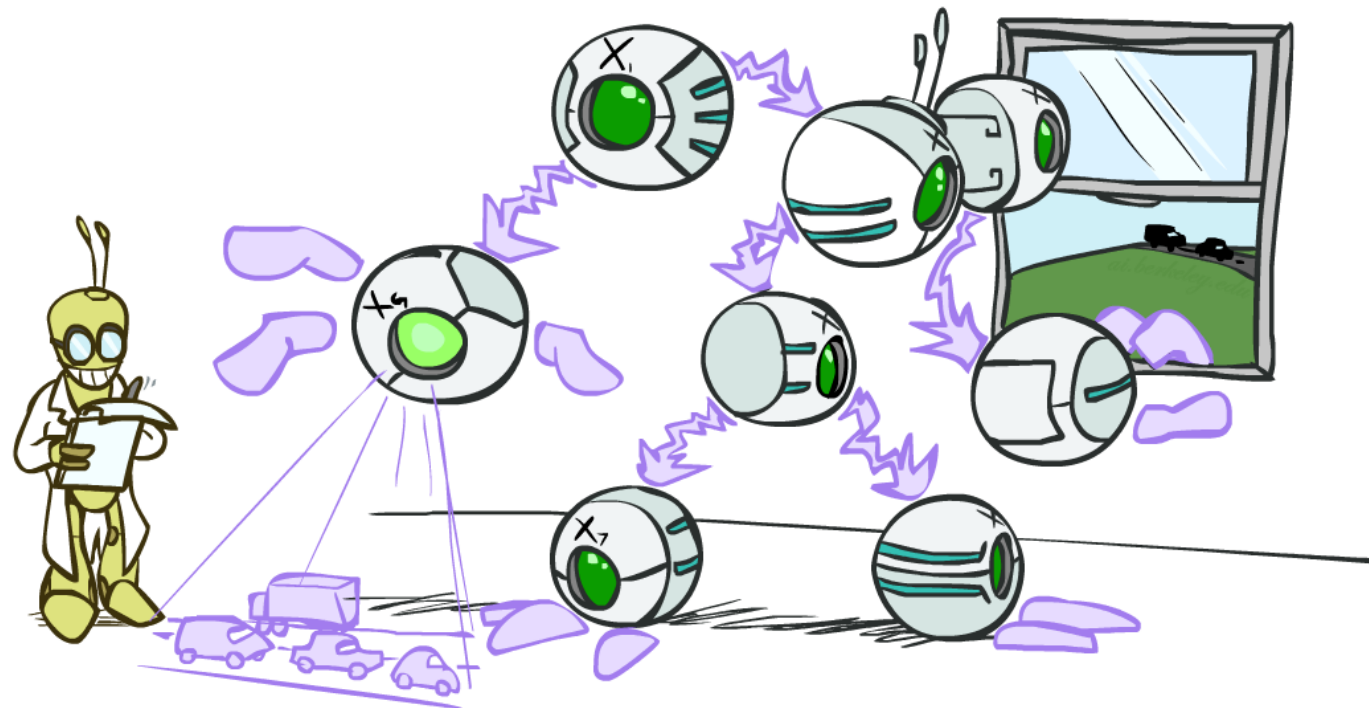


CS 188: Artificial Intelligence

Bayes' Nets: Inference



Instructors: Oliver Grillmeyer and Charlie Snell — University of California, Berkeley

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at <http://ai.berkeley.edu>.]

Announcements

- HW6 is due **Thursday, July 17, 11:59 PM PT**
- Project 3 is due **Friday, July 18, 11:59 PM PT**
- Midterm is **Wednesday July 23, 7-9 PM PT**
- Look at Exam info under Policies link on CS 188 website

Bayes' Net Representation

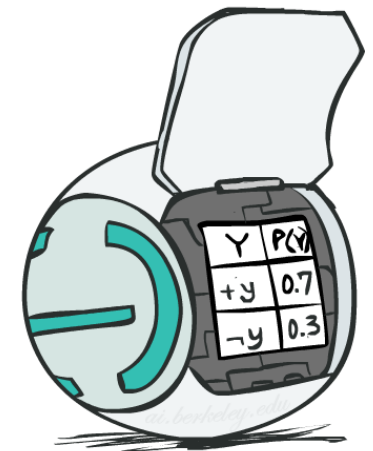
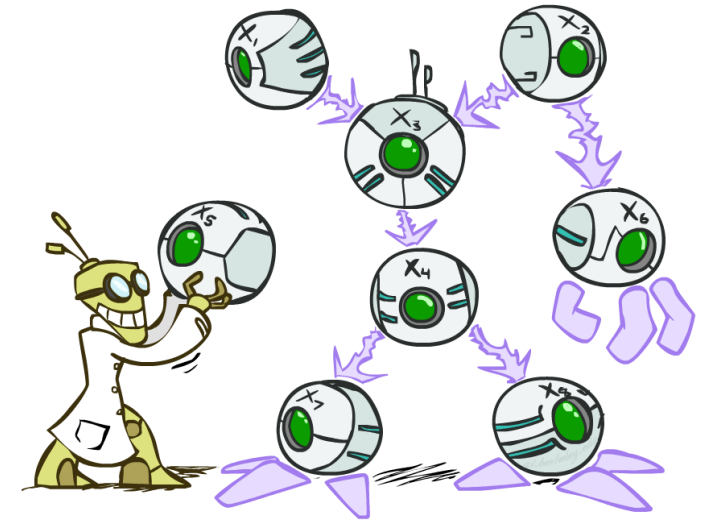
- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
 - A collection of distributions over X , one for each combination of parents' values

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

- Bayes' nets implicitly encode joint distributions

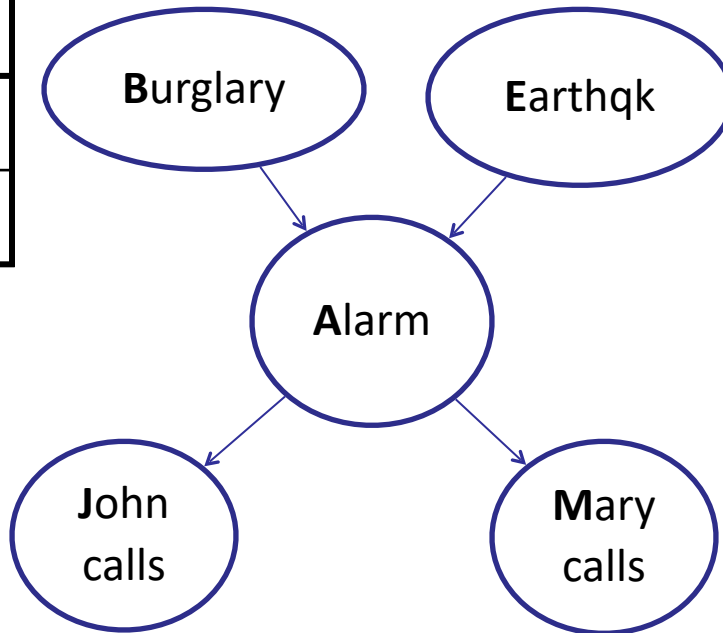
- As a product of local conditional distributions
- To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

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

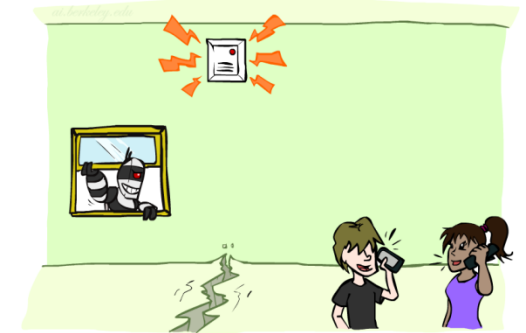


Example: Alarm Network

B	P(B)
+b	0.001
-b	0.999



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

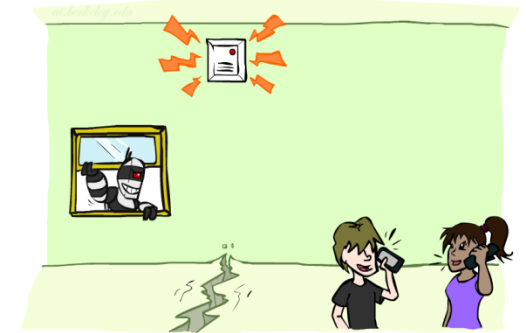
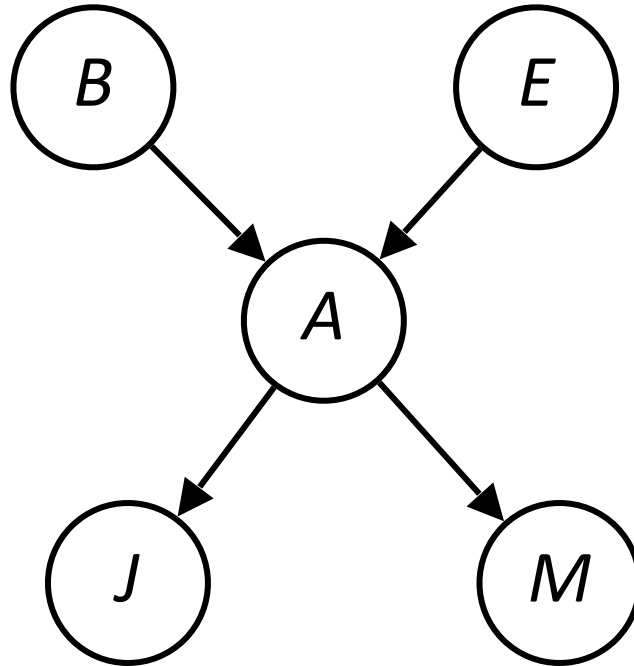
Example: Alarm Network

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

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

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

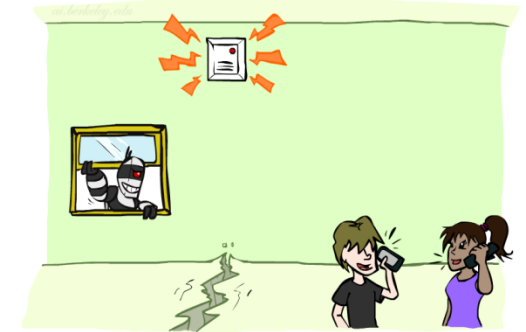
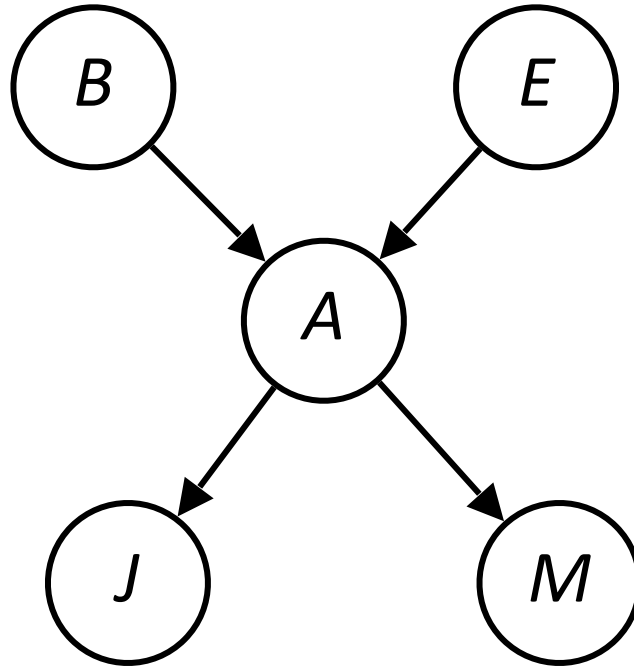
Example: Alarm Network

