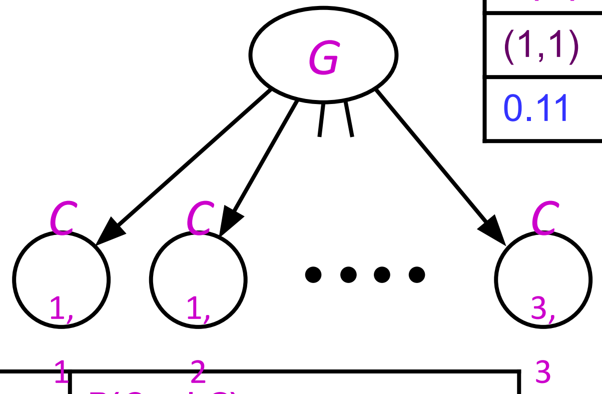
Bayes Net Syntax and Semantics

Bayes Net Syntax

- A set of nodes, one per variable X_i
- A directed, acyclic graph
- A conditional distribution for each node given its ***parent variables*** in the graph
  - ***CPT*** (conditional probability table); each row is a distribution for child given values of its parents



P(*G*)			
(1,1)	(1,2)	(1,3)	…
0.11	0.11	0.11	…

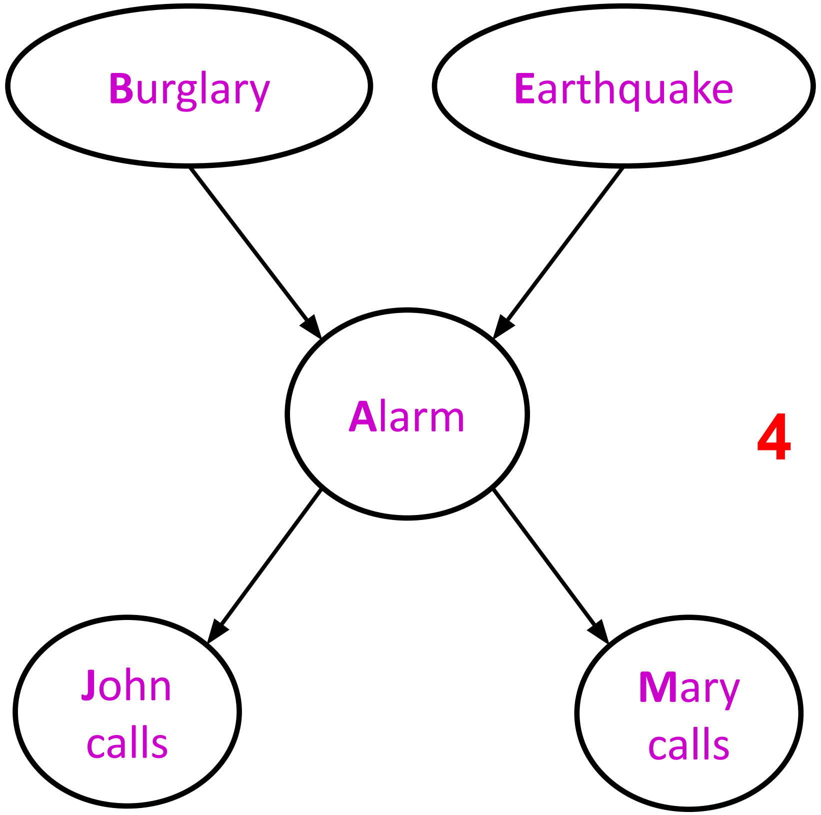
G	**P($C_{1,1}$ | *G*)**			
	g	y	o	r
(1,1)	0.01	0.1	0.3	0.59
(1,2)	0.1	0.3	0.5	0.1
(1,3)	0.3	0.5	0.19	0.01
…				

Bayes net = Topology (graph) + Local Conditional Probabilities

Example: Alarm Network

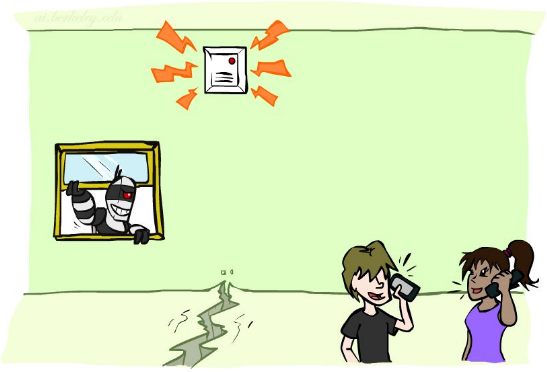
P(B)	
true	false
0.001	0.999

1



1

P(E)	
true	false
0.002	0.998



4

B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

A	P(J|A)	
	true	false
true	0.9	0.1
false	0.05	0.95

2

2

A	P(M|A)	
	true	false
true	0.7	0.3
false	0.01	0.99

Number of ***free parameters*** in each CPT:

Parent range sizes $d_1,\dots,d_k$

Child range size d
Each table row must sum to 1

$(d-1)\ \Pi_i\ d_i$

General formula for sparse BNs

- Suppose
  - n variables
  - Maximum range size is d
  - Maximum number of parents is k
- Full joint distribution has size $O(d^n)$
- Bayes net has size $O(n \cdot d^k)$
  - Linear scaling with n as long as causal structure is local

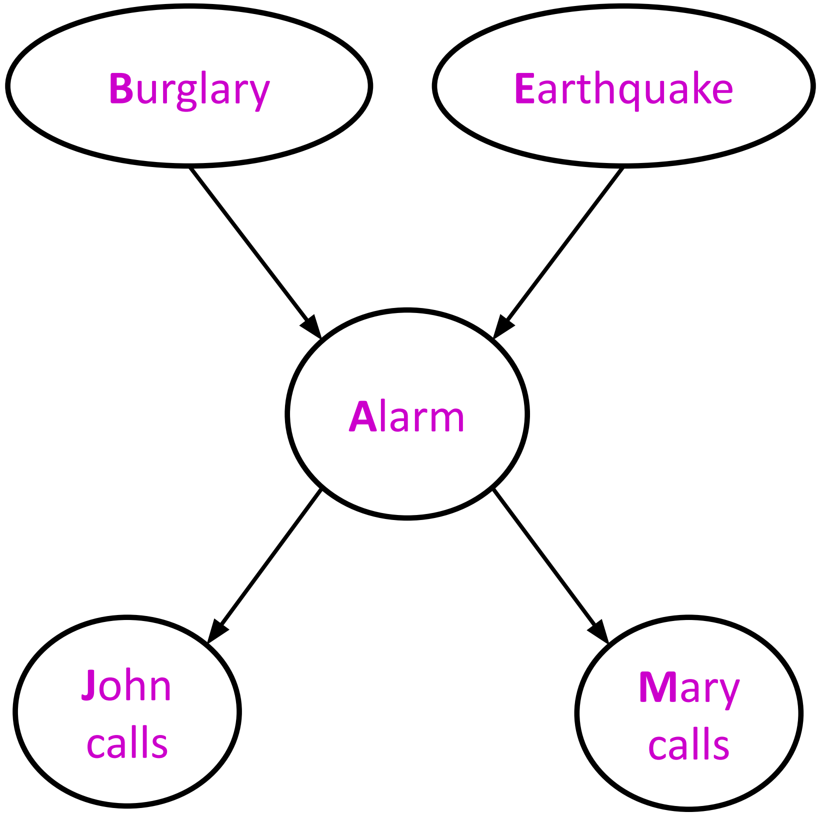
Bayes net global semantics

- Bayes nets encode joint distributions as product of conditional distributions on each variable:

$$P(X_1,..,X_n) = \prod_i P(X_i \mid Parents(X_i))$$

Example

P(B)	
true	false
0.001	0.999



P(E)	
true	false
0.002	0.998

$P(b, \neg e, a, \neg j, \neg m) =$

$P(b)\ P(\neg e)\ P(a|b,\neg e)\ P(\neg j|a)\ P(\neg m|a)$

$= .001 \times .998 \times .94 \times .1 \times .3 = .000028$

B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

A	P(J|A)	
	true	false
true	0.9	0.1
false	0.05	0.95

A	P(M|A)	
	true	false
true	0.7	0.3
false	0.01	0.99

Conditional independence in BNs

- Compare the Bayes net global semantics

$$P(X_1,..,X_n) = \prod_i P(X_i \mid Parents(X_i))$$

  with the chain rule identity

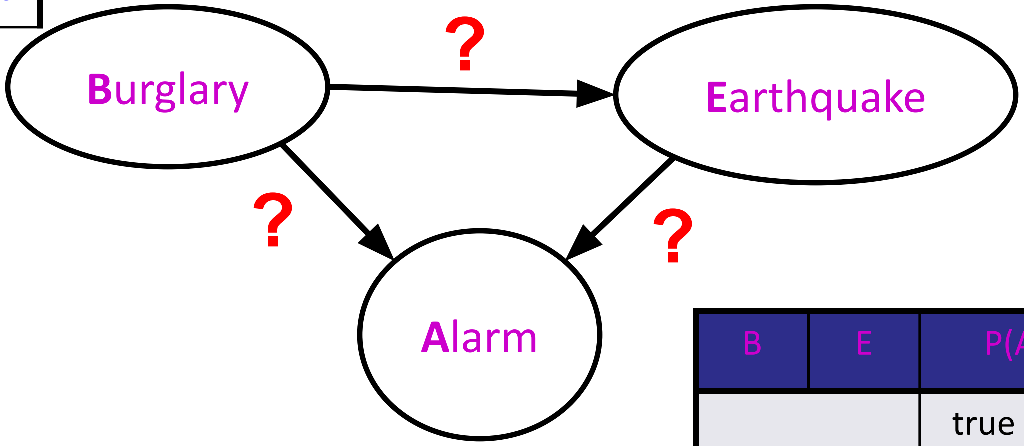
$$P(X_1,..,X_n) = \prod_i P(X_i \mid X_1,\ldots,X_{i-1})$$

