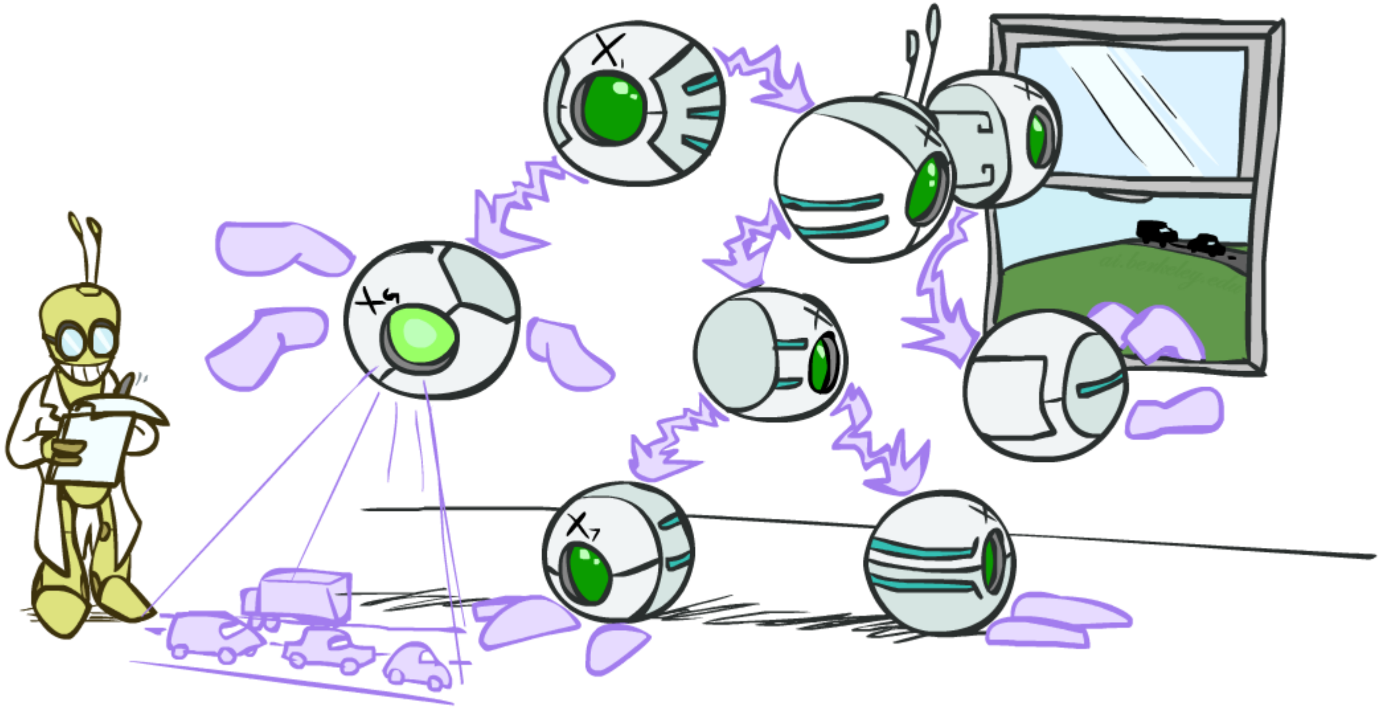
CS 343: Artificial Intelligence

Bayes Nets: Independence and Inference



Profs. Peter Stone and Yuke Zhu — The University of Texas at Austin

[These slides based on those of Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

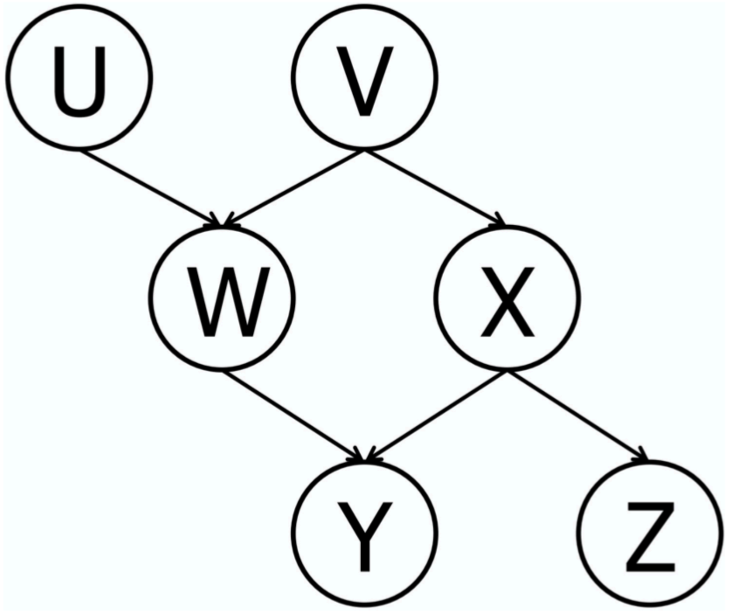
Tuesday In-Class Exercises

Independence

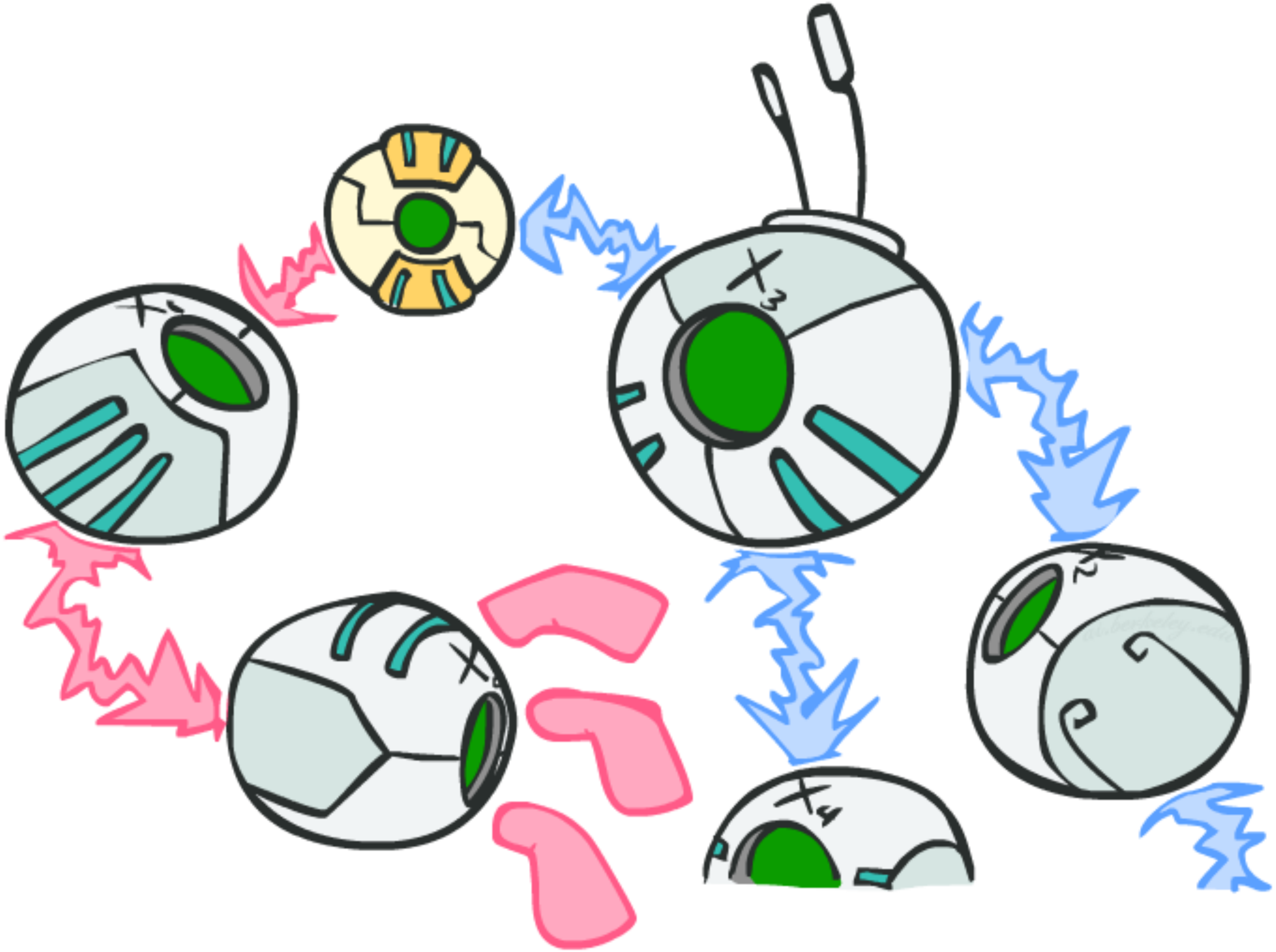
Based only on the structure of the Bayes' Net given right, indicate whether the following conditional independence assertions are either 1) *guaranteed to be true*, 2) *guaranteed to be false*, or 3) *cannot be determined* by the structure alone.

Hint: the meaning of A ⊥ B | C, D is A and B are independent (⊥) of each other other conditioned on (|) C and D

i. U ⊥ V — guaranteed to be true
ii. U ⊥ V | W — cannot be determined
iii. U ⊥ Z | W — cannot be determined
iv. U ⊥ Z | X, W — guaranteed to be true
v. V ⊥ Z | X — guaranteed to be true



D-separation: Outline



D-separation: Outline

- Study independence properties for triples
- Analyze complex cases in terms of member triples
- D-separation: a condition / algorithm for answering such queries

Causal Chains

- This configuration is a "causal chain"

X: Low pressure Y: Rain Z: Traffic

$$P(x,y,z) = P(x)P(y|x)P(z|y)$$

- Guaranteed X independent of Z ? *No!*
  - One example set of CPTs for which X is not independent of Z is sufficient to show this independence is not guaranteed.
  - Example:
    - Low pressure causes rain causes traffic, high pressure causes no rain causes no traffic
    - In numbers:

      P(+y | +x) = 1, P(-y | - x) = 1,
      P(+z | +y) = 1, P(-z | -y) = 1

Causal Chains

- This configuration is a "causal chain"

X: Low pressure   Y: Rain   Z: Traffic

$$P(x,y,z) = P(x)P(y|x)P(z|y)$$

- Guaranteed X independent of Z given Y?

$$P(z|x,y) = \frac{P(x,y,z)}{P(x,y)}$$

$$= \frac{P(x)P(y|x)P(z|y)}{P(x)P(y|x)}$$

$$= P(z|y)$$

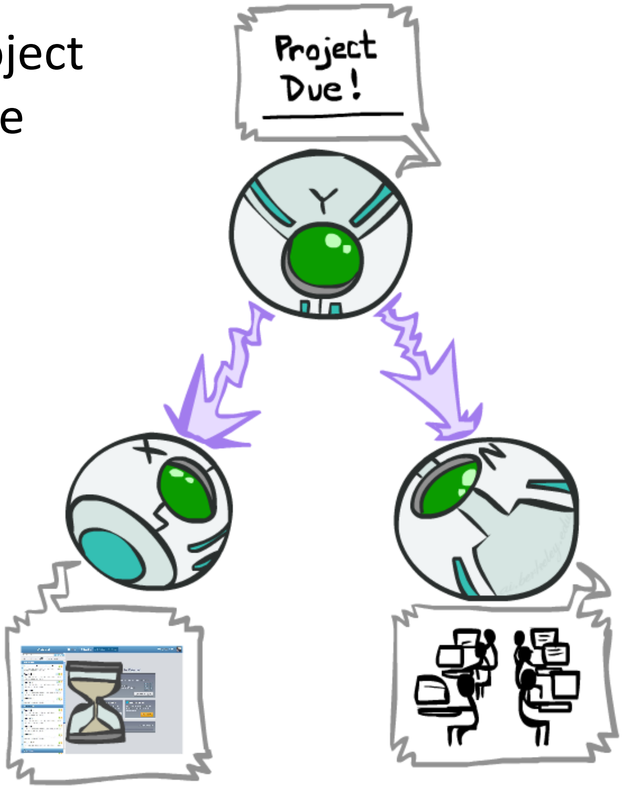
Yes!