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

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

Bayes' Nets

✓ Representation

✓ Conditional Independences

- Probabilistic Inference

- Enumeration (exact, exponential complexity)
- Variable elimination (exact, worst-case exponential complexity, often better)
- Inference is NP-complete
- Sampling (approximate)

- Learning Bayes' Nets from Data

Inference

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

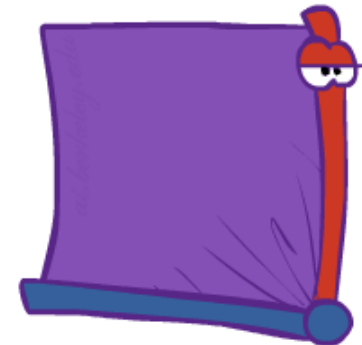
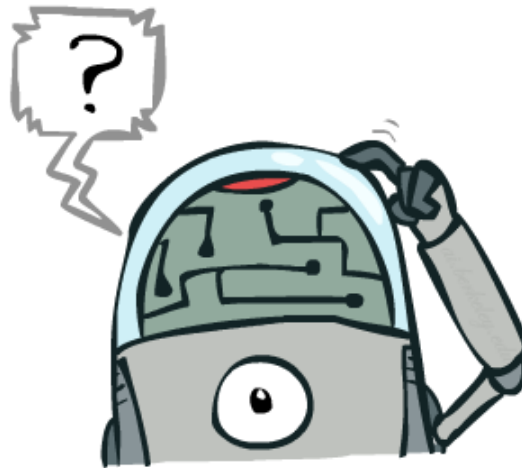
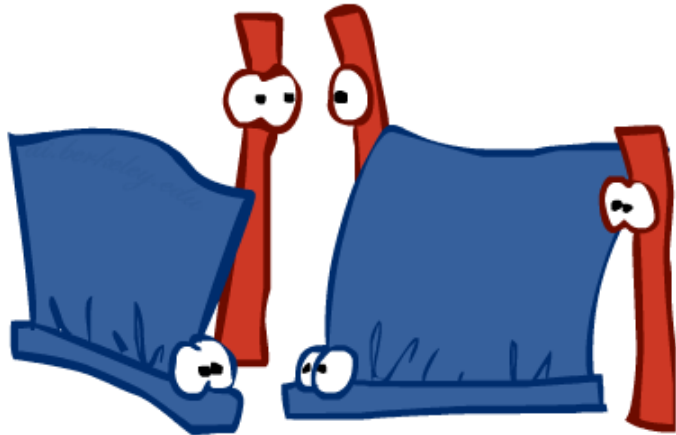
- Examples:

- Posterior probability

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

- Most likely explanation:

$$\operatorname{argmax}_q P(Q = q|E_1 = e_1 \dots)$$



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*

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

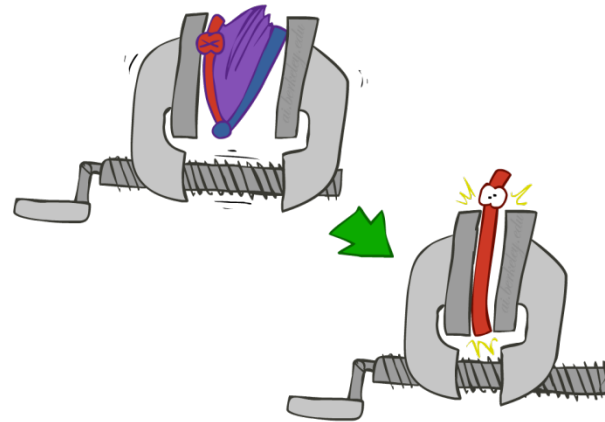
- Step 1: Select the entries consistent with the evidence



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



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

- Step 3: Normalize

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

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

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

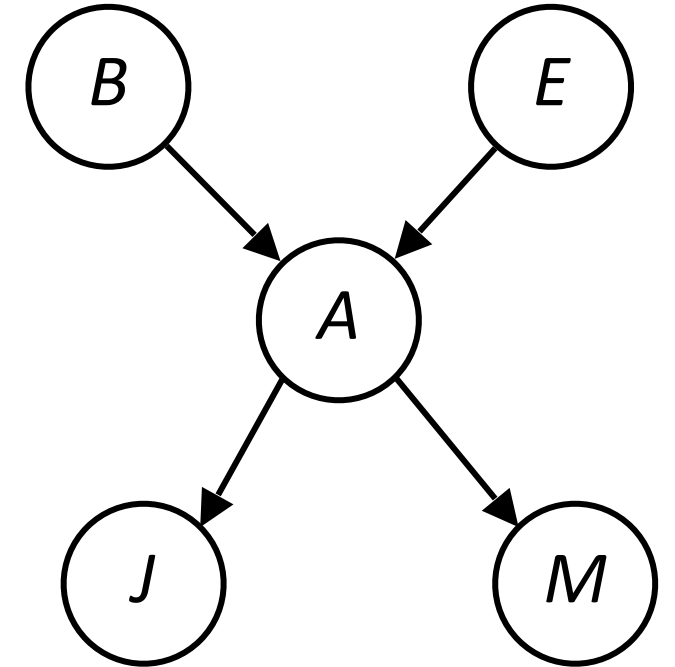
Inference by Enumeration in Bayes' Net

- Given unlimited time, inference in BNs is easy
- Reminder of inference by enumeration by example:

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

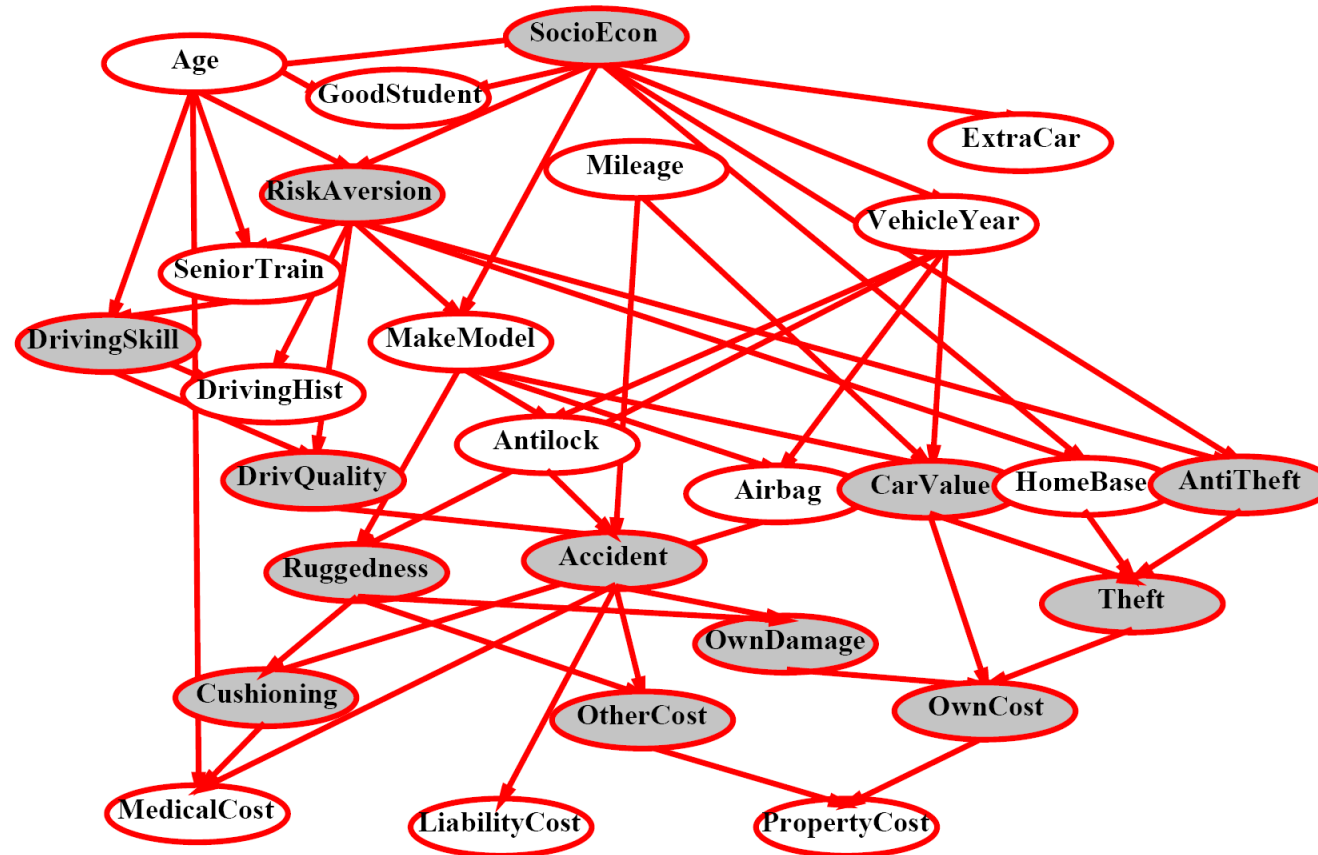
$$= \sum_{e,a} P(B, e, a, +j, +m)$$

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



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

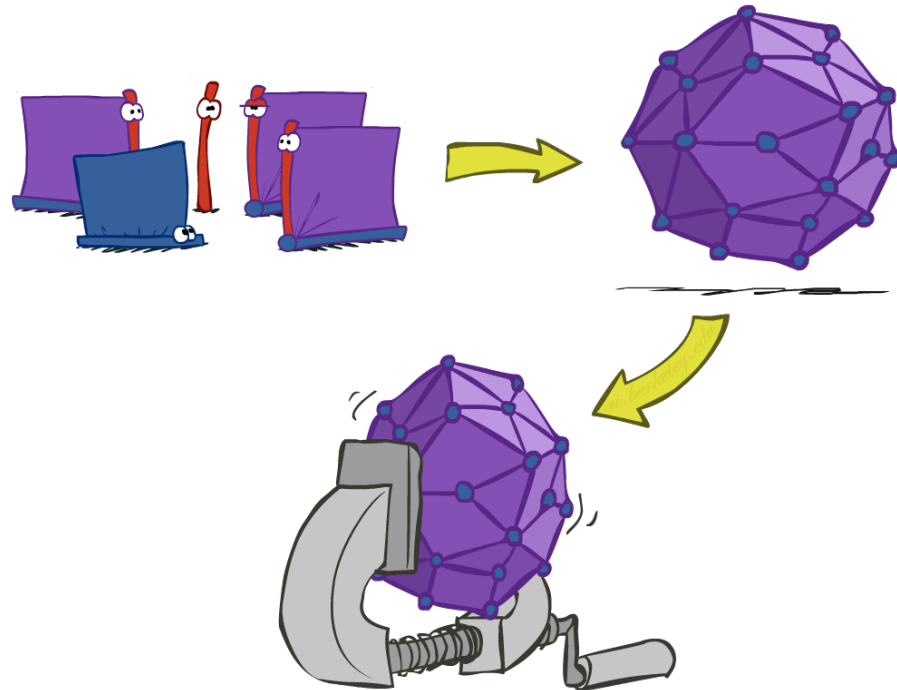
Inference by Enumeration?