- Assume (without loss of generality) that $X_1,..,X_n$ sorted in topological order according to the graph (i.e., parents before children), so $Parents(X_i) \subseteq X_1,\ldots,X_{i-1}$
- So the Bayes net asserts conditional independences $P(X_i \mid X_1,\ldots,X_{i-1}) = P(X_i \mid Parents(X_i))$
  - To ensure these are valid, choose parents for node X_i that "shield" it from other predecessors

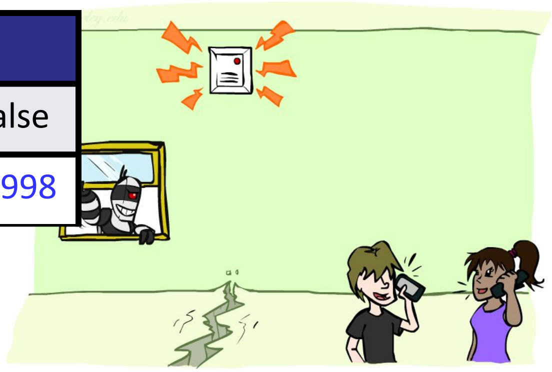
Example: Burglary

- Burglary
- Earthquake
- Alarm

P(B)	
true	false
0.001	0.999



P(E)	
true	false
0.002	0.998

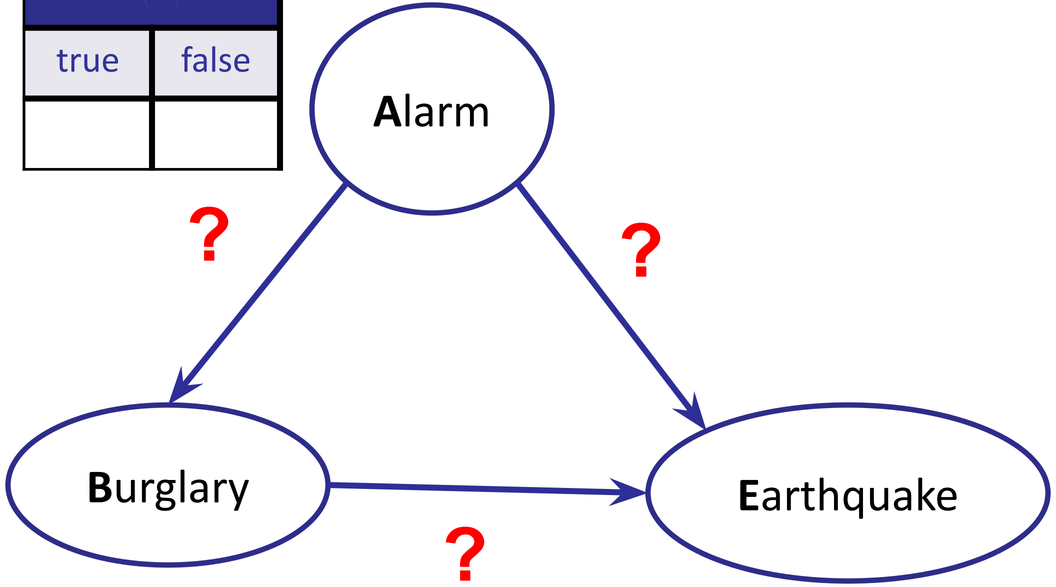


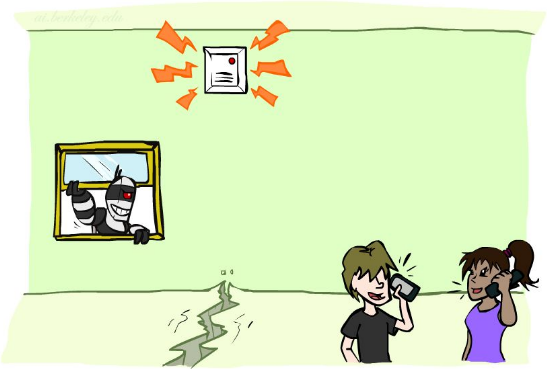
B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

