# CS 486/686: Introduction to Artificial Intelligence

Reasoning Under Uncertainty

# Plan for Today

- Quick review of basic probability
- Bayesian Networks
  - What they are
  - What they mean

# Basics of Probability

- Probability Distribution
  - A specification of a probability for every event in our sample space
  - Probabilities are non-negative and must sum to 1

- Joint Probability Distribution
  - Often the world is described by two or more random variables
  - A joint distribution specifies probabilities for all combinations of events
  - Given two random variables A and B, the joint distribution P(A=a, B=b) for all a,b
  - Marginalization (sumout rule)
    - $P(A = a) = \sum_b P(A = a, B = b)$
    - $P(B = b) = \sum_a P(A = a, B = b)$

# Basics of Probability Distributions

sunny

| | cold | ~cold |
|---|---|---|
| headache | 0.108 | 0.012 |
| ~headache | 0.016 | 0.064 |

~sunny

| | cold | ~cold |
|---|---|---|
| headache | 0.072 | 0.008 |
| ~headache | 0.144 | 0.576 |

P(headache^sunny^cold)=0.108  P(~headache^sunny^~cold)=0.064

P(headache V sunny) = 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28

P(headache)=0.108+0.012+0.072+0.008=0.2

# Basics of Probability

- Conditional Probability
  - P(A|B): fraction of worlds in which B is true that also have A being true

- $P(A|B) = \frac{P(A,b)}{P(B)}$

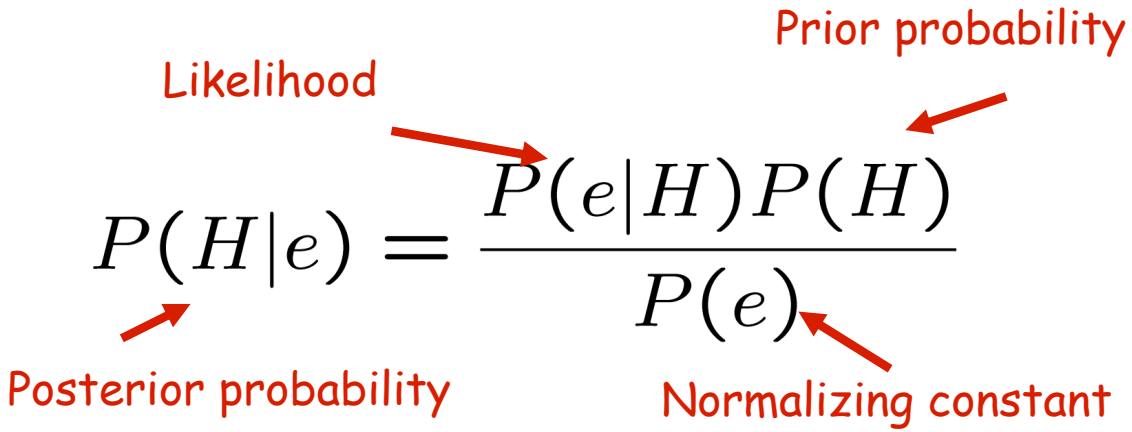
- Chain Rule: $P(A,B) = P(A|B)P(B)$

- Bayes Rule: $P(B|A) = \frac{P(A|B)P(B)}{P(A)}$

**Memorize these!!**

# Bayes Rules and Inference

- Often we want to form a hypothesis about the world given what we have observed
- Bayes rule allows us to compute a belief about hypothesis H, given evidence e



# Basics of Probability

sunny

| | cold | ~cold |
|---|---|---|
| headache | 0.108 | 0.012 |
| ~headache | 0.016 | 0.064 |

~sunny

| | cold | ~cold |
|---|---|---|
| headache | 0.072 | 0.008 |
| ~headache | 0.144 | 0.576 |

P(headache ^ cold| sunny)= P(headache ^ cold ^ sunny)/P(sunny)

= 0.108/(0.108+0.012+0.016+0.064)

= 0. 54

P(headache ^ cold| ~sunny)= P(headache ^ cold ^ ~sunny)/P(~sunny)

= 0.072/(0.072+0.008+0.144+0.576)

= 0.09

# Challenges

- How do we specify the full joint distribution over a set of n random variables?

- What if we want to determine the distribution over a single variable in our joint distribution?

# Independence and Conditional Independence

- Two variables A and B are **independent** if knowledge of A does not change uncertainty of B (and vice versa)
  - P(A|B)=P(A)
  - P(B|A)=P(B)
  - P(A,B)=P(A)P(B) and more generally $P(A_1,A_2,\ldots,A_n)=\prod_i Ai$
- Two variables A and B are conditionally independent given variable C if knowing the value of B does not change the uncertainty of A (and vice versa) ***if the value of C is known***
  - P(A|C)=P(A|B,C)
  - P(B|C)=P(B|A,C)
  - P(A,B|C)=P(A|C)P(B}C)

# Why Do We Care About Independence

- If we have n Boolean independent random variables, we only need n parameters to specify the full joint distribution (instead of $2^n$-1)
  - Furthermore, inference becomes O(n) instead of O($2^n$)!

$4$ independent Boolean random vars $X_1, X_2, X_3, X_4$

$$\Pr(x_1) = 0.4, \Pr(x_2) = 0.2, \Pr(x_3) = 0.5, \Pr(x_4) = 0.8$$

$$\begin{aligned}\Pr(x_1, \sim x_2, x_3, x_4) &= \Pr(x_1)\,(1 - \Pr(x_2))\,\Pr(x_3)\,\Pr(x_4) \\ &= (0.4)(0.8)(0.5)(0.8) \\ &= 0.128\end{aligned}$$

$$\begin{aligned}\Pr(x_1, x_2, x_3 | x_4) &= \Pr(x_1)\,\Pr(x_2)\,\Pr(x_3)\,\mathbf{1} \\ &= (0.4)(0.2)(0.5)(1) \\ &= 0.04\end{aligned}$$

# Leveraging Independence

- While most domains do not exhibit full independence, many do have a fair amount of conditional independence
  - We want to exploit conditional independence for both representation and reasoning

**Bayesian Networks do exactly this**

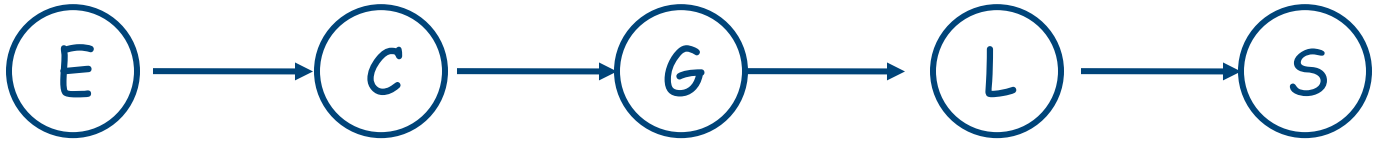
# Notation Break

- P(X) for variable X (or set of variables) refers to (marginal) distribution over X
- Distinguish between P(X) (distribution) and P(x) (numbers)
  - Think of P(X) as a function that accepts any $x_i$ in Dom(X) and returns a number
- P(X|Y) is the **family** of conditional distributions over X (one for each y in Dom(Y)
  - Think of P(X|Y) as a function that accepts any $x_i$ and $y_k$ and returns $P(x_i|y_k)$

