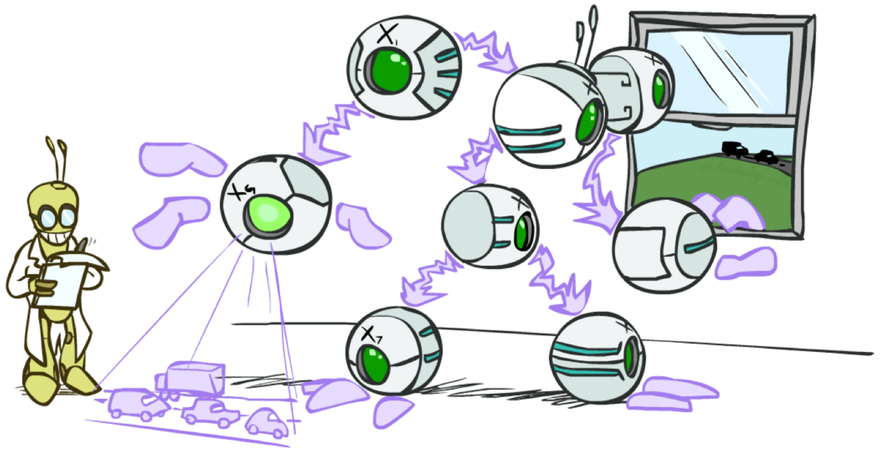
CSE 573: Artificial Intelligence

Hanna Hajishirzi

Bayes Net Inference

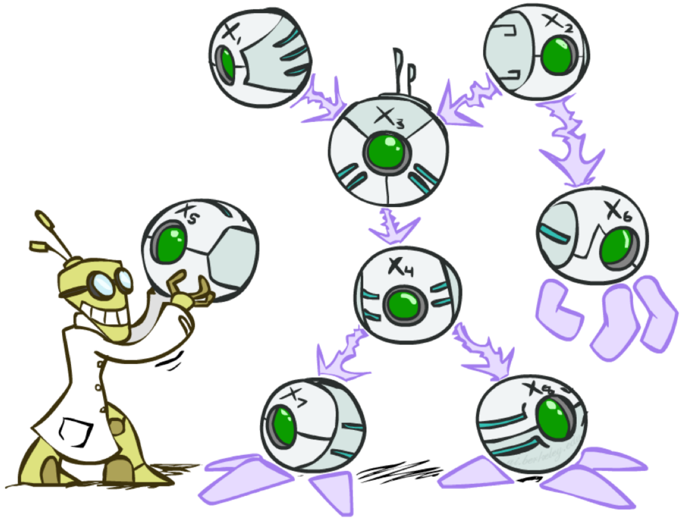


slides adapted from
Dan Klein, Pieter Abbeel ai.berkeley.edu
And Dan Weld, Luke Zettlemoyer

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' values

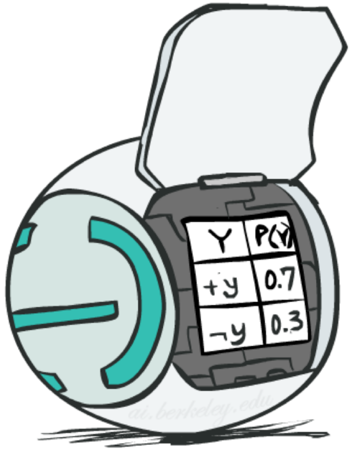
$$P(X|a_1 \ldots a_n)$$



- 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:

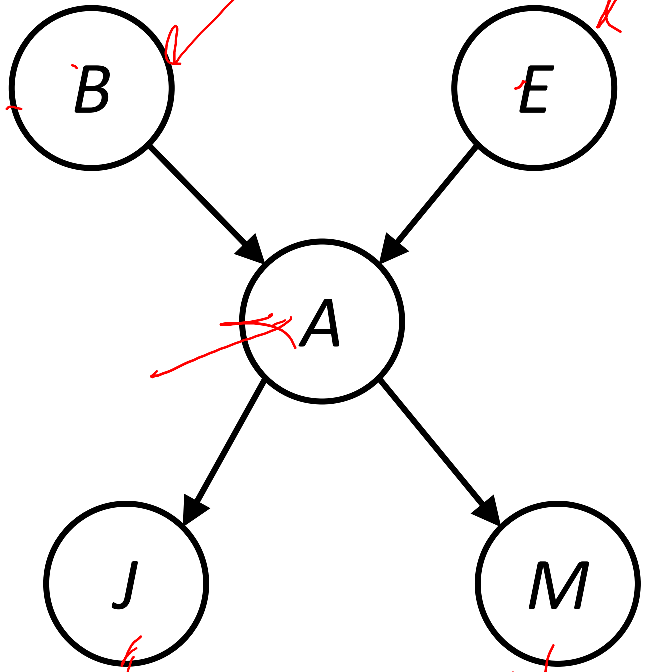
$P(X_i | X_1 \ldots X_{i-1})$

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

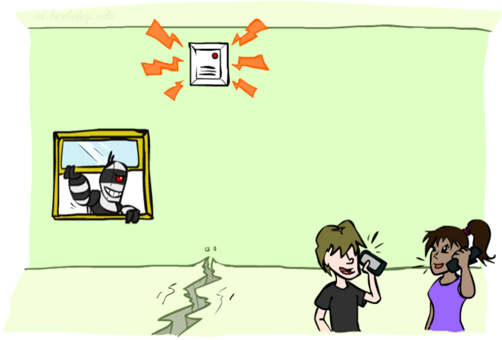


Example: Alarm Network

B	P(B)
+b	0.001
-b	0.999



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

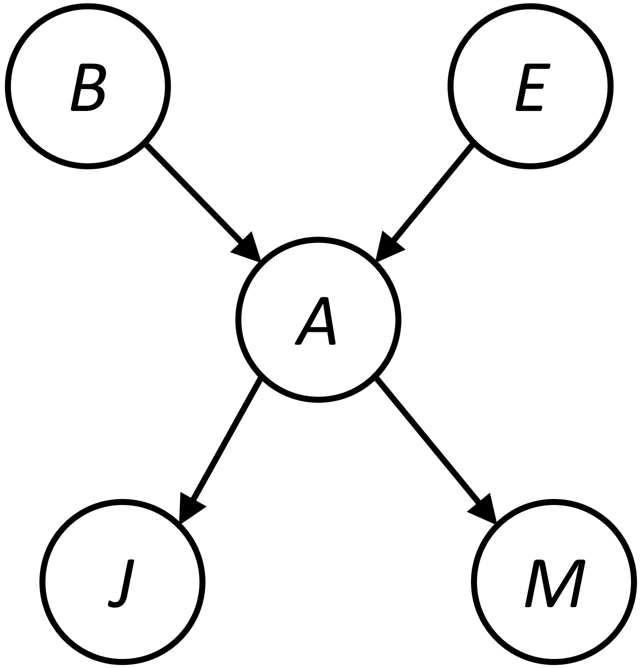
B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	-a	0.05
+b	-e	+a	0.94
+b	-e	-a	0.06
-b	+e	+a	0.29
-b	+e	-a	0.71
-b	-e	+a	0.001
-b	-e	-a	0.999

$$P(+b, -e, +a, -j, +m) =$$

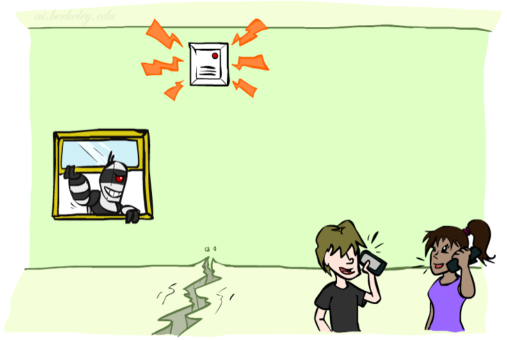
$P(+b)P(-e)P(+a|-e,b)P(-j|a)P(+m|a)$

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

$$P(+b, -e, +a, -j, +m) =$$

$$P(+b)P(-e)P(+a|+b, -e)P(-j|+a)P(+m|+a) =$$

$$0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7$$

Announcements

- Remaining lectures:
  - Today: Inference in BNs
  - Fri: Machine Learning and Neural Net Overview
  - Next Wed: More applied
    - Sequential Neural vs. HMMs
    - Application: Language models and Machine Translation
  - Next Fri: Poster session
    - Stay tuned: might do it virtual

Inference

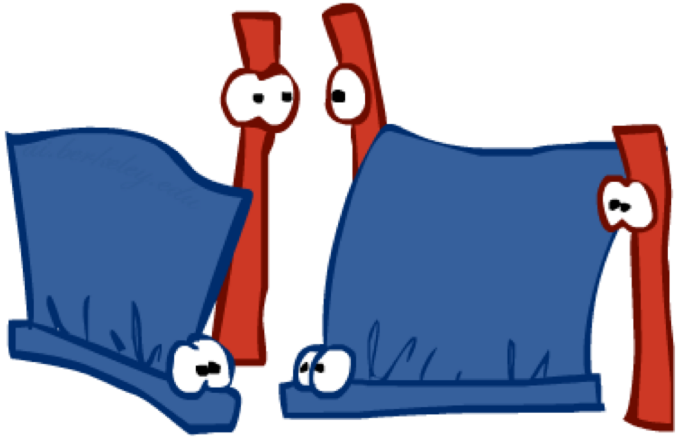
- Inference: calculating some useful quantity from a joint probability distribution

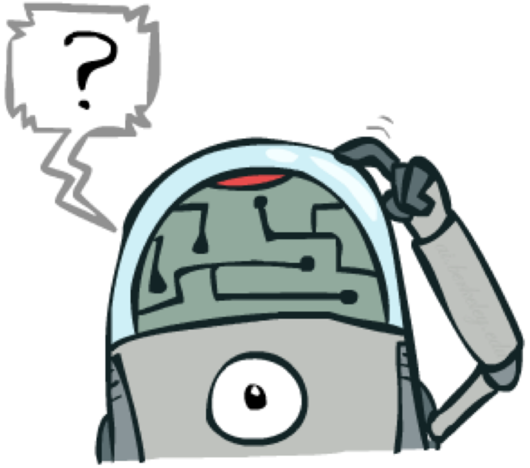
- Examples:
  - Posterior probability

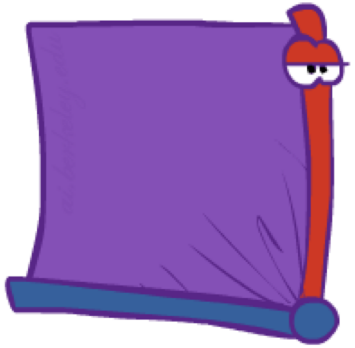
    $$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