Example: Burglary

- Alarm
- Burglary
- Earthquake

P(A)	
true	false



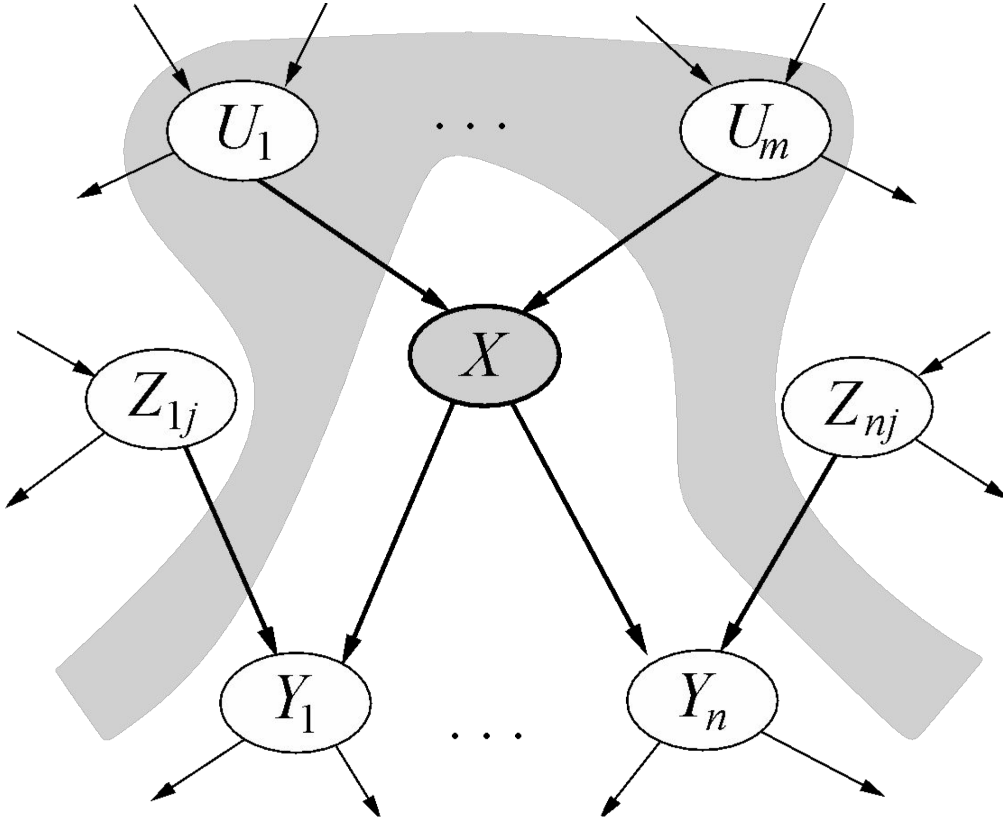


A	P(B|A)	
	true	false
true	?	
false		

A	B	P(E|A,B)	
		true	false
true	true	?	
true	false		
false	true		
false	false		

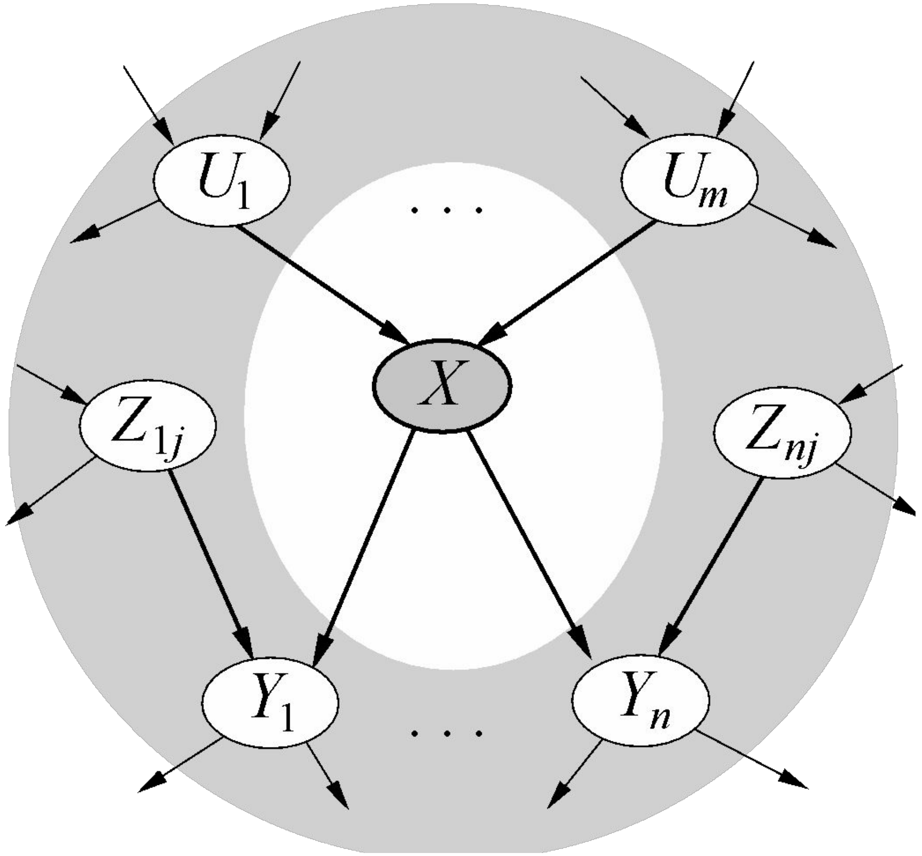
Conditional independence semantics

- ***Every variable is conditionally independent of its non-descendants given its parents***
- Conditional independence semantics <=> global semantics



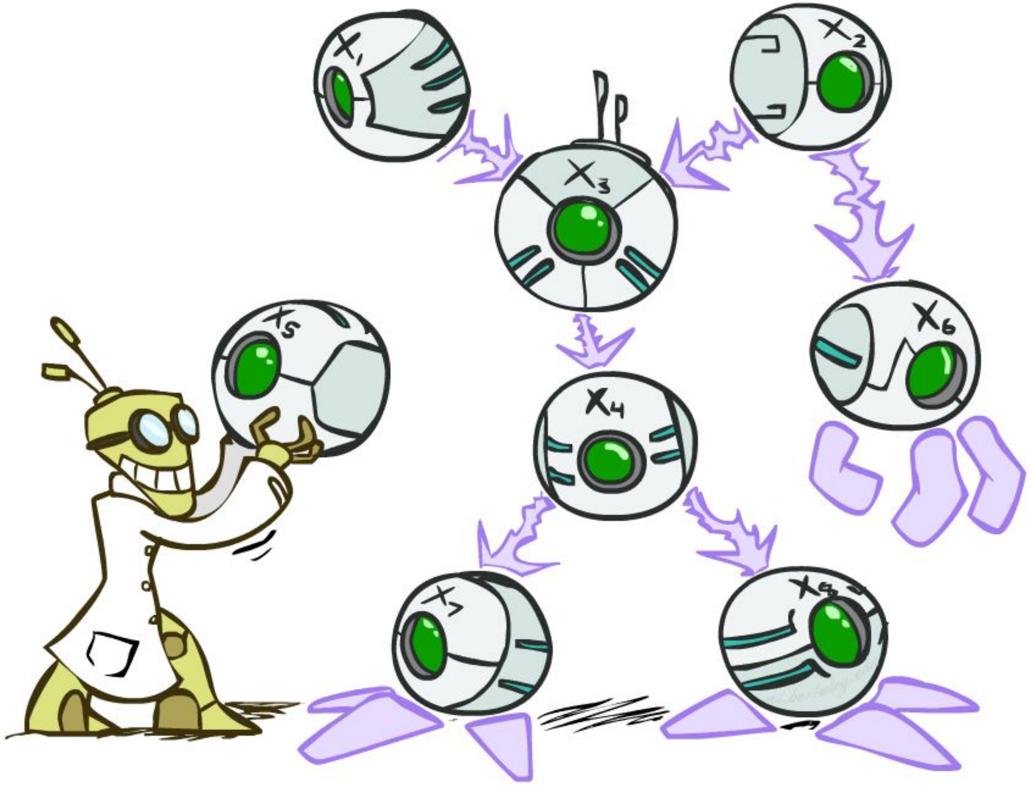
Markov blanket

- A variable's Markov blanket consists of parents, children, children's other parents
- ***Every variable is conditionally independent of all other variables given its Markov blanket***



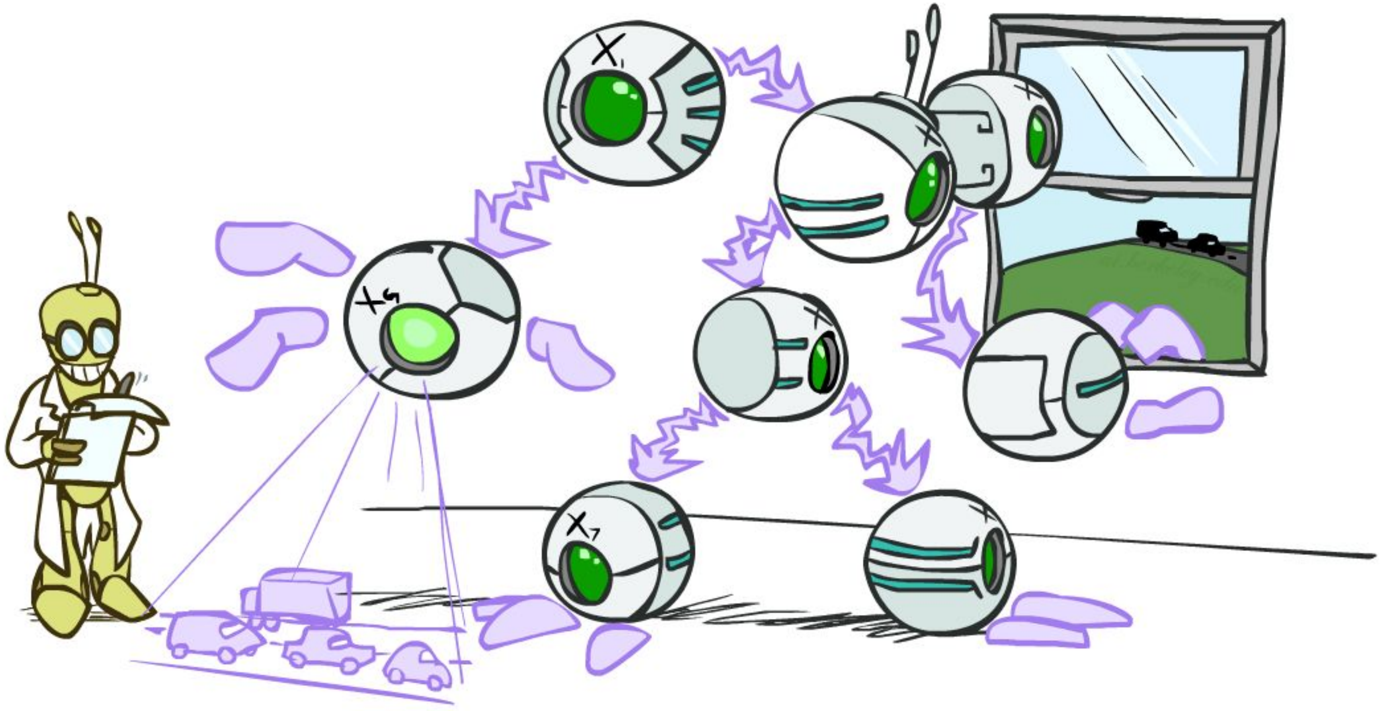
Summary

- Independence and conditional independence are important forms of probabilistic knowledge
- Bayes net encode joint distributions efficiently by taking advantage of conditional independence
  - Global joint probability = product of local conditionals
  - Local causality => exponential reduction in total size



CS 188: Artificial Intelligence

Bayes Nets: Exact Inference



Instructor: Stuart Russell and Dawn Song --- University of California, Berkeley

Bayes Nets

✔ Part I: Representation

Part II: Exact inference

- Enumeration (always exponential complexity)
- Variable elimination (worst-case exponential complexity, often better)
- Inference is NP-hard in general