# Exploiting Conditional Independence

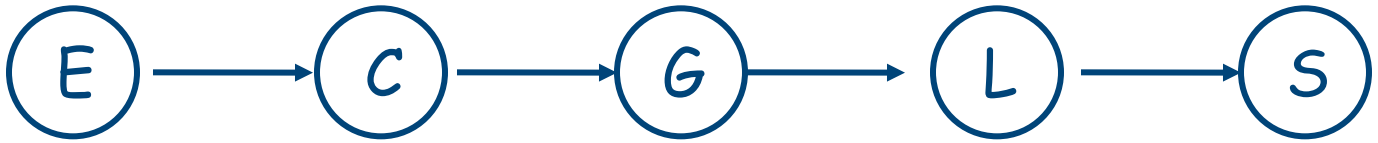
## Consider the following story

If Kate woke up too early (E), she probably needs coffee (C); if Kate needs coffee (C), she is likely to be grumpy (G). If she is grumpy, then it's possible that the lecture won't go smoothly (L). If the lecture does not go smoothly, then the students will likely be sad (S).



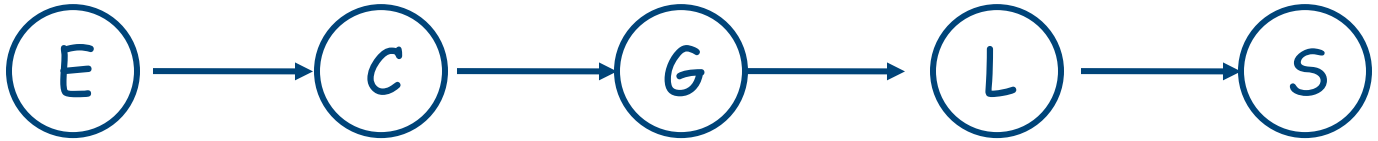

E – Kate woke too early  G – Kate is grumpy  S – Students are sad
C – Kate needs coffee  L– The lecture did not go smoothly

# Conditional Independence




- If you learned any of E, C, G, or L then your assessment of P(S) would change
  - if any of these are seen to be true, you would increase P(s) and decrease P(~s)
  - So S is **not independent** of E, C, G, or L

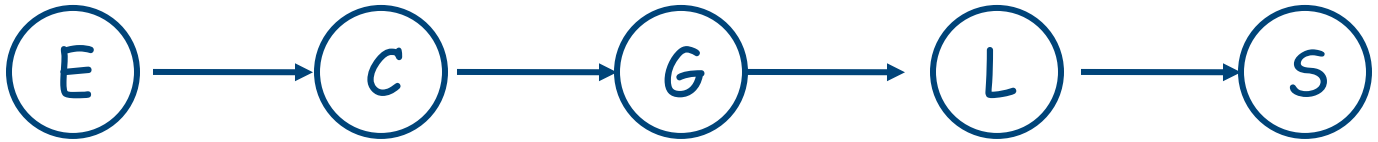
# Conditional Independence




But if you knew the value of L (true or false) then learning the values of E, C, or G would not influence P(S)

- Students are not sad because Kate did not have a coffee, they are sad because of the lecture
- **So S is independent of E, C, and G, given L**

# Conditional Independence



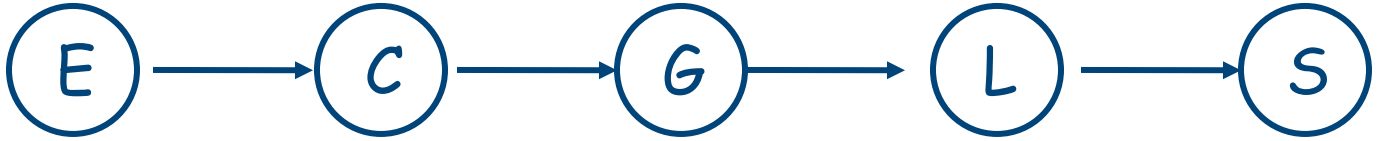

Similarly

- L is independent of E and C, given G
- G is independent of E given C

This means that

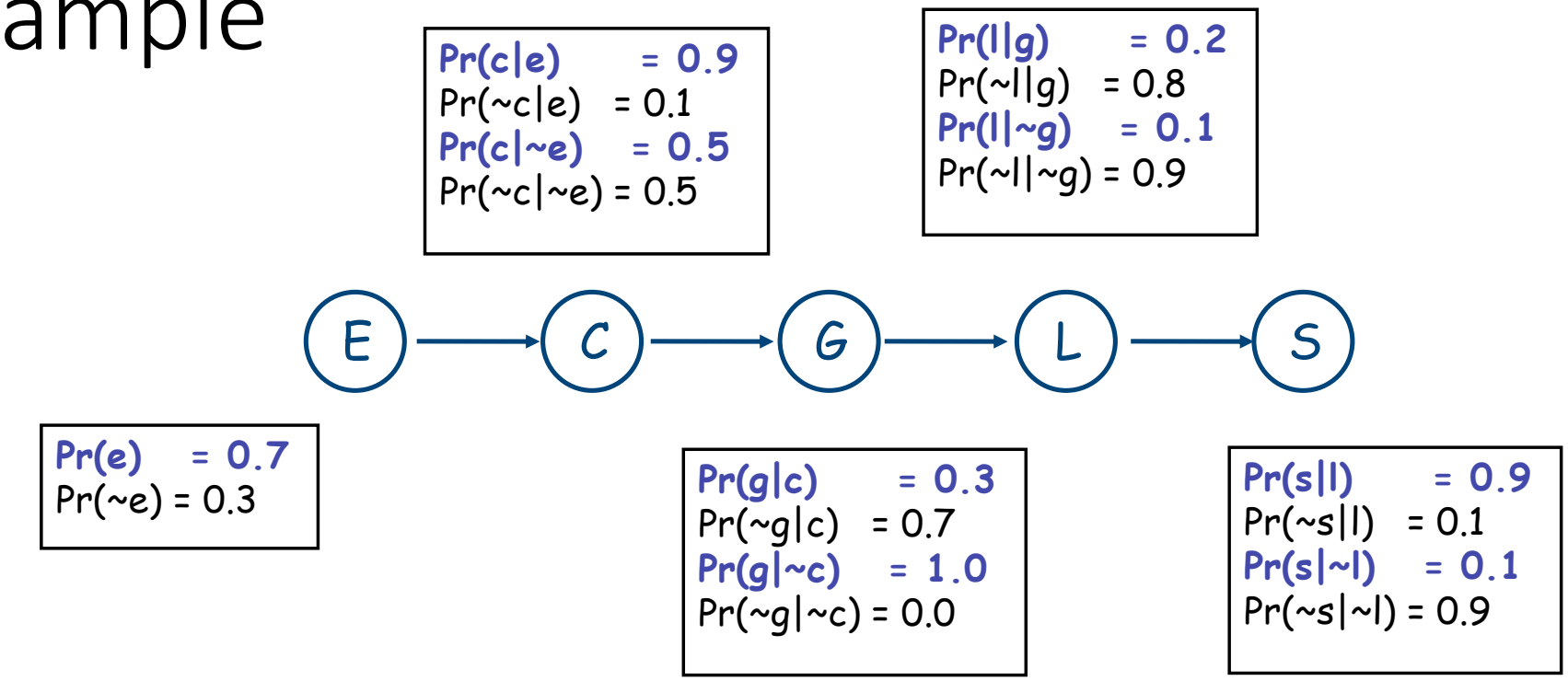
- P(S|L,{G,C,E})=
- P(L|G, {C,E})=
- P(G|C,{E})=
- P(C|E)=
- P(E)=