- Evidence along the chain "blocks" the influence

Common Cause

- This configuration is a “common cause”

Y: Project due



X: Forums busy

Z: Lab full

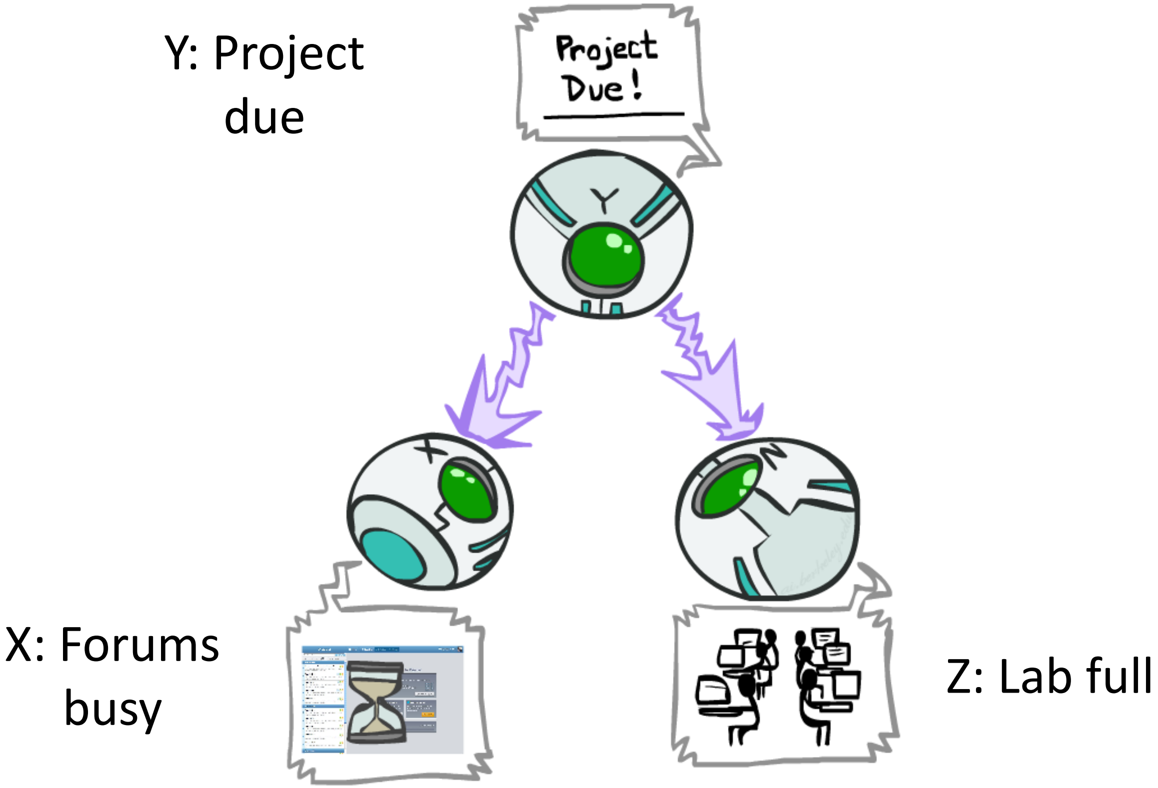
$$P(x, y, z) = P(y)P(x|y)P(z|y)$$

- Guaranteed X independent of Z ? *No!*
  - One example set of CPTs for which X is not independent of Z is sufficient to show this independence is not guaranteed.
  - Example:
    - Project due causes both forums busy and lab full
    - In numbers:

      P(+x | +y) = 1, P(-x | -y) = 1,
      P(+z | +y) = 1, P(-z | -y) = 1

Common Cause

- This configuration is a "common cause"



$$P(x, y, z) = P(y)P(x|y)P(z|y)$$

- Guaranteed X and Z independent given Y?

$$P(z|x, y) = \frac{P(x, y, z)}{P(x, y)}$$

$$= \frac{P(y)P(x|y)P(z|y)}{P(y)P(x|y)}$$

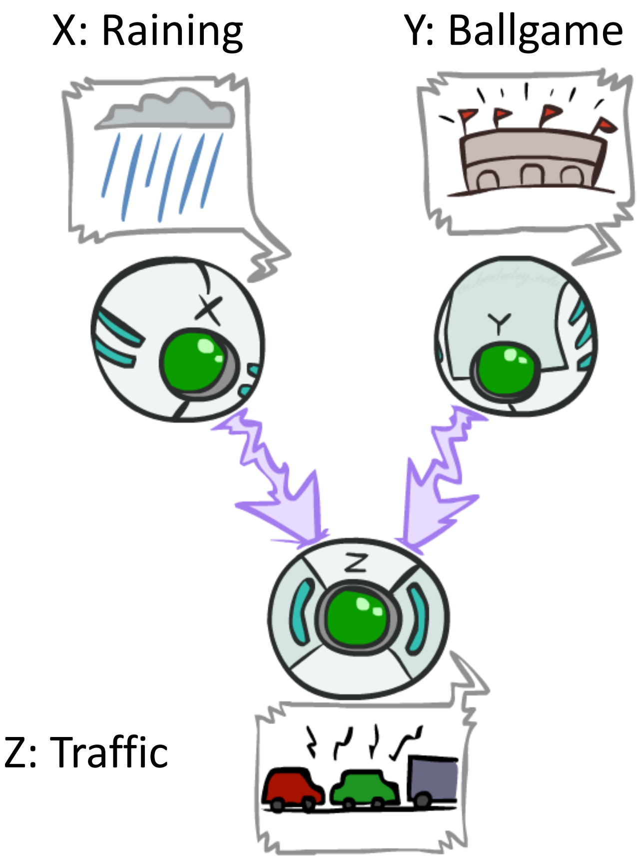
$$= P(z|y)$$

Yes!

- Observing the cause blocks influence between effects.

Common Effect

- Last configuration: two causes of one effect (v-structures)

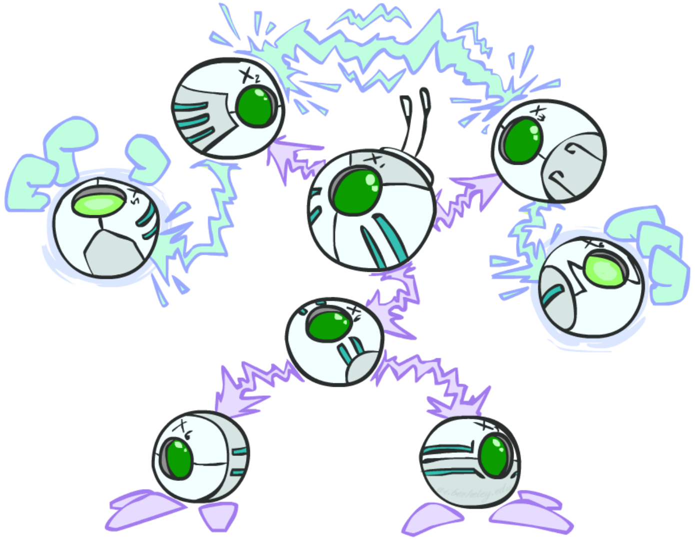


- Are X and Y independent?
  - *Yes*: the ballgame and the rain cause traffic, but they are not correlated
  - Still need to prove they must be (try it!)
- Are X and Y independent given Z?
  - *No*: seeing traffic puts the rain and the ballgame in competition as explanation.
- This is backwards from the other cases
  - Observing an effect activates influence between possible causes.

The General Case

The General Case

- General question: in a given BN, are two variables independent (given evidence)?
- Solution: analyze the graph
- Any complex example can be broken into repetitions of the three canonical cases



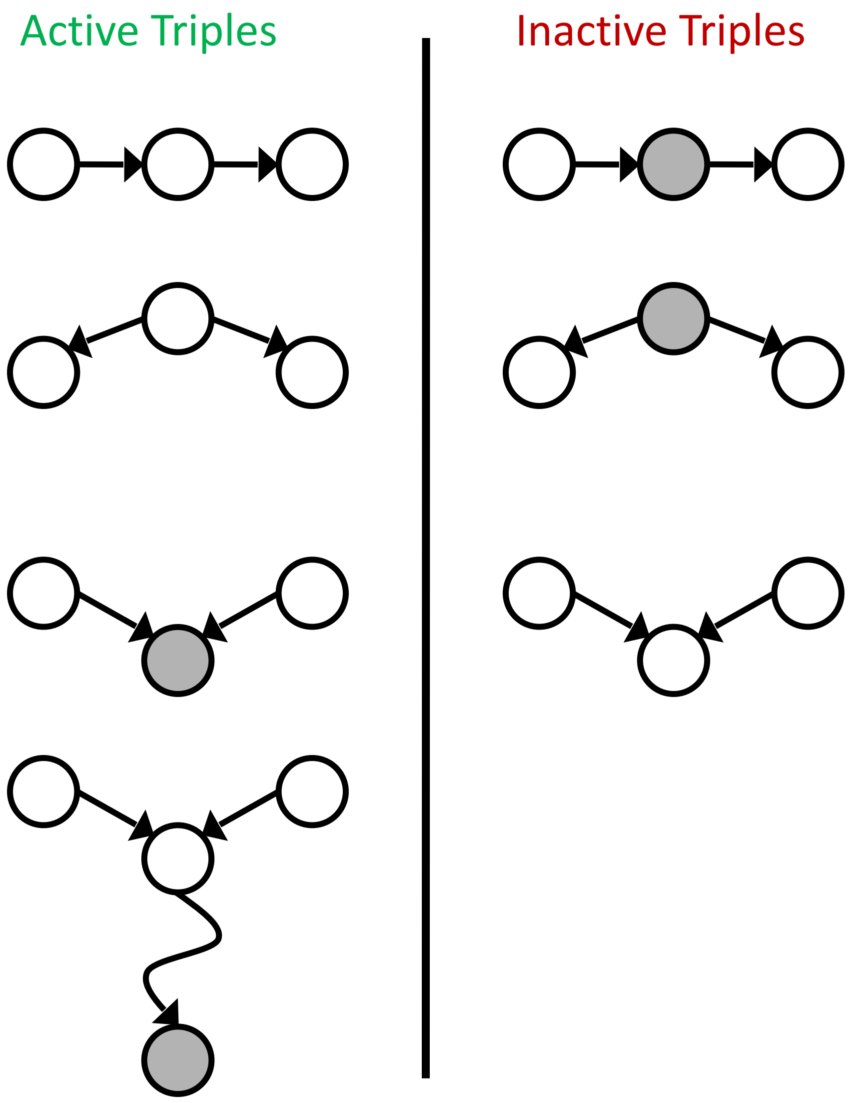
Active / Inactive Paths

- Question: Are X and Y conditionally independent given evidence variables {Z}?
  - Yes, if X and Y "d-separated" by Z
  - Consider all (undirected) paths from X to Y
  - No active paths = conditional independence!

