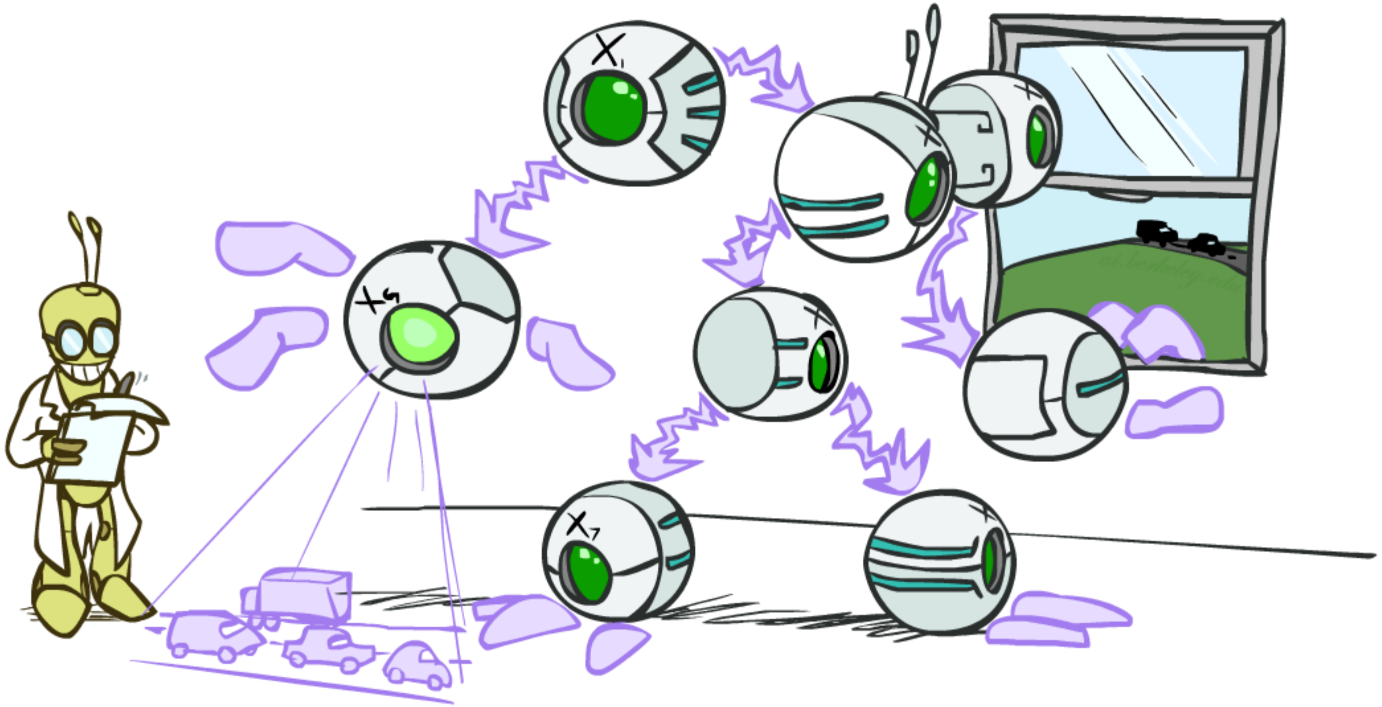
CS 188: Artificial Intelligence

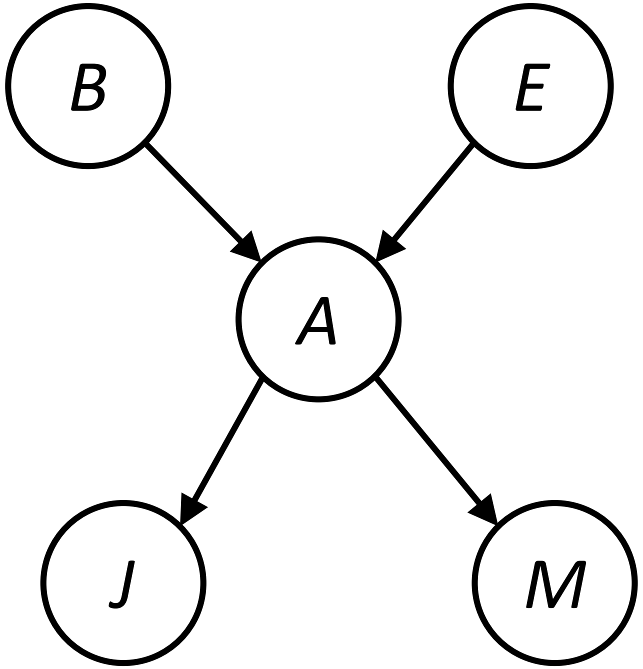
Bayes Nets: Exact Inference

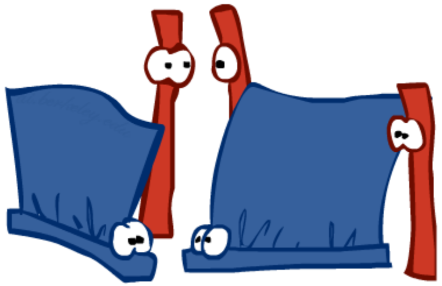


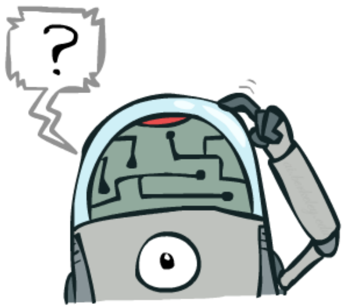
Instructor: Stuart Russell and Peyrin Kao

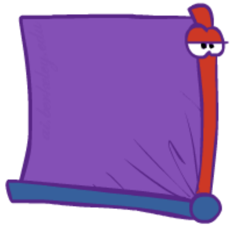
Inference by Enumeration in Bayes Net

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









Can we do better?