# Conditional Independence




- By the chain rule
  - P(S,L,G,C,E)=?
- By our independence assumptions
  - P(S,L,G,C,E)=?
- We can specify the full joint by specifying five **conditional distributions**: P(S|L), P(L|G), P(G|C), P(C|E) and P(E)

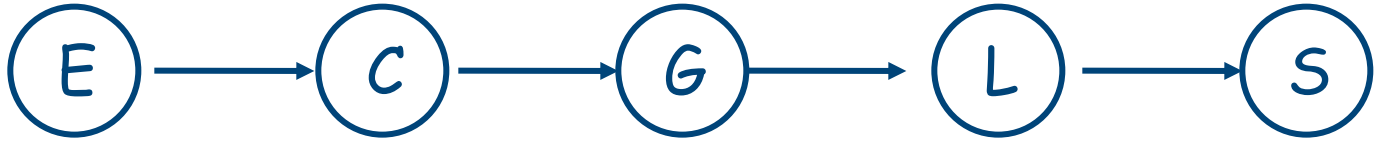
# Example




Specifying the joint requires only 9 parameters instead of 31 for explicit representation

- linear in number of vars instead of exponential
- linear in general if dependence has a chain structure

# Inference is Easy




$$P(g) = \sum_{c_i \in Dom(C)} \Pr(g \mid c_i) \Pr(c_i)$$

$$= \sum_{c_i \in Dom(C)} \Pr(g \mid c_i) \sum_{e_i \in Dom(E)} \Pr(c_i \mid e_i) \Pr(e_i)$$

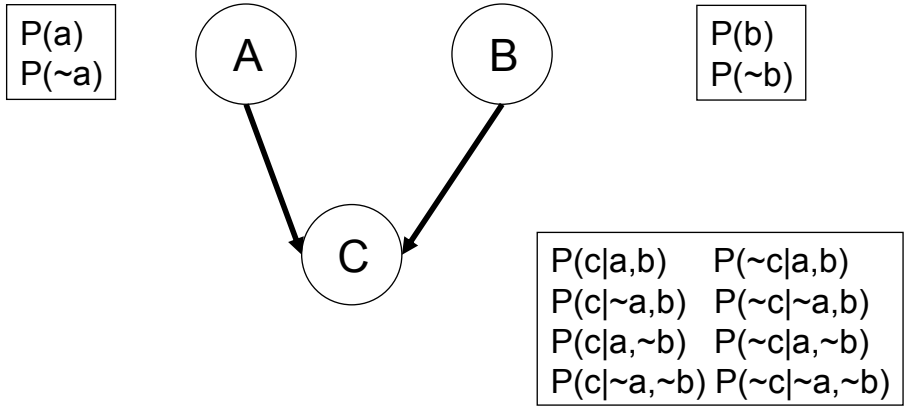
These are all terms specified in our local distributions!

# Bayesian Networks

- A Bayesian network is a **graphical representation** of direct dependencies over a set of variables + a **set of conditional probability distributions** (CPTs) quantifying the strength of the influences

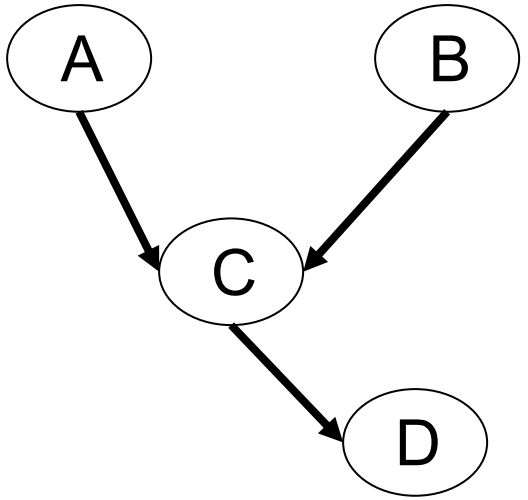
A BN over a set of variables $\{X_1,..,X_n\}$ consists of

- A direct acyclic graph whose nodes are the variables
- A set of CPTs $P(X_i|\text{Parents}(X_i))$ for each $X_i$



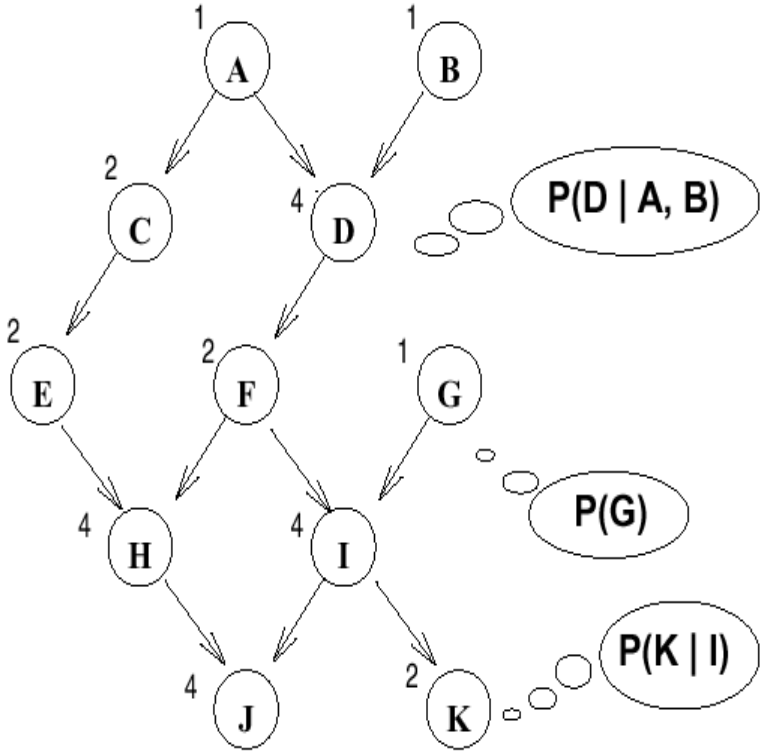
# Bayesian Networks

- Parents of a node
- Children of a node
- Descendents of a node
- Ancestors of a node
- Family: set of nodes consisting of X and its parents
  - CPTs are defined over families




Parents(C)={A,B}
Children(A)={C}
Descendents(B)={C,D}
Ancestors{D}={A,B,C}
Family{C}={C,A,B}

# Bayesian Network




- A couple CPTS are "shown"
- Explicit joint requires $2^{11}$ -1 =2047 params
- BN requires only 27 parms (the number of entries for each CPT is listed)

# Semantics

The structure of Bayesian Network means: every $X_i$ is conditionally independent of all of its non-descendants given its parents

