflect the state of the world suggested by the evidence

Likelihood weighting doesn't solve all our problems

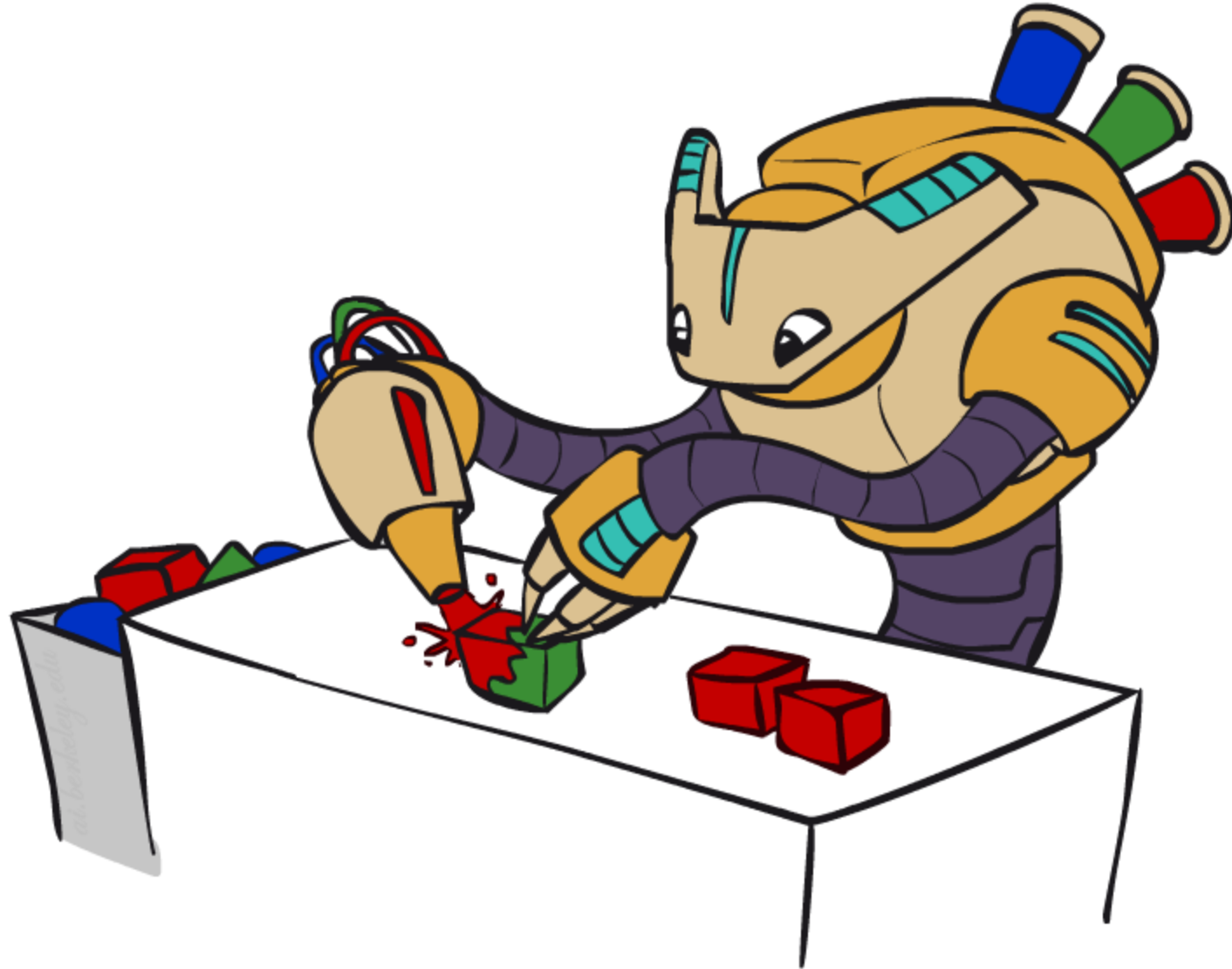
- Evidence influences the choice of downstream variables, but not upstream ones ( $C$  isn't more likely to get a value matching the evidence)

We would like to consider evidence when we sample every variable

→ Gibbs sampling



# Gibbs sampling



# Gibbs sampling

- *Procedure:* keep track of a full instantiation  $x_1, x_2, \dots, x_n$ . Start with an arbitrary instantiation consistent with the evidence. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed. Keep repeating this for a long time.