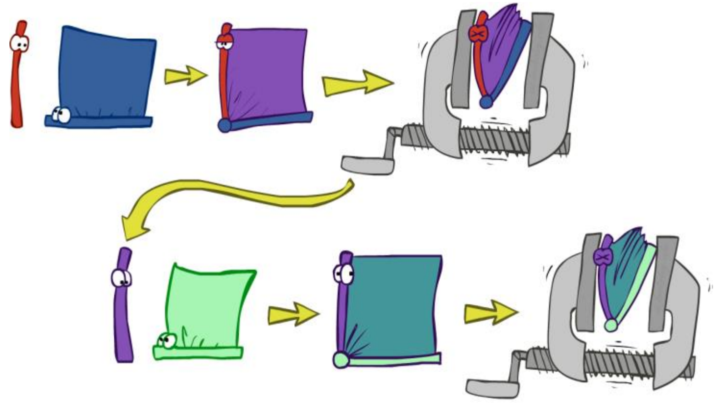
- Consider **uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz**
  - 16 multiplies, 7 adds
  - Lots of repeated subexpressions!
- Rewrite as **(u+v)(w+x)(y+z)**
  - 2 multiplies, 3 adds
- $\sum_{e,a} P(B)\ P(e)\ P(a|B,e)\ P(j|a)\ P(m|a)$
- $= P(B)P(e)P(a|B,e)P(j|a)P(m|a) + P(B)P(\neg e)P(a|B,\neg e)P(j|a)P(m|a)$

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

  Lots of repeated subexpressions!