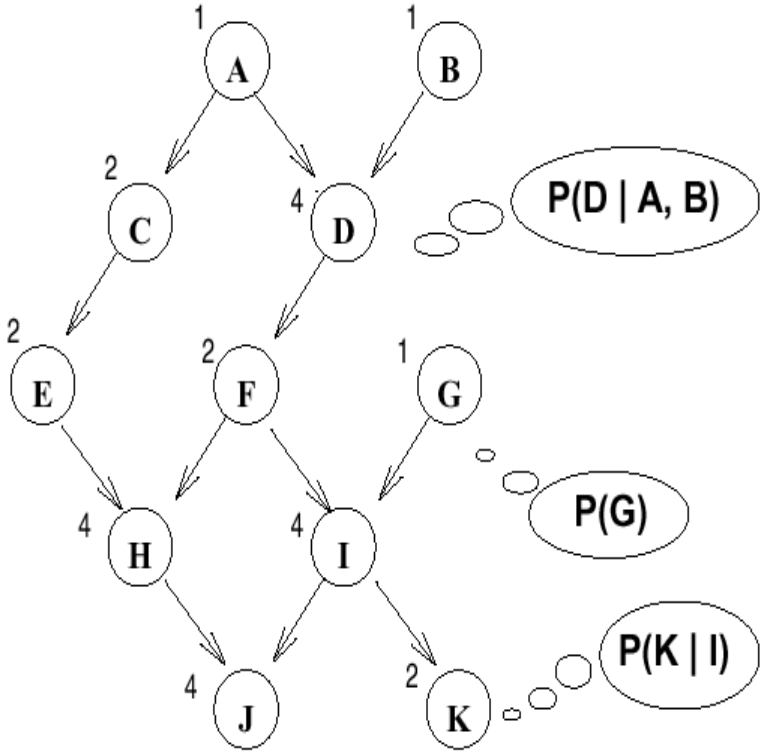
$$\Pr(X_i \mid S \cup \text{Par}(X_i)) = \Pr(X_i \mid \text{Par}(X_i))$$

for any subset S of the NonDescendants($X_i$)

# Semantics

- Query $P(x_1,x_2,\ldots,x_n)$
  - $= P(x_n|x_{n-1},\ldots,x_1)P(x_{n-1}|x_{n-2},\ldots,x_1)P(x_{n-2}|x_{n-3},\ldots,x_1)\ldots P(x_1)$
  - $= P(x_n|Par(x_n))P(x_{n-1}|Par(x_{n-1}))\ldots P(x_1)$

# Bayesian Network




- A couple CPTS are "shown"
- Explicit joint requires $2^{11}$ -1 =2047 params
- BN requires only 27 parms (the number of entries for each CPT is listed)

# Constructing a BN

Given any distribution over variables $X_1, X_2, \ldots, X_n$, we can construct a BN that faithfully represents that distribution

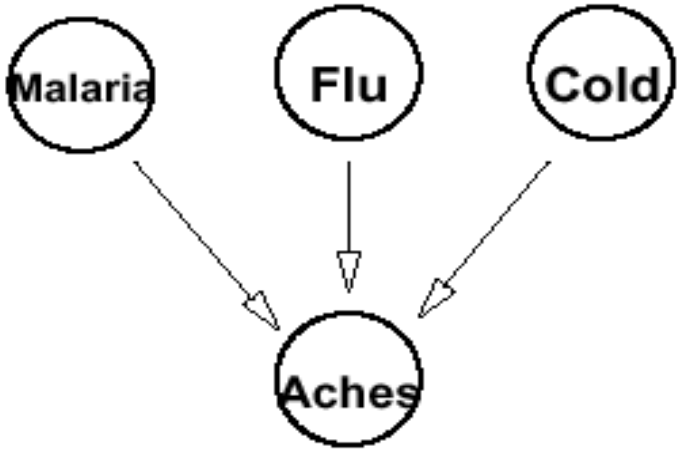
Take any ordering of the variables (say, the order given), and go through the following procedure for $X_n$ down to $X_1$. Let Par($X_n$) be any subset S $\subseteq$ $\{X_1, \ldots, X_{n-1}\}$ such that $X_n$ is independent of $\{X_1, \ldots, X_{n-1}\}$ - S given S. Such a subset must exist. Then determine the parents of $X_{n-1}$ in the same way, finding a similar S $\subseteq$ $\{X_1, \ldots, X_{n-2}\}$, and so on. In the end, a DAG is produced and the BN semantics must hold by construction.

# Causal Intuitions

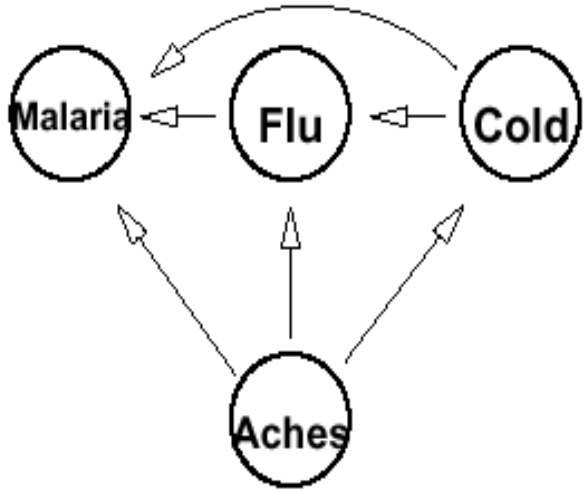
The construction of a BN is simple

- Works with arbitrary orderings of variable set
- But some orderings are much better than others
- Generally, if ordering/dependence structure reflects causal intuitions, we get a more compact BN

# Causal Intuitions



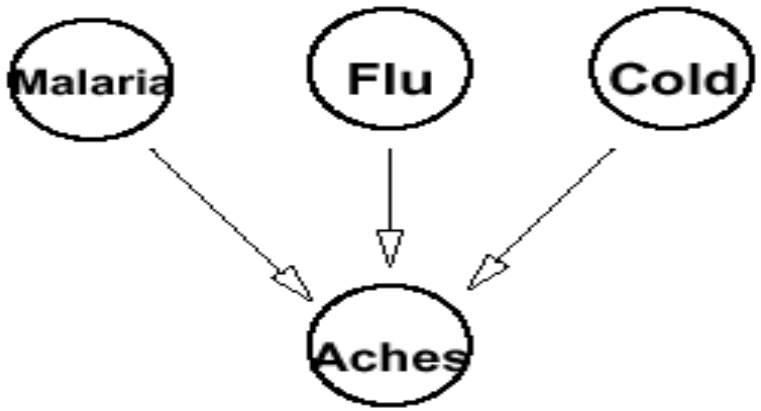

In this BN we've used the ordering Malaria, Cold, Flu, Aches to build BN for distribution P.
Note the variables can only have parents that come earlier in the ordering.