# Gibbs sampling

- *Procedure:* keep track of a full instantiation  $x_1, x_2, \dots, x_n$ . Start with an arbitrary instantiation consistent with the evidence. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed. Keep repeating this for a long time.
- *Property:* in the limit of repeating this infinitely many times the resulting sample is coming from the correct distribution

# Gibbs sampling

- *Procedure:* keep track of a full instantiation  $x_1, x_2, \dots, x_n$ . Start with an arbitrary instantiation consistent with the evidence. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed. Keep repeating this for a long time.
- *Property:* in the limit of repeating this infinitely many times the resulting sample is coming from the correct distribution
- *Rationale:* both upstream and downstream variables condition on evidence.

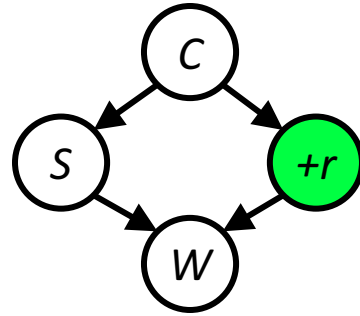
# Gibbs sampling

- *Procedure*: keep track of a full instantiation  $x_1, x_2, \dots, x_n$ . Start with an arbitrary instantiation consistent with the evidence. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed. Keep repeating this for a long time.
- *Property*: in the limit of repeating this infinitely many times the resulting sample is coming from the correct distribution
- *Rationale*: both upstream and downstream variables condition on evidence.
- In contrast: likelihood weighting only conditions on upstream evidence, and hence weights obtained in likelihood weighting can sometimes be very small. Sum of weights over all samples is indicative of how many “effective” samples were obtained, so want high weight.

# Gibbs sampling example: $P(S \mid +r)$

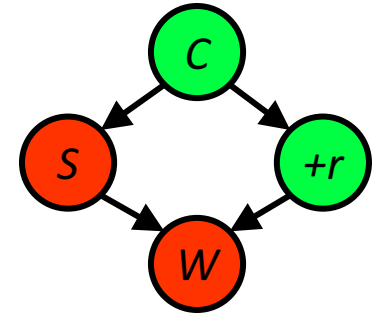
Step 1: Fix evidence

$R = +r$



Step 2: Initialize other variables

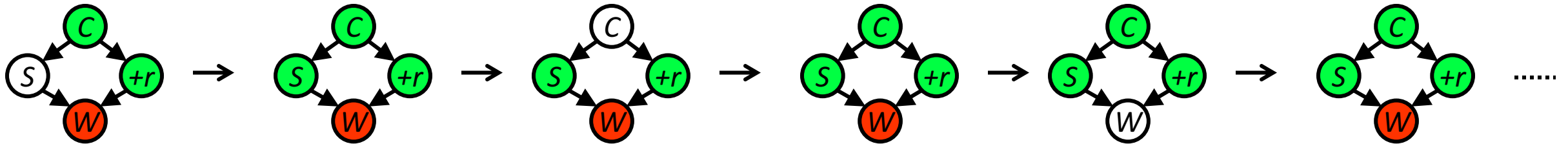
Randomly



Steps 3: Repeat

Choose a non-evidence variable  $X$

Resample  $X$  from  $P(X \mid \text{all other variables})$



Sample from  $P(S \mid +c, -w, +r)$

Sample from  $P(C \mid +s, -w, +r)$

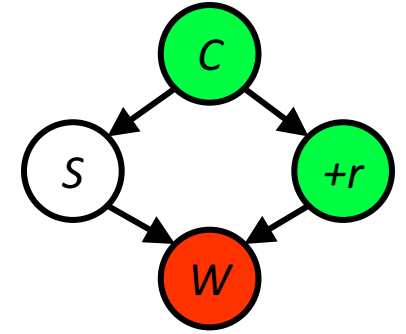
Sample from  $P(W \mid +s, +c, +r)$



# Efficient resampling of one variable

Sample from  $P(S \mid +c, +r, -w)$

$$\begin{aligned} P(S \mid +c, +r, -w) &= \frac{P(S, +c, +r, -w)}{P(+c, +r, -w)} \\ &= \frac{P(S, +c, +r, -w)}{\sum_s P(s, +c, +r, -w)} \\ &= \frac{P(+c)P(S \mid +c)P(+r \mid +c)P(-w \mid S, +r)}{\sum_s P(+c)P(s \mid +c)P(+r \mid +c)P(-w \mid s, +r)} \\ &= \frac{P(+c)P(S \mid +c)P(+r \mid +c)P(-w \mid S, +r)}{P(+c)P(+r \mid +c) \sum_s P(s \mid +c)P(-w \mid s, +r)} \\ &= \frac{P(S \mid +c)P(-w \mid S, +r)}{\sum_s P(s \mid +c)P(-w \mid s, +r)} \end{aligned}$$