  - Most likely explanation:

    $$\text{argmax}_q \; P(Q = q|E_1 = e_1 \ldots)$$

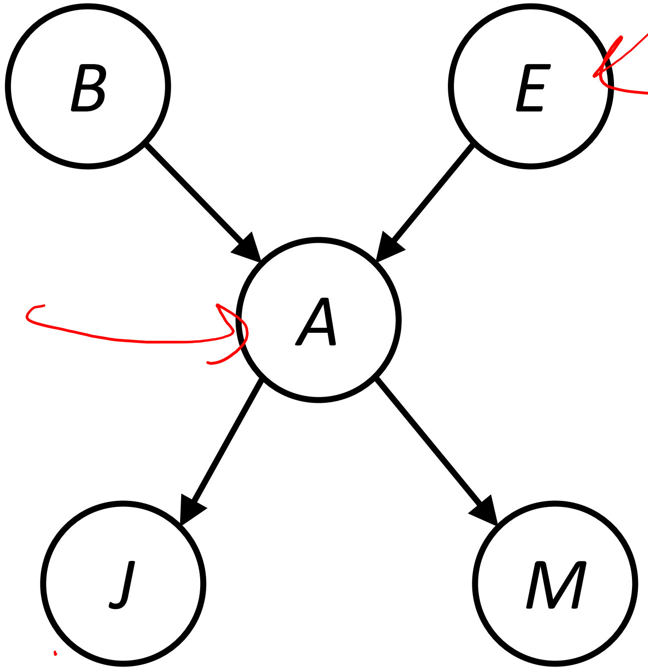






Inference by Enumeration in Bayes' Net

- Given unlimited time, inference in BNs is easy



$$P(B \mid +j, +m) \propto_B P(B, +j, +m)$$

$$= \sum_{e,a} P(B, e, a, +j, +m)$$

$$= \sum_{e,a} P(B)P(e)P(a|B,e)P(+j|a)P(+m|a)$$

$$=P(B)P(+e)P(+a|B,+e)P(+j|+a)P(+m|+a) + P(B)P(+e)P(-a|B,+e)P(+j|-a)P(+m|-a)$$
$$P(B)P(-e)P(+a|B,-e)P(+j|+a)P(+m|+a) + P(B)P(-e)P(-a|B,-e)P(+j|-a)P(+m|-a)$$

Example: Traffic Domain

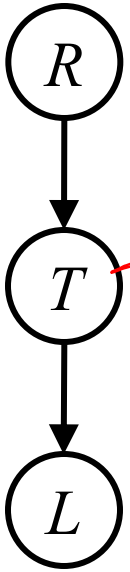
- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class!

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

$$P(L) = ?$$

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

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



$P(R)$

+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)

$P(R)$

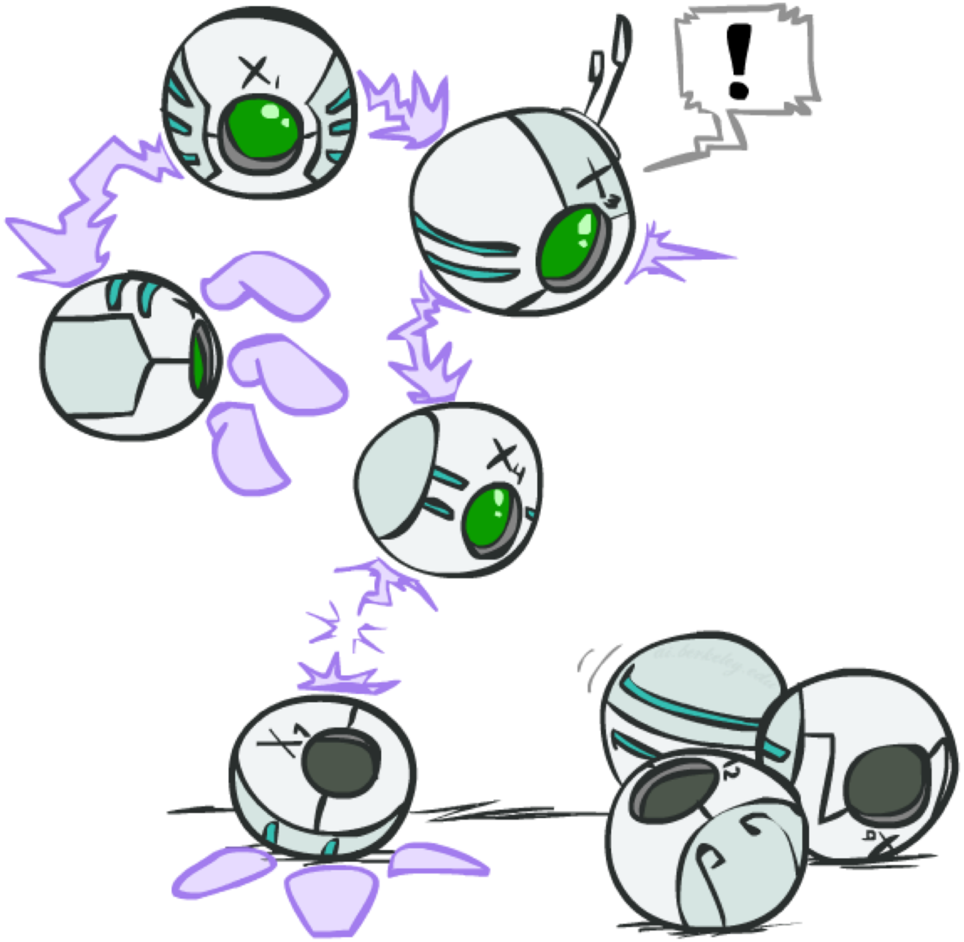
+r	0.1
-r	0.9

$P(T|R)$

+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

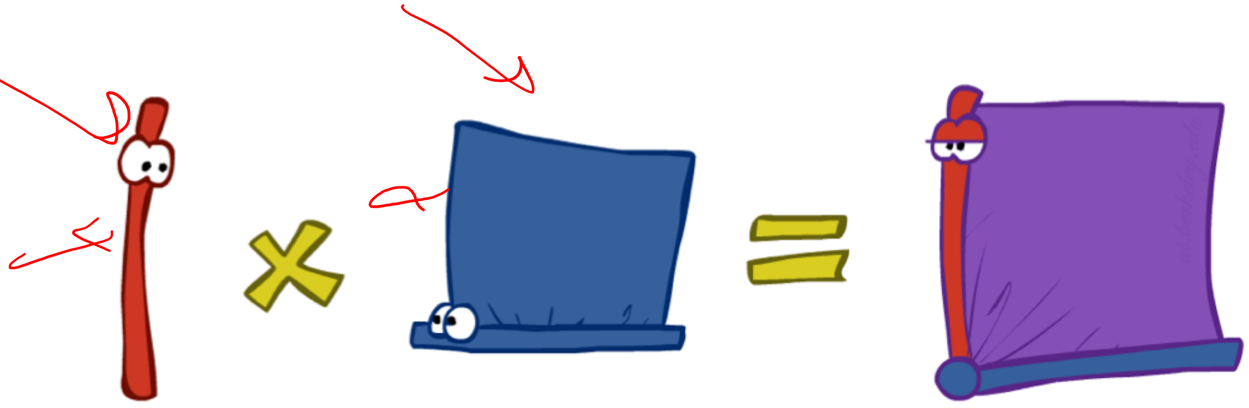
$P(+\ell|T)$

+t	+l	0.3
-t	+l	0.1

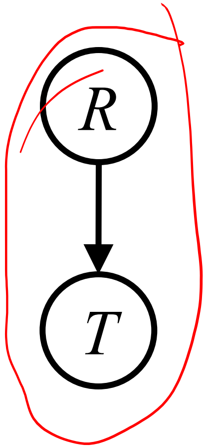
- Procedure: Join all factors, then sum out all hidden variables

Operation 1: Join Factors

- First basic operation: joining factors
- Combining factors:
  - Just 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