Inference by Enumeration vs. Variable Elimination

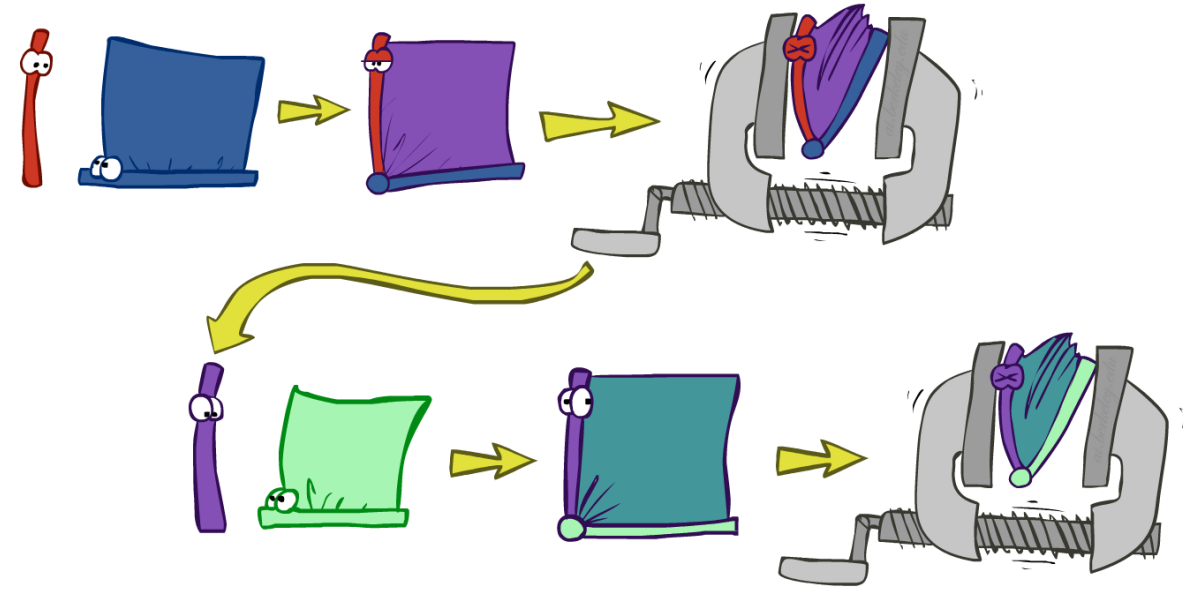
- Why is inference by enumeration so slow?

- 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

Example: Traffic Domain

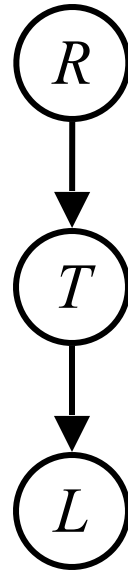
■ Random Variables

- 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

$$P(R)$$

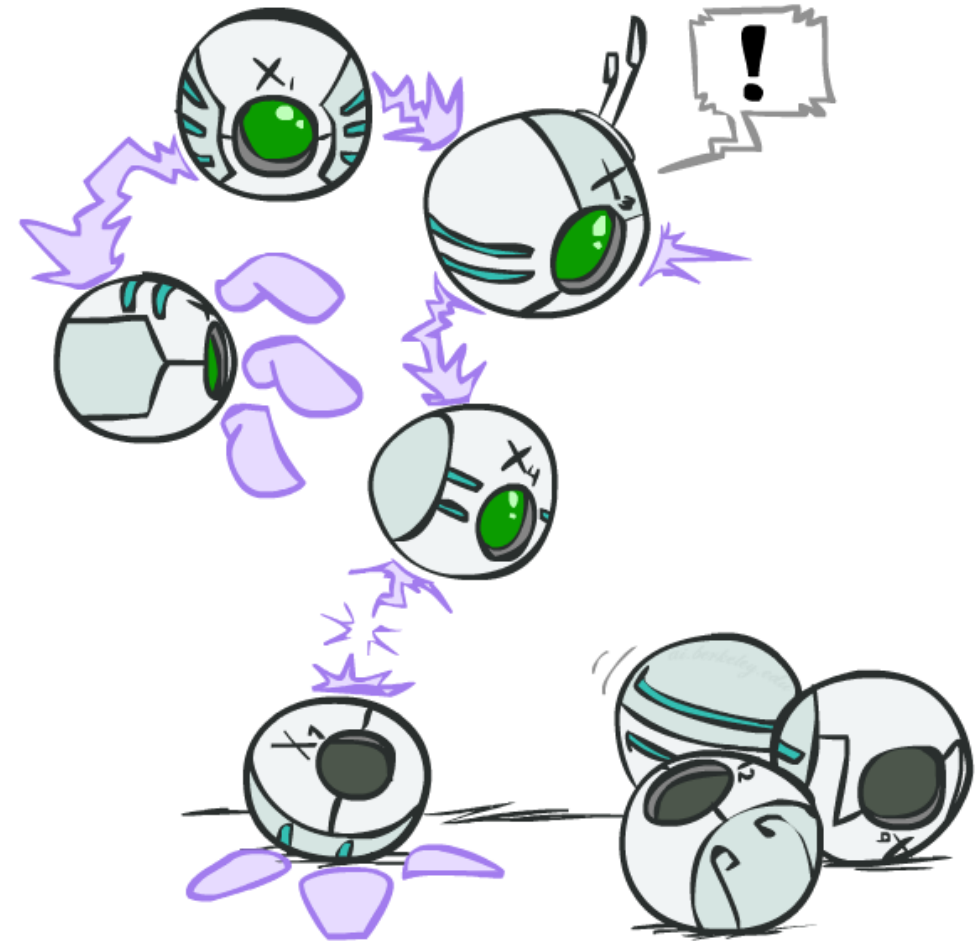
+r	0.1
-r	0.9

$$P(T|R)$$

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

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

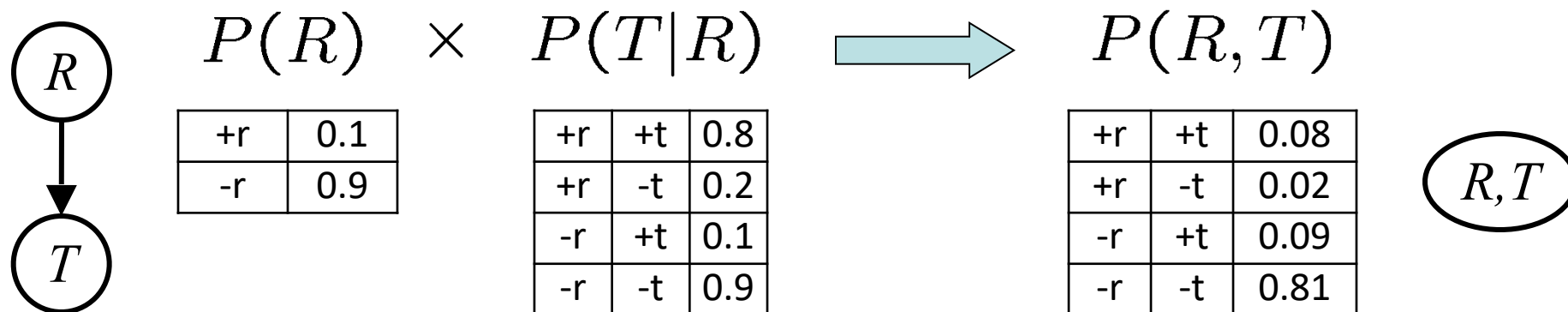
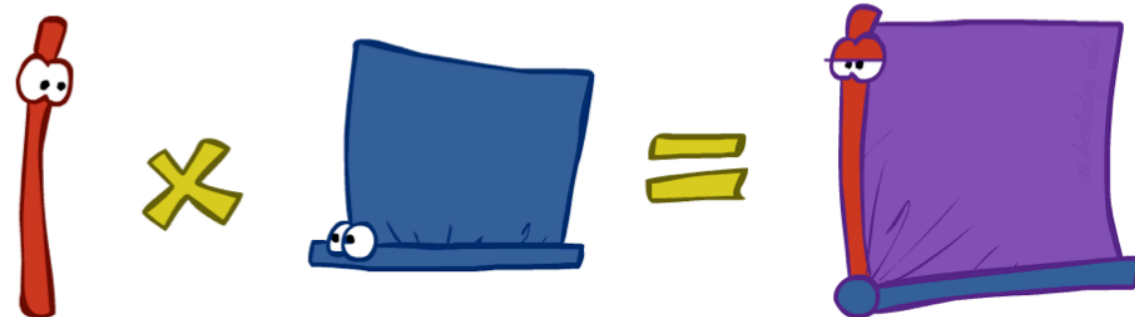
+t	+l	0.3
-t	+l	0.1



- Procedure: Join all factors, eliminate all hidden variables, normalize

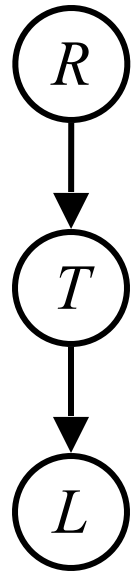
Operation 1: Join Factors

- 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



- 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

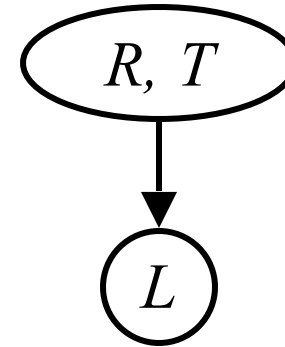


$P(R, T)$

+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



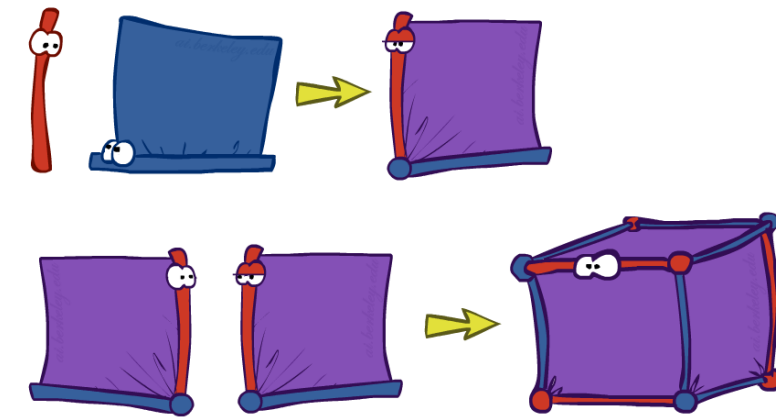
Join T



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:

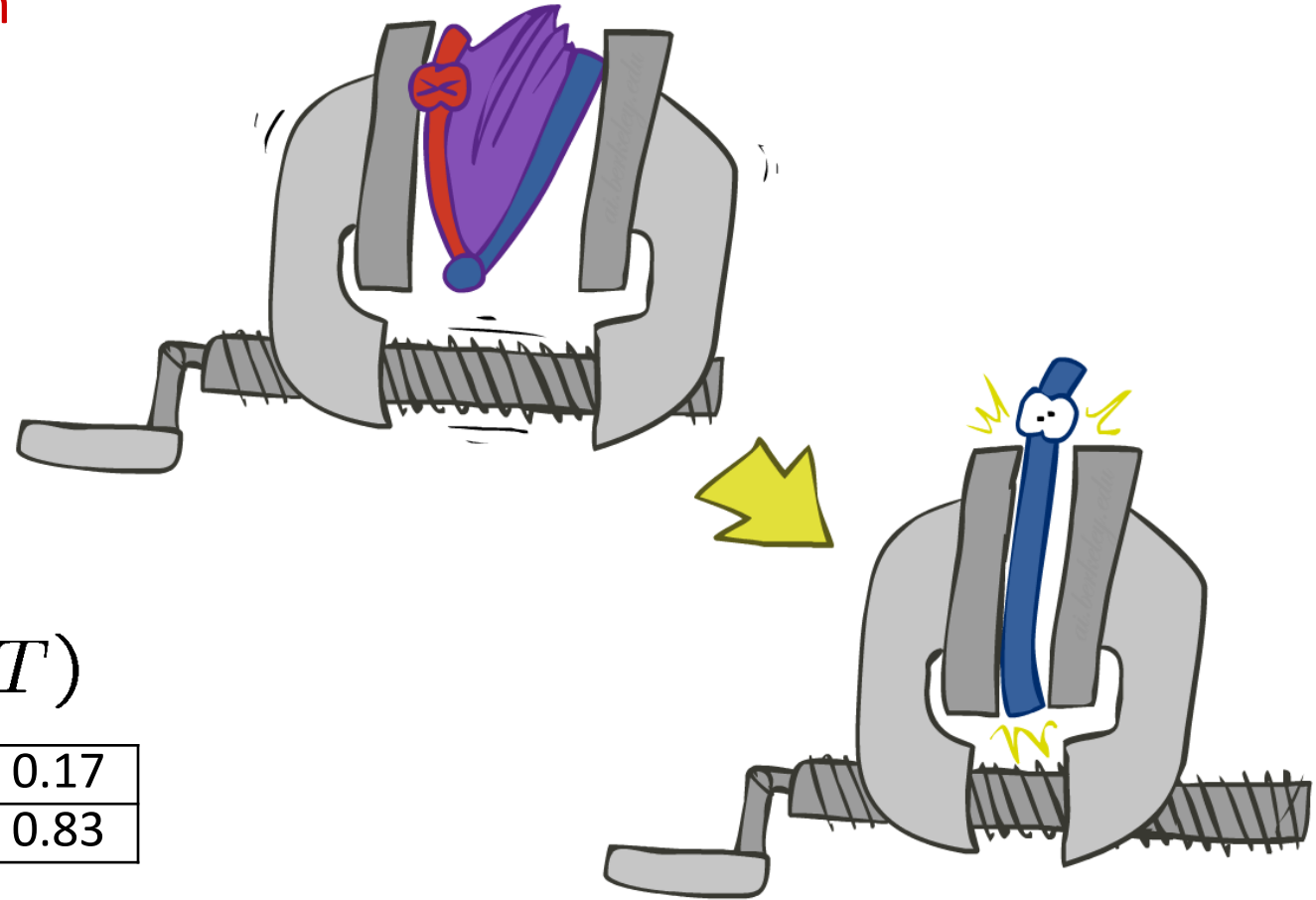
$$P(R, T)$$

+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

$P(R, T, L)$

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

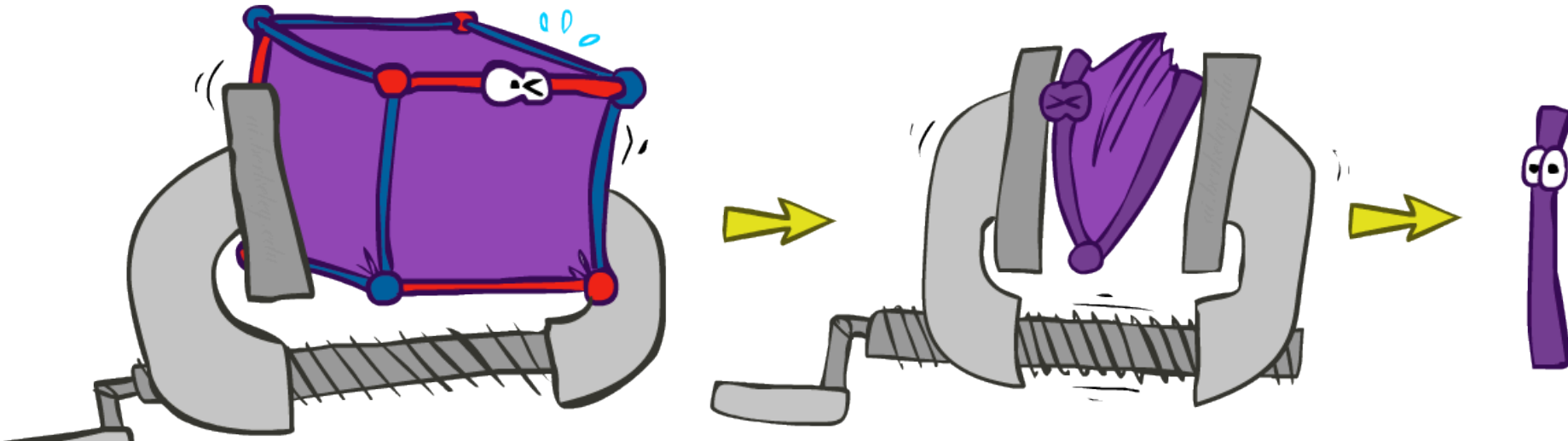
$P(T, L)$

T, L		
+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

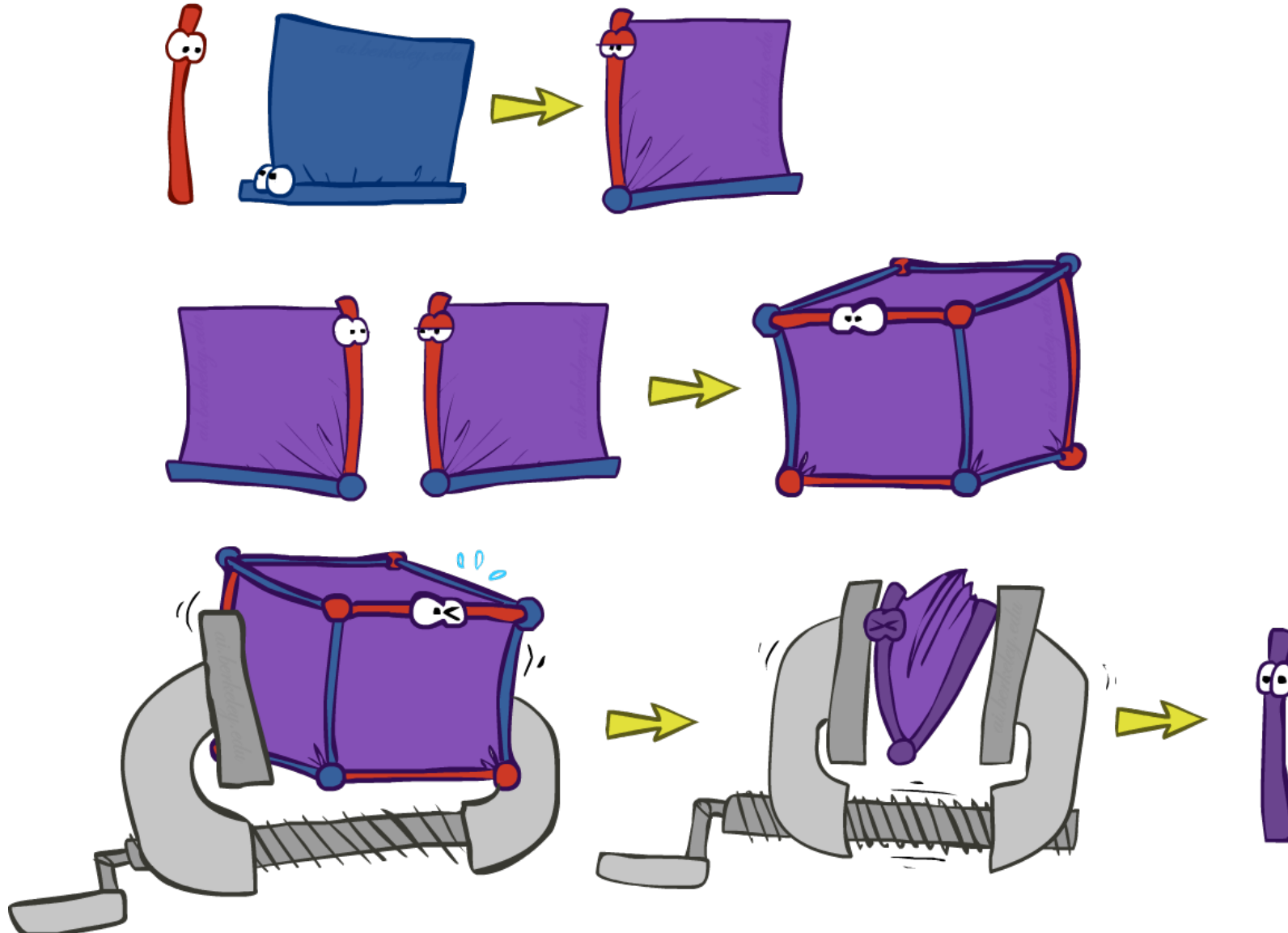
Sum out T

$P(L)$

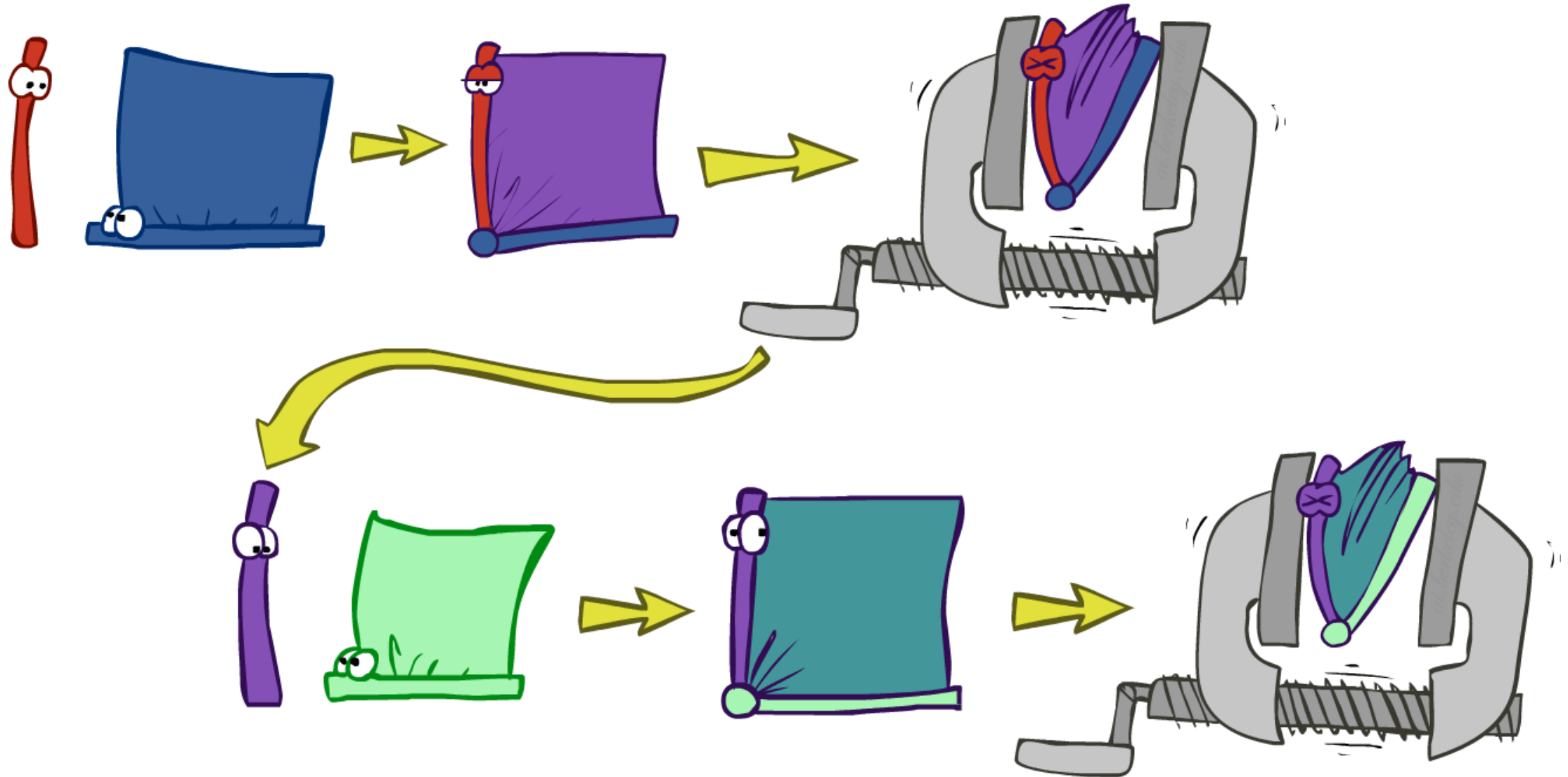
L	
+l	0.134
-l	0.886



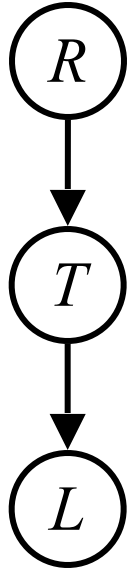
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)



Marginalizing Early (= Variable Elimination)