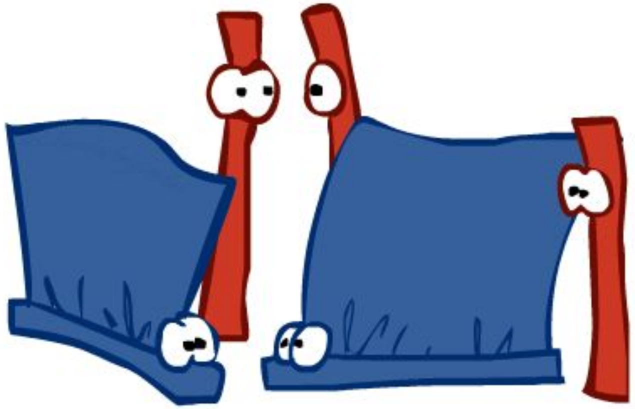
Part III: Approximate Inference

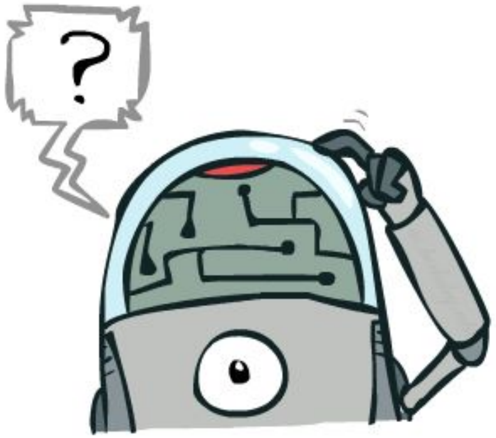
Later: Learning Bayes nets from data

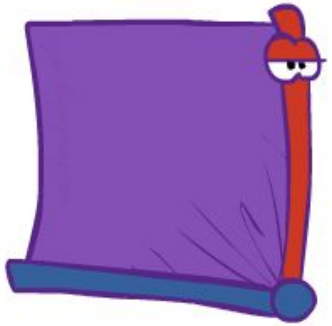
Inference

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

- Examples:
  - Posterior marginal probability
    - $P(Q|e_1,..,e_k)$
    - E.g., what disease might I have?
  - Most likely explanation:
    - $\text{argmax}_{q,r,s}\, P(Q=q,R=r,S=s|e_1,..,e_k)$
    - E.g., what did he say?

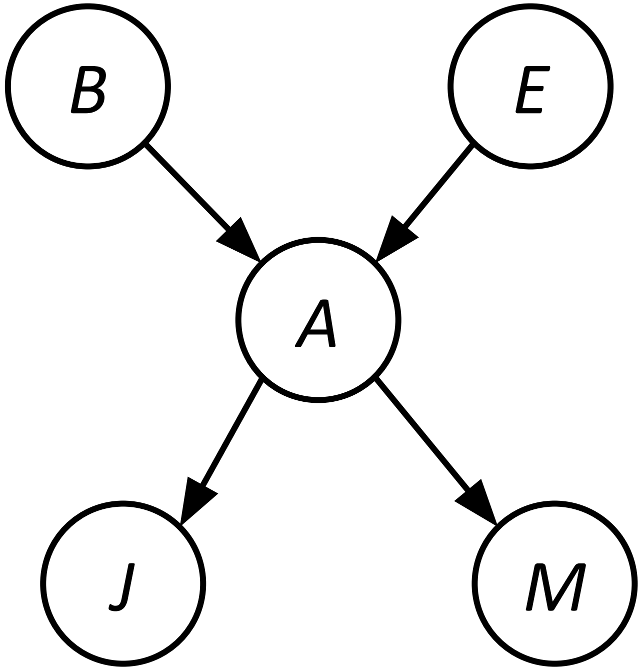


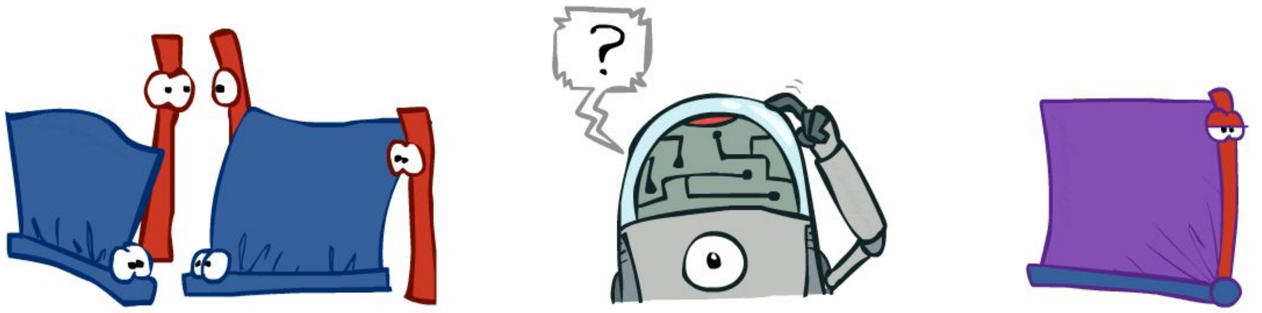




Inference by Enumeration in Bayes Net

- Reminder of inference by enumeration:
  - Any probability of interest can be computed by summing entries from the joint distribution: $P(\mathbf{Q} \mid \mathbf{e}) = \alpha \sum_{\mathbf{h}} P(\mathbf{Q}, \mathbf{h}, \mathbf{e})$
  - Entries from the joint distribution can be obtained from a BN by multiplying the corresponding conditional probabilities
- $P(B \mid j, m) = \alpha \sum_{e,a} P(B, e, a, j, m)$
- $= \alpha \sum_{e,a} P(B)\, P(e)\, P(a|B,e)\, P(j|a)\, P(m|a)$
- So inference in Bayes nets means computing sums of products of numbers: sounds easy!!
- Problem: sums of ***exponentially many*** products!





Can we do better?

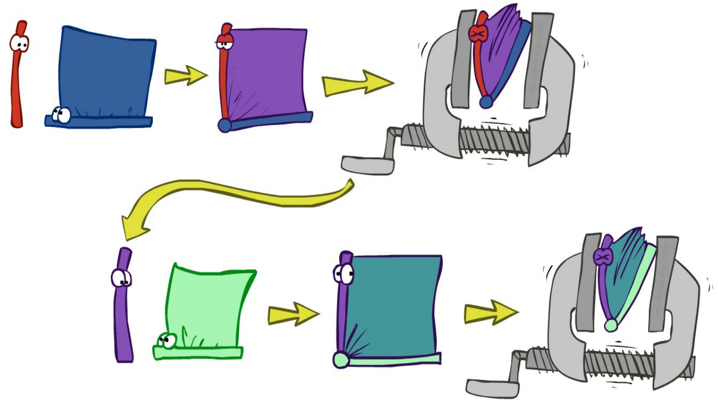
- Consider **uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz**
  - 16 multiplies, 7 adds
  - Lots of repeated subexpressions!
- Rewrite as **(u+v)(w+x)(y+z)**
  - 2 multiplies, 3 adds
- $\sum_{e,a} P(B)\ \mathrm{P}(\mathrm{e})\ \mathrm{P}(\mathrm{a}|B,e)\ P(j|a)\ P(m|a)$
- $= P(B)P(e)P(\mathrm{a}|B,e)P(j|a)P(m|a) + P(B)P(\neg e)P(\mathrm{a}|B,\neg e)P(j|a)P(m|a)$

  $+ P(B)P(e)P(\neg \mathrm{a}|B,e)P(j|\neg a)P(m|\neg a) + P(B)P(\neg e)P(\neg \mathrm{a}|B,\neg e)P(j|\neg a)P(m|\neg a)$

  Lots of repeated subexpressions!