- A path is active if each triple is active:
  - Causal chain A → B → C where B is unobserved (either direction)
  - Common cause A ← B → C where B is unobserved
  - Common effect (aka v-structure)

    A → B ← C where B *or one of its descendants* is observed

- All it takes to block a path is a single inactive segment



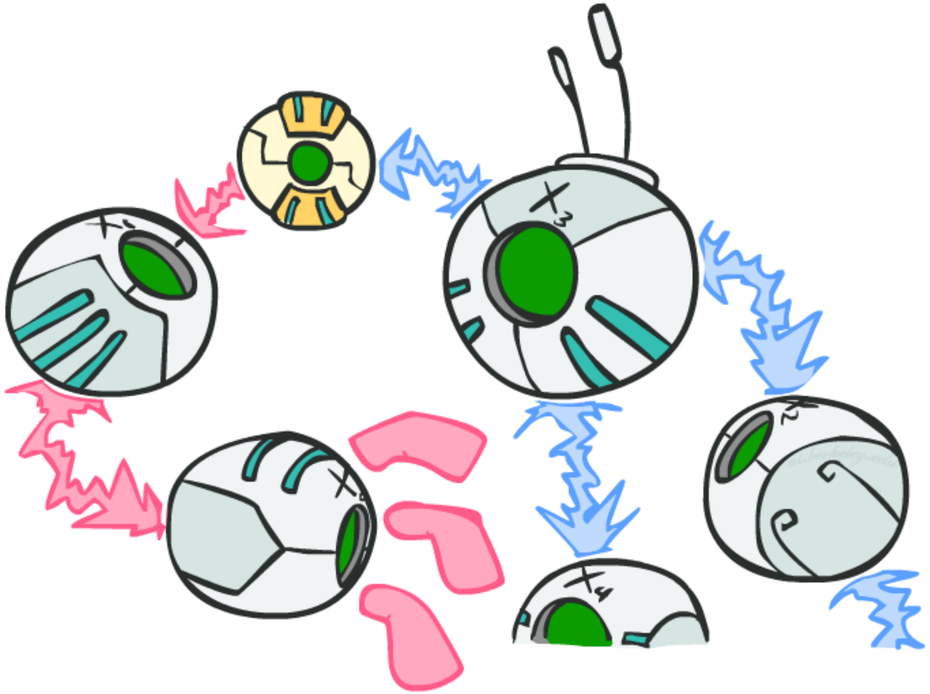
D-Separation

- Query: $X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$?
- Check all (undirected!) paths between X_i and X_j
  - If one or more active, then independence not guaranteed

$$X_i \not\!\perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

  - Otherwise (i.e. if all paths are inactive), then independence is guaranteed

$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

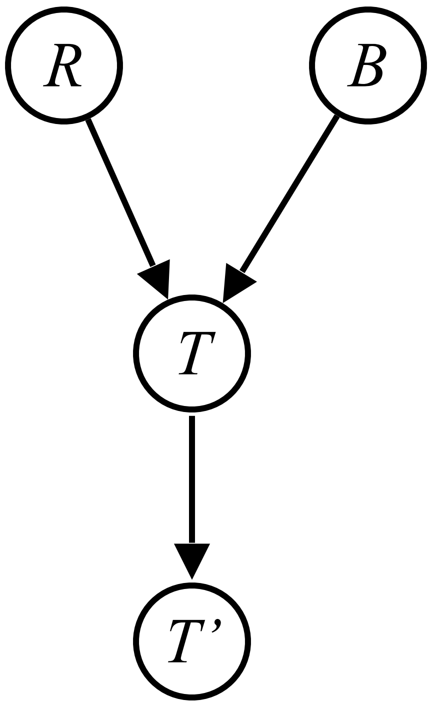


Example

$R \perp\!\!\!\perp B$ *Yes*

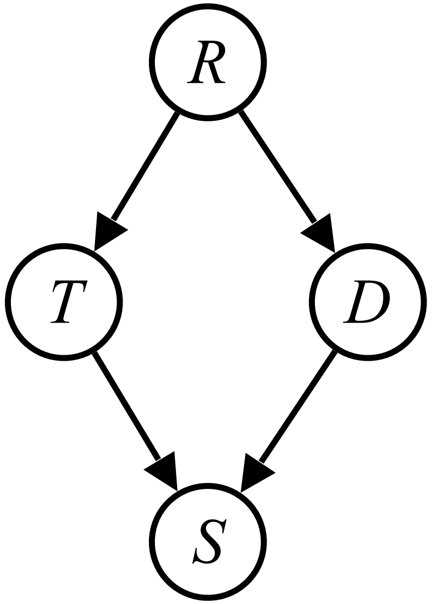
$R \perp\!\!\!\perp B | T$ *Not guaranteed*

$R \perp\!\!\!\perp B | T'$ *Not guaranteed*



Example

- Variables:
  - R: Raining
  - T: Traffic
  - D: Roof drips
  - S: I'm sad



- Questions:

$T \perp\!\!\!\perp D$ *Not guaranteed*

$T \perp\!\!\!\perp D | R$ *Yes*

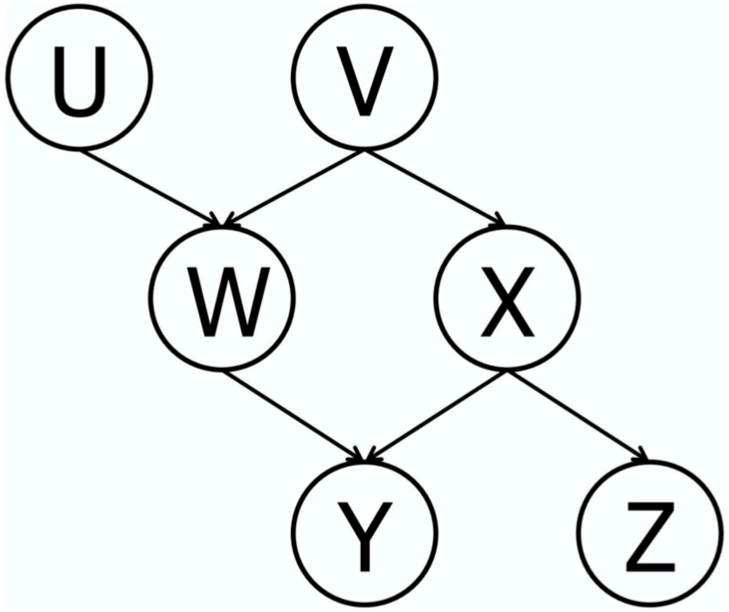
$T \perp\!\!\!\perp D | R, S$ *Not guaranteed*

Tuesday In-Class Exercises

Independence

Based only on the structure of the Bayes' Net given right, indicate whether the following conditional independence assertions are either 1) *guaranteed to be true*, 2) *guaranteed to be false*, or 3) *cannot be determined* by the structure alone.

Hint: the meaning of $A \perp B \mid C, D$ is A and B are independent ($\perp$) of each other other conditioned on (|) C and D



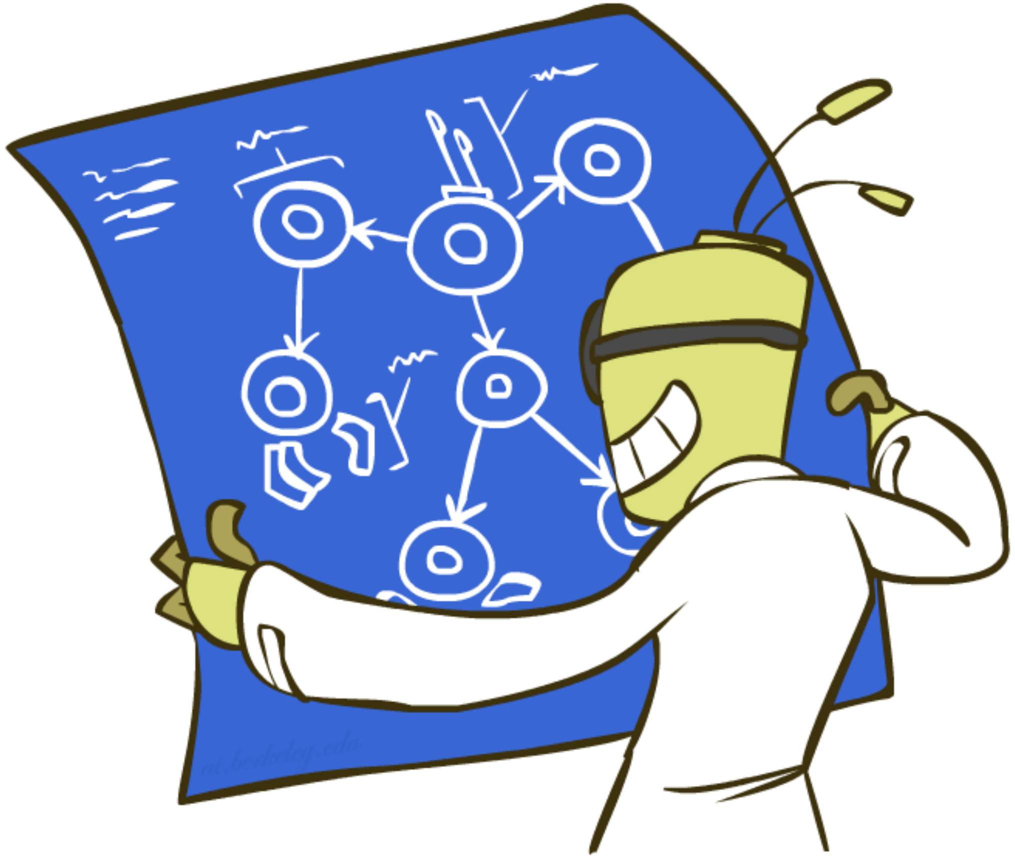
i. $U \perp V$ — guaranteed to be true
ii. $U \perp V \mid W$ — cannot be determined
iii. $U \perp Z \mid W$ — cannot be determined
iv. $U \perp Z \mid X, W$ — guaranteed to be true
v. $V \perp Z \mid X$ — guaranteed to be true