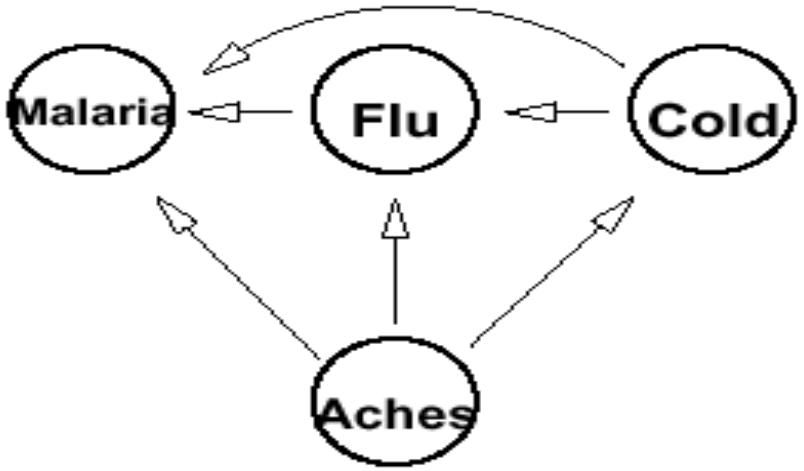

We could have used the ordering Aches, Cold, Flu, Malaria. Note that the CPTs would be different but this would still capture the same joint distribution.

# Compactness of the Representation

- In general, if each random Boolean variable is directly influenced by at most k others, then each CPT will be of size at most $2^k$. Thus, the entire network can be specified by $n2^k$ parameters.




1+1+1+8=11 numbers




1+2+4+8=15 numbers

# Testing Independence

We can use the structure of a BN to recognize variable independence given some evidence E.

**D-separation:** A set of variables, E, d-separates X and Y if it **blocks** every undirected path between X and Y.

X and Y are conditionally independent given E if E d-separates X and Y.

# Blocking

P is an undirected path from X to Y in BN. Let **E** be evidence set. **E** blocks path P iff there is some node in Z on the path such that

- **Case 1**: one arc on P goes into Z and one goes out of Z and Z in **E**, or
- **Case 2**: both arcs on P leave Z and Z in **E**, or
- **Case 3:** both arcs on P enter Z and neither Z, nor any of its descendents, are in **E**

# Blocking

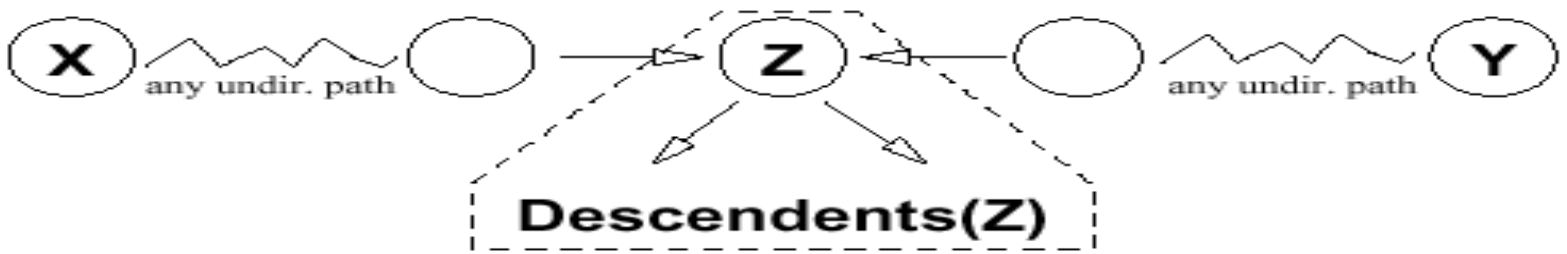
(1)


If Z in evidence, the path between X and Y blocked

(2)

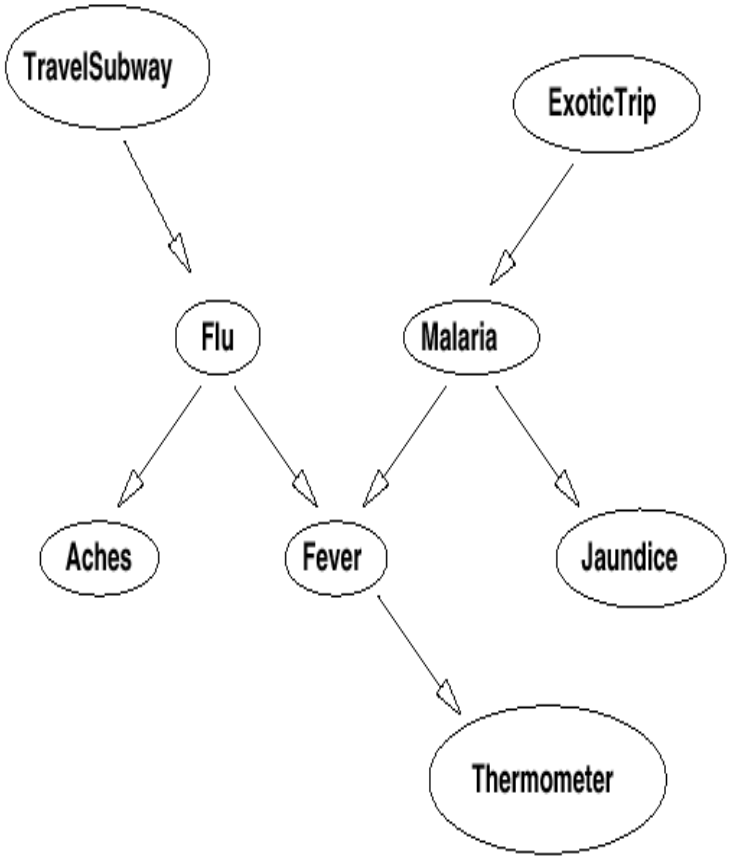

If Z in evidence, the path between X and Y blocked

(3)




If Z is ***not*** in evidence and***no*** descendent of Z is in evidence, then the path between X and Y is blocked

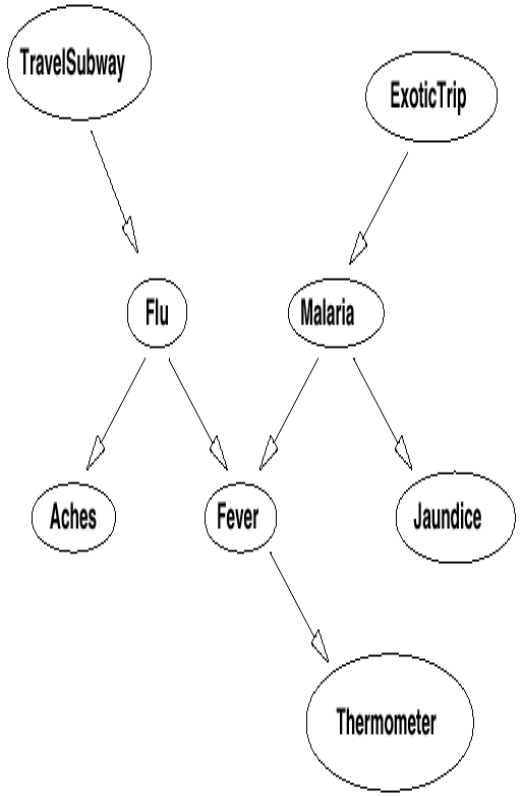
# Example




1. Subway and Thermometer?
2. Aches and Fever?
3. Aches and Thermometer?
4. Flu and Malaria?
5. Subway and ExoticTrip?

# Inference in Bayesian Networks

- Independence allows us to compute prior and posterior probabilities effectively.