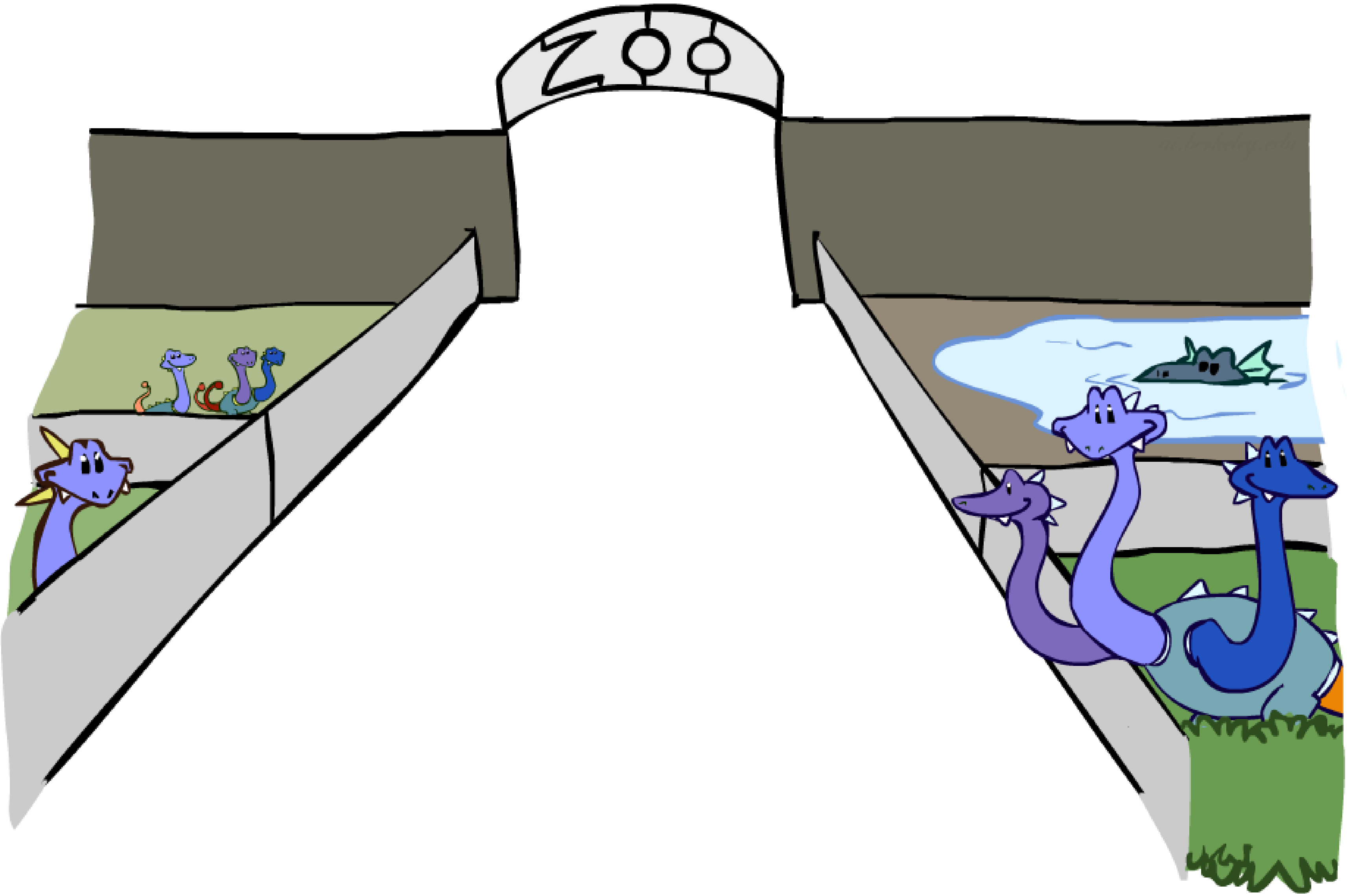
Variable elimination: The basic ideas

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



- 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

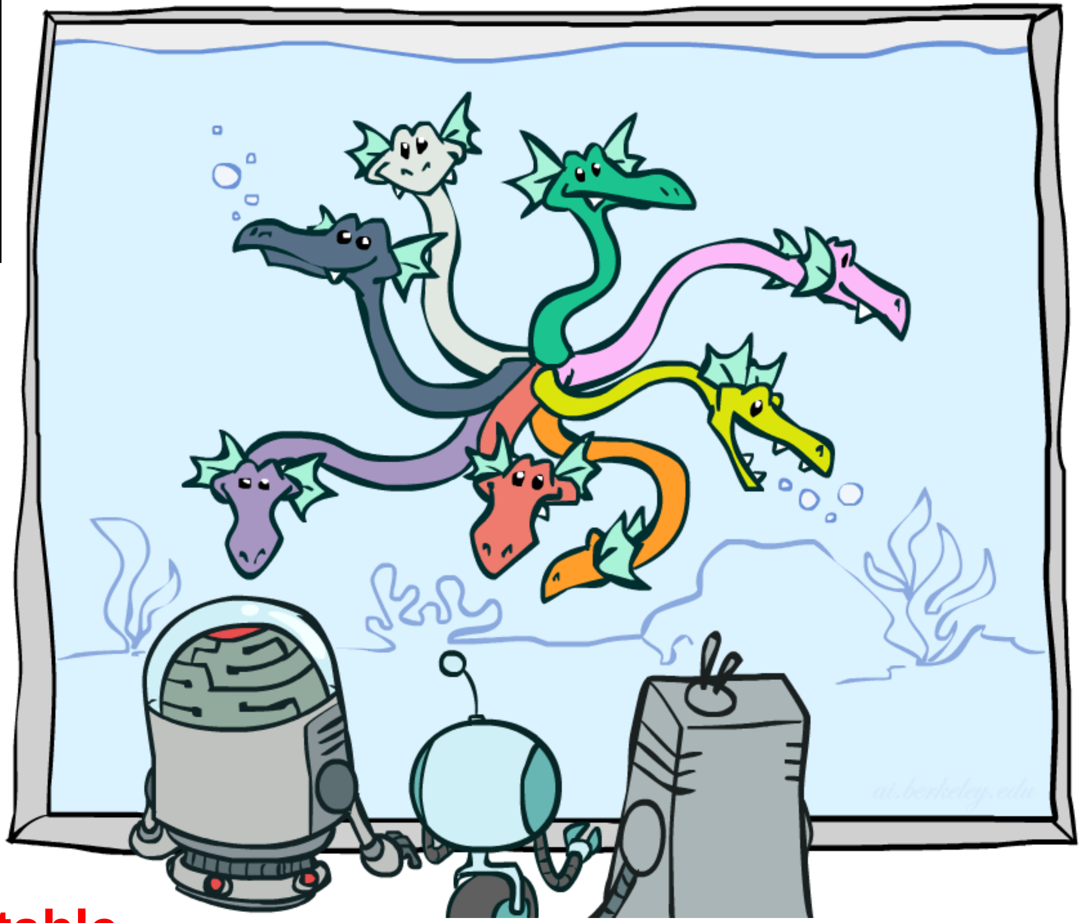
$P(A,J)$

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)

$P(a,J) = P_a(J)$

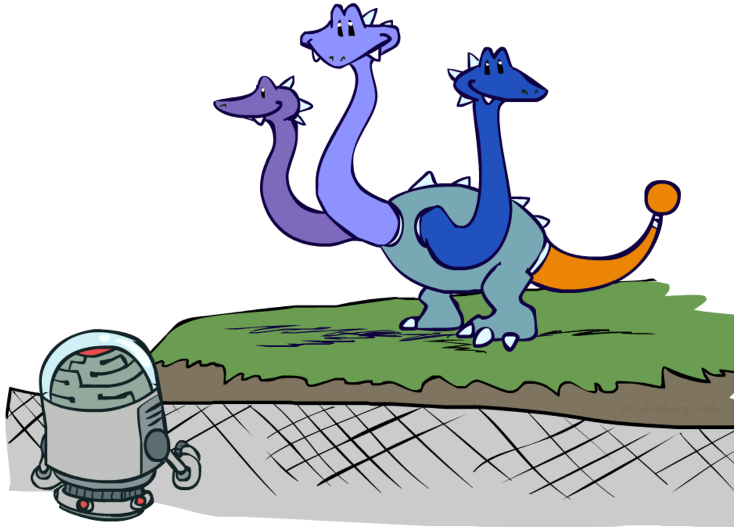
A \ J	true	false
true	0.09	0.01



Number of variables (capitals) = dimensionality of the table

Factor Zoo II

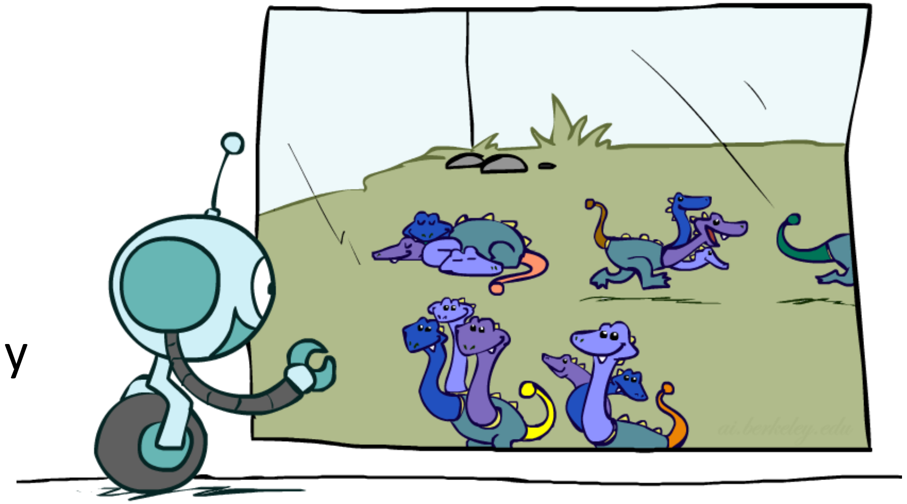
- Single conditional: $P(Y \mid x)$
  - Entries $P(y \mid x)$ for fixed x, all y
  - Sums to 1