Structure Implications

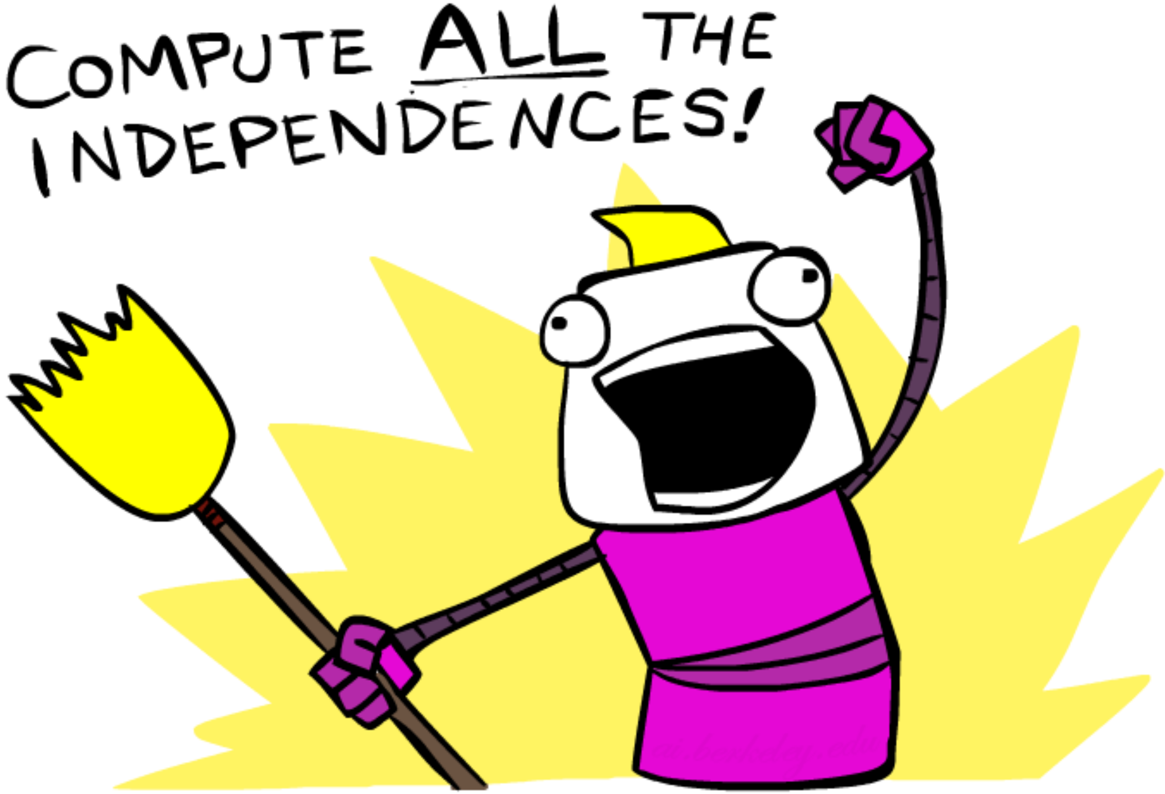
- Given a Bayes net structure, can run d-separation algorithm to build a complete list of conditional independences that are necessarily true of the form

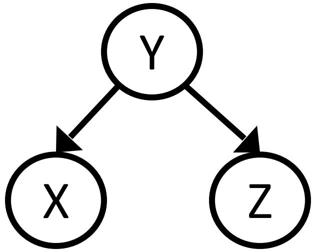
$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

- This list determines the set of probability distributions that can be represented

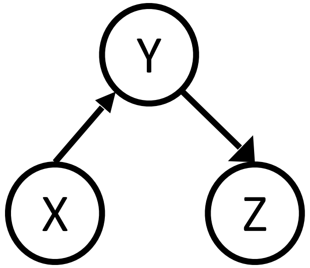


Computing All Independences

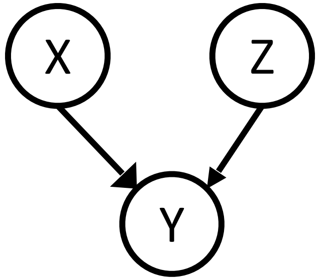




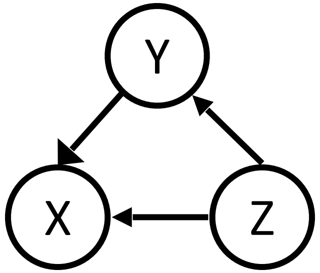
$X \perp\!\!\!\perp Z \mid Y$



$X \perp\!\!\!\perp Z \mid Y$



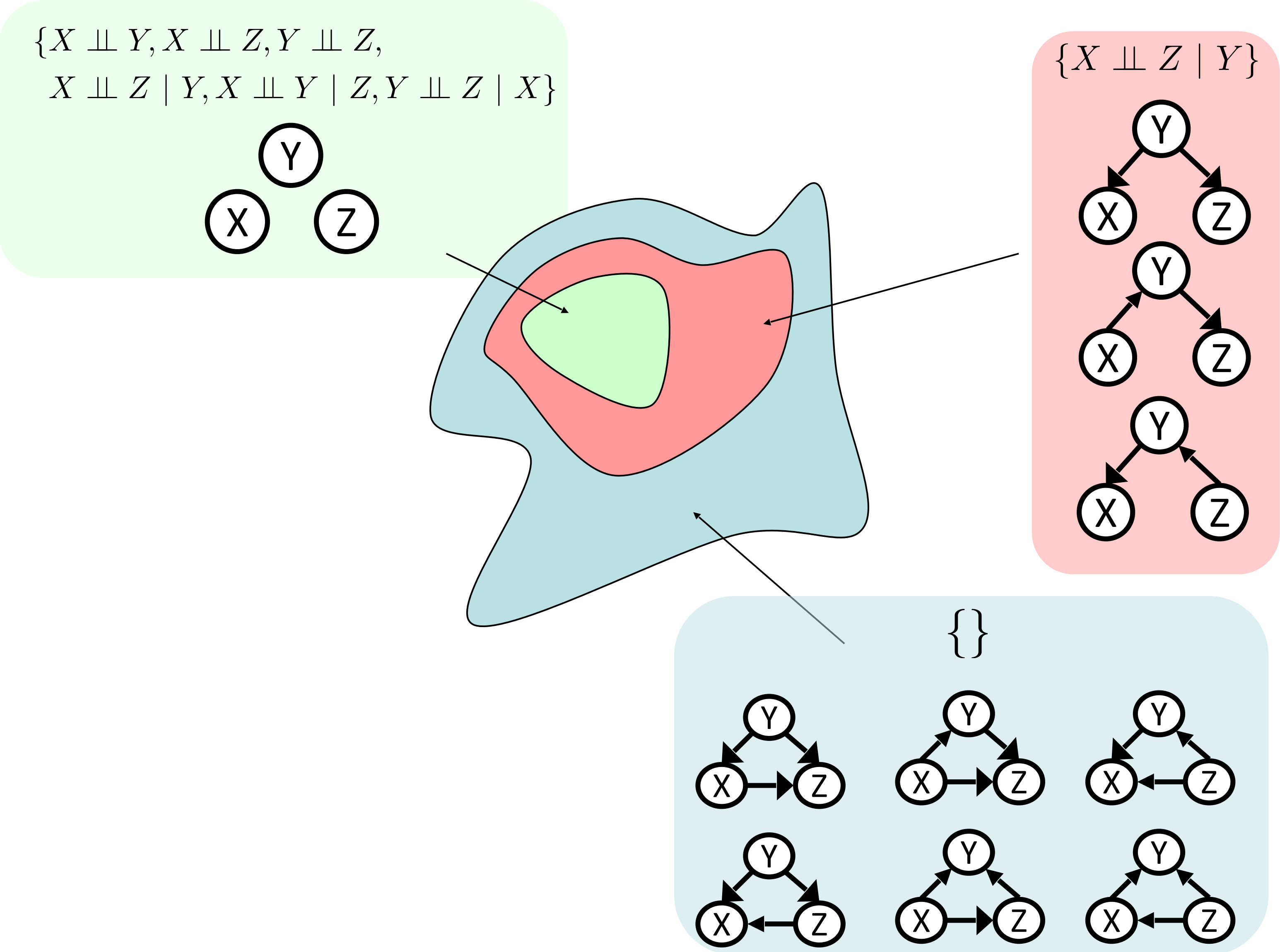
$X \perp\!\!\!\perp Z$



None!

Topology Limits Distributions

- Given some graph topology G, only certain joint distributions can be encoded
- The graph structure guarantees certain (conditional) independences
- (There might be more independence)
- Adding arcs increases the set of distributions, but has several costs
- Full conditioning can encode any distribution



Bayes Nets Representation Summary

- Bayes nets compactly encode joint distributions
- Guaranteed independencies of distributions can be deduced from BN graph structure
- D-separation gives precise conditional independence guarantees from graph alone
- A Bayes net's joint distribution may have further (conditional) independence that is not detectable until you inspect its specific distribution

Inference

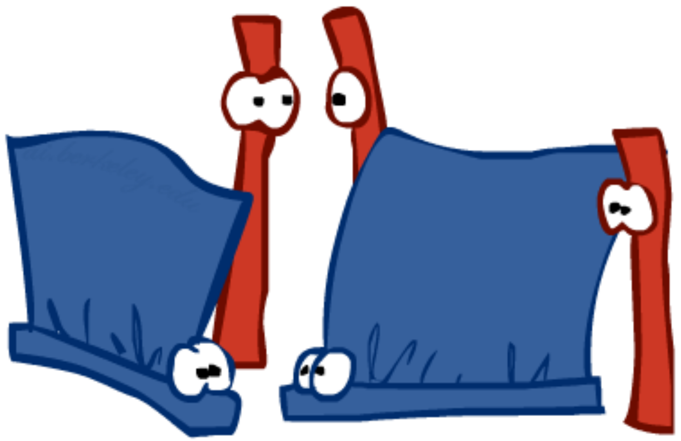
- Inference: calculating some useful quantity from a joint probability distribution

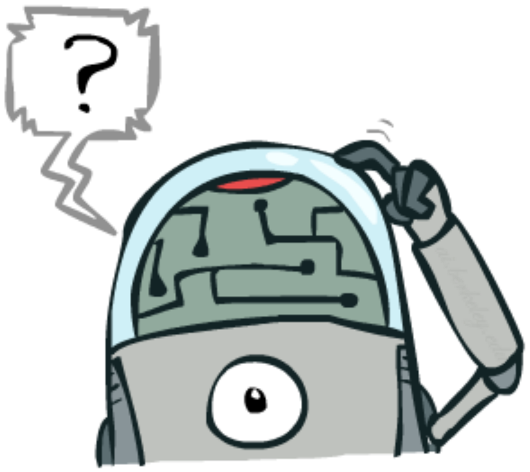
- Examples:
  - Posterior probability

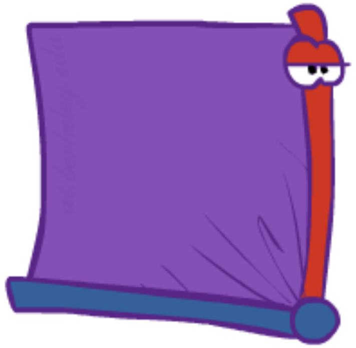
    $$P(Q|E_1 = e_1, \ldots E_k = e_k)$$

  - Most likely explanation:

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







Inference by Enumeration

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

$$\left.\begin{array}{l} E_1 \dots E_k = e_1 \dots e_k \\ Q \\ H_1 \dots H_r \end{array}\right\} X_1, X_2, \dots X_n$$