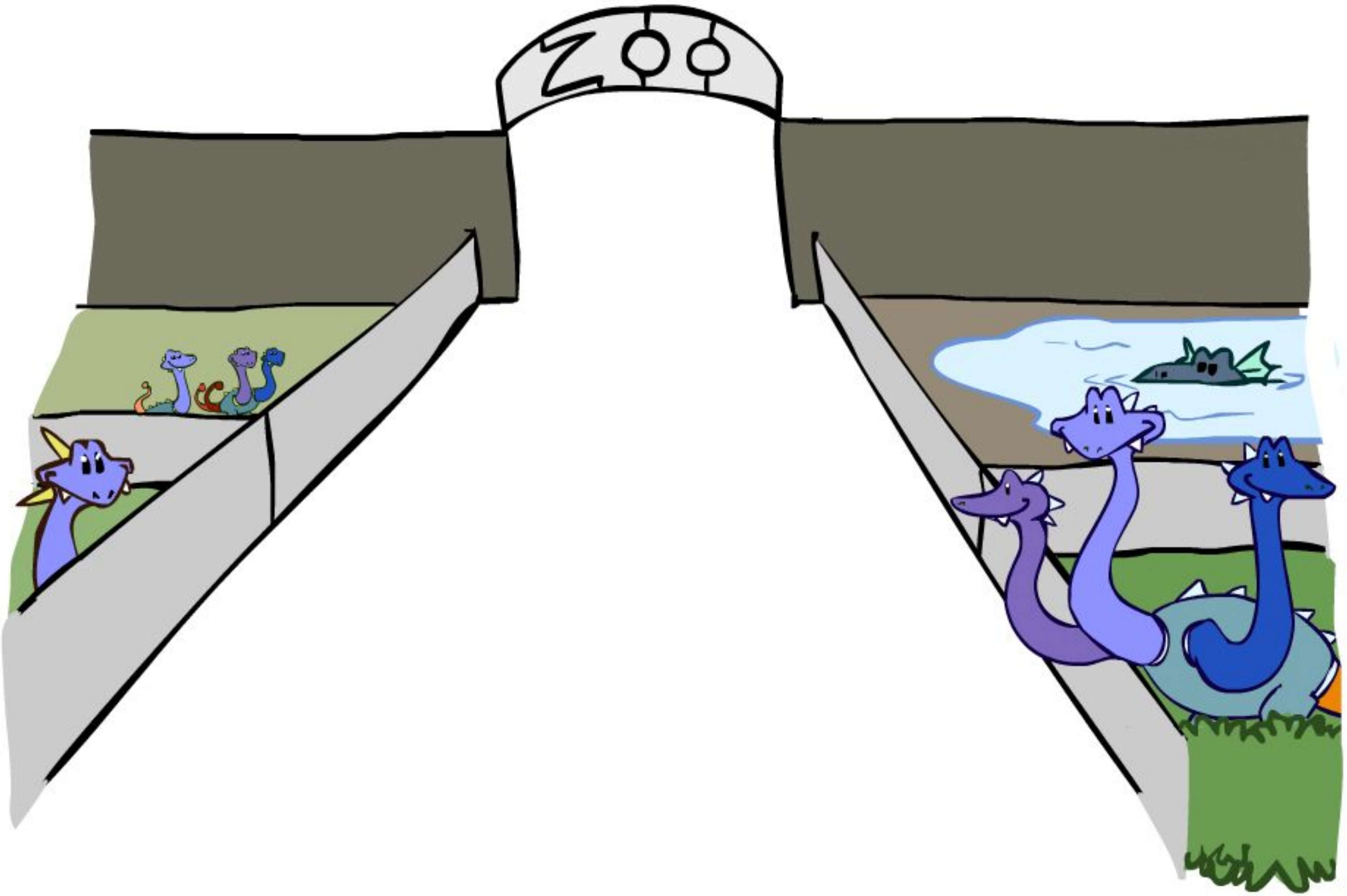
Variable elimination: The basic ideas

- Move summations inwards as far as possible
  - $P(B \mid j, m) = \alpha \sum_{e,a} P(B)\, P(e)\, P(a|B,e)\, P(j|a)\, P(m|a)$
  - $\qquad\qquad\qquad = \alpha\, P(B) \sum_{e} P(e) \sum_{a} P(a|B,e)\, P(j|a)\, P(m|a)$



- Do the calculation from the inside out
  - I.e., sum over *a* first, then sum over *e*
  - Problem: $P(a|B,e)$ isn't a single number, it's a bunch of different numbers depending on the values of *B* and *e*
  - Solution: use arrays of numbers (of various dimensions) with appropriate operations on them; these are called ***factors***

Factor Zoo

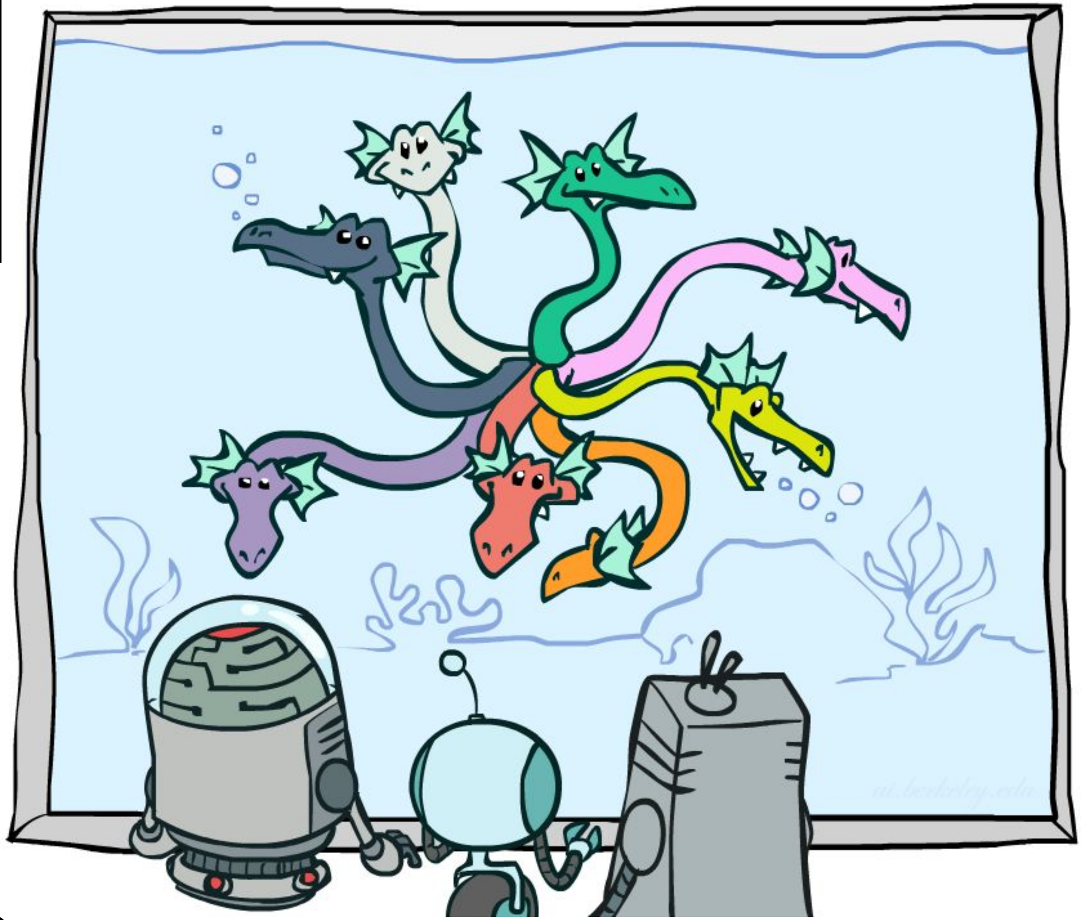


Factor Zoo I

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

P(A,J)

A \ J	true	false
true	0.09	0.01
false	0.045	0.855



- Projected joint: P(x,Y)
  - A slice of the joint distribution
  - Entries P(x,y) for one x, all y
  - |Y|-element vector
  - Sums to P(x)

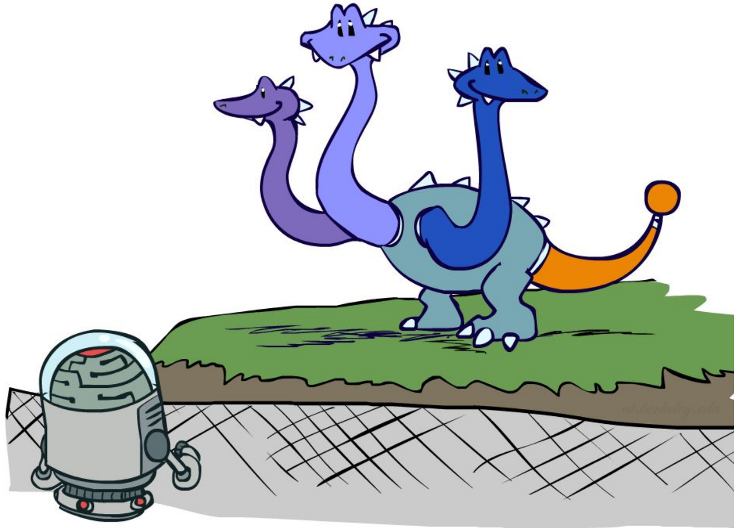
P(a,J)

A \ J	true	false
true	0.09	0.01

Number of variables (capitals) = dimensionality of the table

Factor Zoo II

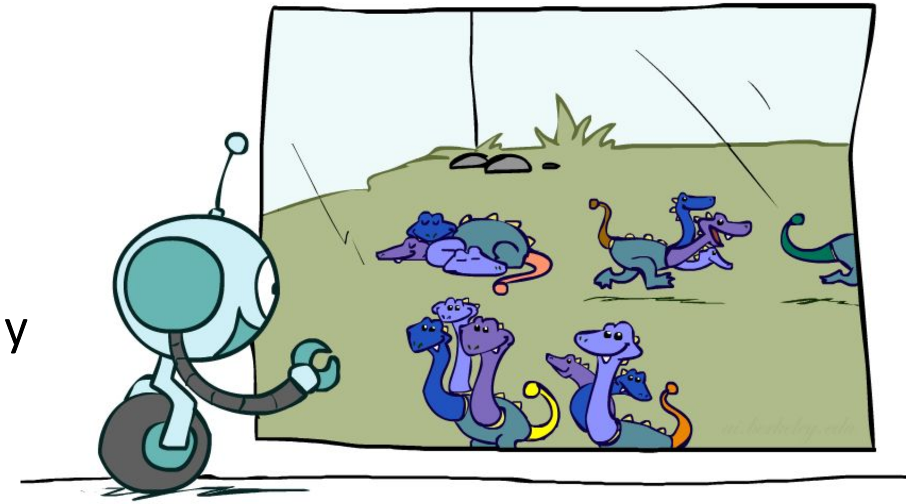
- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all y
  - Sums to 1