Traffic Domain

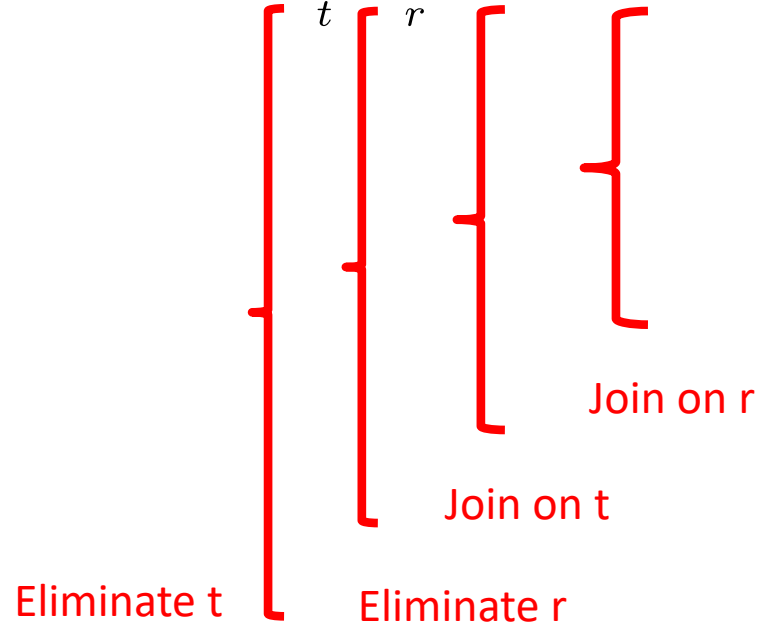


$$P(L) = ?$$

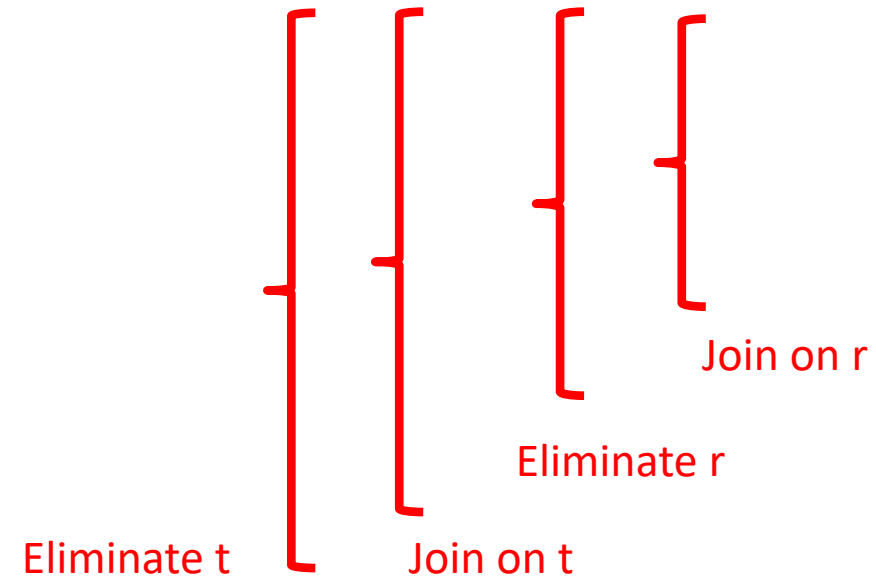
■ Inference by Enumeration

■ Variable Elimination

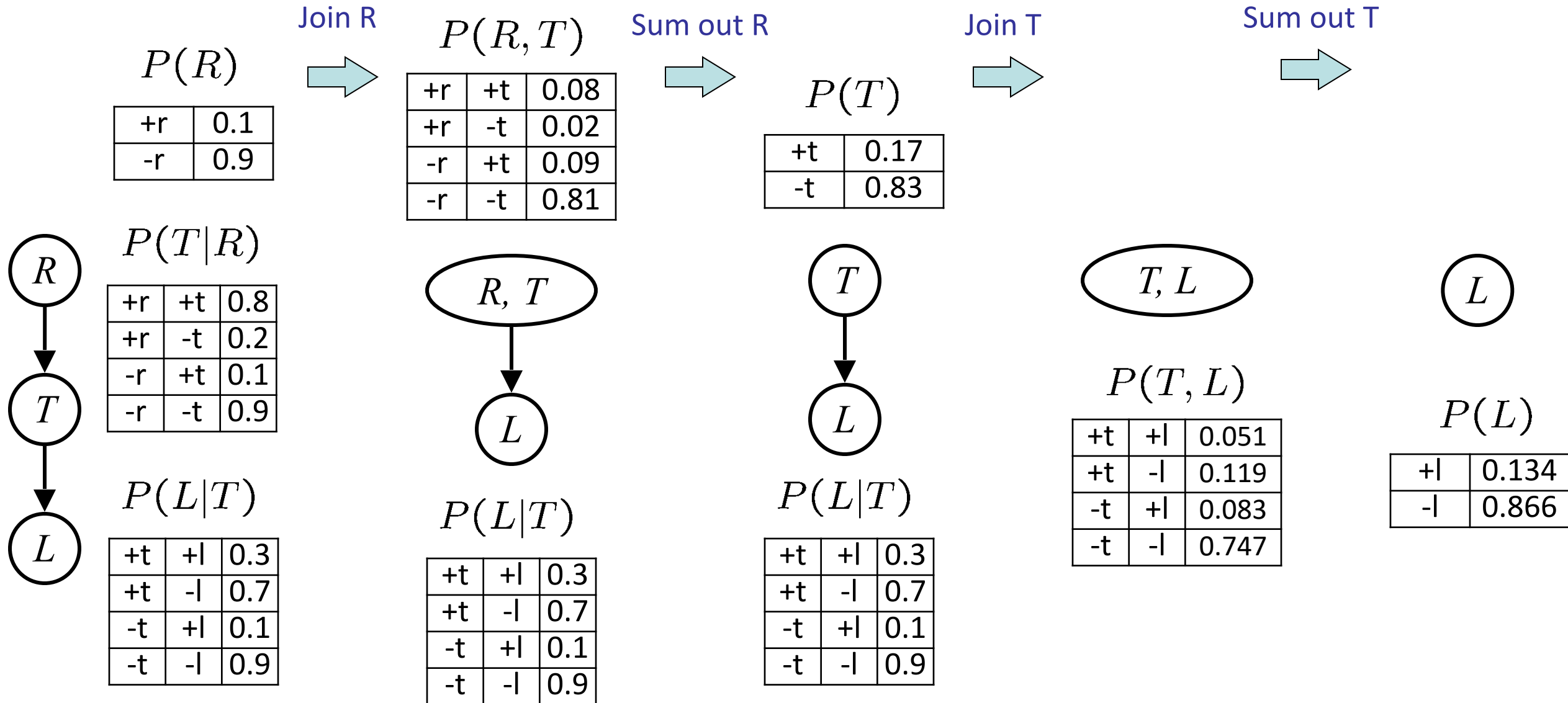
$$= \sum_t \sum_r P(L|t)P(r)P(t|r)$$



$$= \sum_t P(L|t) \sum_r P(r)P(t|r)$$



Marginalizing Early! (aka VE)



Evidence

- If evidence, start with factors that select that evidence
 - With no evidence, these are the 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

- For computing. $P(L|+r)$, the initial factors become:

$$P(+r)$$

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

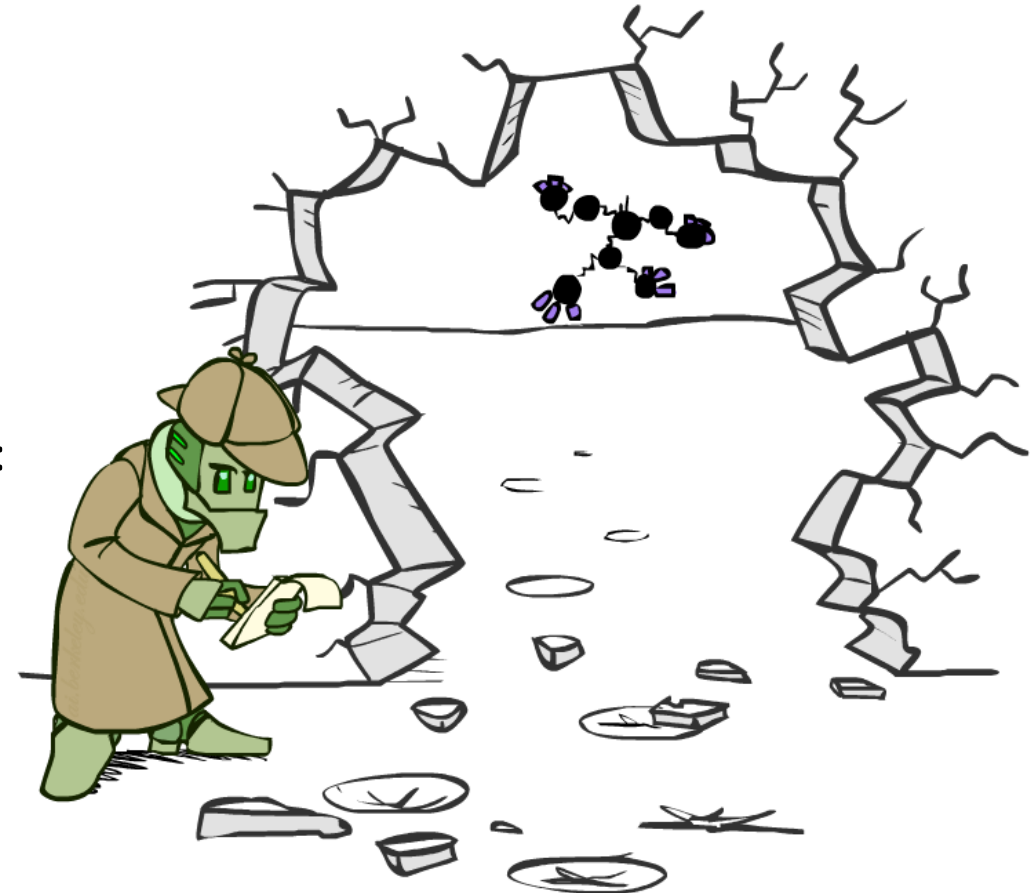
$$P(T|+r)$$

+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



Evidence II

- 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

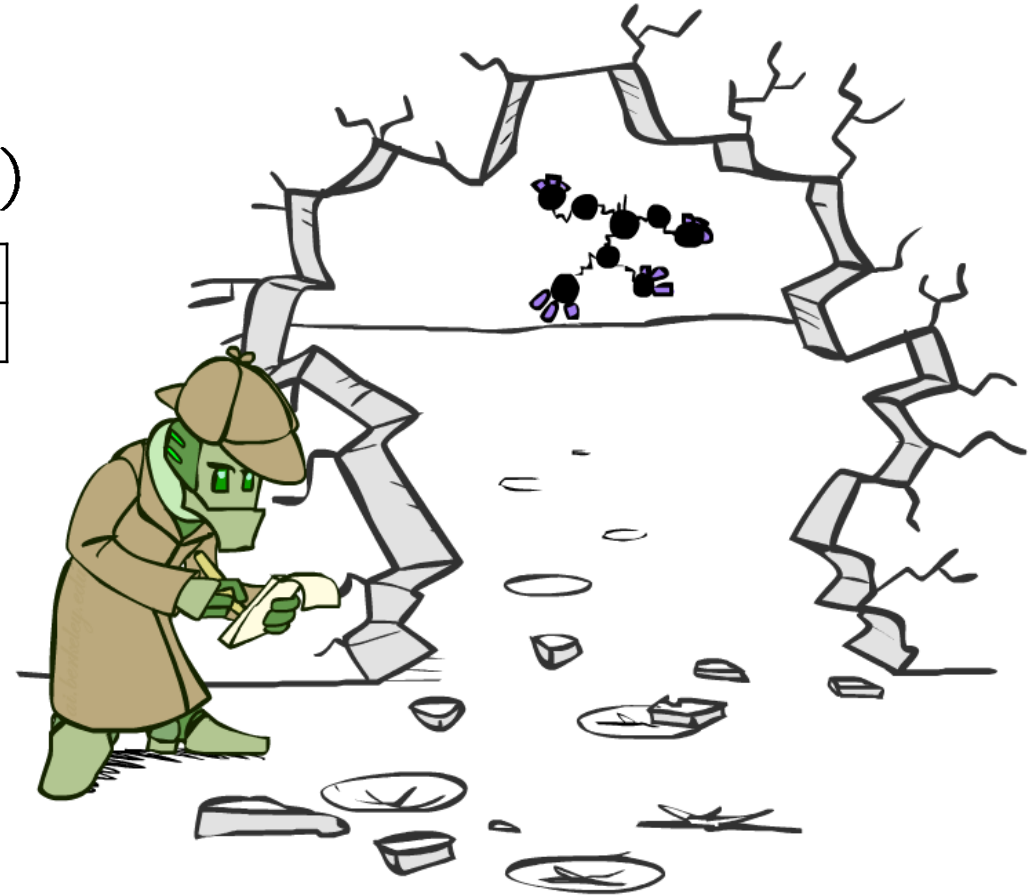
Normalize



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

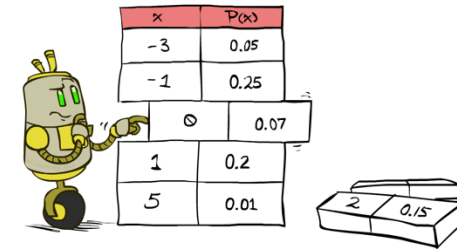
+l	0.26
-l	0.74

- To get our answer, just normalize this!
- That's it!



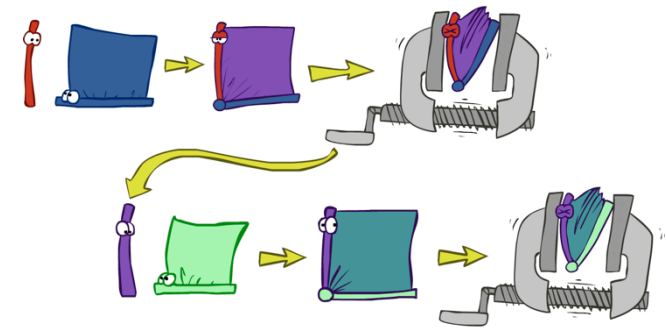
General Variable Elimination

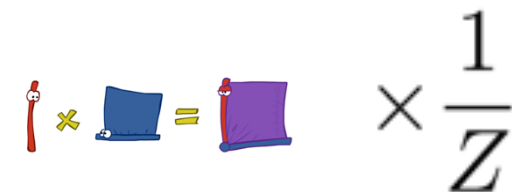
- Query: $P(Q|E_1 = e_1, \dots, E_k = e_k)$
- Start with initial factors:
 - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not Q or evidence):
 - Pick a hidden variable H_i
 - Join all factors mentioning H_i
 - Eliminate (sum out) H_i
- Join all remaining factors and normalize



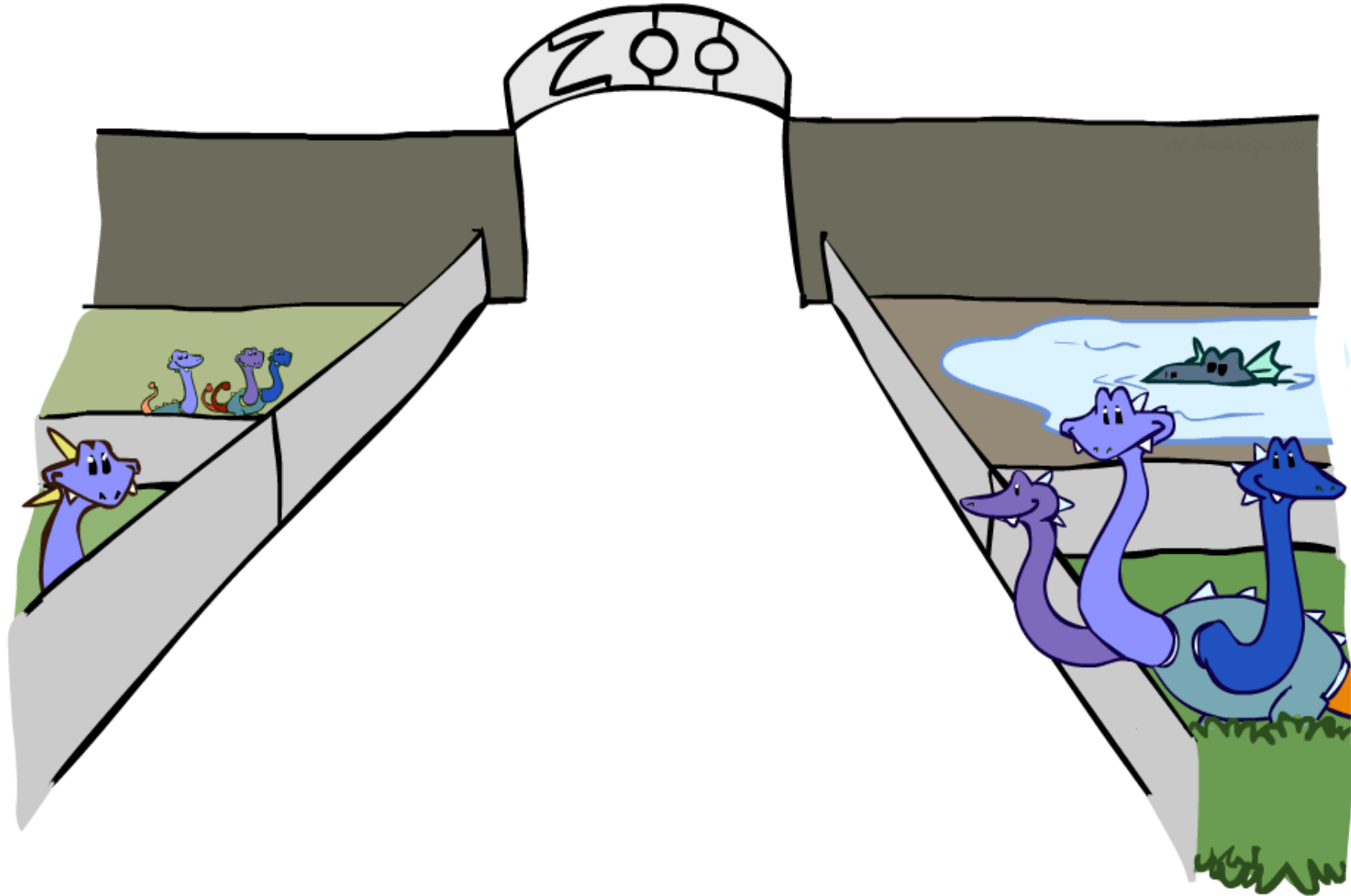
x	P(x)
-3	0.05
-1	0.25
0	0.07
1	0.2
5	0.01

A small box with the value 0.15 is shown next to the table.




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

Factor Zoo



Factor Zoo I

- 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 $P(x)$

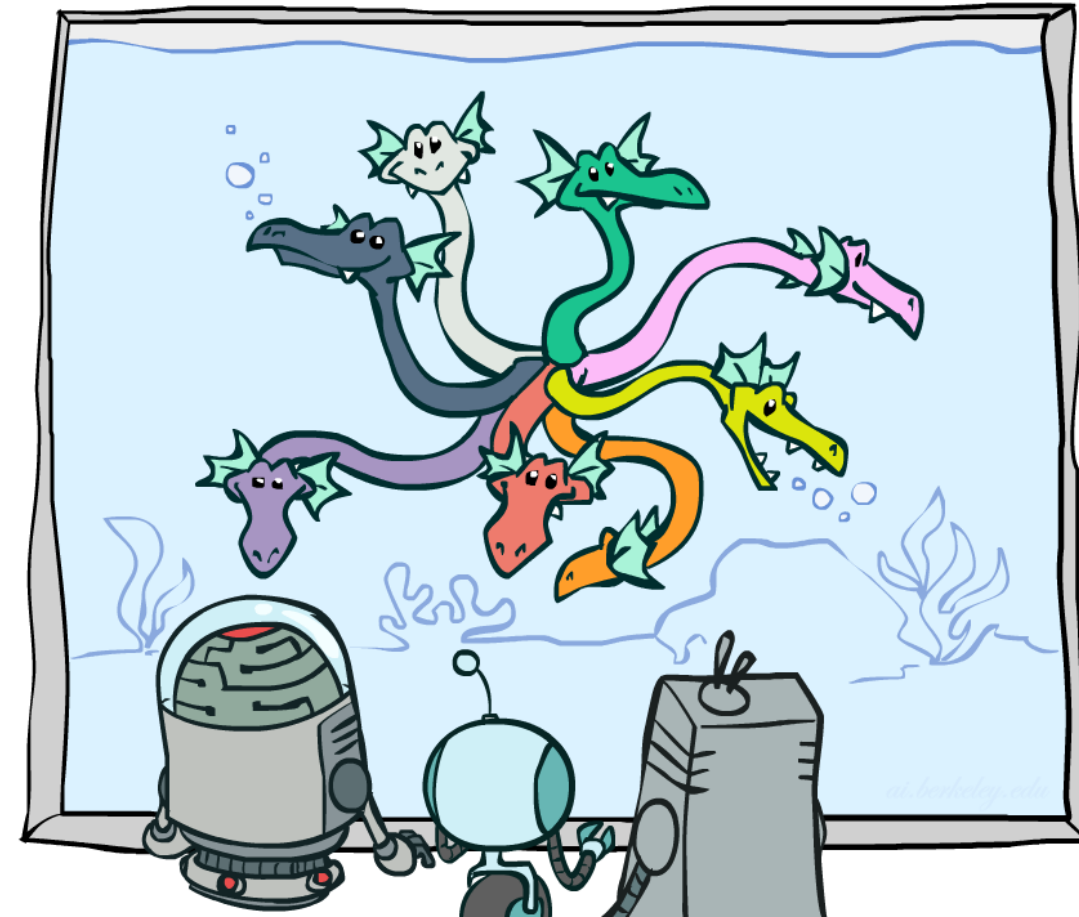
- Number of capitals =
dimensionality of the table