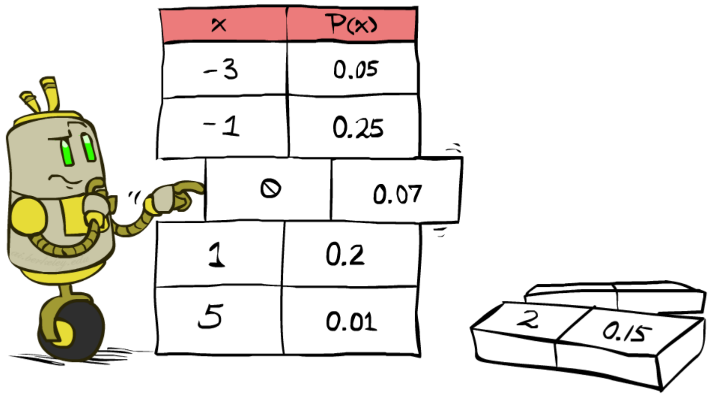
All variables

- We want:

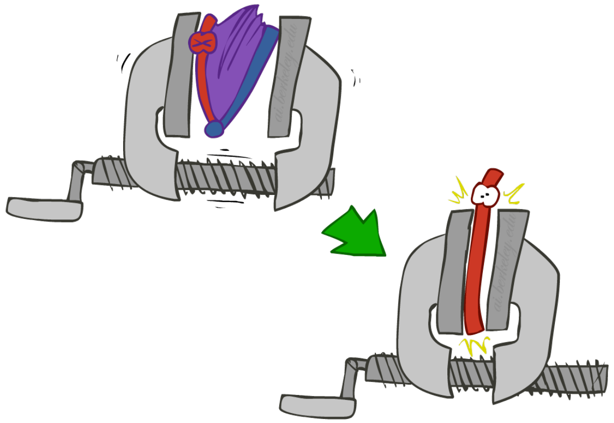
** Works fine with multiple query variables, too*

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

- Step 1: Select the entries consistent with the evidence



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



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

- Step 3: Normalize

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

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

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

Inference by Enumeration

- P(W)?

p(W=sun) = 0.3 + 0.1 + 0.1 + 0.15 = 0.65
p(W=rain) = 0.05 + 0.05 + 0.05 + 0.2 = 0.35

- P(W | winter)?

p(W=sun , winter) = 0.1 + 0.15 = 0.25
p(W=rain , winter) = 0.05 + 0.2 = 0.25
p(W=sun | winter) = 0.25 / (0.25 + 0.25) = 0.5
p(W=rain | winter) = 0.25 / (0.25 + 0.25) = 0.5

- P(W | winter, hot)?

p(W=sun , winter, hot) = 0.1
p(W=rain , winter, hot) = 0.05
p(W=sun | winter, hot) = 0.1 / (0.1 + 0.05) = 2/3
p(W=rain | winter, hot) = 0.05 / (0.1 + 0.05) = 1/3

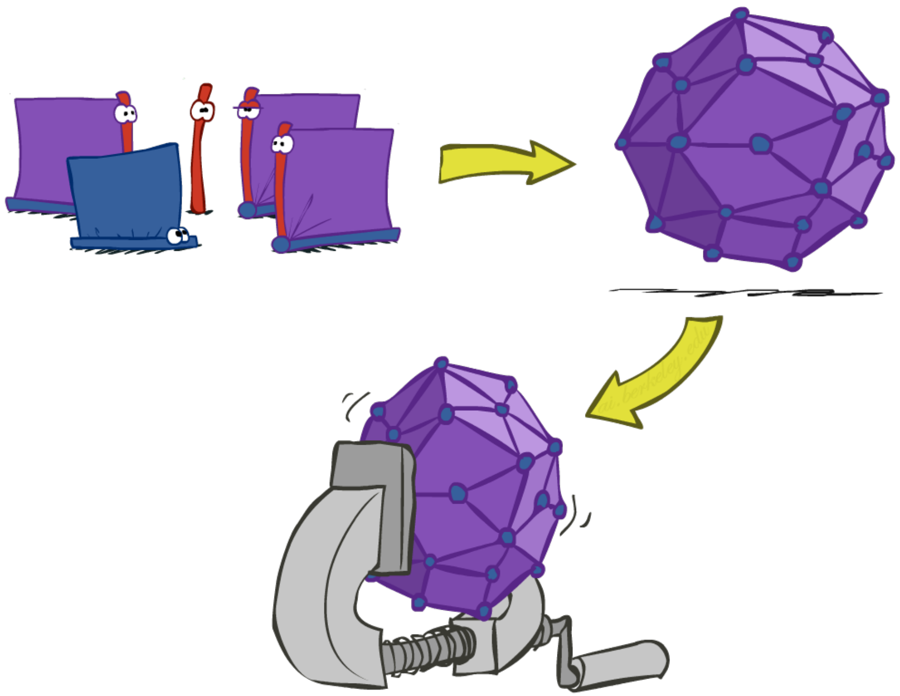
S	T	W	P
summer	hot	sun	0.30
summer	hot	rain	0.05
summer	cold	sun	0.10
summer	cold	rain	0.05
winter	hot	sun	0.10
winter	hot	rain	0.05
winter	cold	sun	0.15
winter	cold	rain	0.20

Inference by Enumeration

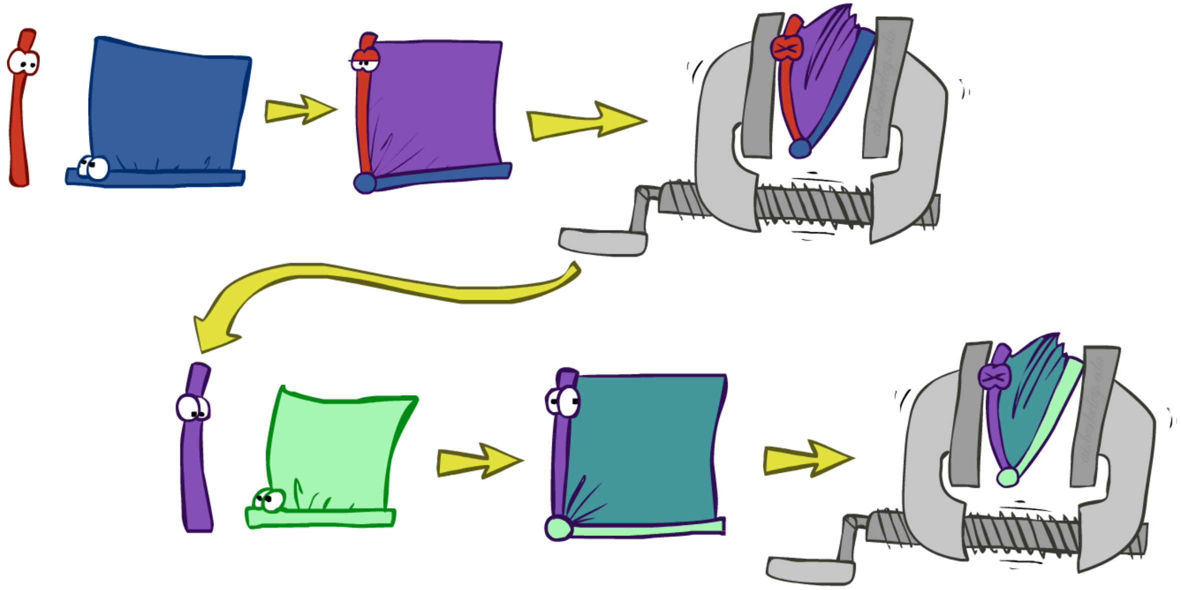
- Obvious problems:
  - Worst-case time complexity $O(d^n)$
  - Space complexity $O(d^n)$ to store the joint distribution
  - What about continuous distributions?

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

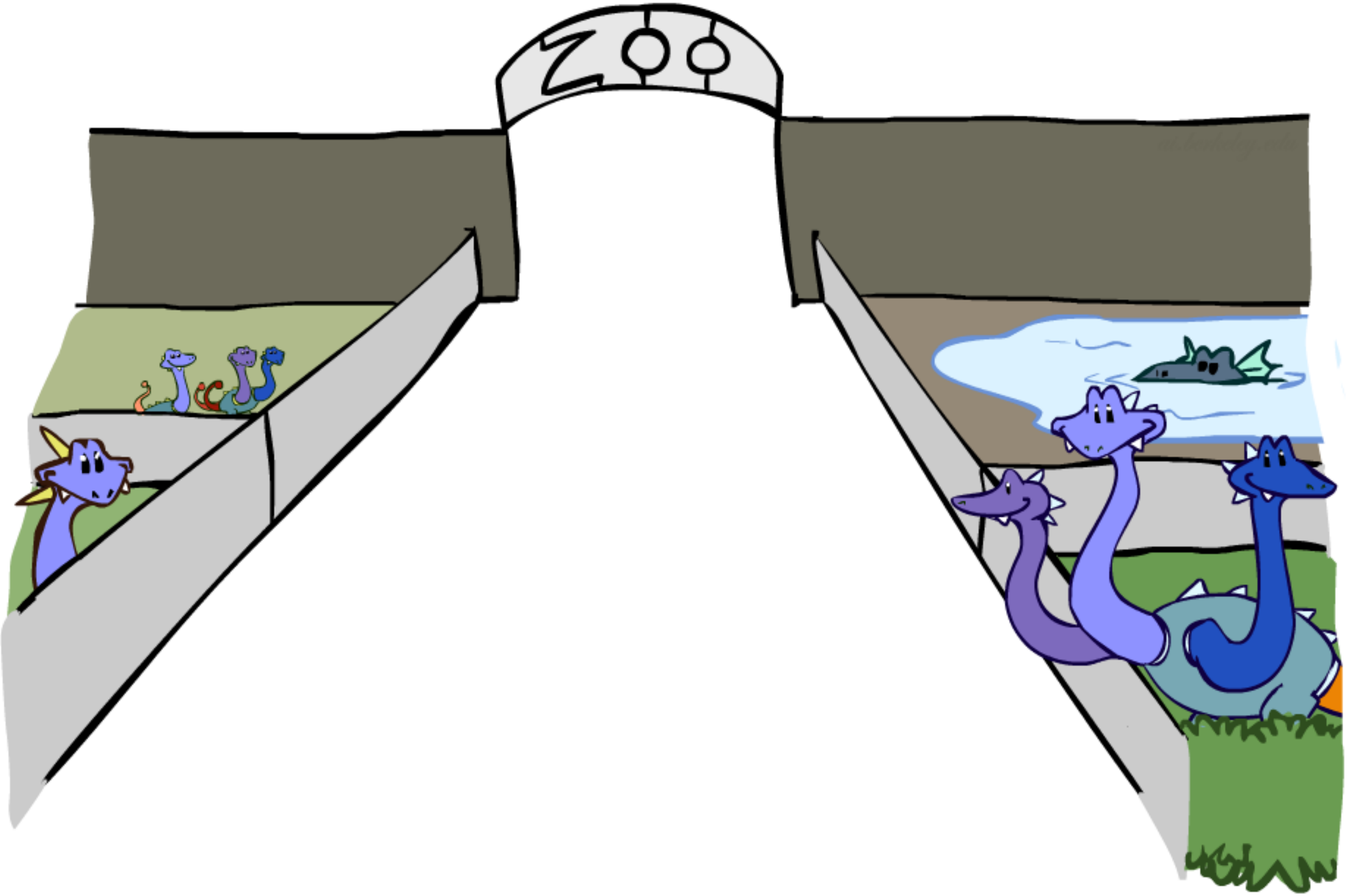


- Idea: interleave joining and marginalizing!
  - Called "Variable Elimination"
  - Still NP-hard, but usually much faster than inference by enumeration