$P(J|a)$

A \ J	true	false
true	0.9	0.1

- Family of conditionals:
  P(X |Y)
  - Multiple conditionals
  - Entries P(x | y) for all x, y
  - Sums to |Y|

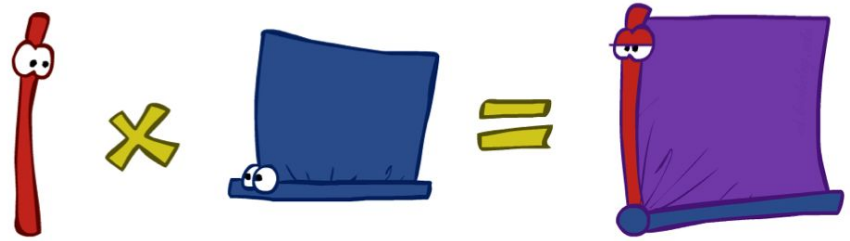


$P(J|A)$

A \ J	true	false	
true	0.9	0.1	} - $P(J|a)$
false	0.05	0.95	} - $P(J|\neg a)$

Operation 1: Pointwise product

- First basic operation: ***pointwise product*** of factors (similar to a ***database join***, ***not*** matrix multiply!)
  - New factor has ***union*** of variables of the two original factors
  - Each entry is the product of the corresponding entries from the original factors



- Example: $P(J|A) \times P(A) = P(A,J)$

$P(A)$

true	0.1
false	0.9

x

$P(J|A)$

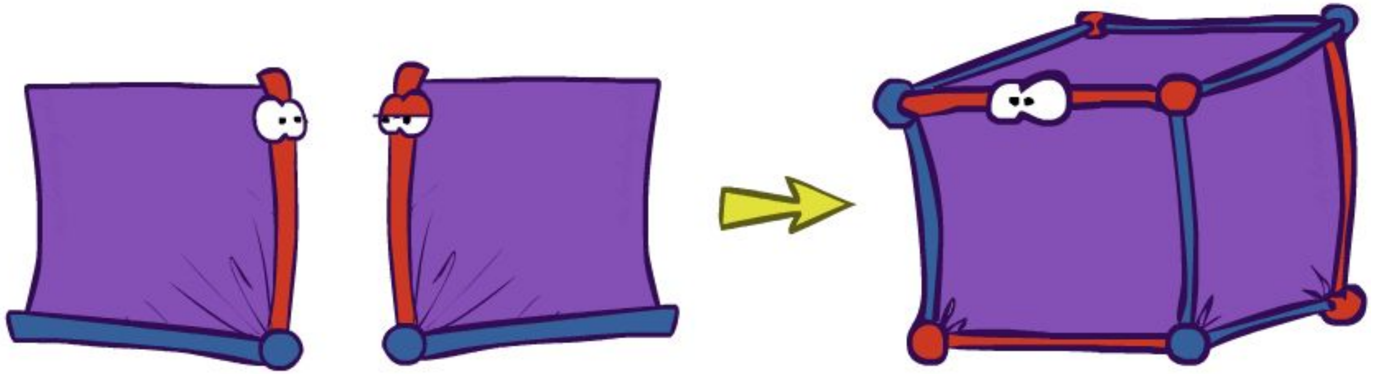
A \ J	true	false
true	0.9	0.1
false	0.05	0.95

=

$P(A,J)$

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

Example: Making larger factors



- Example: $P(A,J)$ x $P(A,M)$ = $P(A,J,M)$

$P(A,J)$

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

x

$P(A,M)$

A \ M	true	false
true	0.07	0.03
false	0.009	0.891

=

$P(A,J,M)$

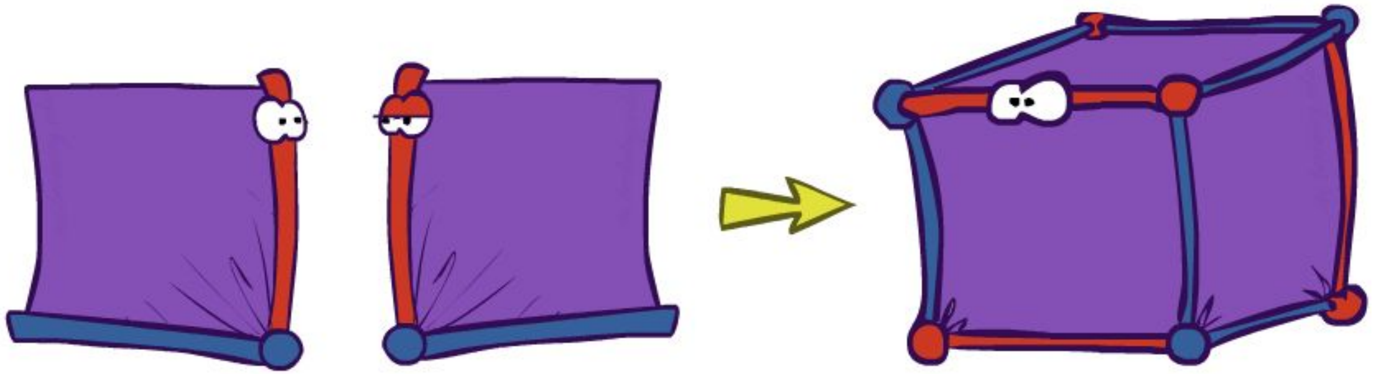
A=false

J \ M	true	false
		18

A=true

J \ M	true	false
true		
false		.0003

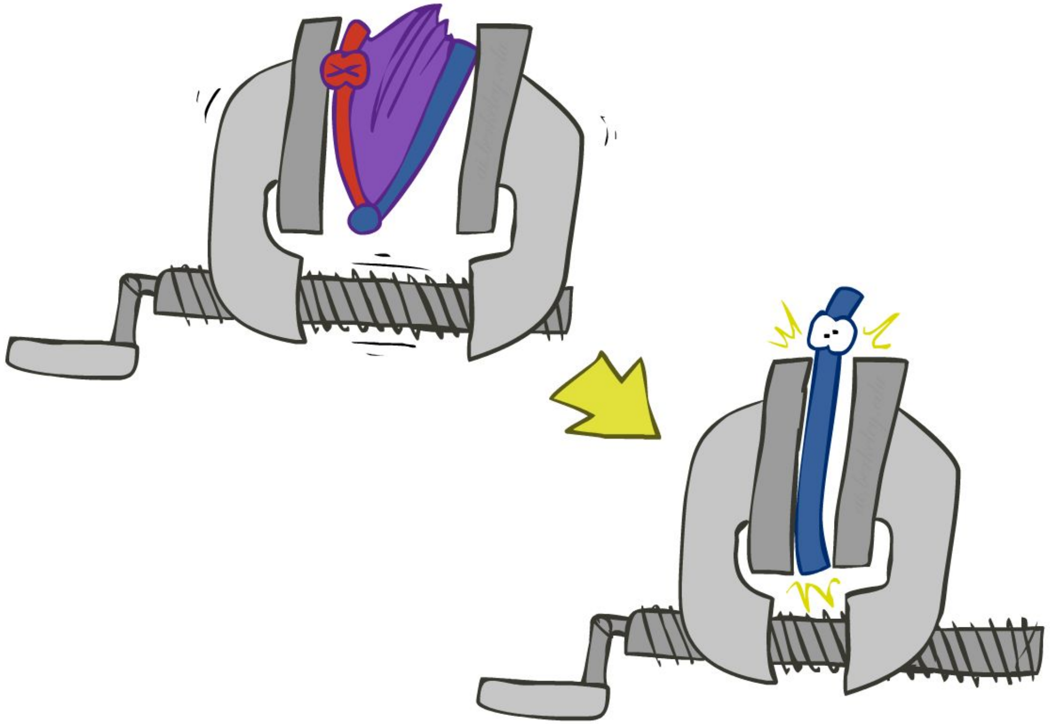
Example: Making larger factors



- Example: $P(U,V)$ x $P(V,W)$ x $P(W,X)$ = $P(U,V,W,X)$
- Sizes: [10,10] x [10,10] x [10,10] = [10,10,10,10]
- I.e., 300 numbers blows up to 10,000 numbers!
- Factor blowup can make VE very expensive

Operation 2: Summing out a variable

- Second basic operation: ***summing out*** (or eliminating) a variable from a factor
  - Shrinks a factor to a smaller one
- Example: $\sum_j P(A,J) = P(A,j) + P(A,\neg j) = P(A)$