$P(R)$

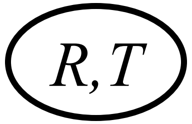
+r	0.1
-r	0.9

$\times$ $P(T|R)$

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

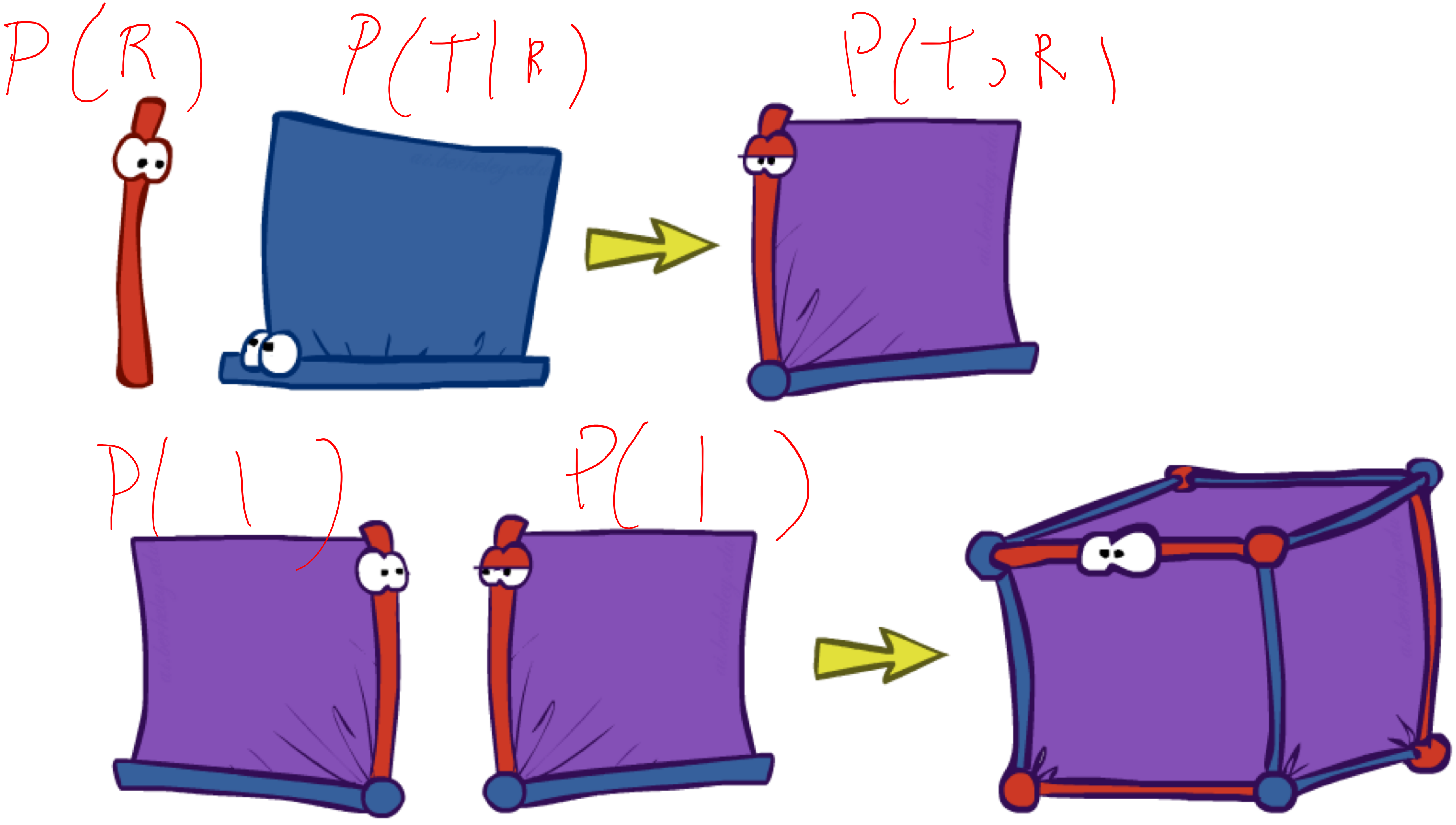
$\rightarrow$ $P(R,T)$

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

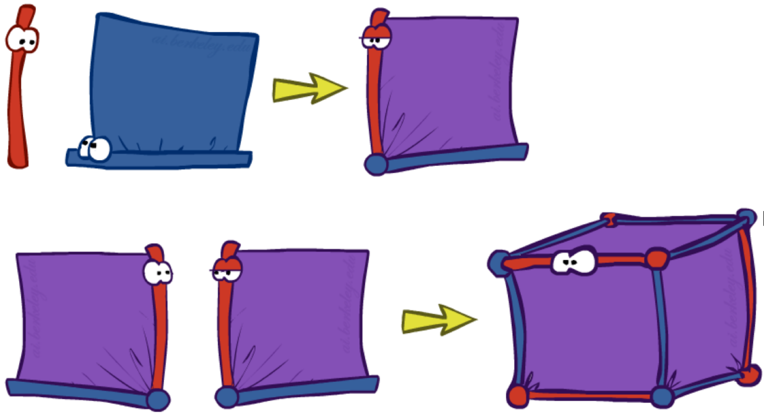


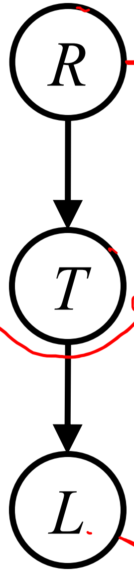
- Computation for each entry: pointwise products $\forall r, t: \quad P(r,t) = P(r) \cdot P(t|r)$

Example: Multiple Joins



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

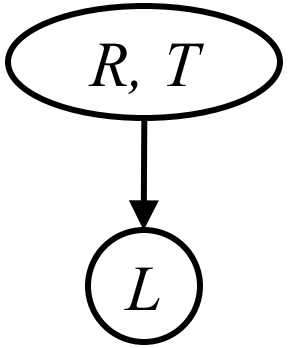
Join R

$P(R,T)$

+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

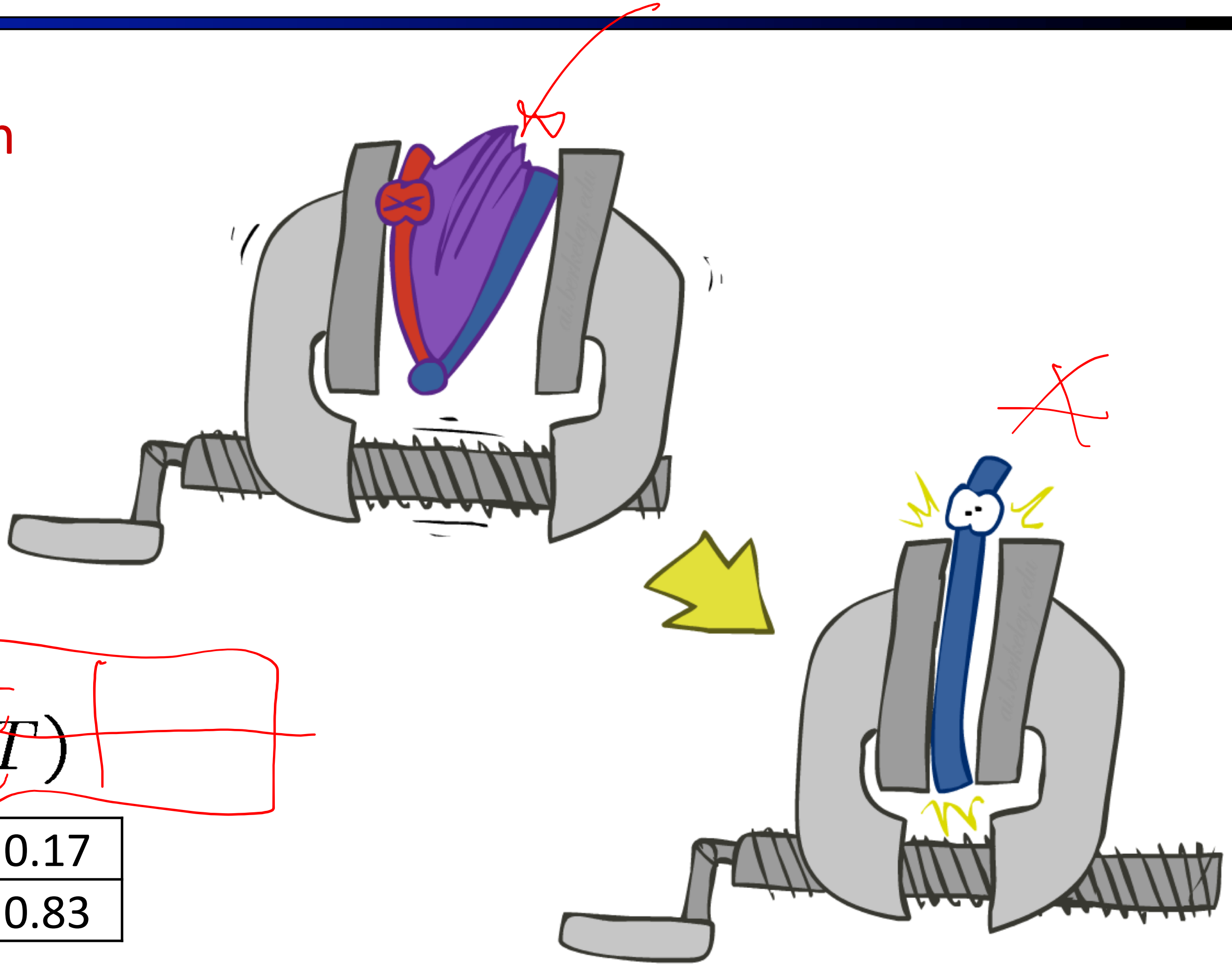
$P(R,T) \times P(L|T) = P(R,T,L)$

$P(R,T,L)$

+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

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

Sum out R

T, L

+t +l
+t -l
-t +l
-t -l

$P(T,L)$

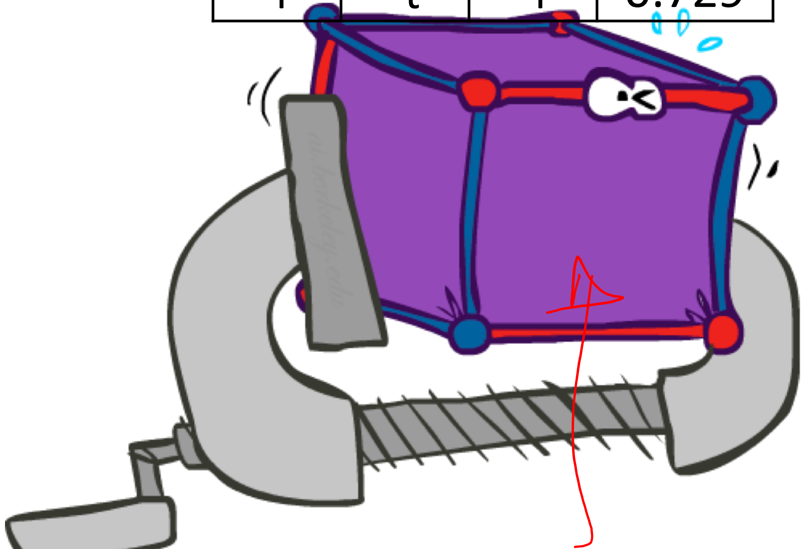
+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

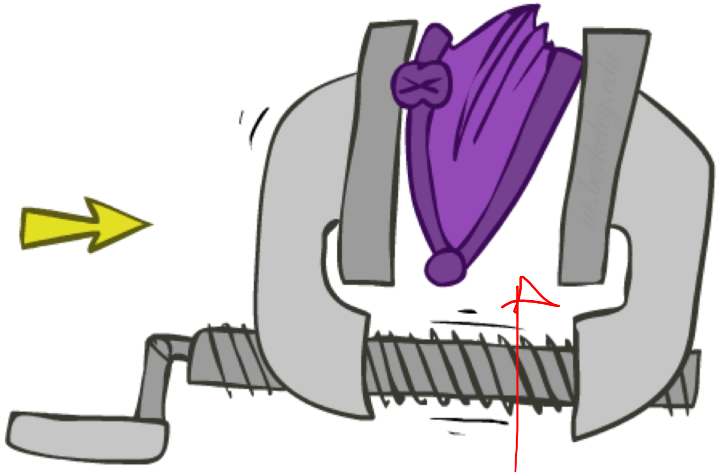
Sum out T

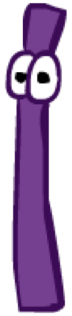
L

$P(L)$

+l	0.134
-l	0.866





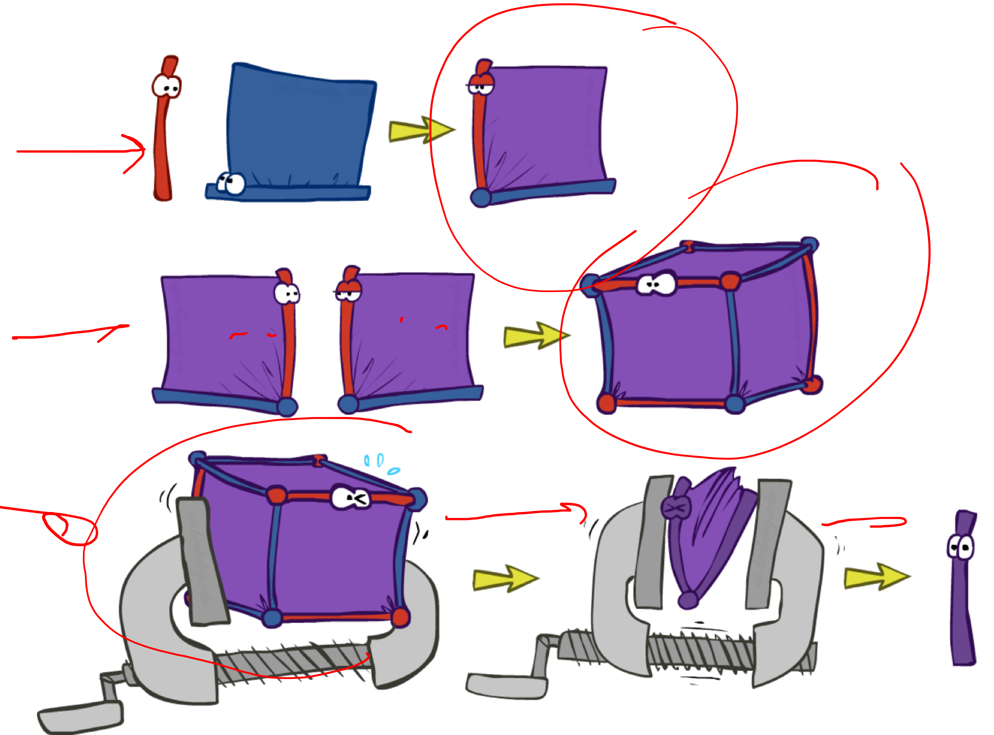


Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)