  - First we'll need some new notation: factors

Factor Zoo



Factor Zoo I

- Joint distribution: P(X,Y)
  - Entries P(x,y) for all x, y
  - Sums to 1

$$P(T, W)$$

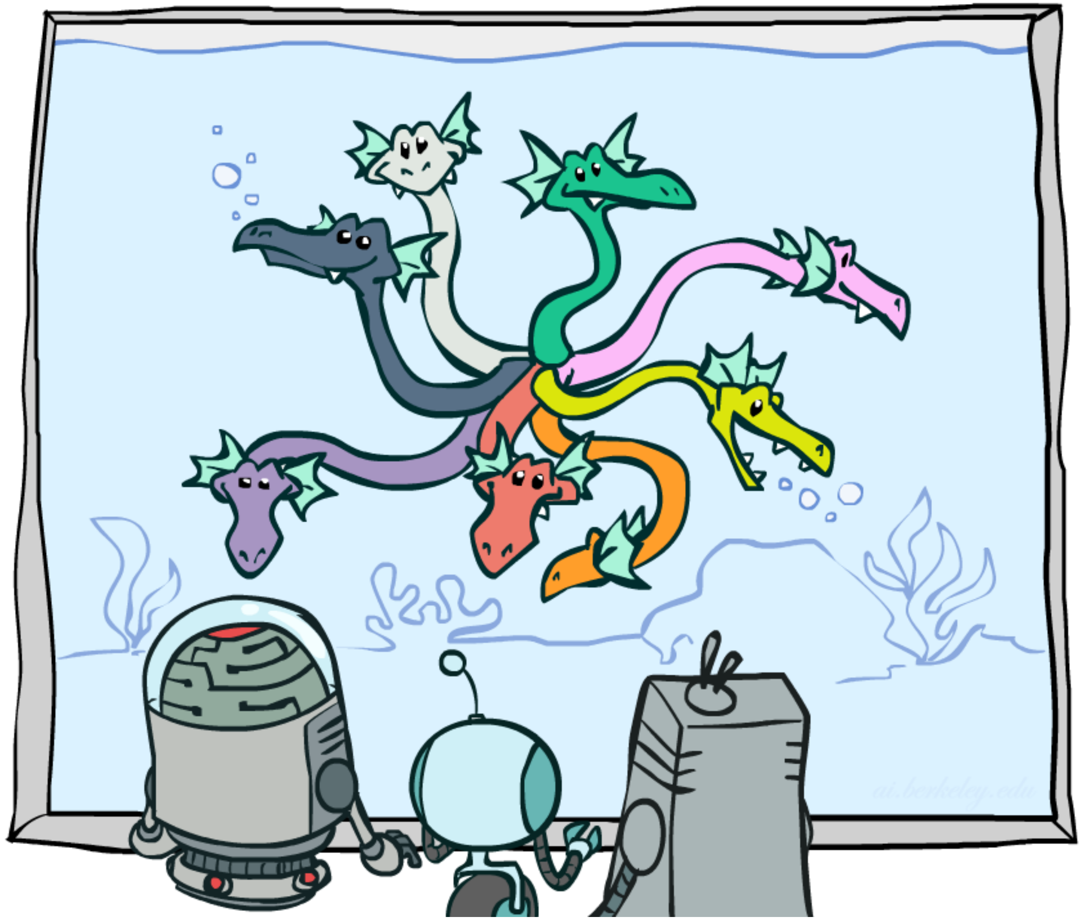
T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

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

$$P(cold, W)$$

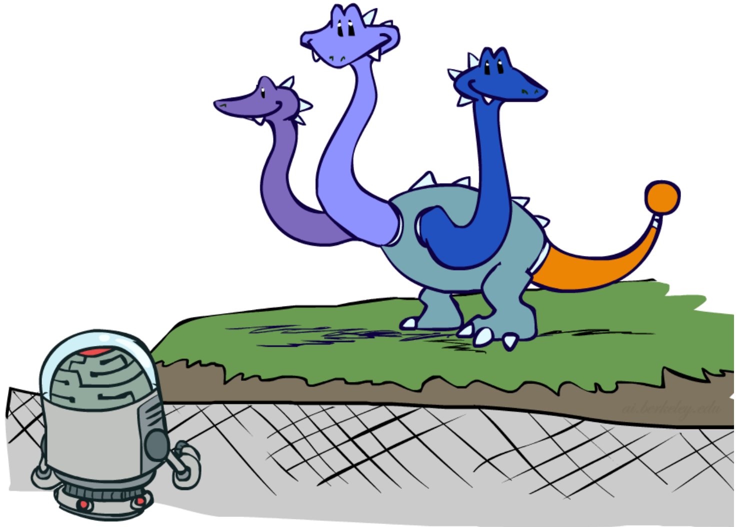
T	W	P
cold	sun	0.2
cold	rain	0.3

- Number of capitals = dimensionality of the table



Factor Zoo II

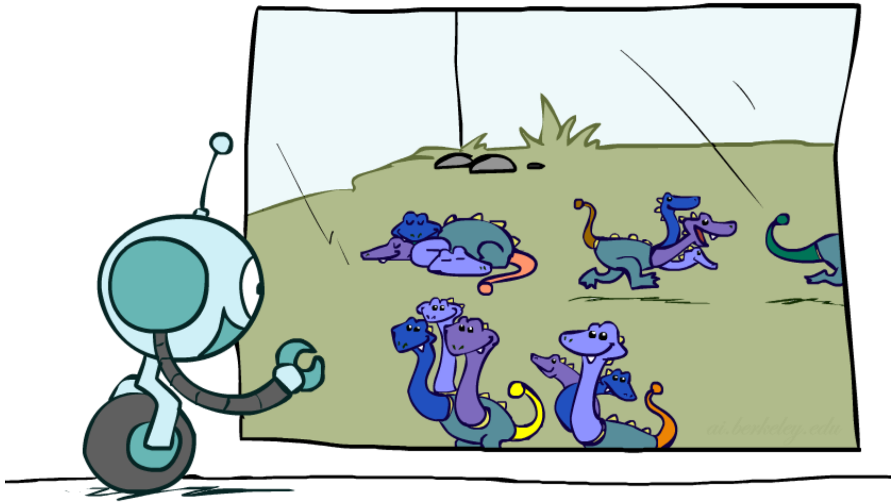
- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all y
  - Sums to 1



$P(W|cold)$

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

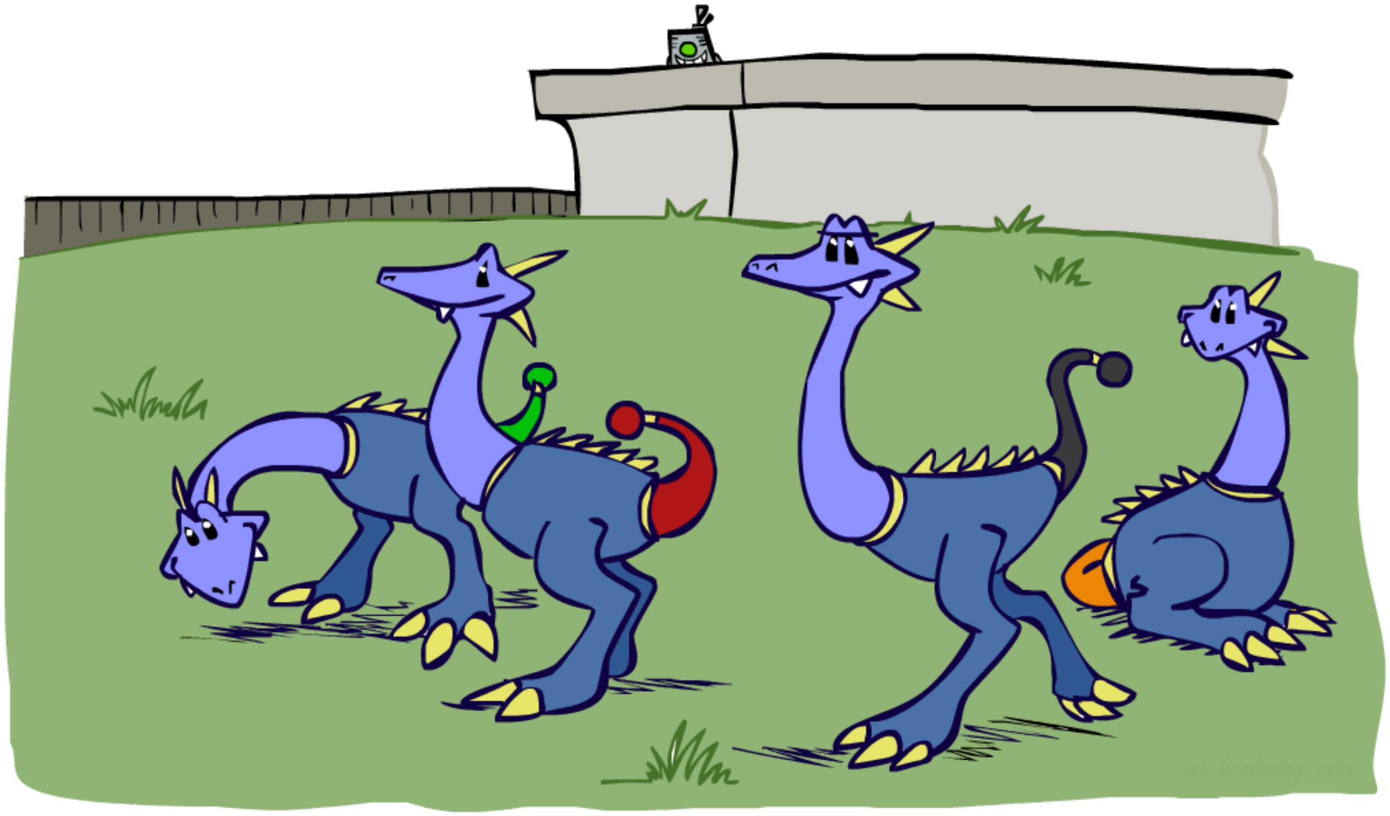
- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!