P(J)=

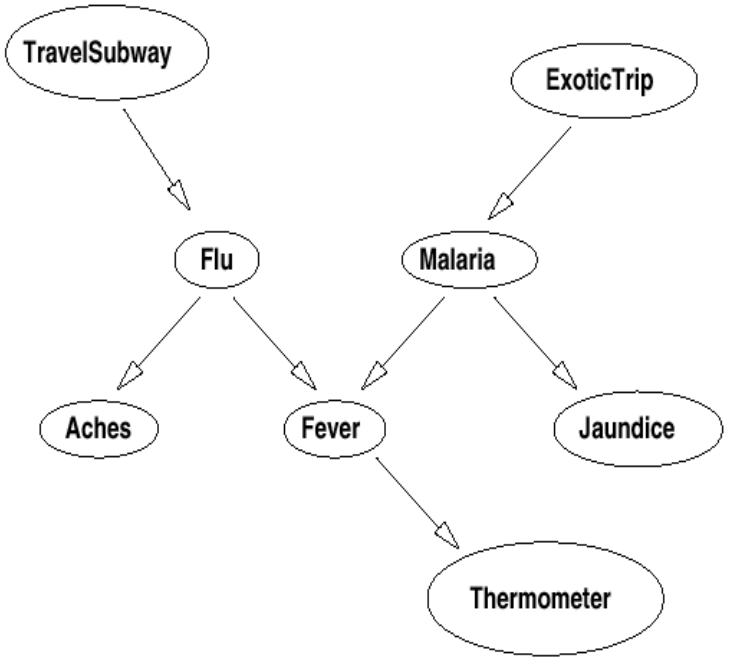
P(J|ET) =

P(Fev)=

P(Fev| TS=ts, Mal=~m) =

# Simple Backward Inference

When evidence is “downstream” of a query variable then you must use Bayes Rule.

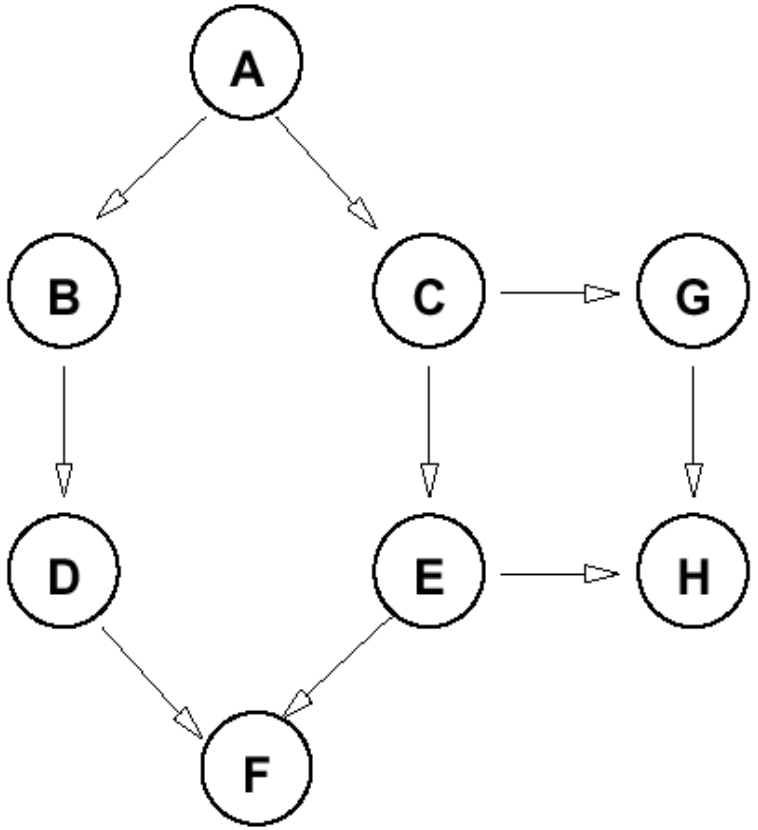



P(ET|J=j)=

P(ET|J=j, Fev=fev) =

# Variable Elimination

What about general BNs?




P(H|A,F)=?

# Variable Elimination

An inference procedure that simply applies the summing-out rule (marginalization) repeatedly

Exploits independence in network and distributes the sum inward

- Basically doing dynamic programming

# Factors

- A function $f(X_1,...,X_k)$ is called a **factor**
  - View this as a table of numbers, one for each instantiation of the variables
  - Exponential in k
- Each CPT in a BN is a factor
  - P(C|A,B) is a function of 3 variables, A, B, C
    - Represented as f(A,B,C)
- Notation: f(**X**,**Y**) denotes a factor over variables **X**∪**Y**
  - **X** and **Y** are sets of variables

# Product of Factors

- Let f(**X,Y**) and g(**Y,Z**) be two factors with variables **Y** in common
- The product of f and g, denoted by h=fg is
  - h(**X,Y,Z**)=f(**X,Y**) x g(**Y,Z**)

| f(A,B) | | g(B,C) | | h(A,B,C) | | | |
|---|---|---|---|---|---|---|---|
| ab | 0.9 | bc | 0.7 | abc | 0.63 | ab~c | 0.27 |
| a~b | 0.1 | b~c | 0.3 | a~bc | 0.08 | a~b~c | 0.02 |
| ~ab | 0.4 | ~bc | 0.8 | ~abc | 0.28 | ~ab~c | 0.12 |
| ~a~b | 0.6 | ~b~c | 0.2 | ~a~bc | 0.48 | ~a~b~c | 0.12 |

# Summing a Variable Out

- Let f(X,**Y**) be a factor with variable X and variable set **Y**
- We sum out variable X from f to produce h=$\sum_X$f where h(**Y**)=$\sum_{x \in Dom(X)}$ f(x,**Y**)

| f(A,B) | | h(B) | |
|---|---|---|---|
| ab | 0.9 | b | 1.3 |
| a~b | 0.1 | ~b | 0.7 |
| ~ab | 0.4 | | |
| ~a~b | 0.6 | | |

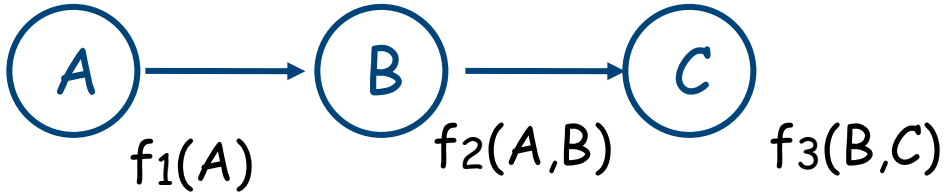
# Restrict a Factor

- Let f(X,**Y**) be a factor with variable X
- We restrict factor f to X=x by setting X to the value x and "deleting". Define $h=f_{X=x}$ as: h(**Y**)=f(x,**Y)**

| f(A,B) | | $h(B) = f_{A=a}$ | |
|---|---|---|---|
| ab | 0.9 | b | 0.9 |
| a~b | 0.1 | ~b | 0.1 |
| ~ab | 0.4 | | |
| ~a~b | 0.6 | | |