$P(R)$

$P(T|R) \longrightarrow P(R,T,L) \longrightarrow P(L)$

$P(L|T)$

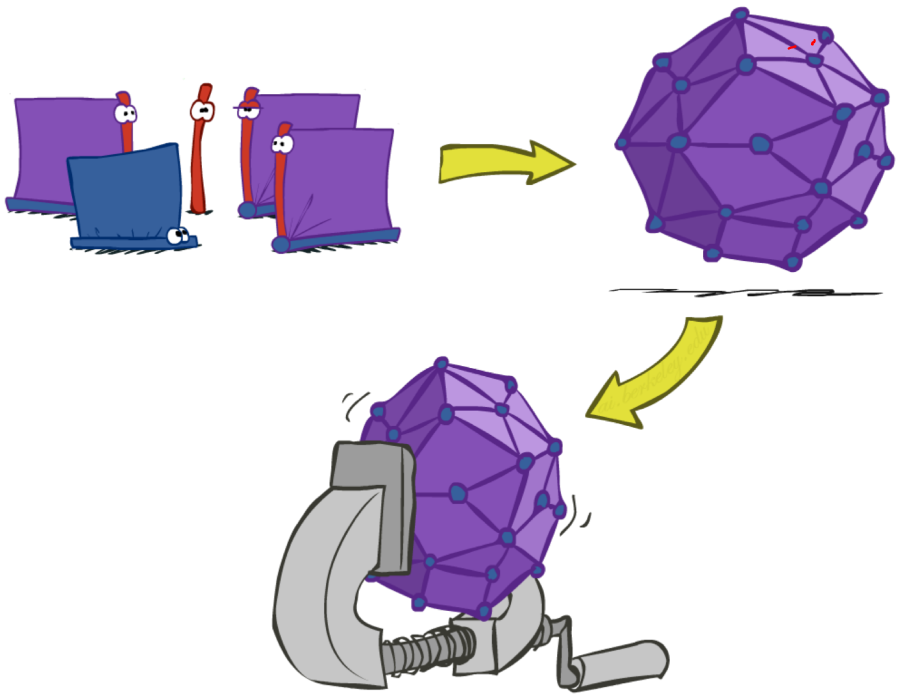


K

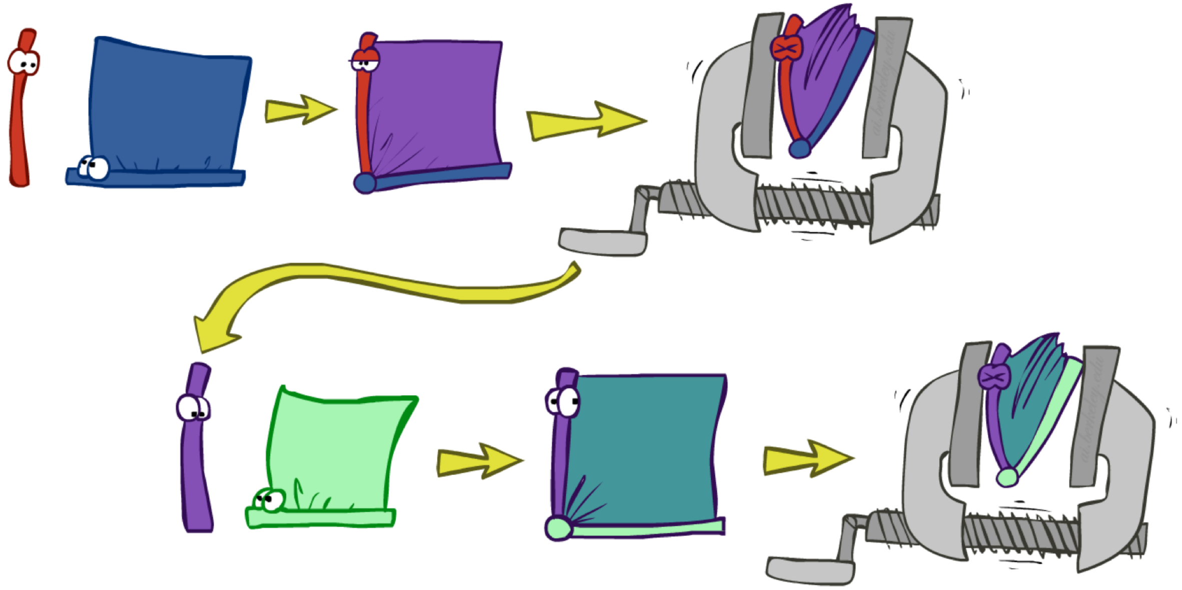
2

Inference by Enumeration vs. Variable Elimination

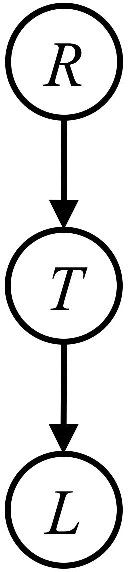
- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables



- Idea: interleave joining and marginalizing!
  - Called "Variable Elimination"
  - Still NP-hard, but usually much faster than inference by enumeration



Marginalizing Early! (aka VE)



$P(R)$

+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

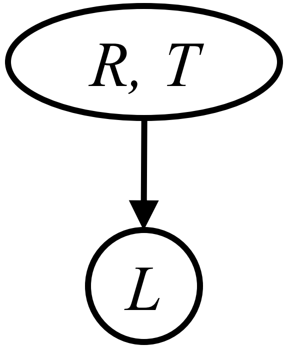
$P(L|T)$

+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



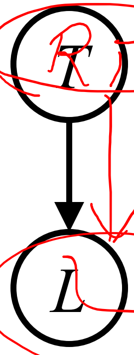
$P(L|T)$

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

Sum out R

$P(T)$

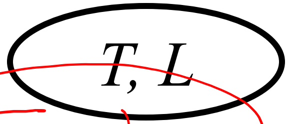
+t	0.17
-t	0.83



$P(L|T)$

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

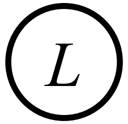
Join T



$P(T, L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

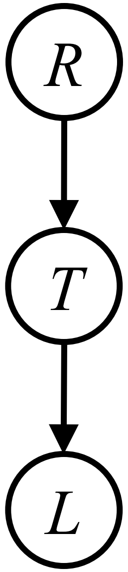
Sum out T



$P(L)$

+l	0.134
-l	0.866

Traffic Domain



$$P(L) = ?$$

- Inference by Enumeration

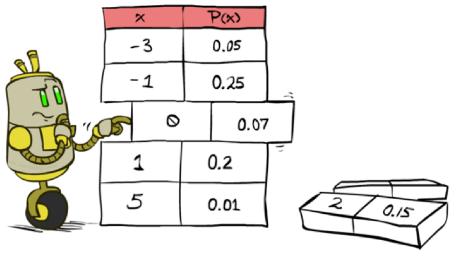
$$= \underbrace{\sum_t \underbrace{\sum_r \underbrace{P(L|t)\underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Join on t}}}_{\text{Eliminate r}}}_{\text{Eliminate t}}$$

- Variable Elimination

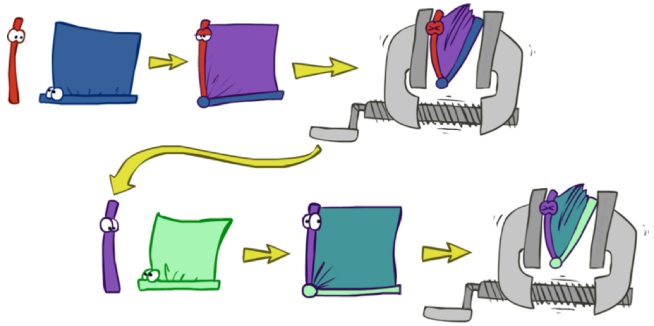
$$= \underbrace{\sum_t \underbrace{P(L|t)\underbrace{\sum_r \underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Eliminate r}}}_{\text{Join on t}}}_{\text{Eliminate t}}$$

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)



- 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

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

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

  - Computing $P(L|+r)$ the initial factors become:

$P(+r)$

+r	0.1

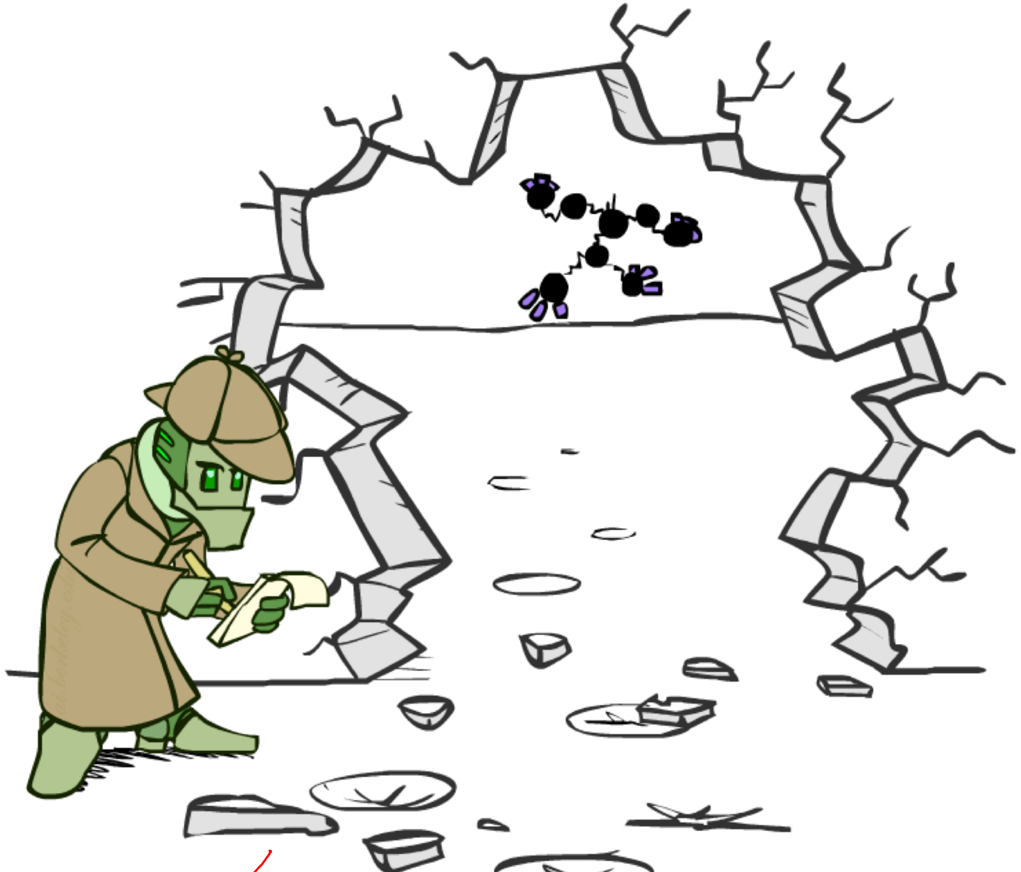
$P(T|+r)$

+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



$P(T, L|+r)$

Evidence II

- Result will be a selected joint of query and evidence
  - E.g. for P(L | +r), we would end up with:

$$P(+r, L)$$

+r	+l	0.026
+r	-l	0.074

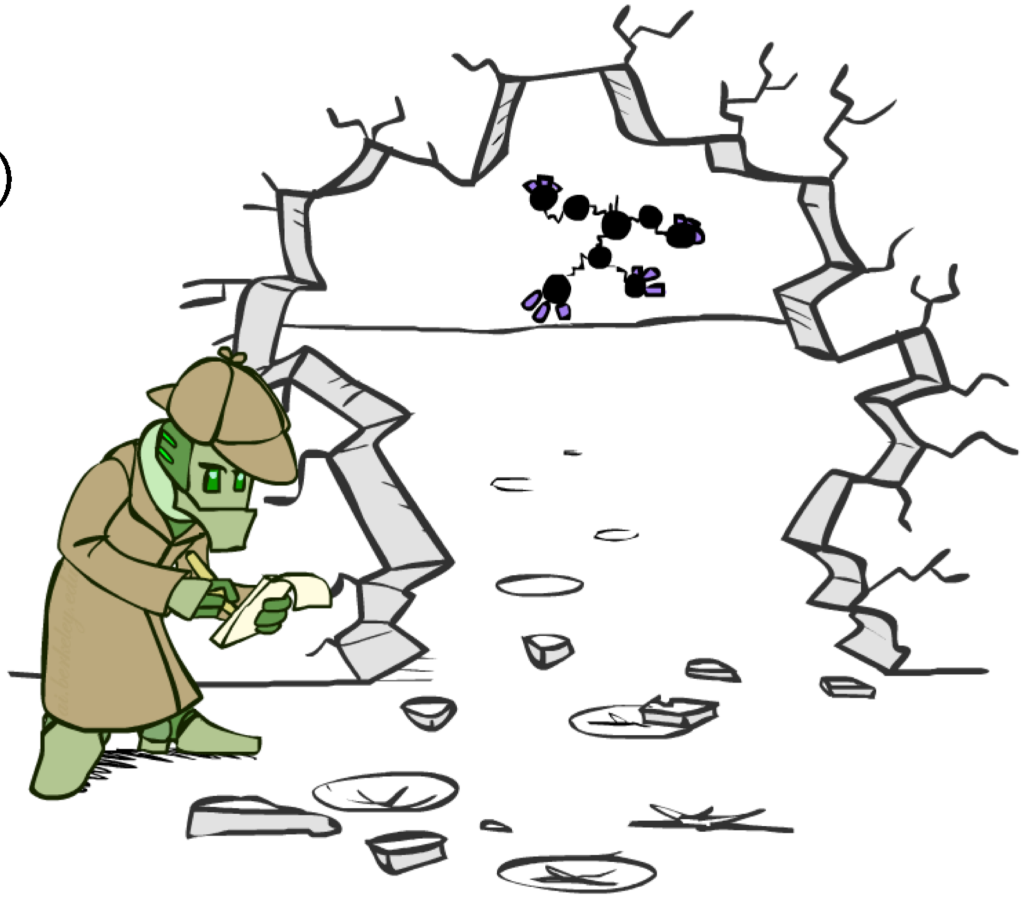
Normalize

$$P(L|+r)$$

+l	0.26
-l	0.74

- To get our answer, just normalize this!

- That ’s it!



Inference by Enumeration

- General case:
  - Evidence variables: $E_1 \ldots E_k = e_1 \ldots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \ldots H_r$

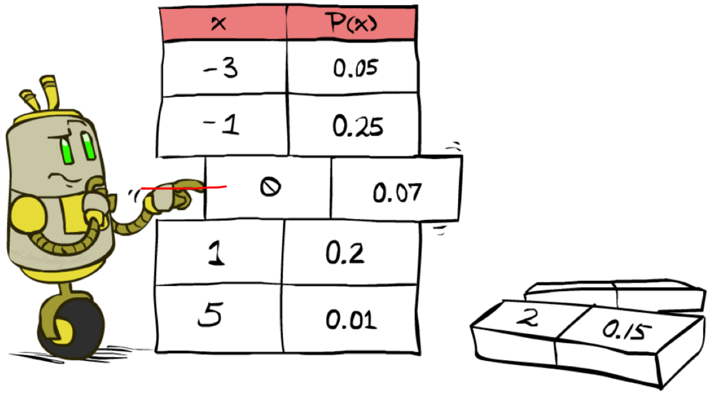
$\left.\begin{matrix} E_1 \ldots E_k = e_1 \ldots e_k \\ Q \\ H_1 \ldots H_r \end{matrix}\right\} \begin{matrix} X_1, X_2, \ldots X_n \\ \textit{All variables} \end{matrix}$

- We want:

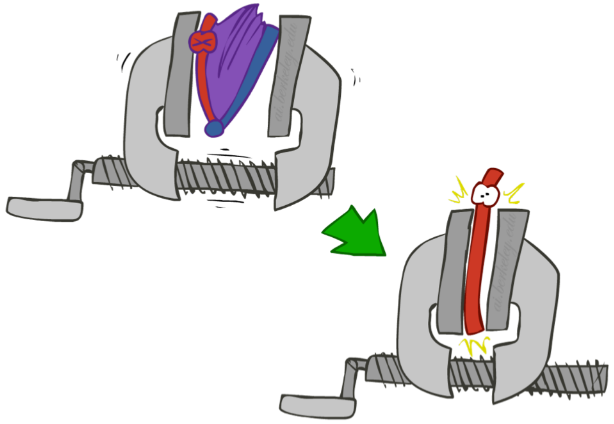
* Works fine with multiple query variables, too

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

- Step 1: Select the entries consistent with the evidence



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



- Compute joint

$$P(Q, e_1 \ldots e_k) = \sum_{h_1 \ldots h_r} \underbrace{P(Q, h_1 \ldots h_r, e_1 \ldots e_k)}$$

- Sum out hidden variables $X_1, X_2, \ldots X_n$

- Step 3: Normalize

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

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

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

Variable Elimination

- General case:
  - Evidence variables: $E_1 \dots E_k = e_1 \dots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \dots H_r$

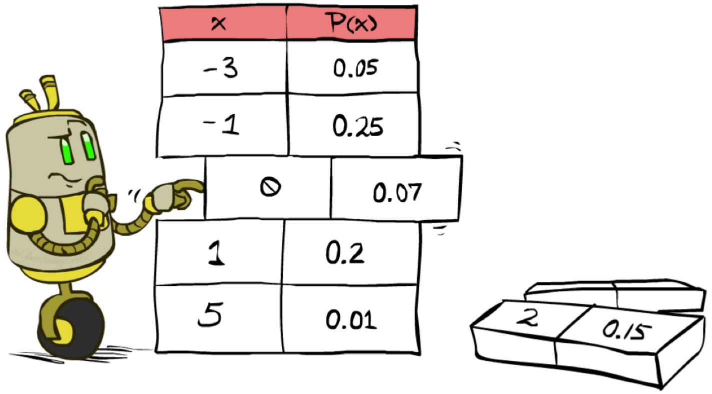
$\left.\right\}$ $X_1, X_2, \dots X_n$ *All variables*

- We want:

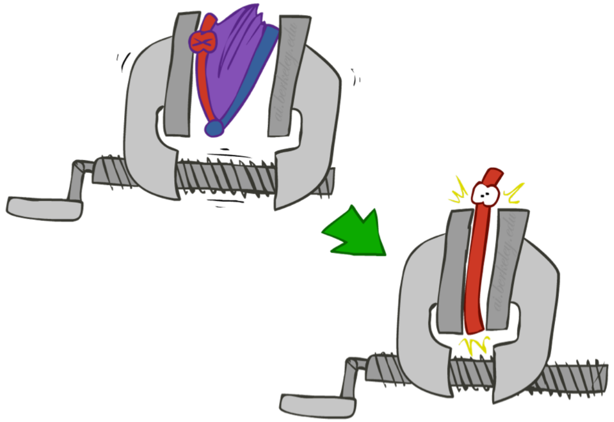
** Works fine with multiple query variables, too*

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

- Step 1: Select the entries consistent with the evidence



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



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

- Step 3: Normalize

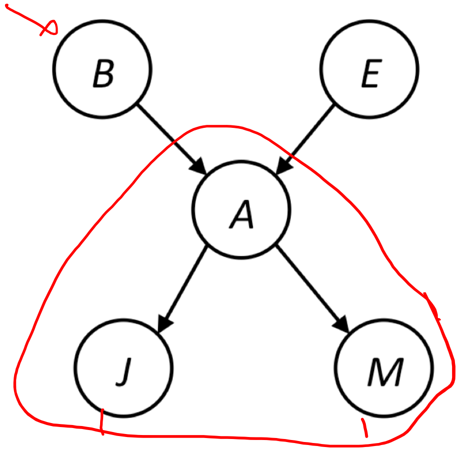
$$\times \frac{1}{Z}$$

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

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

- Interleave joining and summing out

Example



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

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

$P(A, j, m \mid B, E)$

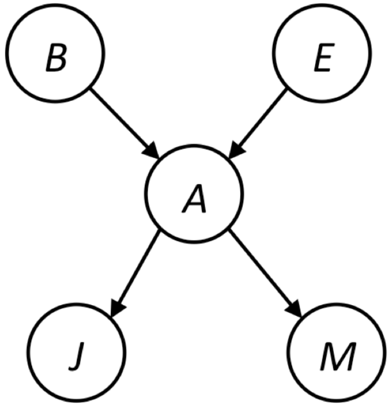
Choose A

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

$$\xrightarrow{\times} P(j,m,A|B,E) \xrightarrow{\Sigma} P(j,m|B,E)$$

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

Example



$$\boxed{P(B) \qquad P(E) \qquad P(j,m|B,E)}$$

Choose E

$$P(E),\ P(j,m|B,E) \xrightarrow{\times} P(j,m,E|B) \xrightarrow{\sum} P(j,m|B)$$