$P(rain|T)$

T	W	P
hot	rain	0.2
cold	rain	0.6

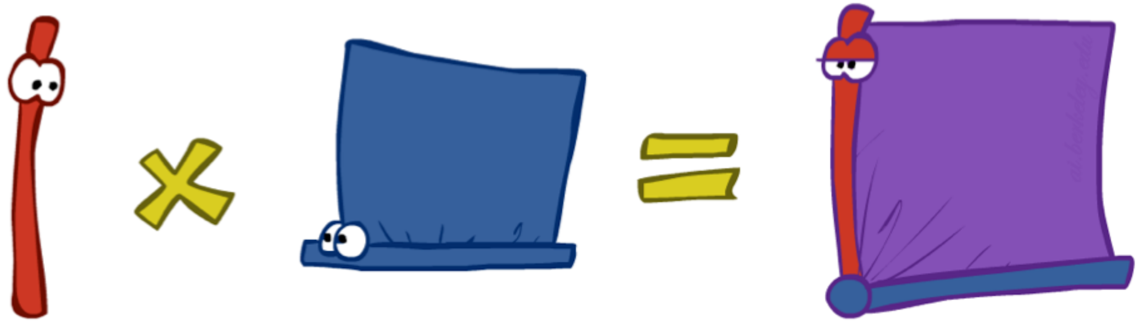
$P(rain|hot)$

$P(rain|cold)$

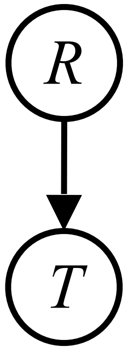


Operation 1: Join Factors

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

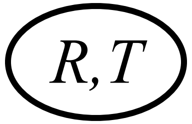
$\times$

$P(T|R)$

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

$P(R,T)$

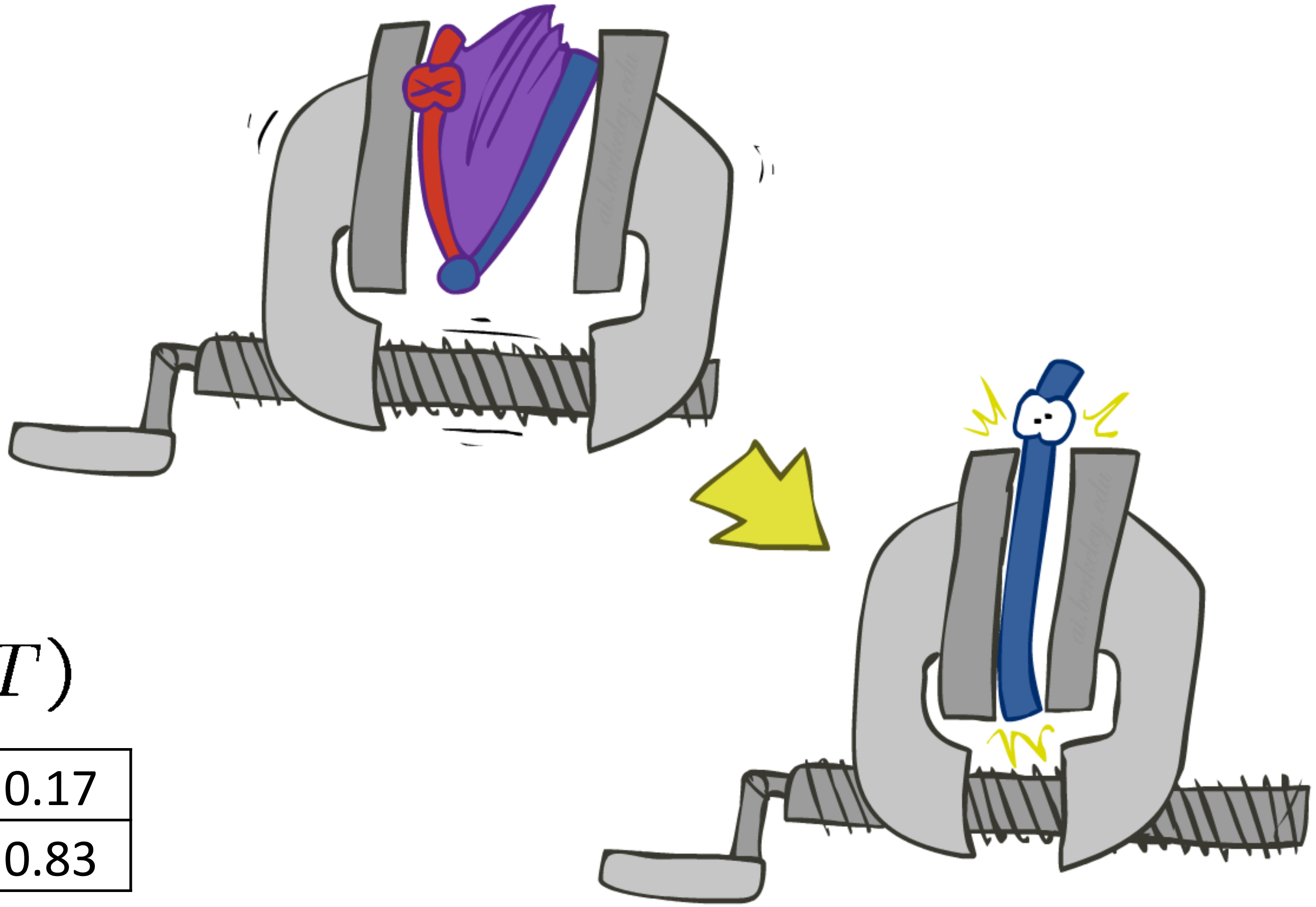
+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



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

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 $\rightarrow$

$P(T)$

+t	0.17
-t	0.83

Inference by Enumeration: Procedural Outline

- Track objects called factors
- Initial factors are local CPTs (one per node)

$P(R)$

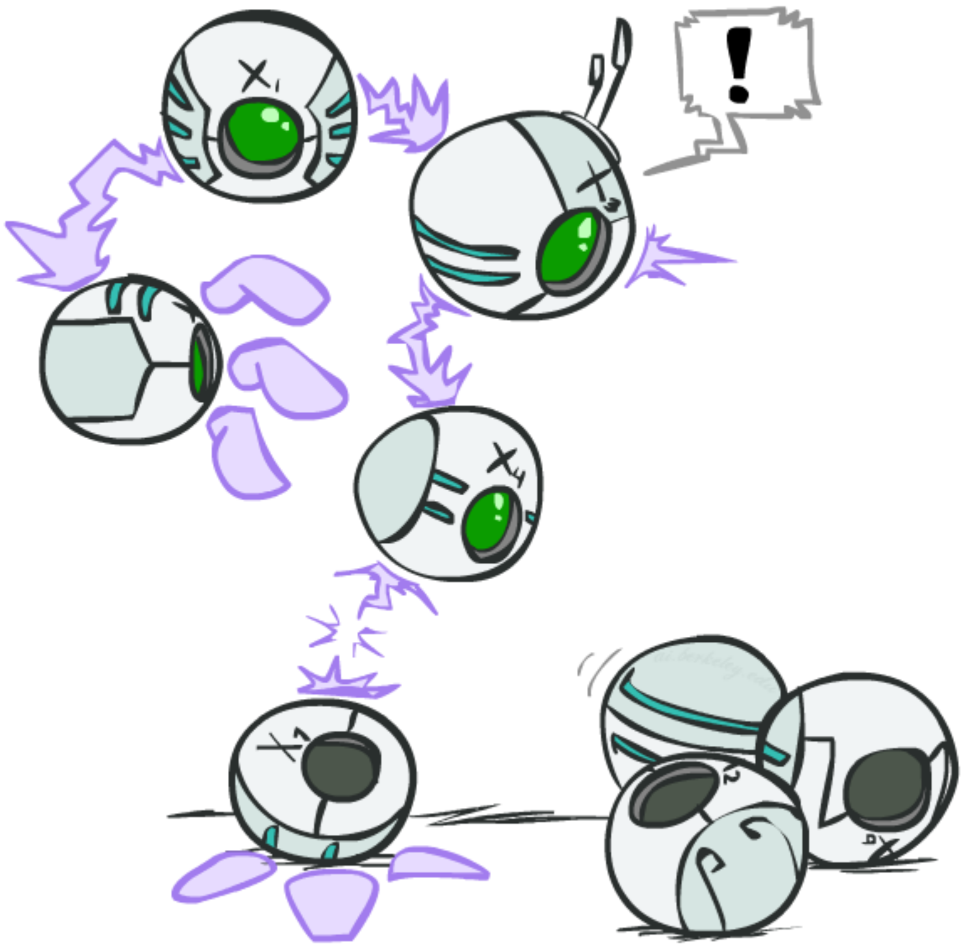
+r	0.1
-r	0.9

$P(T|R)$

+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 $L = +\ell$ then the initial factors are:

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

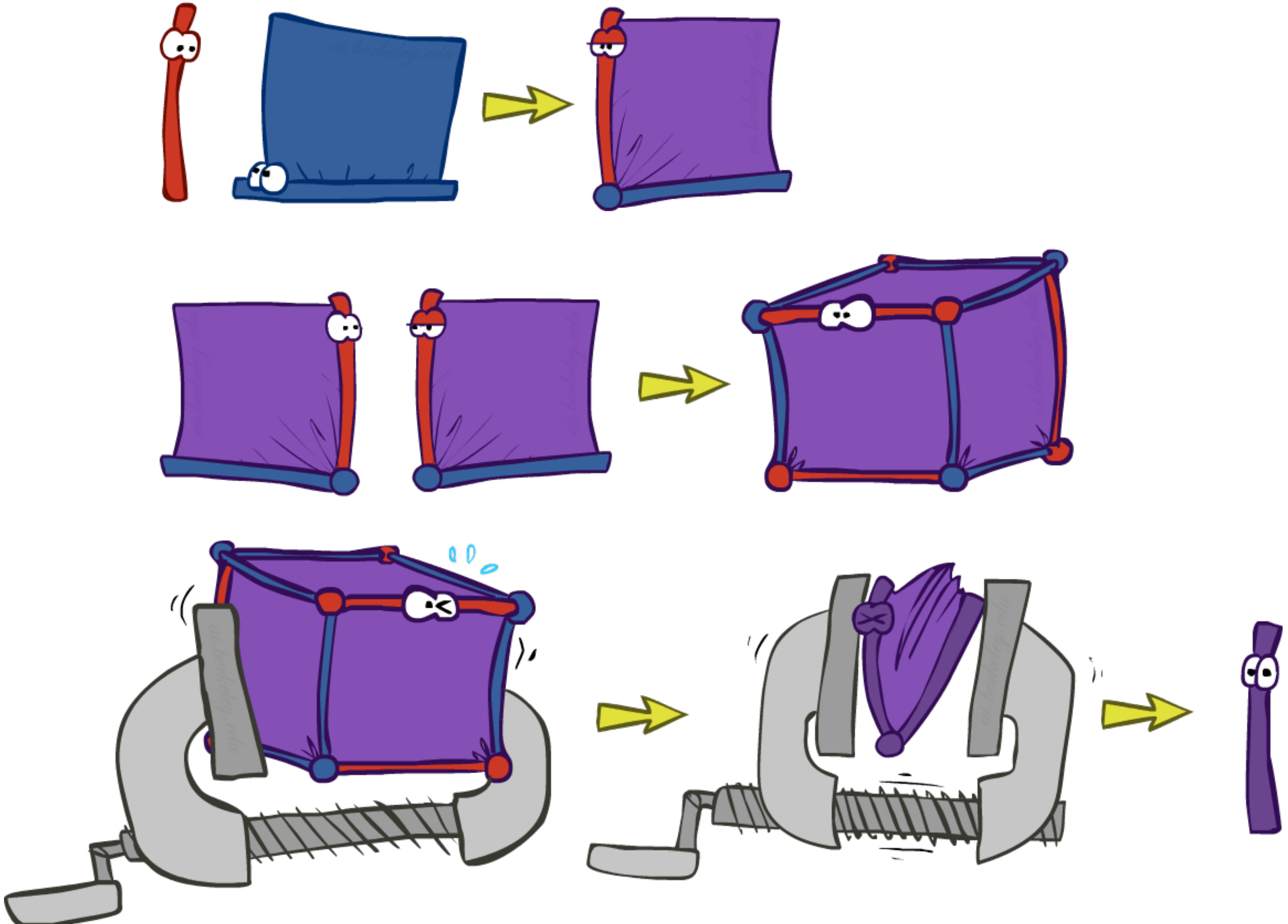
+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(+\ell|T)$