# Variable Elimination: No Evidence

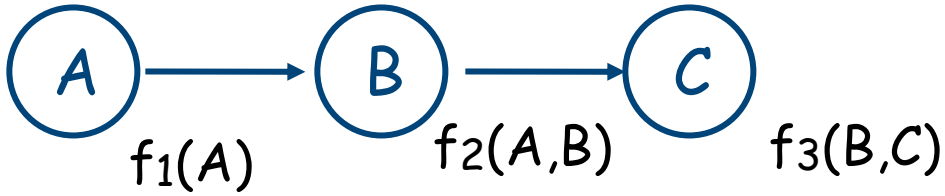
- Computing prior probability of query variable X can be seen as applying these operations on factors




- $P(C) = \Sigma_{A,B}\, P(C|B)\, P(B|A)\, P(A)$
  $= \Sigma_B\, P(C|B)\, \Sigma_A\, P(B|A)\, P(A)$
  $= \Sigma_B\, f_3(B,C)\, \Sigma_A\, f_2(A,B)\, f_1(A)$
  $= \Sigma_B\, f_3(B,C)\, f_4(B)$
  $= f_5(C)$

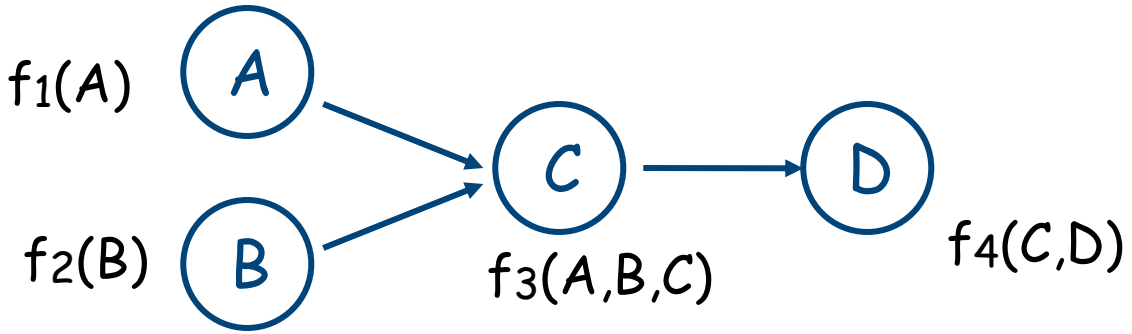
Define new factors: $f_4(B) = \Sigma_A\, f_2(A,B)\, f_1(A)$ and $f_5(C) = \Sigma_B\, f_3(B,C)\, f_4(B)$

# Variable Elimination: No Evidence




| $f_1(A)$ | | $f_2(A,B)$ | | $f_3(B,C)$ | | $f_4(B)$ | | $f_5(C)$ | |
|---|---|---|---|---|---|---|---|---|---|
| a | 0.9 | ab | 0.9 | bc | 0.7 | b | 0.85 | c | 0.625 |
| ~a | 0.1 | a~b | 0.1 | b~c | 0.3 | ~b | 0.15 | ~c | 0.375 |
| | | ~ab | 0.4 | ~bc | 0.2 | | | | |
| | | ~a~b | 0.6 | ~b~c | 0.8 | | | | |

# Variable Elimination: No Evidence




$P(D) = \Sigma_{A,B,C}\ P(D|C)\ P(C|B,A)\ P(B)\ P(A)$

$= \Sigma_C\ P(D|C)\ \Sigma_B\ P(B)\ \Sigma_A\ P(C|B,A)\ P(A)$

$= \Sigma_C\ f_4(C,D)\ \Sigma_B\ f_2(B)\ \Sigma_A\ f_3(A,B,C)\ f_1(A)$

$= \Sigma_C\ f_4(C,D)\ \Sigma_B\ f_2(B)\ f_5(B,C)$

$= \Sigma_C\ f_4(C,D)\ f_6(C)$

$= f_7(D)$

Define new factors: $f_5(B,C)$, $f_6(C)$, $f_7(D)$, in the obvious way

# VE Algorithm

Given query variable Q, remaining variables Z. Let F be the set of factors corresponding to CPTs for Q and Z.

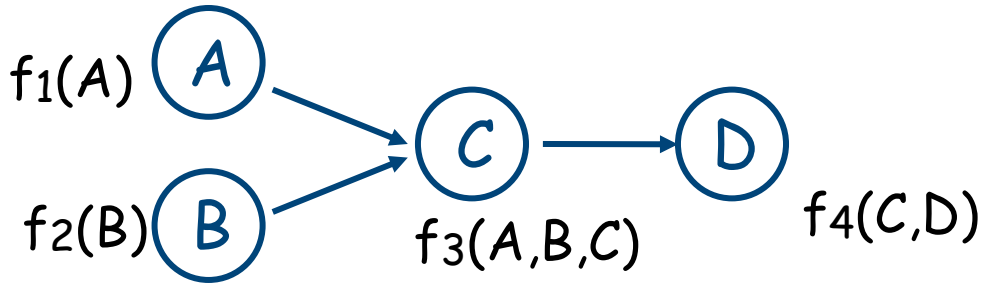
1. Choose an elimination ordering $Z_1, \ldots, Z_n$ of variables in **Z**.
2. For each $Z_j$ -- in the order given -- eliminate $Z_j \in$ **Z**
   as follows:
   (a) Compute new factor $g_j = \Sigma_{Zj} f_1 \times f_2 \times \ldots \times f_k$,
   where the $f_i$ are the factors in F that include $Z_j$
   (b) Remove the factors $f_i$ (that mention $Z_j$ ) from F
   and add new factor $g_j$ to F
3. The remaining factors refer only to the query variable Q.
   Take their product and normalize to produce P(Q)

# Revisiting the Example

**Factors:** $f_1(A)$ $f_2(B)$ $f_3(A,B,C)$ $f_4(C,D)$

**Query:** P(D)?

**Elim. Order:** A, B, C




Step 1: Add $f_5(B,C) = \sum_A f_3(A,B,C)\, f_1(A)$
Remove: $f_1(A)$, $f_3(A,B,C)$

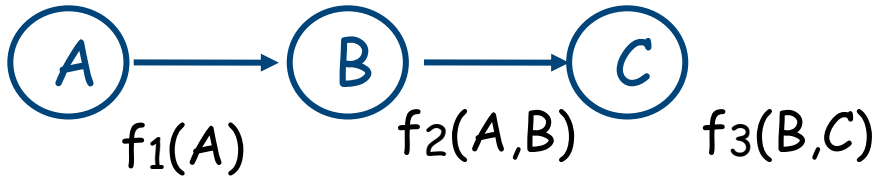
Step 2: Add $f_6(C) = \sum_B f_2(B)\, f_5(B,C)$
Remove: $f_2(B)$ , $f_5(B,C)$

Step 3: Add $f_7(D) = \sum_C f_4(C,D)\, f_6(C)$
Remove: $f_4(C,D)$, $f_6(C)$

Last factor $f_7(D)$ is (possibly unnormalized) probability P(D)

# VE: Evidence

Computing posterior of query variable given evidence is similar; suppose we observe C=c:




$$
\begin{aligned}
P(A|c) &= \alpha\, P(A)\, P(c|A) \\
&= \alpha\, P(A)\, \Sigma_B\, P(c|B)\, P(B|A) \\
&= \alpha\, f_1(A)\, \Sigma_B\, f_3(B,c)\, f_2(A,B) \\
&= \alpha\, f_1(A)\, \Sigma_B\, f_4(B)\, f_2(A,B) \\
&= \alpha\, f_1(A)\, f_5(A) \\
&= \alpha\, f_6(A)
\end{aligned}
$$