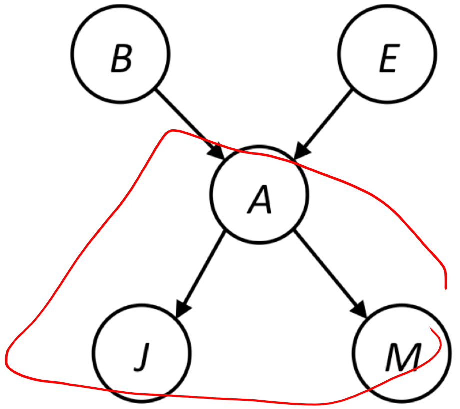
$$\boxed{P(B) \qquad P(j,m|B)}$$

Finish with B

$$P(B),\ P(j,m|B) \xrightarrow{\times} P(j,m,B) \xrightarrow{\text{Normalize}} P(B|j,m)$$

Example

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



$$\boxed{P(B) \qquad P(E) \qquad P(A|B,E) \qquad P(j|A) \qquad P(m|A)}$$

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

marginal can be obtained from joint by summing out

use Bayes' net joint distribution expression

joining on a, and then summing out gives f_1

joining on e, and then summing out gives f_2

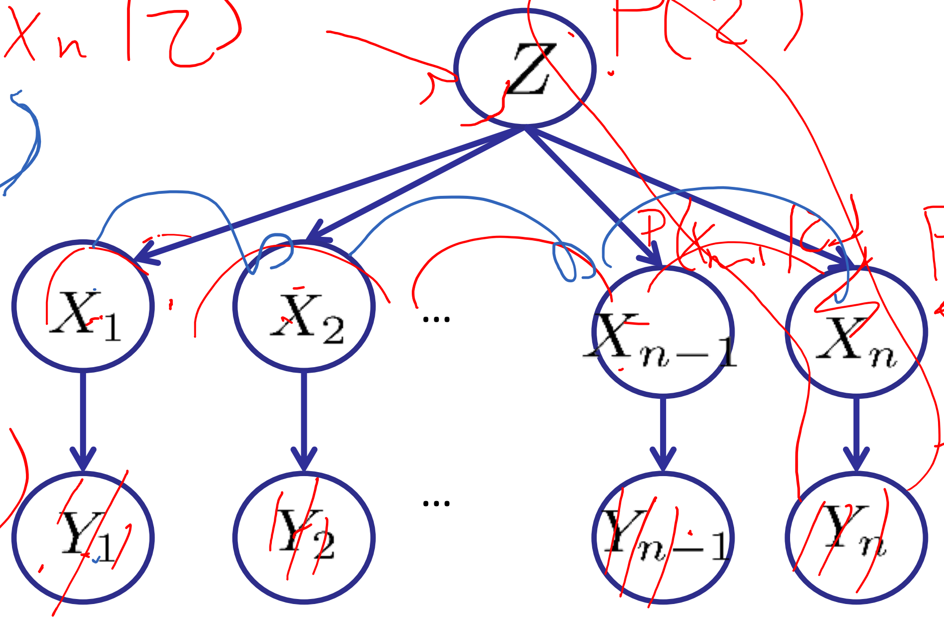
Variable Elimination Ordering

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

$P(X_1, Z)$

$Z: P(Z)P(X_1|Z)P(X_2|Z)P(X_n|Z)$

$P(Z, X_1, X_2, \ldots, X_n)$



$X_1: P(X_1|Z)P(Y_1|X_1)$

join $P(X_1, y_1|Z)$

el.: $P(y_1|Z)$

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

VE: Computational and Space Complexity

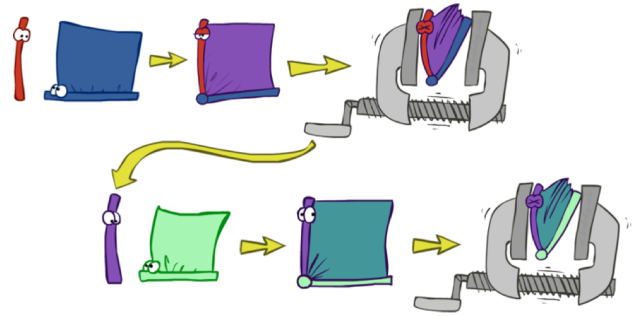
- The computational and space complexity of variable elimination is determined by the largest factor

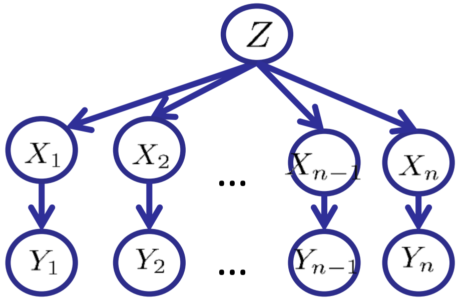
- The elimination ordering can greatly affect the size of the largest factor.
  - E.g., previous slide's example 2^n vs. 2

- Does there always exist an ordering that only results in small factors?
  - No!

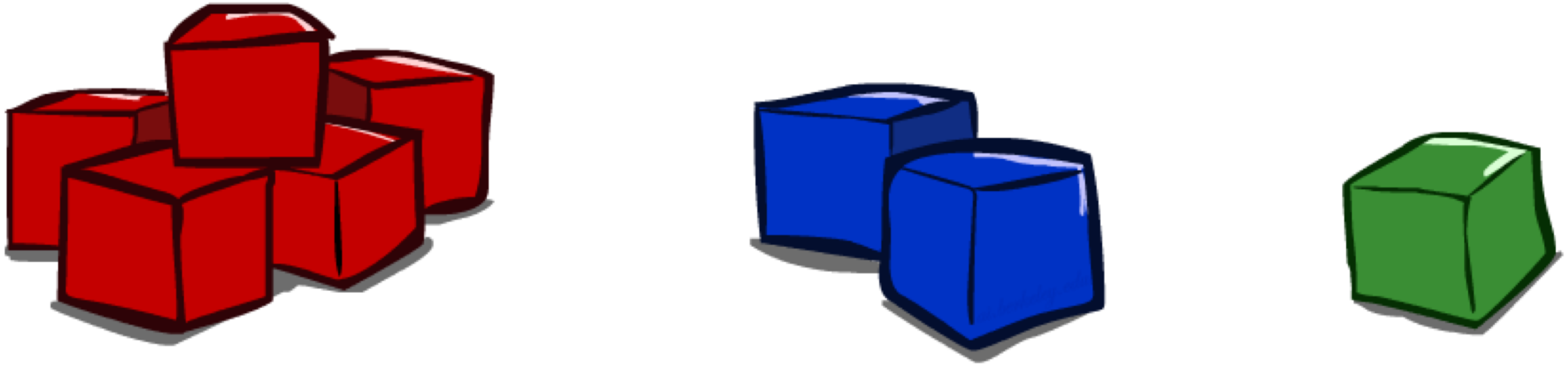
Variable Elimination

- Interleave joining and marginalizing
- d^k entries computed for a factor over k variables with domain sizes d
- Ordering of elimination of hidden variables can affect size of factors generated
- Worst case: running time exponential in the size of the Bayes' net



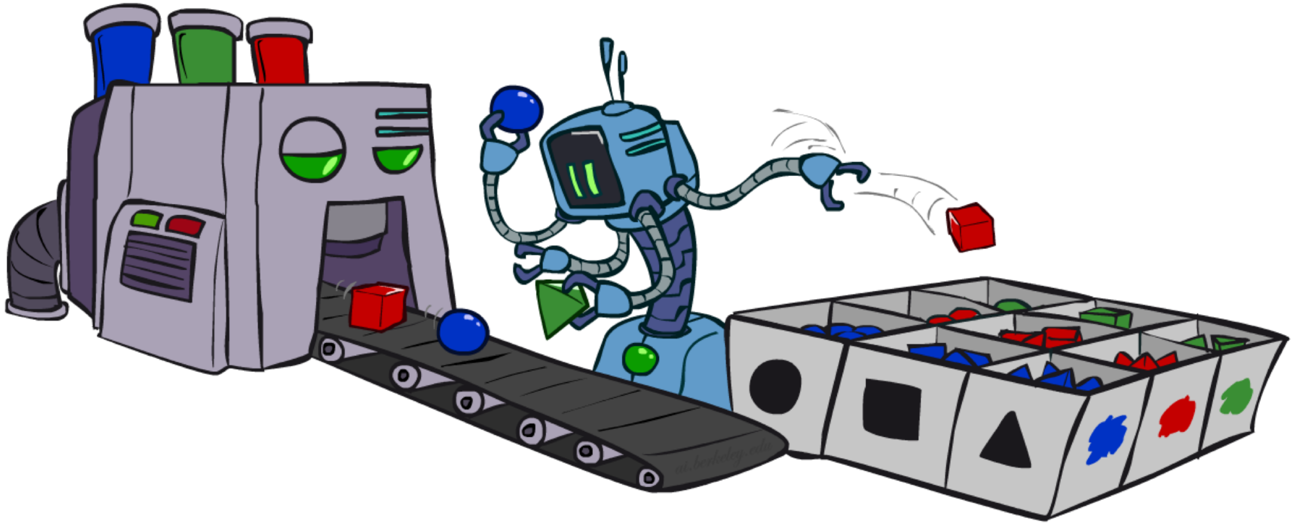


Approximate Inference: Sampling



Sampling

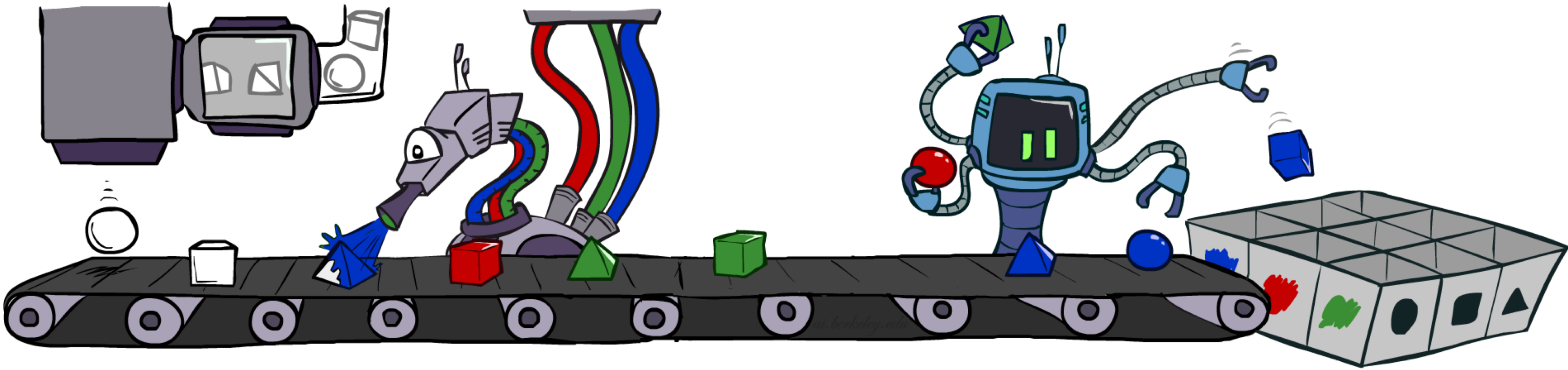
- Sampling is a lot like repeated simulation
  - Predicting the weather, basketball games, …
- Basic idea
  - Draw N samples from a sampling distribution S
  - Compute an approximate probability
- Why sample?
  - Learning: get samples from a distribution you don't know
  - Inference: getting a sample is faster than computing the right answer