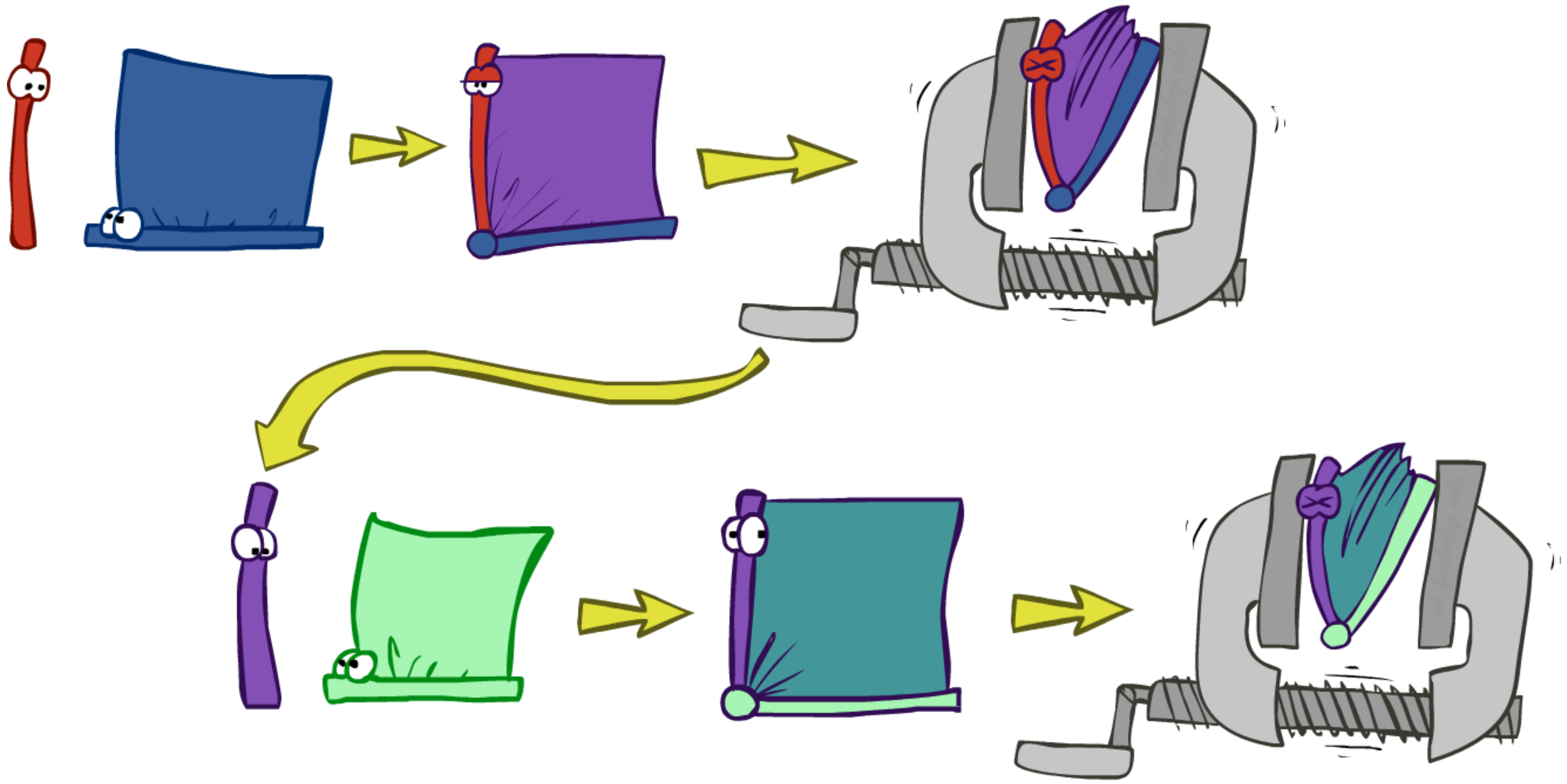
+t	+l	0.3
-t	+l	0.1

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

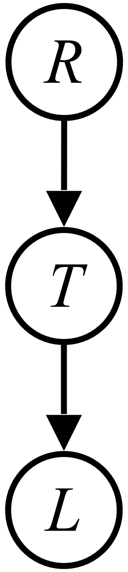
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)



Marginalizing Early (= Variable Elimination)



Traffic Domain



$$P(L) = \,?$$

- Inference by Enumeration

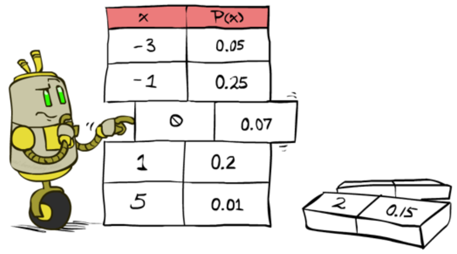
$$= \underbrace{\sum_t \underbrace{\sum_r \underbrace{P(L|t)\underbrace{P(r)P(t|r)}_{\text{Join on r}}}_{\text{Join on t}}}_{\text{Eliminate r}}}_{\text{Eliminate t}}$$

- Variable Elimination

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

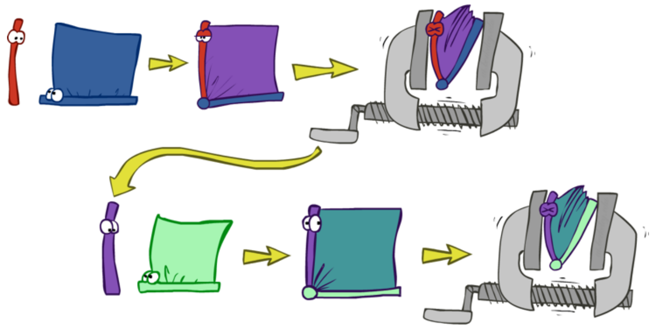
General Variable Elimination

- 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

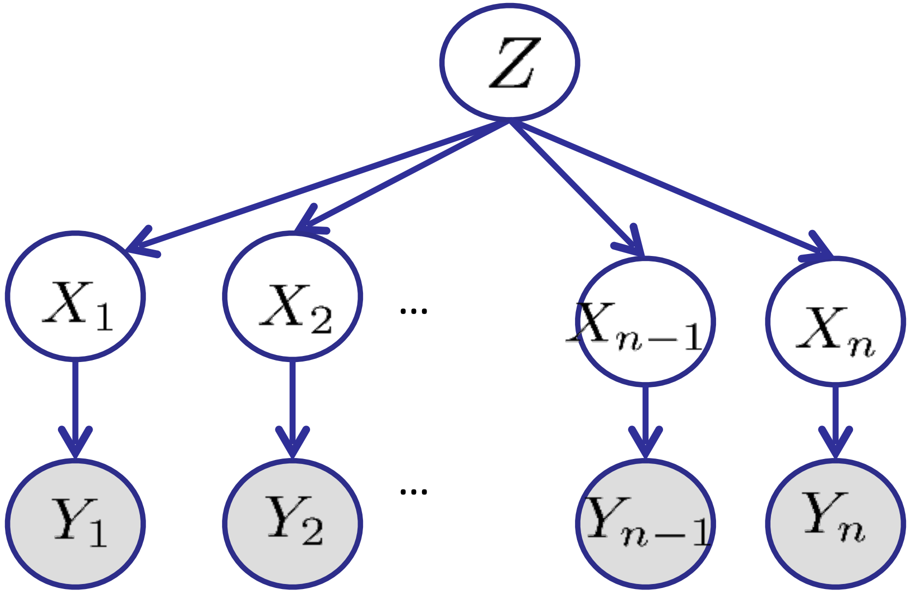


- Join all remaining factors and normalize

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

Variable Elimination Ordering

- For the query $P(X_n \mid y_1,\ldots,y_n)$ work through the following two different orderings: $Z, X_1, \ldots, X_{n-1}$ and $X_1, \ldots, X_{n-1}, Z$. 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

- All we are doing is changing the ordering of the variables that are eliminated…
- …but it can (sometimes) reduce storage and complexity to linear w.r.t. number of variables!
- 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:

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

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

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

$$\ldots$$

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

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

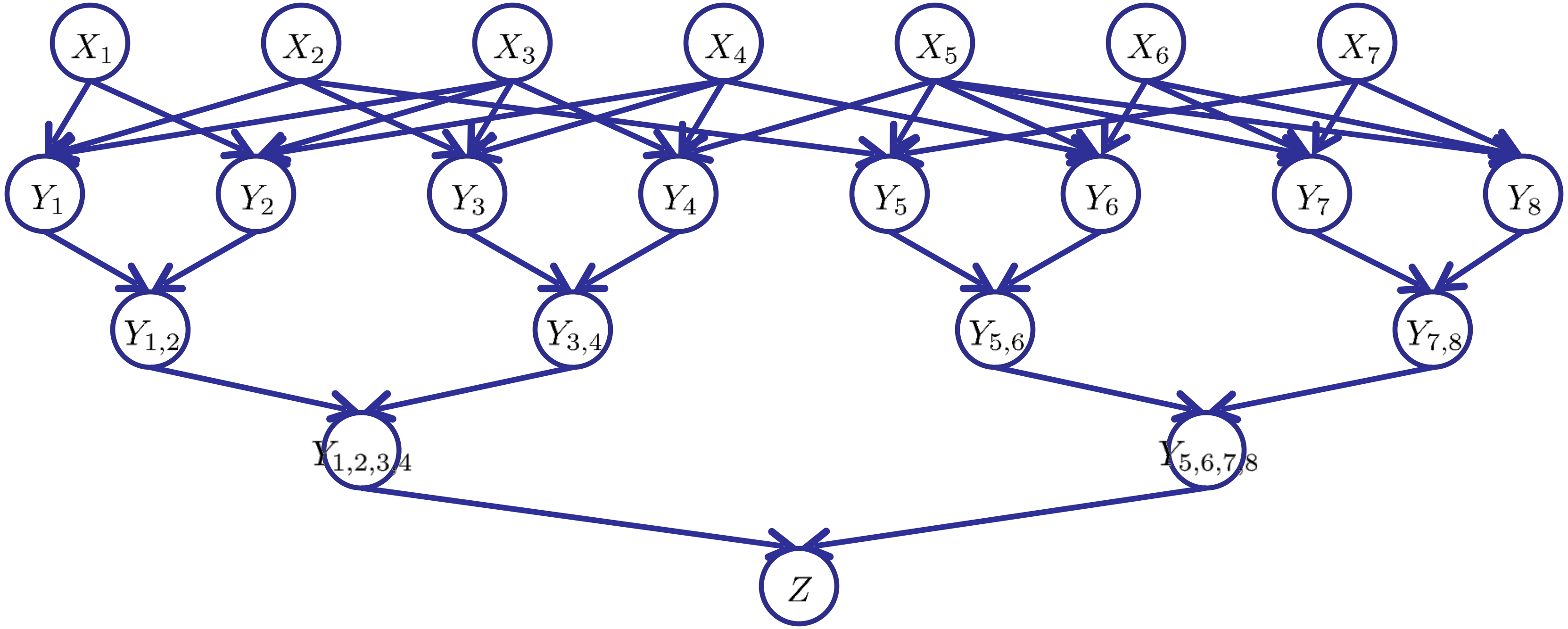
$$\ldots$$

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

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

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

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



- 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.

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