P(J|a)

A \ J	true	false
true	0.9	0.1

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

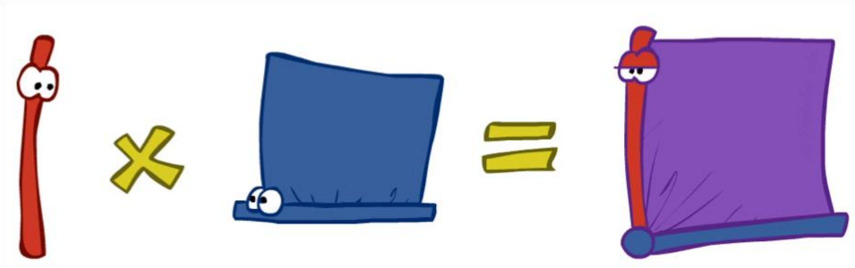


P(J|A)

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

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

$P(J|A)$

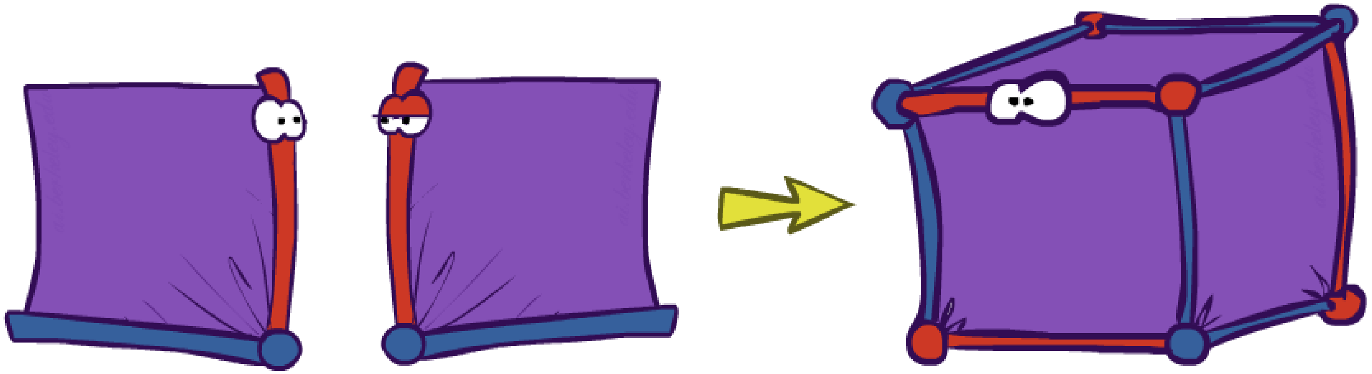
A \ J	true	false
true	0.9	0.1
false	0.05	0.95