New factors: $f_4(B) = f_3(B,c)$; $f_5(A) = \Sigma_B\, f_2(A,B)\, f_4(B)$;

$f_6(A) = f_1(A)\, f_5(A)$

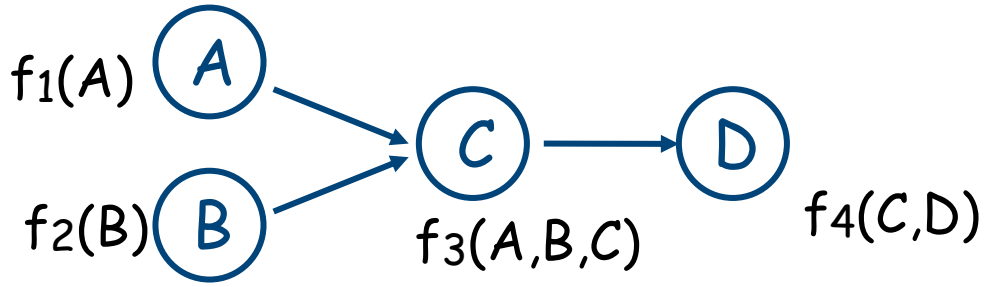
# VE Algorithm

Given query variable Q, evidence **E=e,** remaining variables Z. Let F be the set of factors corresponding to CPTs for Q and Z.

1. Replace each factor $f \in F$ that mentions a variable(s) in E with its restriction $f_{E=e}$ (somewhat abusing notation)
2. Choose an elimination ordering $Z_1, \ldots, Z_n$ of variables in **Z**.
3. Run VE as before.
4. Until remaining factors refer only to Q. Take their product and normalize to produce P(Q).

# Example

**Factors:** $f_1(A)$ $f_2(B)$
$f_3(A,B,C)$ $f_4(C,D)$
**Query:** P(A)?
*Evidence*: D = d
**Elim. Order:** C, B

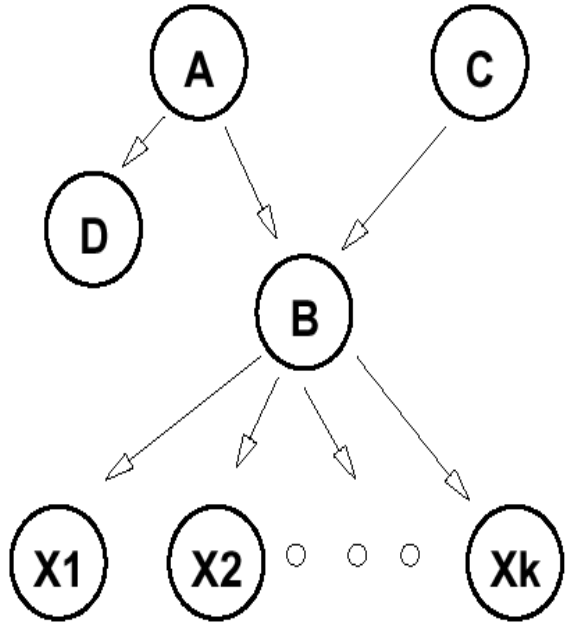


# VE Notes

- Each iteration eliminates one variable
  - No factor contains evidence variables after the initial restriction
  - Number of iterations is linear in number of variables
- Complexity is linear in number of variables but exponential in the size of the largest factor
  - Recall each factor is exponential in its number of variables
  - Can’t do better than size of the BN (since its original CPTs are part of the factor set)
  - But when we create new factors we might make significantly larger factors.

# Elimination Ordering: Polytrees

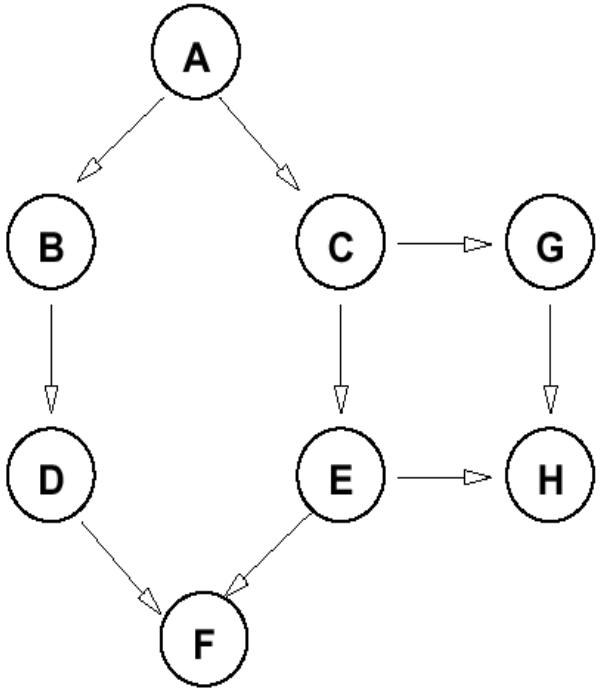
- Inference is linear in the size of the network
  - Ordering: Eliminate “singly-connected” nodes
  - Result: No factor ever grows larger than original factors




- What happens if we eliminated B first?

# Effects of Different Orderings

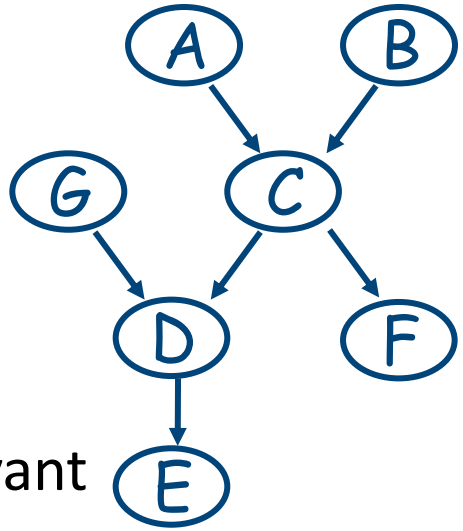
- Suppose query variable is D. Consider different orderings for this network
  - A,F,H,G,B,C,E: Good
  - E,C,A,B,G,H,F: Bad



# Relevance

- Certain variables have no impact on the query
  - In ABC network, computing P(A) with no evidence requires elimination of B and C
    - But when you sum out these variables, you compute a trivial factor
    - Eliminating C: $g(C)=\sum_C f(B,C)=\sum_C Pr(C|B)$.
    - Note that $P(c|b)+P(\sim c|b)=1$ and $P(c|\sim b)+P(\sim c|\sim b)=1$

- Can restrict ourselves to **relevant** variables
  - Given query Q, evidence E
  - Q is relevant
  - If any node Z is relevant, its parents are relevant
  - If $E \in E$ is a descendant of a relevant node, then E is relevant



# Where do BNs Come From?

- Handcrafted
  - Interact with domain expert to identify dependencies among variables (causal structure) and quantify local distributions (CPTs)
- Empirical data with human expertise used as a guide

- Recent emphasis on learning BNs directly from data