$P(A,J)$

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

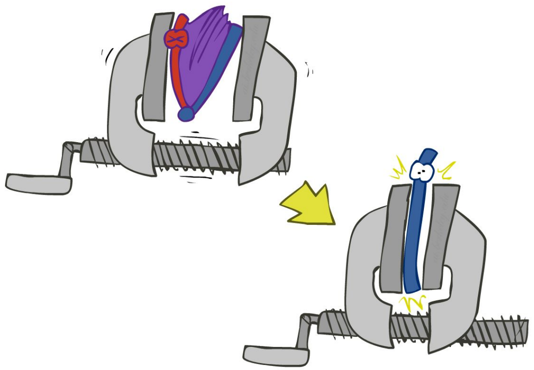
Sum out J

$P(A)$

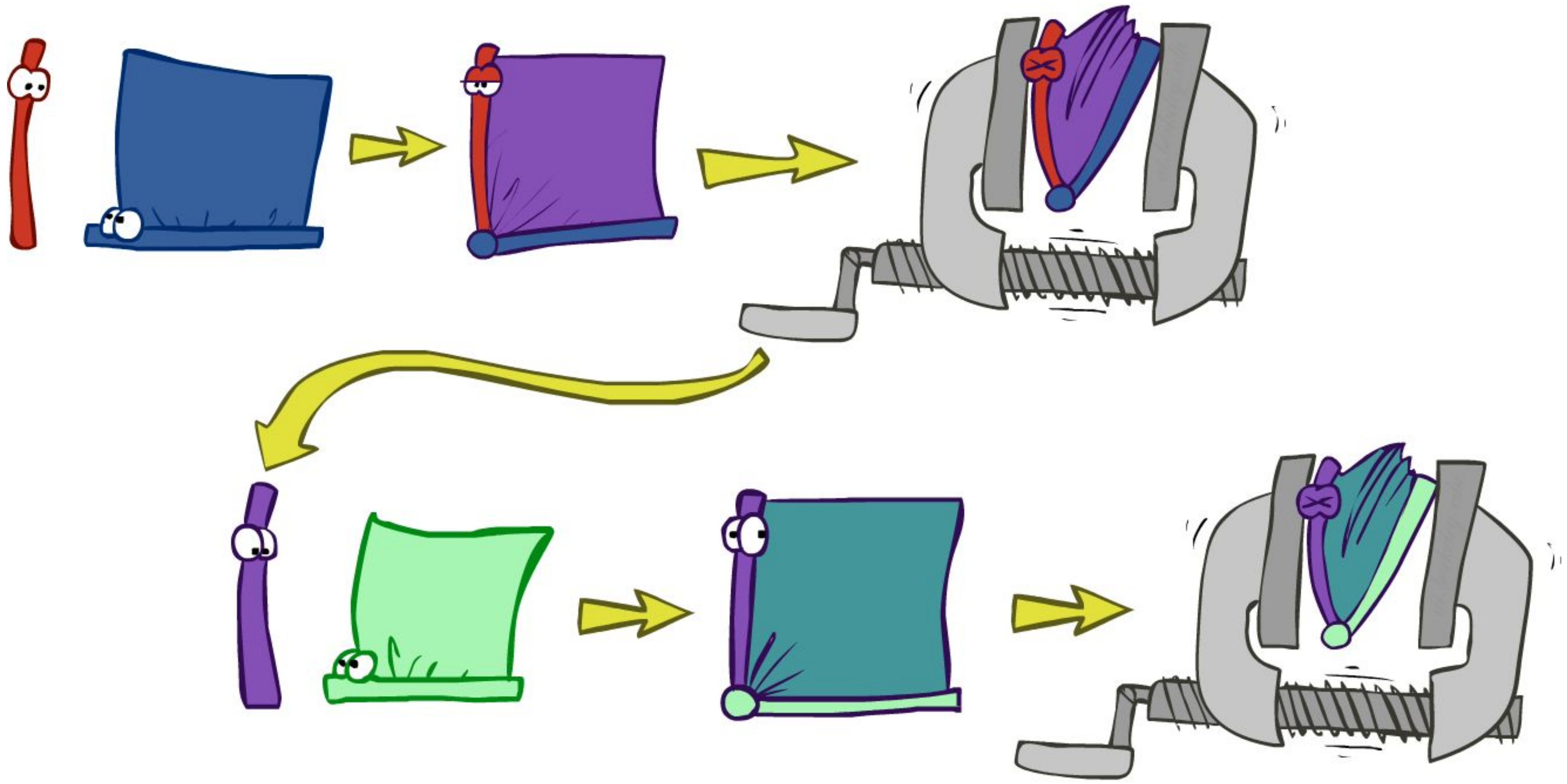
true	0.1
false	0.9

Summing out from a product of factors

- Project the factors each way first, then sum the products
- Example: $\sum_a P(a|B,e) \times P(j|a) \times P(m|a)$
- $= P(a|B,e) \times P(j|a) \times P(m|a) +$
- $P(\neg a|B,e) \times P(j|\neg a) \times P(m|\neg a)$

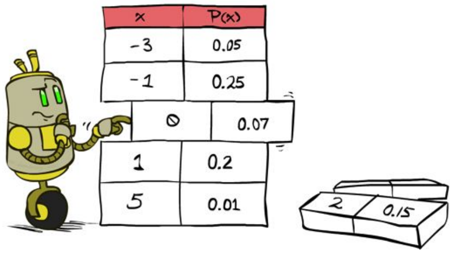


Variable Elimination

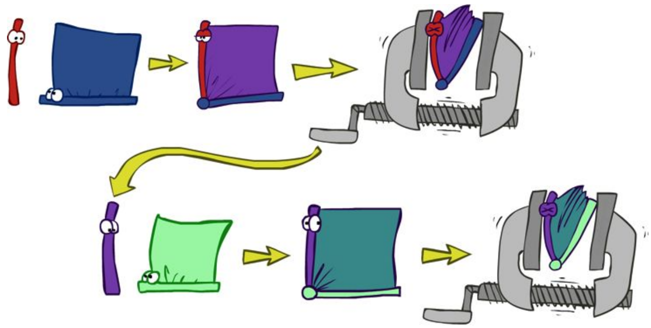


Variable Elimination

- Query: $P(Q|E_1=e_1,.., E_k=e_k)$

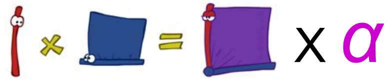


- Start with initial factors:
  - Local CPTs (but instantiated by evidence)



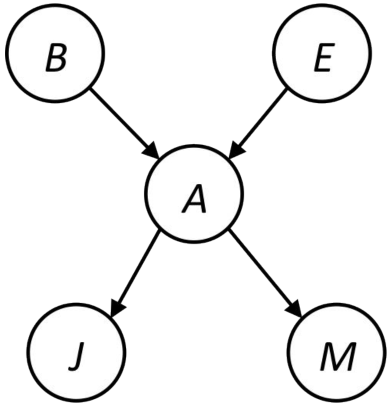
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H_j
  - Eliminate (sum out) H_j from the product of all factors mentioning H_j

- Join all remaining factors and normalize



Example

Query $P(B \mid j,m)$



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

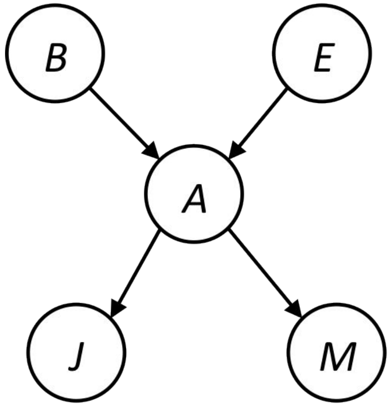
Choose A

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

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

Example

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



Choose E

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

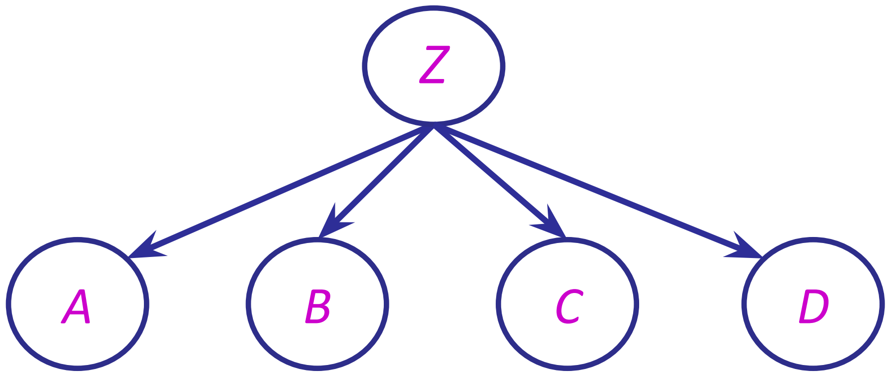
$P(B) \quad P(j,m|B)$

Finish with B

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

Order matters

- Order the terms Z, A, B C, D
  - $P(D) = \alpha \sum_{z,a,b,c} P(z)\, P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)$
  - $\quad = \alpha \sum_z P(z) \sum_a P(a|z) \sum_b P(b|z) \sum_c P(c|z)\, P(D|z)$
  - Largest factor has 2 variables (D,Z)
- Order the terms A, B C, D, Z
  - $P(D) = \alpha \sum_{a,b,c,z} P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)\, P(z)$
  - $\quad = \alpha \sum_a \sum_b \sum_c \sum_z P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)\, P(z)$
  - Largest factor has 4 variables (A,B,C,D)
  - In general, with n leaves, factor of size 2^n