=

$P(A,J)$

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

Example: Making larger factors



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

$P(A,J)$

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

=

$P(A,J,M)$

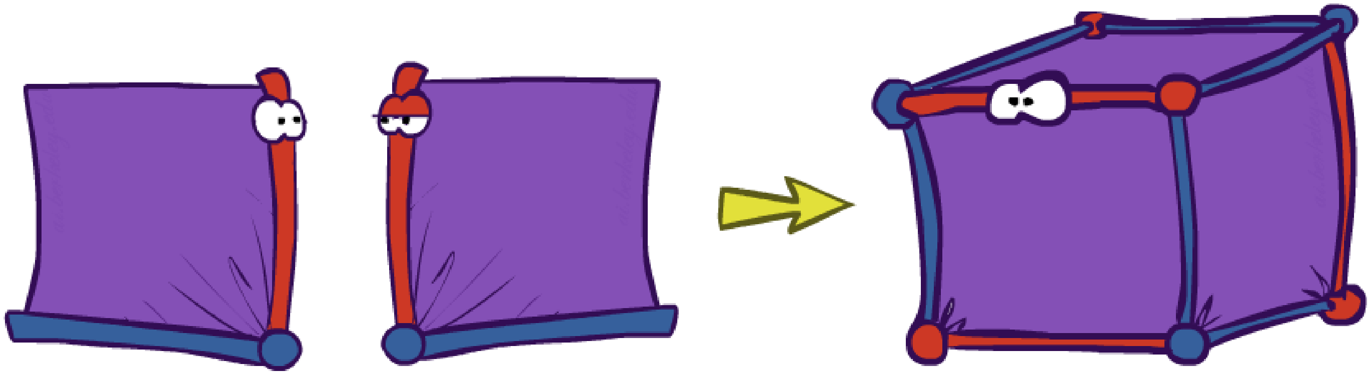
A=false

J \ M	true	false
		18

A=true

J \ M	true	false
true		
false		.0003

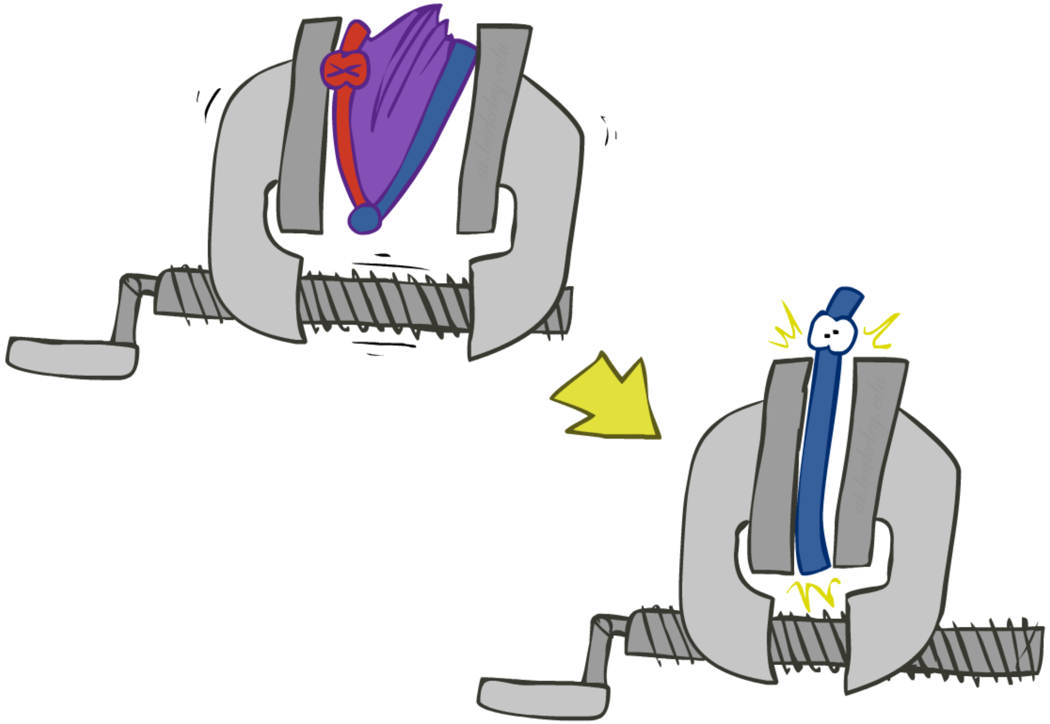
Example: Making larger factors



- Example: $P(U,V) \times P(V,W) \times P(W,X) = P(U,V,W,X)$
- 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