Sampling in Bayes' Nets

- Prior Sampling
- Rejection Sampling
- Likelihood Weighting

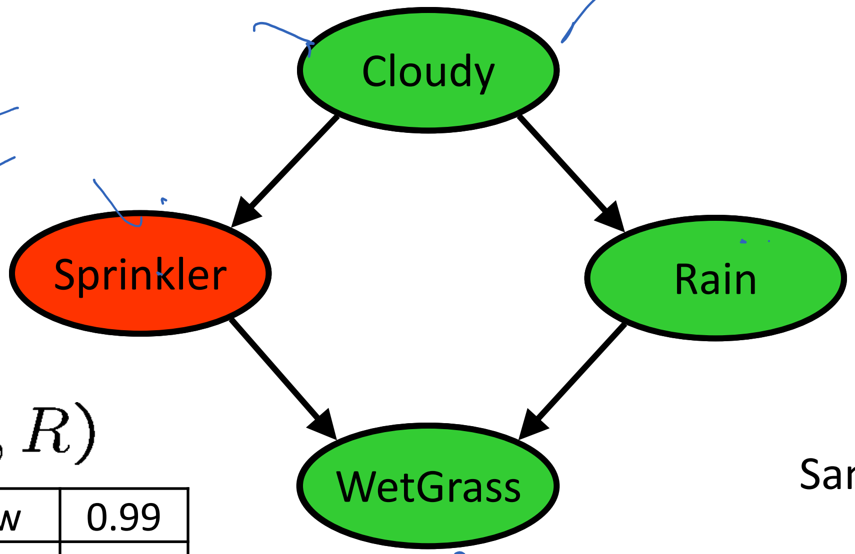
Prior Sampling



Prior Sampling

$P(C)$

+c	0.5
-c	0.5



$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5

$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S, R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

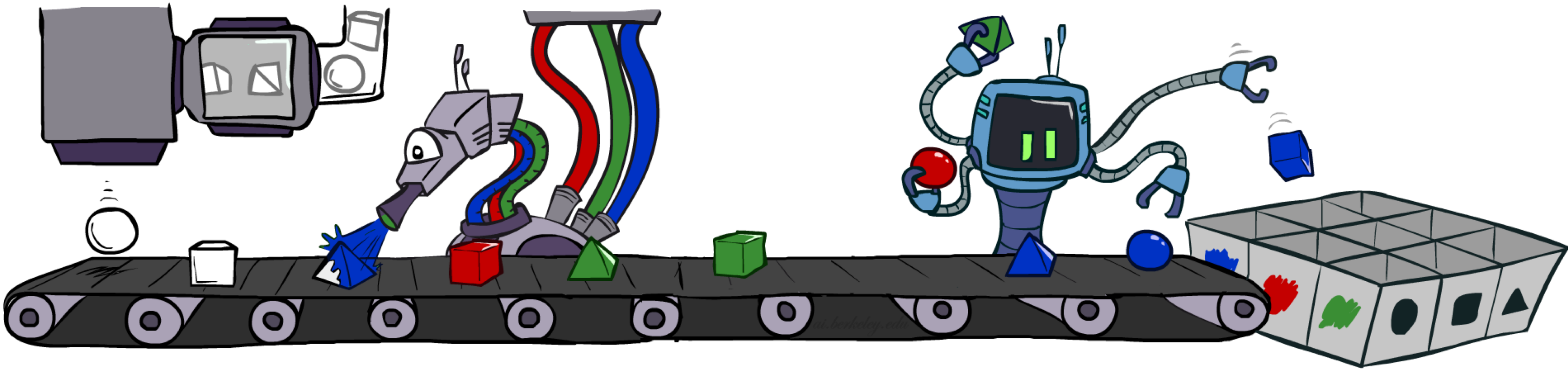
+c, -s, +r, +w

-c, +s, -r, +w

...

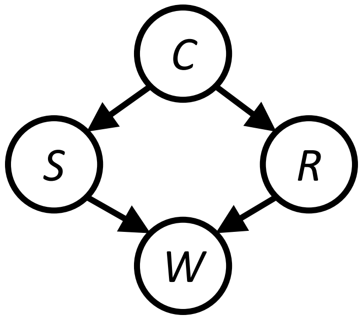
Prior Sampling

- For i = 1, 2, …, n
  - Sample x_i from $P(X_i \mid Parents(X_i))$
- Return $(x_1, x_2, \ldots, x_n)$



Example

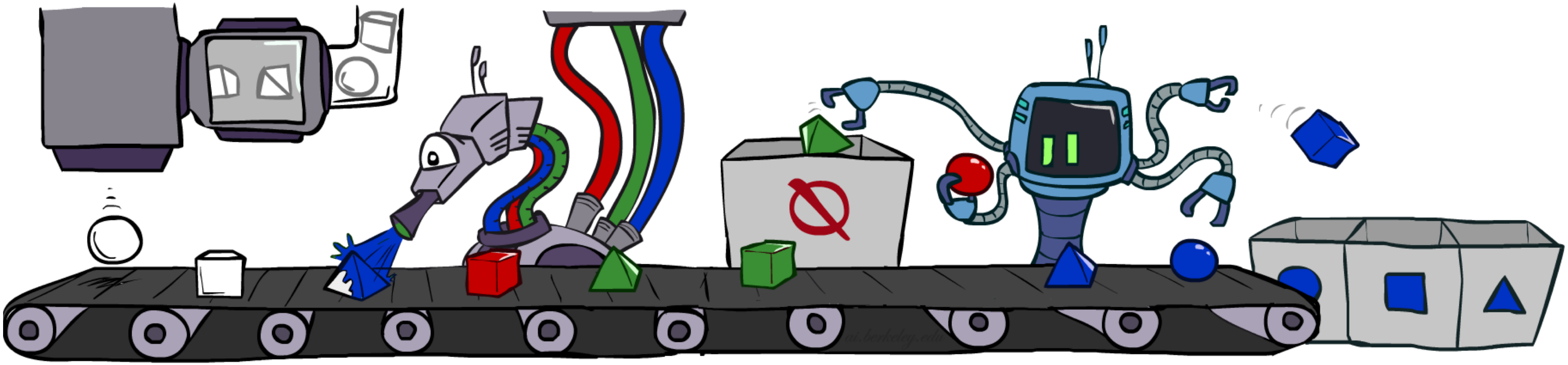
- We'll get a bunch of samples from the BN:
  - +c, -s, +r, +w
  - +c, +s, +r, +w
  - ~~-c, +s, +r, -w~~
  - +c, -s, +r, +w
  - -c, -s, -r, +w



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

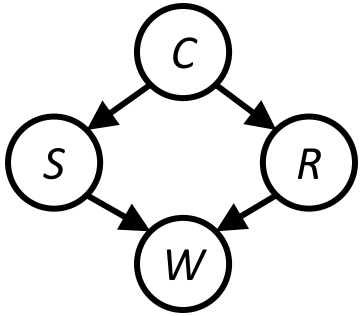
3/4 1/4

Rejection Sampling



Rejection Sampling

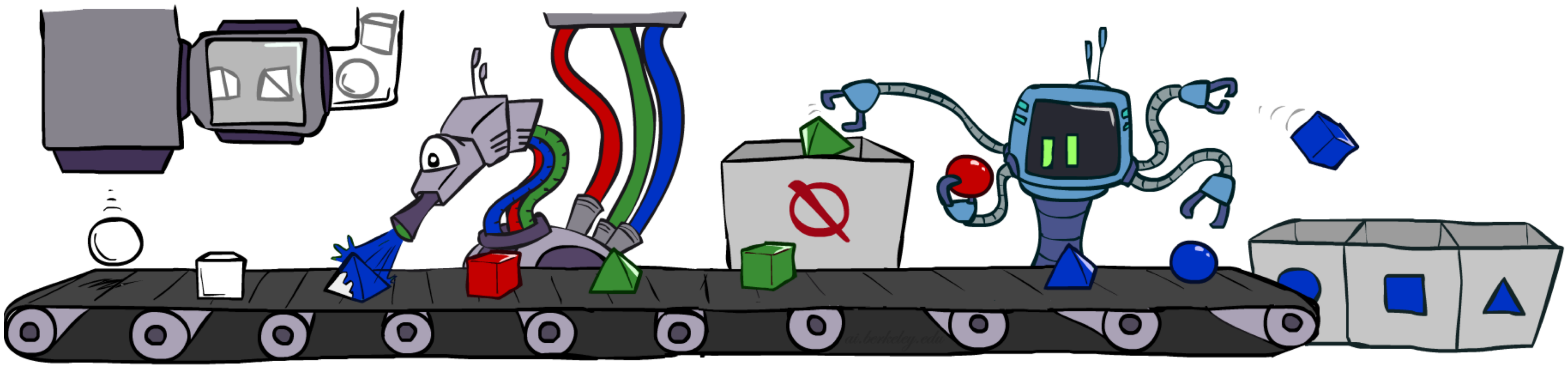
- Let's say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C as we go
- Let's say we want P(C | +s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don't have S=+s
  - This is called rejection sampling
  - It is also consistent for conditional probabilities (i.e., correct in the limit)