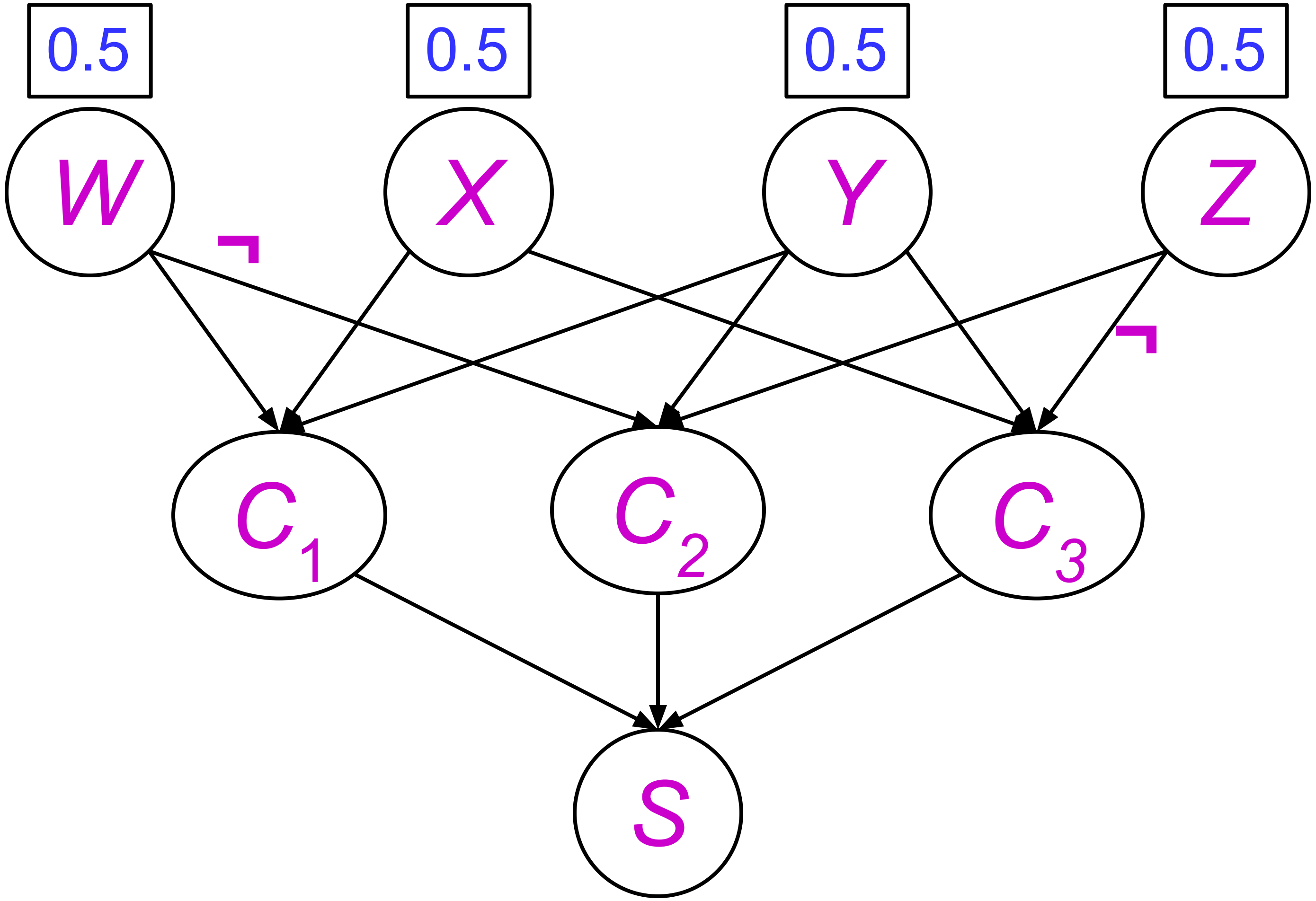
VE: Computational and Space Complexity

- The computational and space complexity of variable elimination is determined by the largest factor (and it's space that kills you)

- 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!

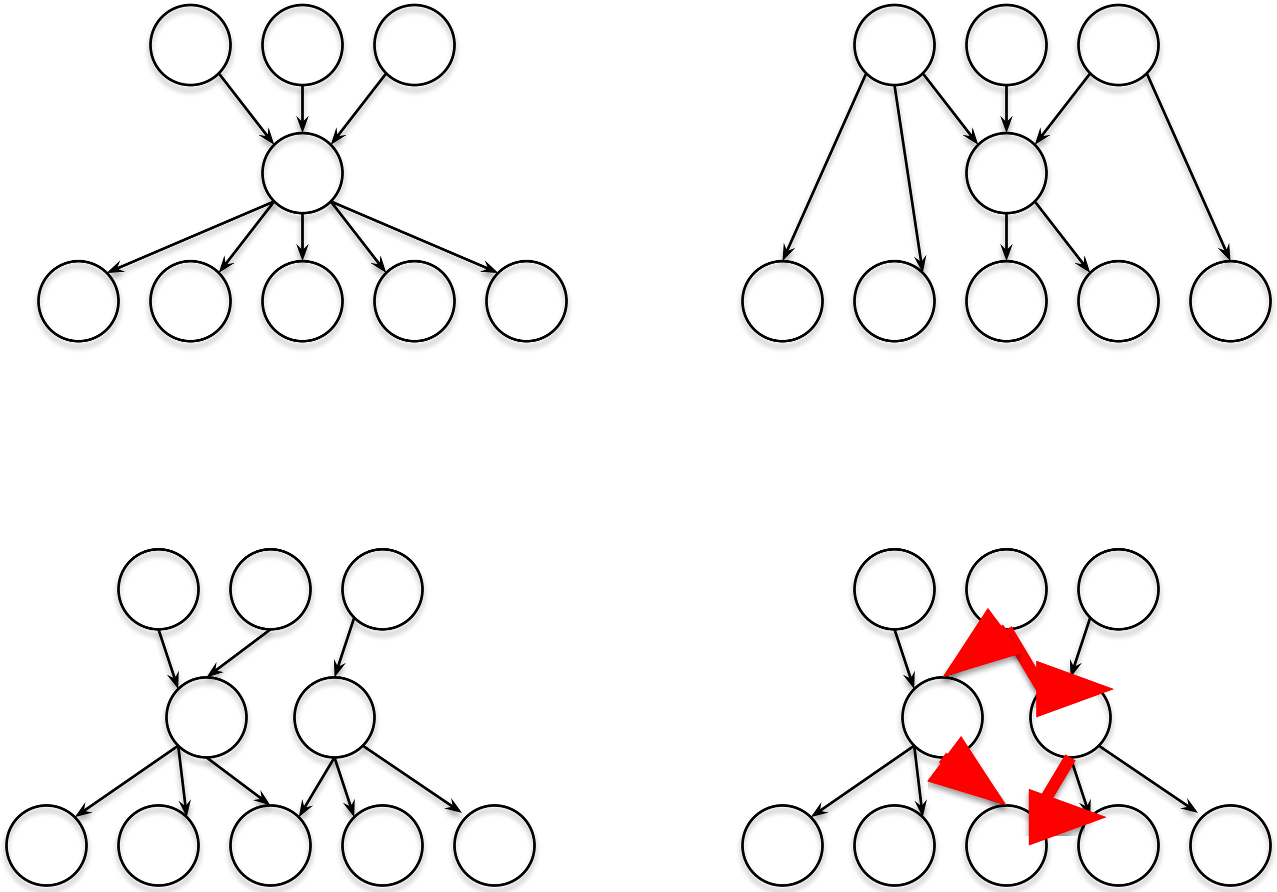
Worst Case Complexity? Reduction from SAT



- Variables: *W, X, Y, Z*
- CNF clauses:
  1. $C_1 = W \vee X \vee Y$
  2. $C_2 = Y \vee Z \vee \neg W$
  3. $C_3 = X \vee Y \vee \neg Z$
- Sentence $S = C_1 \wedge C_2 \wedge C_3$
- $P(S) > 0$ iff S is satisfiable
  - => ***NP-hard***
- $P(S) = K \times 0.5^n$ where K is the number of satisfying assignments for clauses
  - => ***#P-hard***

Polytrees

- A polytree is a directed graph with no undirected cycles
- For poly-trees the complexity of variable elimination is ***linear in the network size*** if you eliminate from the leave towards the roots



Bayes Nets

- ✓ Part I: Representation

- ✓ Part II: Exact inference
  - ✓ Enumeration (always exponential complexity)
  - ✓ Variable elimination (worst-case exponential complexity, often better)
  - ✓ Inference is NP-hard in general

- Part III: Approximate Inference

- Later: Learning Bayes nets from data