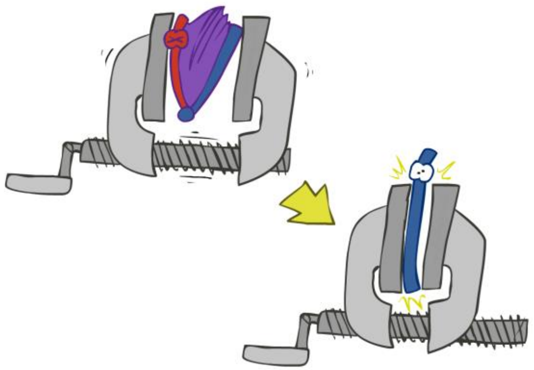
Sum out J

$P(A)$

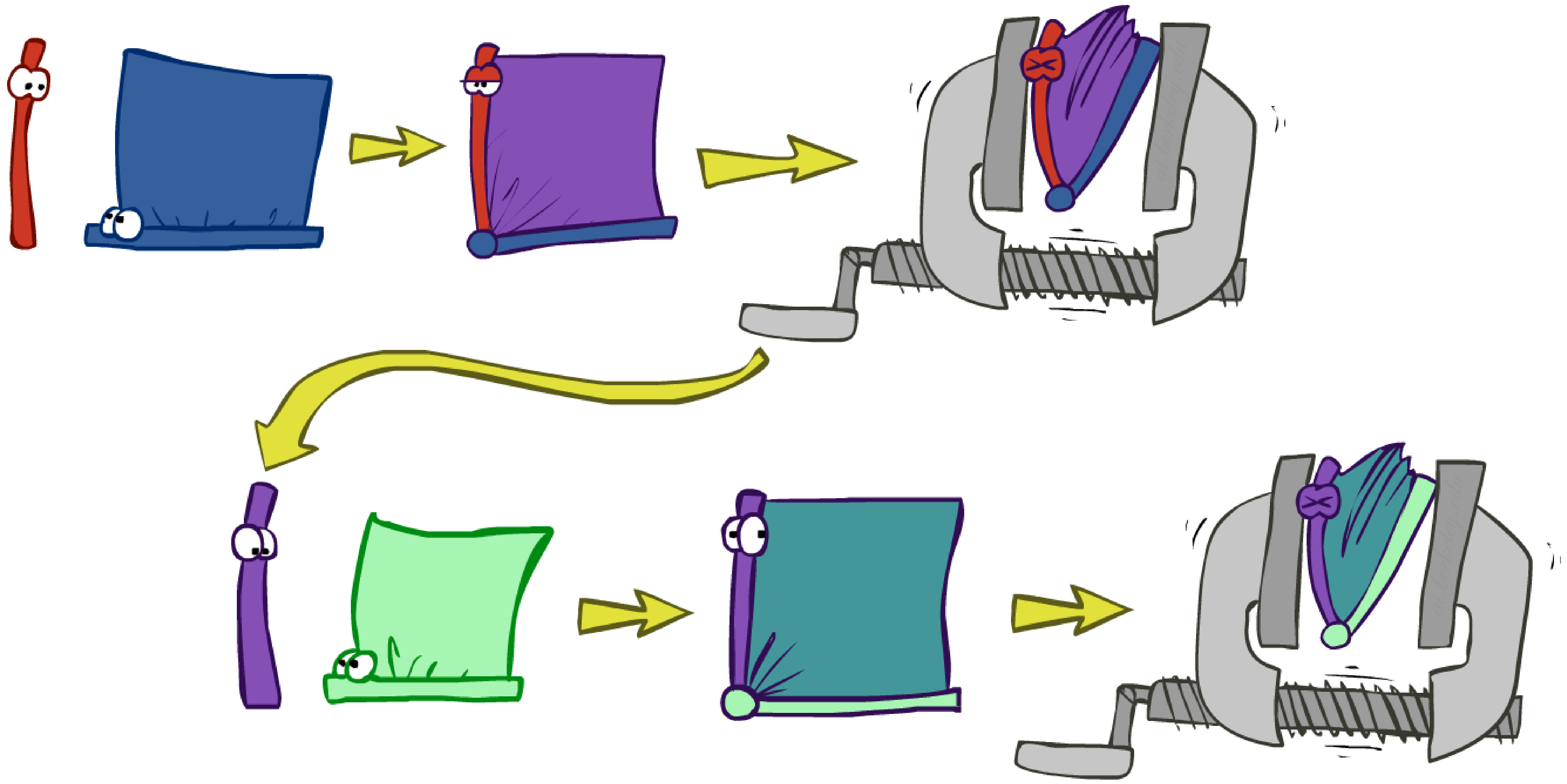
true	0.1
false	0.9

Summing out from a product of factors

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

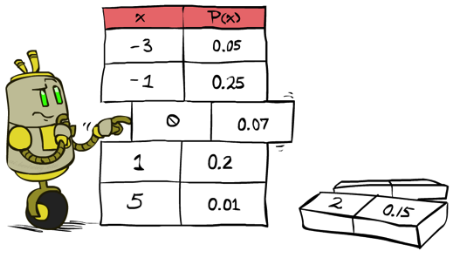


Variable Elimination



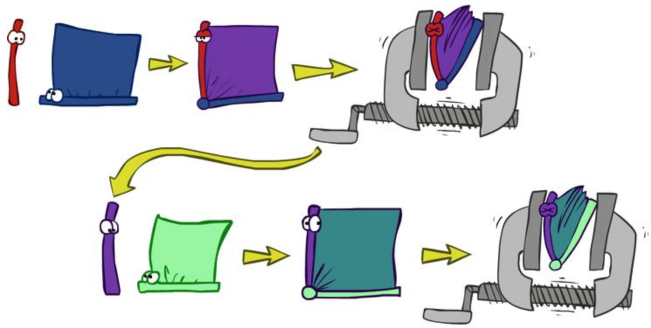
Variable Elimination

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

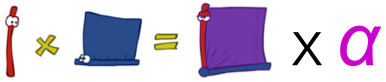


- Start with initial factors:
  - Local CPTs (but instantiated by evidence)

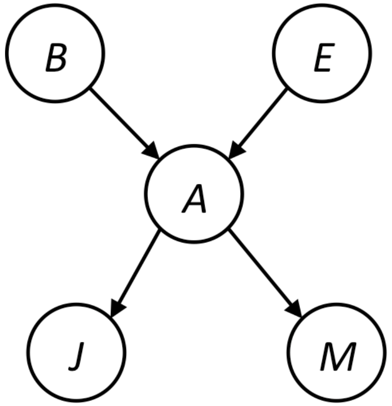
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H_j
  - Eliminate (sum out) H_j from the product of all factors mentioning H_j



- 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

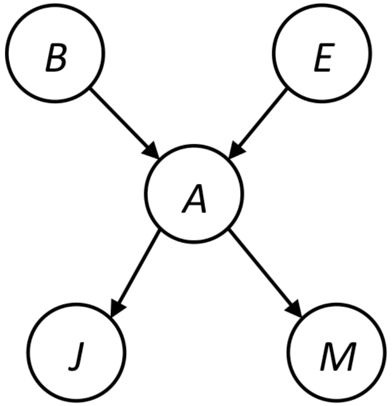
$P(A|B,E)$
$P(j|A)$
$P(m|A)$

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

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

Example

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



Choose E

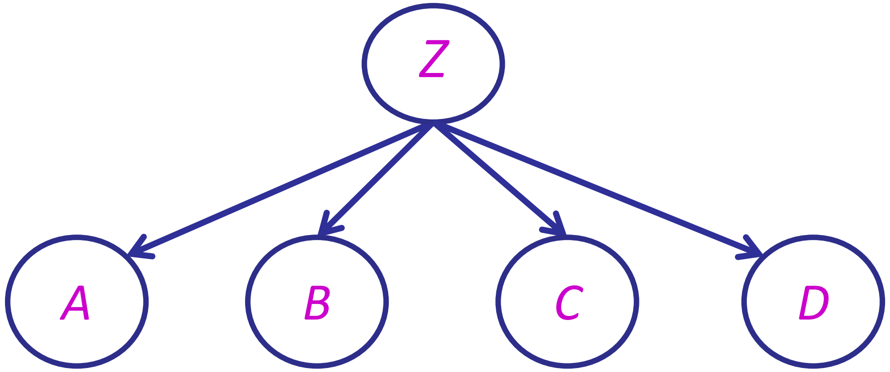
$P(E)$
$P(j,m|B,E)$ $\times$ Σ $P(j,m|B)$