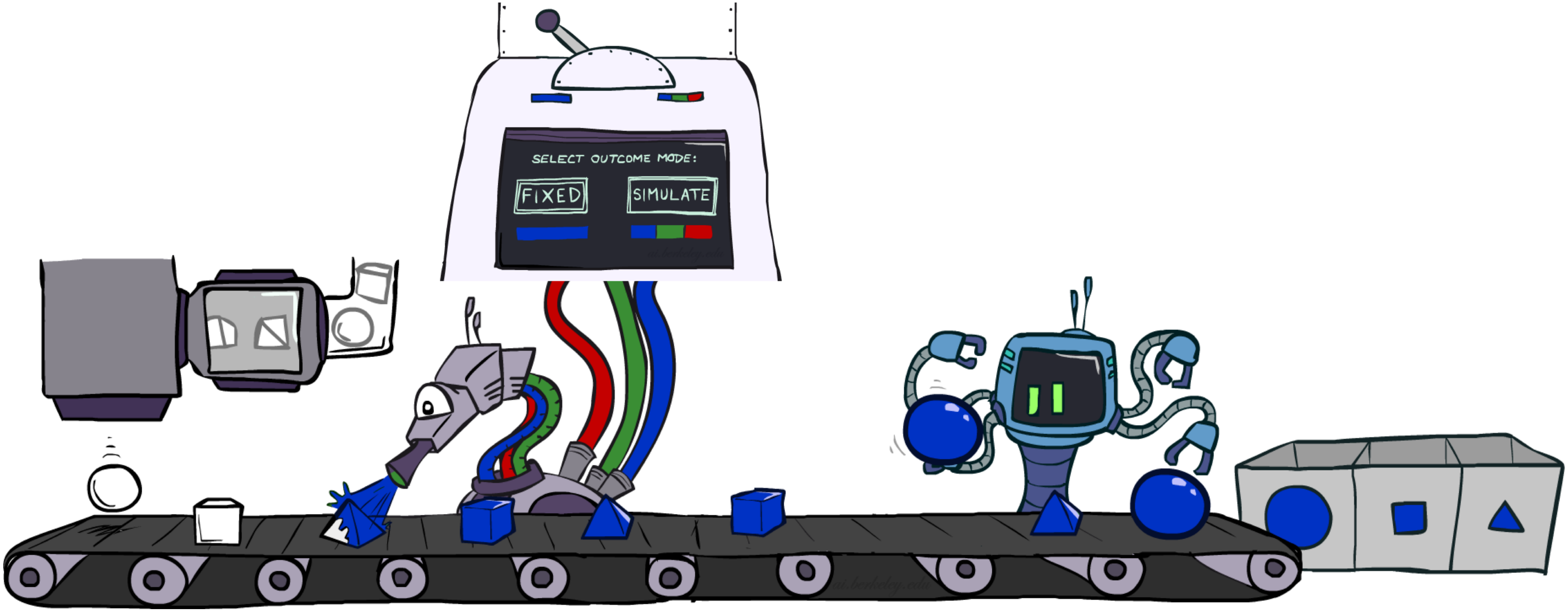
+c, -s, +r, +w
+c, +s, +r, +w
-c, +s, +r, -w
+c, -s, +r, +w
-c, -s, -r, +w

Rejection Sampling

- Input: evidence instantiation
- For i = 1, 2, …, n
  - Sample x_i from $P(X_i \mid \text{Parents}(X_i))$
  - If x_i not consistent with evidence
    - Reject: return – no sample is generated in this cycle
- Return $(x_1, x_2, \ldots, x_n)$

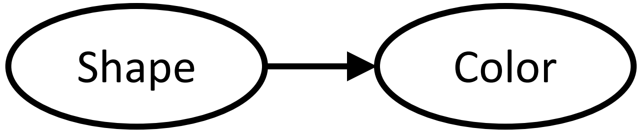


Likelihood Weighting

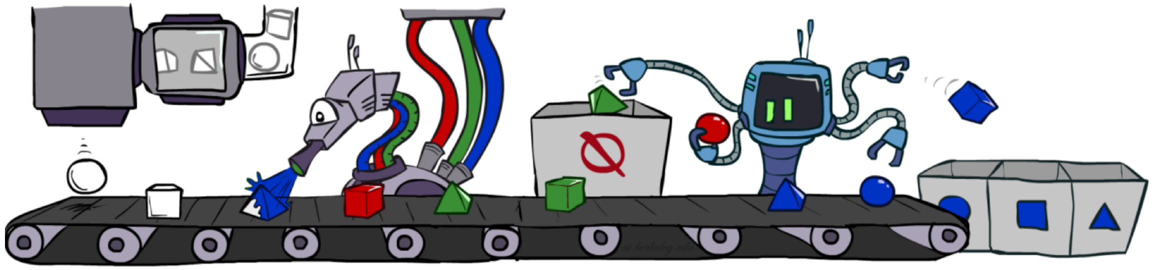


Likelihood Weighting

- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider P(Shape | blue)

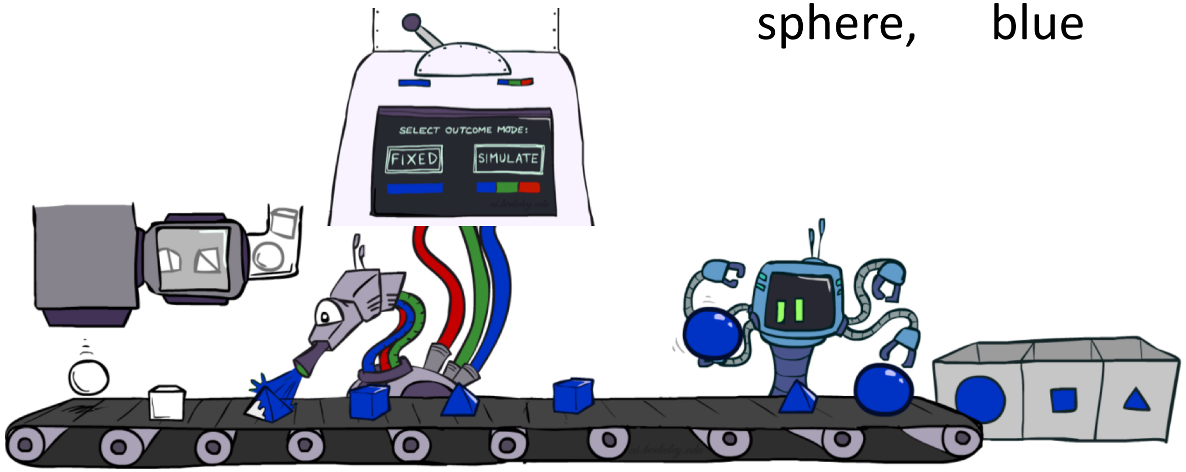


~~pyramid, green~~
~~pyramid, red~~
sphere, blue
~~cube, red~~
~~sphere, green~~



- 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



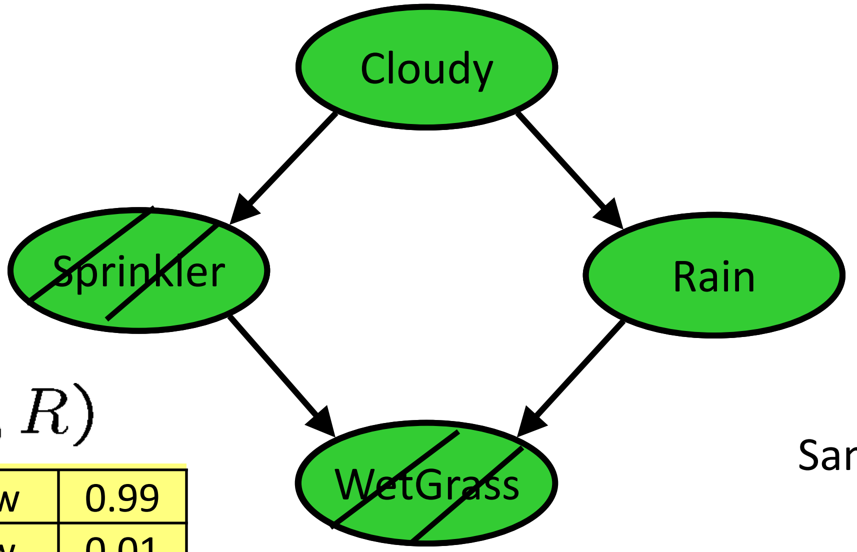
Likelihood Weighting

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

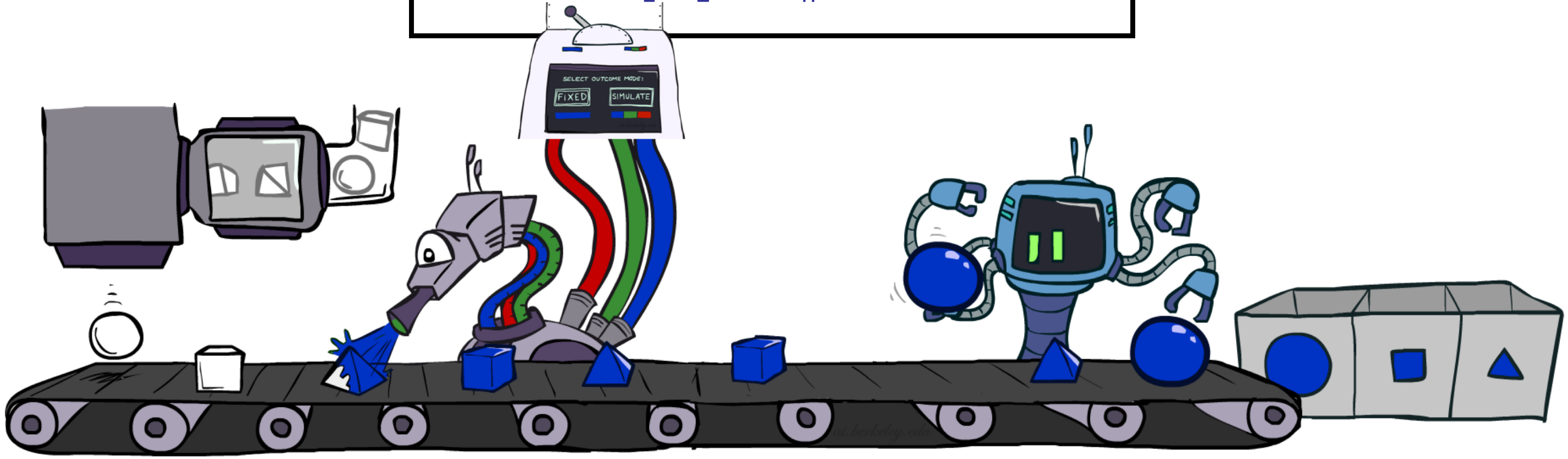
+c, +s, +r, +w

…

$$w = 1.0 \times 0.1 \times 0.99$$

Likelihood Weighting

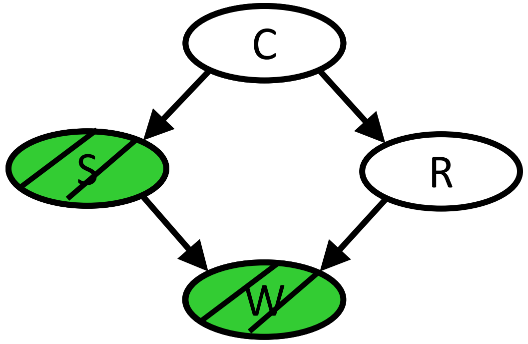
- Input: evidence instantiation
- $w = 1.0$
- for $i = 1, 2, \ldots, n$
  - if X_i is an evidence variable
    - X_i = observation x_i for X_i
    - Set $w = w * P(x_i \mid \text{Parents}(X_i))$
  - else
    - Sample x_i from $P(X_i \mid \text{Parents}(X_i))$
- return $(x_1, x_2, \ldots, x_n)$, w



Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

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



- Now, samples have weights

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

- Together, weighted sampling distribution is consistent

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

$$= P(\mathbf{z},\mathbf{e})$$

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