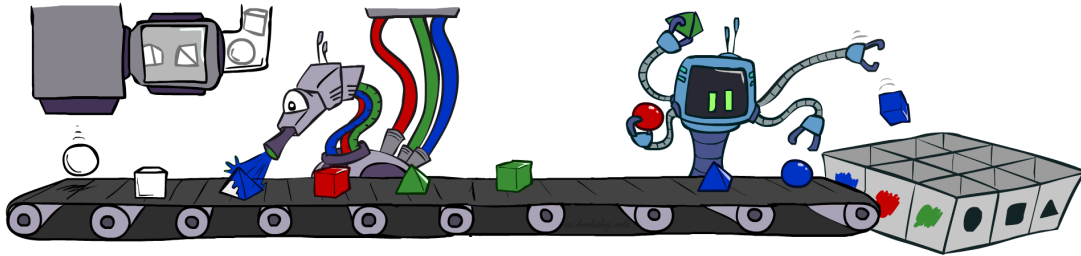


Many things cancel out – only CPTs with S remain!

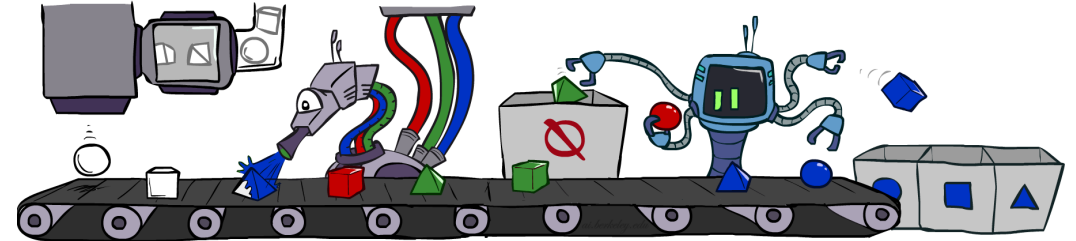
More generally: only CPTs that have resampled variable need to be considered, and joined together

# Bayes' net sampling summary

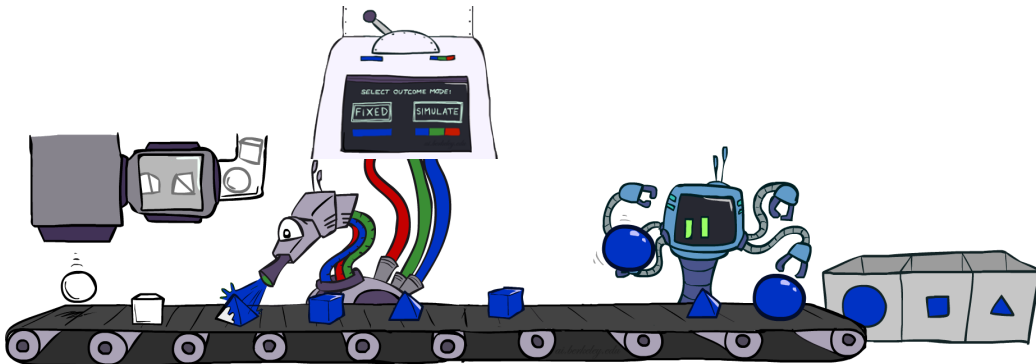
Prior Sampling  $P$



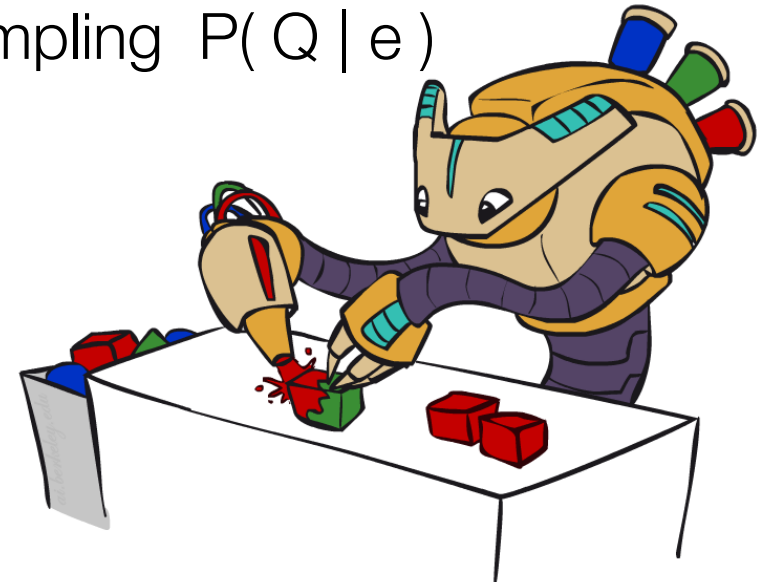
Rejection Sampling  $P(Q | e)$



Likelihood Weighting  $P(Q | e)$



Gibbs Sampling  $P(Q | e)$



OK, that's all for today!

- Up next: supervised machine learning!