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

Finish with B

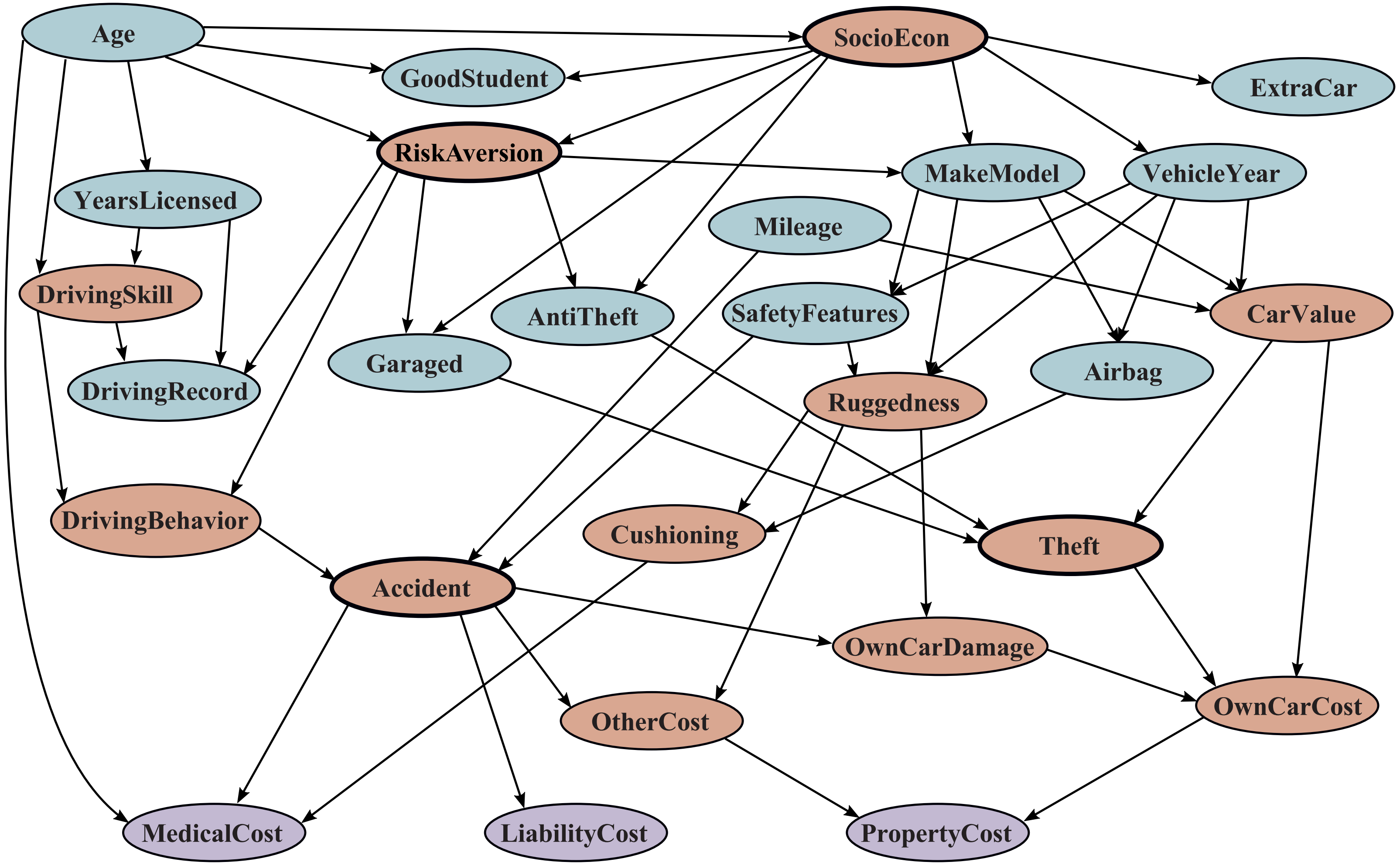
$P(B)$
$P(j,m|B)$ $\times$ $P(j,m,B)$ Normalize $P(B \mid j,m)$