$P(T, W)$

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

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

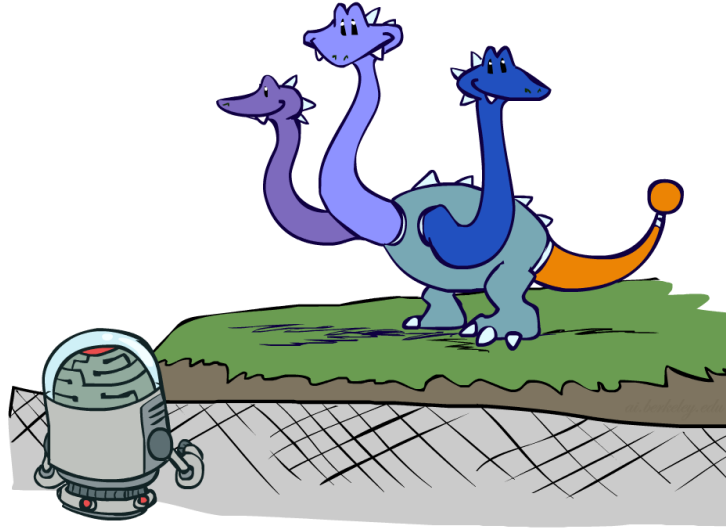
T	W	P
cold	sun	0.2
cold	rain	0.3



Factor Zoo II

- Single conditional: $P(Y | x)$

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



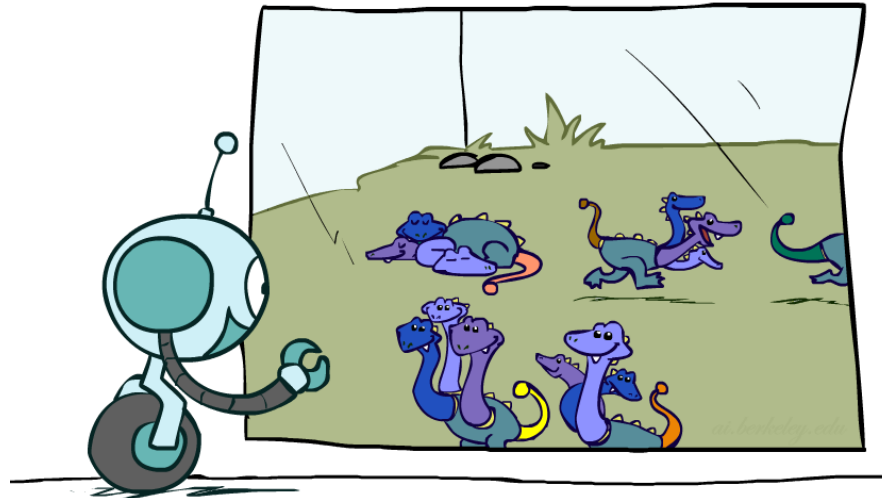
$$P(W|cold)$$

T	W	P
cold	sun	0.4
cold	rain	0.6

- Family of conditionals:

$$P(Y | X)$$

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



$$P(W|T)$$

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 III

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

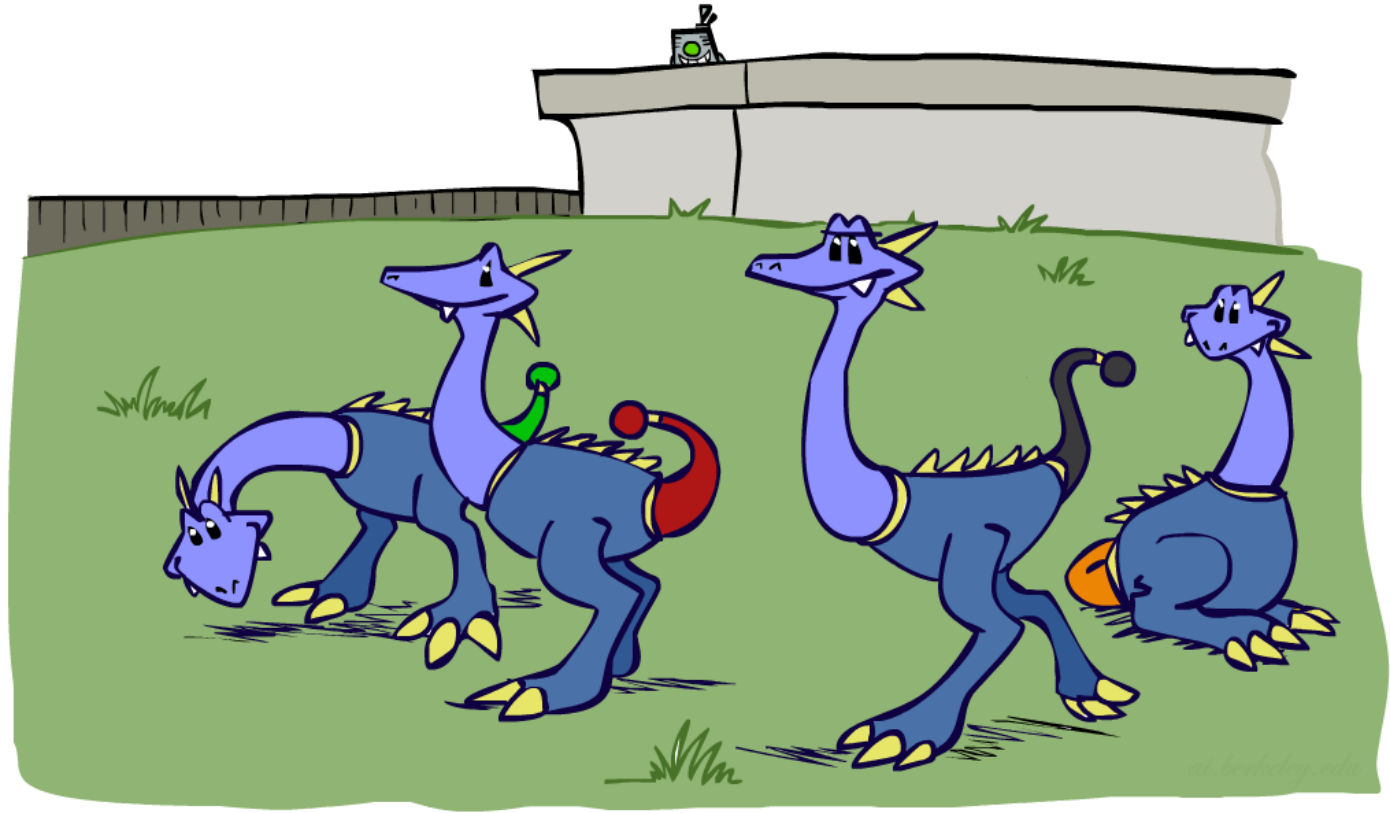
- Entries $P(y \mid x)$ for fixed y ,
but for all x
- Sums to ... who knows!

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

T	W	P
hot	rain	0.2
cold	rain	0.6

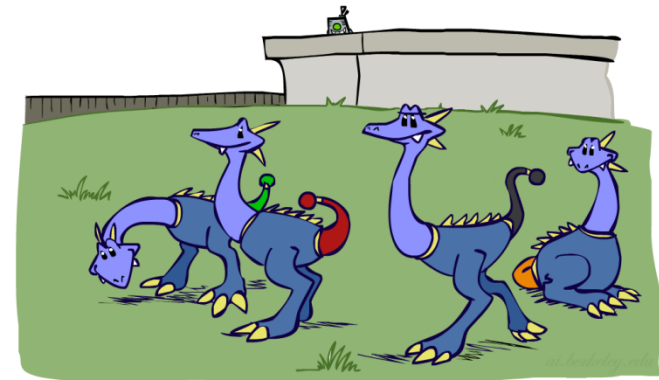
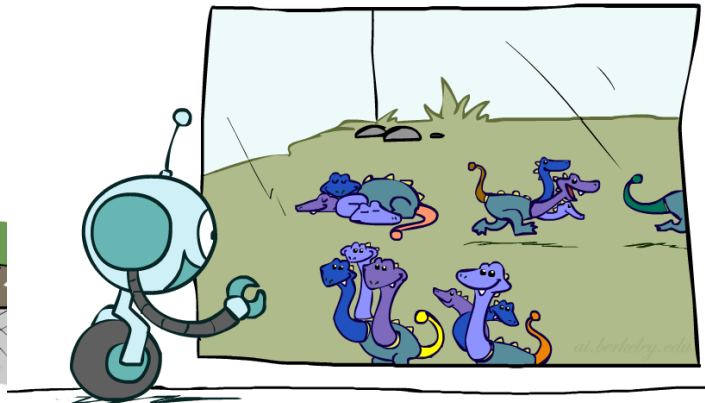
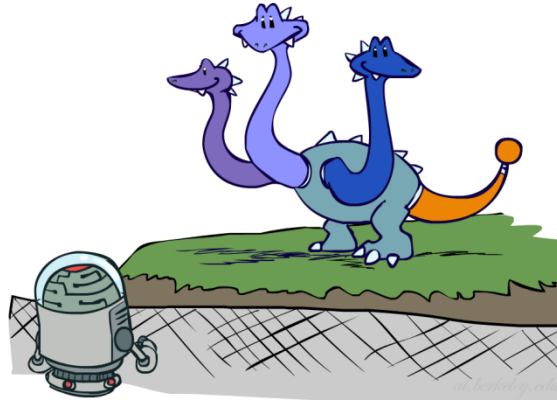
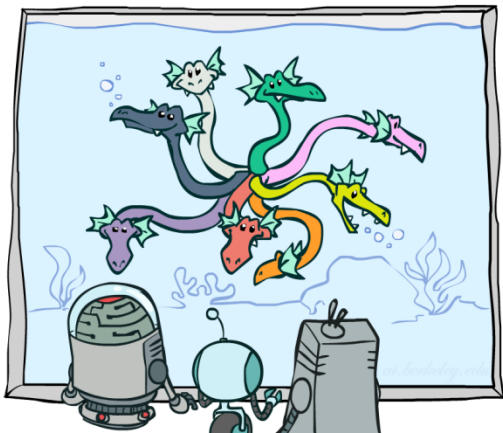
$$\left. \begin{array}{c} P(\text{rain} \mid \text{hot}) \\ P(\text{rain} \mid \text{cold}) \end{array} \right\}$$