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

Example Bayes' Net: Car Insurance



Enumeration: 227M operations

Elimination: 221K operations

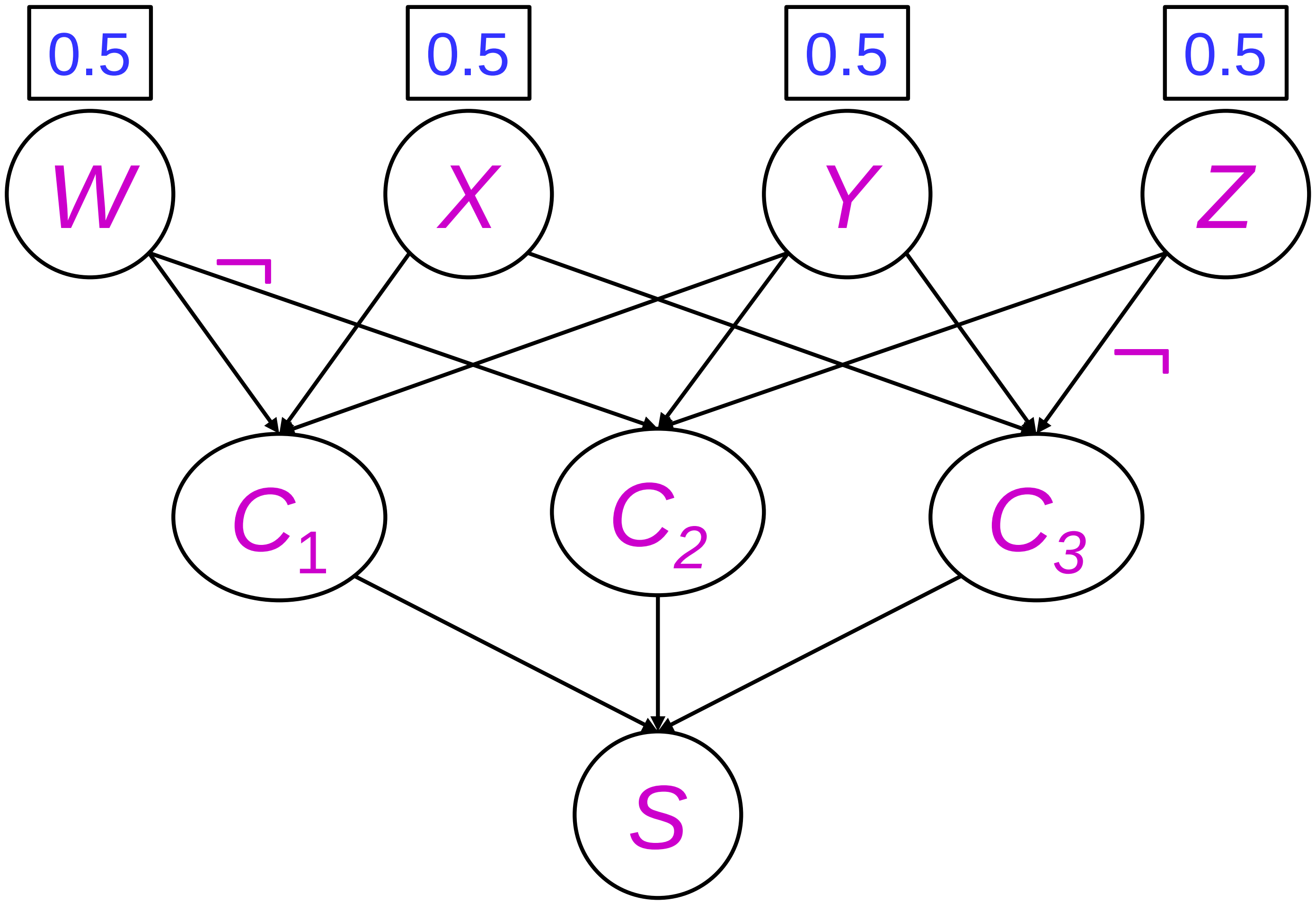
Computational and Space Complexity

- 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., ZABCD example 2^n vs. 2

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

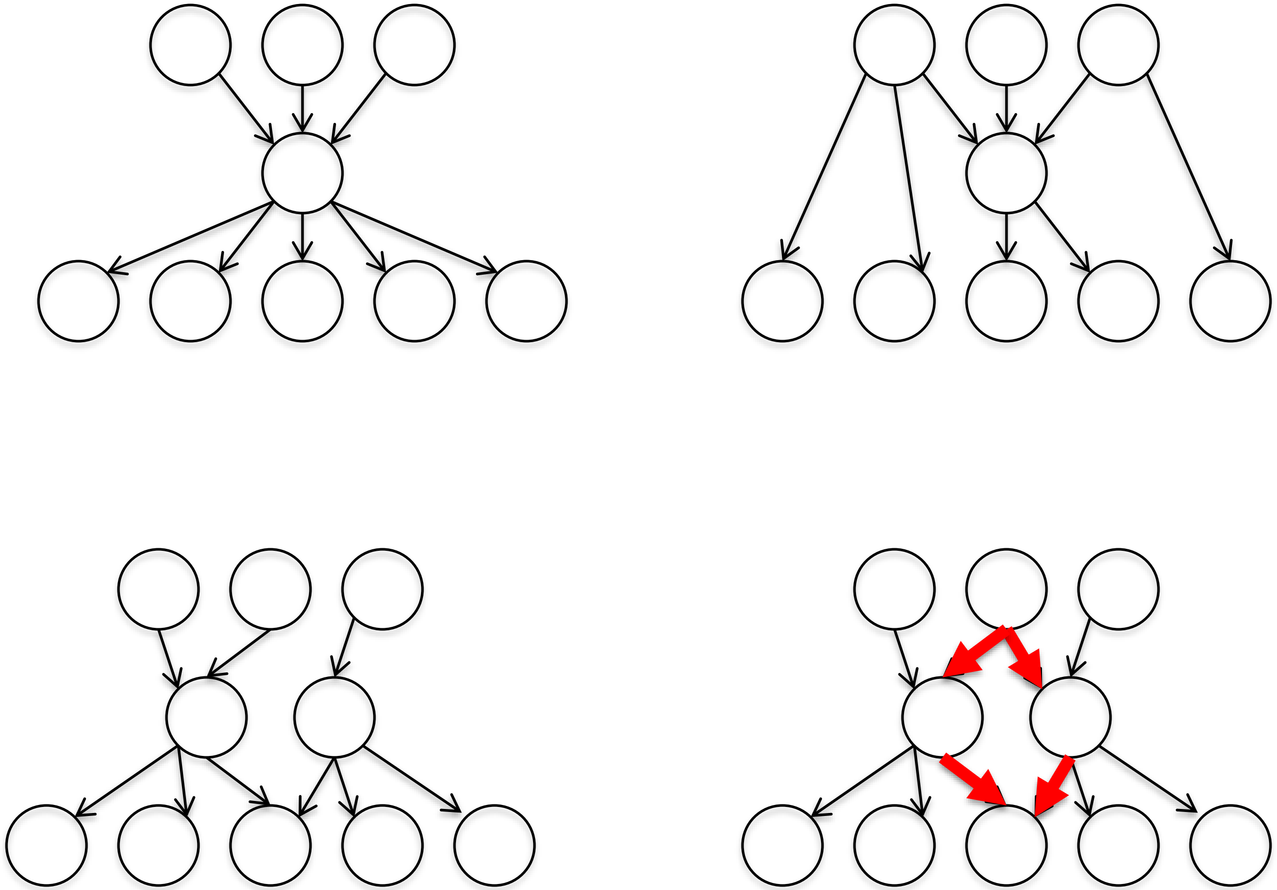
Worst Case Complexity? Reduction from SAT



- 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 leaves towards the roots



Summary

- Exact inference = sums of products of conditional probabilities from the network
- Enumeration is always exponential
- Variable elimination reduces this by avoiding the recomputation of repeated subexpressions
  - Massive speedups in practice
  - Linear time for polytrees

- Exact inference is #P-hard

- Next: approximate inference