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



Factor Zoo Summary

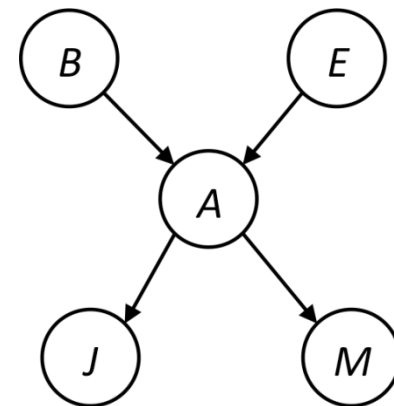
- In general, when we write $P(Y_1 \dots Y_N \mid X_1 \dots X_M)$
 - It is a “factor,” a multi-dimensional array
 - Its values are $P(y_1 \dots y_N \mid x_1 \dots x_M)$
 - Any assigned (=lower-case) X or Y is a dimension selected from the array



Example

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

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

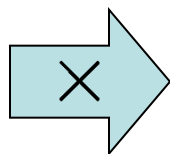


Choose A

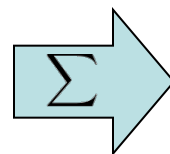
$$P(A|B, E)$$

$$P(j|A)$$

$$P(m|A)$$



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



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

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

Example

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

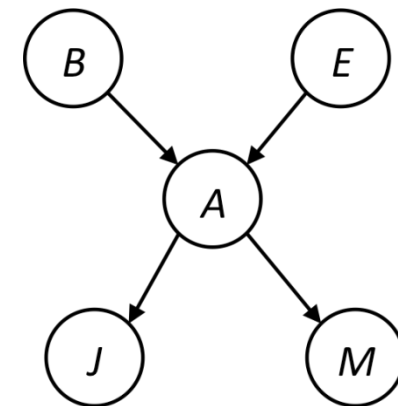
Choose E

$$\begin{array}{c} P(E) \\ P(j, m|B, E) \end{array} \xrightarrow{\times} P(j, m, E|B) \xrightarrow{\Sigma} P(j, m|B)$$

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

Finish with B

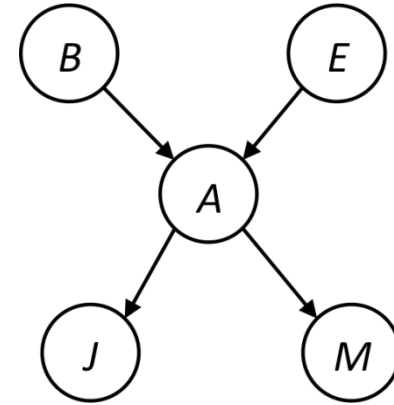
$$\begin{array}{c} P(B) \\ P(j, m|B) \end{array} \xrightarrow{\times} P(j, m, B) \xrightarrow{\text{Normalize}} P(B|j, m)$$



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 obtained from joint by summing out

use Bayes' net joint distribution expression

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

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

All we are doing is exploiting $uwv + uwz + uxy + uxz + vwy + vwz + vxy + vxz = (u+v)(w+x)(y+z)$ to improve computational efficiency!

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:

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

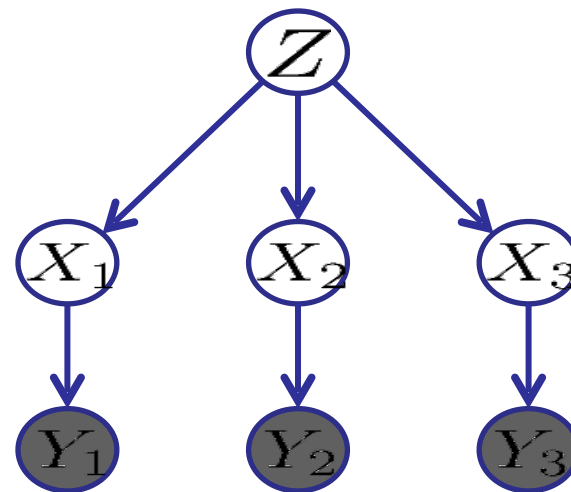
Eliminate Z , this introduces the factor $f_3(y_1, y_2, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z)$, and we are left:

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

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

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

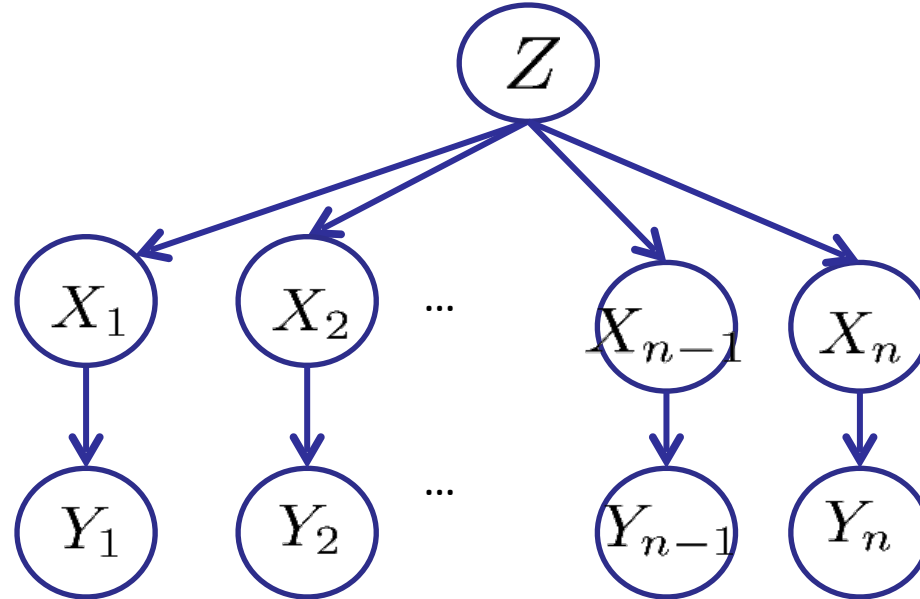
Normalizing over X_3 gives $P(X_3|y_1, y_2, y_3)$.



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!

Worst Case Complexity?

- CSP:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5) \wedge (x_2 \vee x_5 \vee x_7) \wedge (x_4 \vee x_5 \vee x_6) \wedge (\neg x_5 \vee x_6 \vee \neg x_7) \wedge (\neg x_5 \vee \neg x_6 \vee x_7)$$

$$P(X_i = 0) = P(X_i = 1) = 0.5$$

$$Y_1 = X_1 \vee X_2 \vee \neg X_3$$

...

$$Y_8 = \neg X_5 \vee X_6 \vee X_7$$

$$Y_{1,2} = Y_1 \wedge Y_2$$

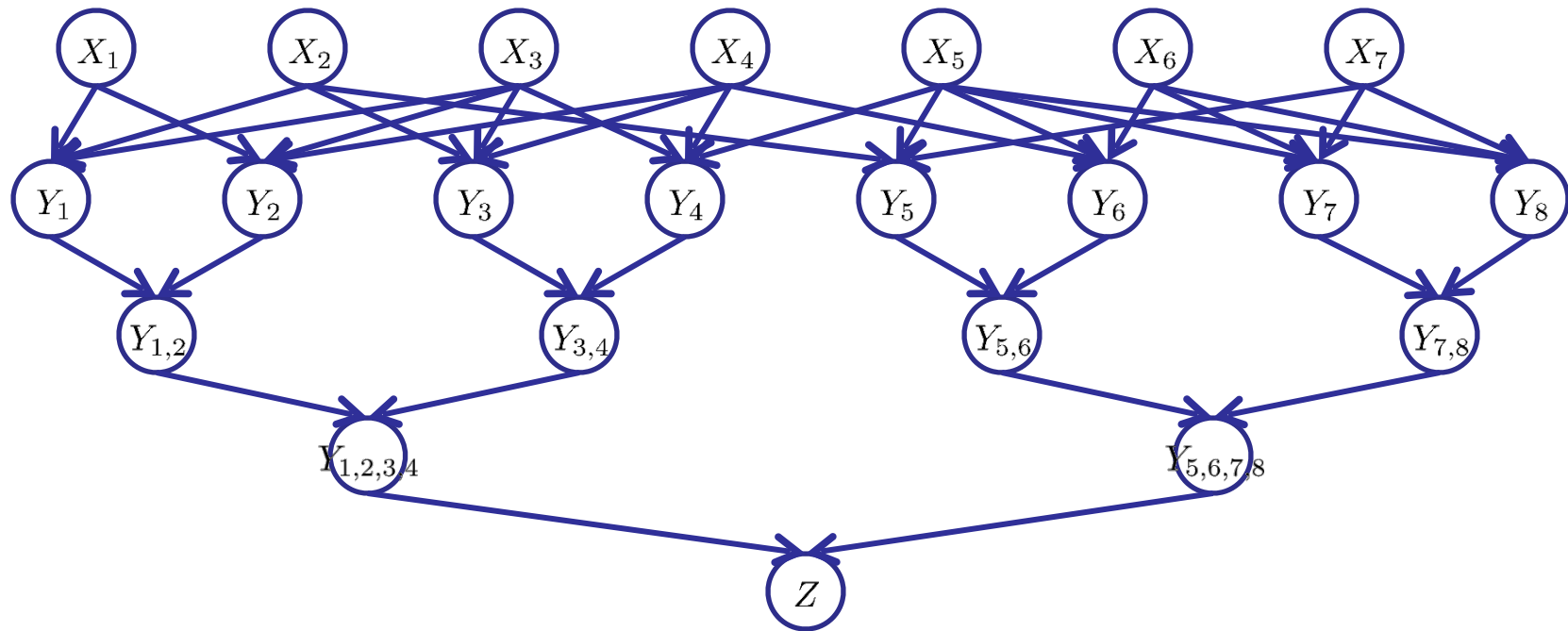
...

$$Y_{7,8} = Y_7 \wedge Y_8$$

$$Y_{1,2,3,4} = Y_{1,2} \wedge Y_{3,4}$$

$$Y_{5,6,7,8} = Y_{5,6} \wedge Y_{7,8}$$

$$Z = Y_{1,2,3,4} \wedge Y_{5,6,7,8}$$



- If we can answer $P(z)$ equal to zero or not, we answered whether the 3-SAT problem has a solution.
- Hence inference in Bayes' nets is NP-hard. No known efficient probabilistic inference in general.

Polytrees

- A polytree is a directed graph with no undirected cycles
- For poly-trees you can always find an ordering that is efficient
 - Try it!!
- Cut-set conditioning for Bayes' net inference
 - Choose set of variables such that if removed only a polytree remains
 - Exercise: Think about how the specifics would work out!

Bayes' Nets

- ✓ Representation

- ✓ Conditional Independences

- Probabilistic Inference

- ✓ Enumeration (exact, exponential complexity)

- Variable elimination (exact, worst-case
✓ exponential complexity, often better)

- Inference is NP-complete

- ✓ Sampling (approximate)

- Learning Bayes' Nets from Data