Bayesian Networks

Today we'll introduce Bayesian Networks.

This material is covered in chapters 13 and 14. Chapter 13 gives basic background on probability and Chapter 14 talks about Bayesian Networks. This includes methods for exact reasoning in Bayes Nets as well as approximate reasoning.

Benefits of Independence

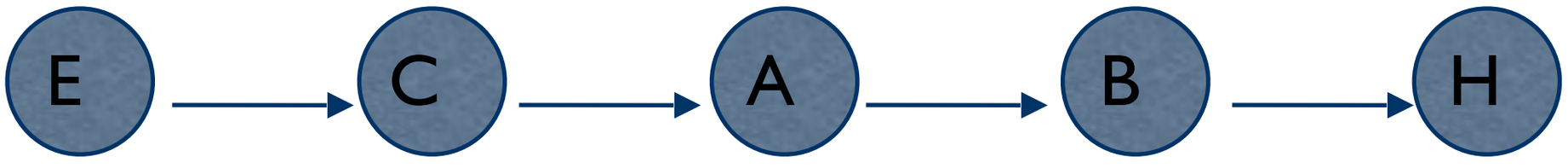
- Complete independence reduces both representation of joint and inference from $O(2^n)$ to $O(n)$!
- Unfortunately, such complete mutual independence is very rare. Most realistic domains do not exhibit this property.
- Fortunately, most domains do exhibit a fair amount of conditional independence. And we can exploit conditional independence for representation and inference as well.
- **Bayesian networks** do just this.

Exploiting Conditional Independence

Let's see what conditional independence buys us, computationally

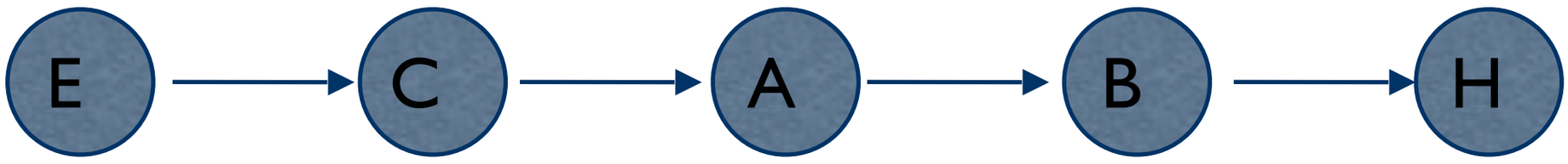
Consider a story:

"If Craig woke up too early (E is true), Craig probably needs coffee (C); if Craig needs coffee, he's likely angry (A). If he is angry, he has an increased chance of bursting a brain vessel (B). If he bursts a brain vessel, Craig is quite likely to be hospitalized (H)."



E – Craig woke too early A – Craig is angry H – Craig hospitalized
C – Craig needs coffee B – Craig burst a blood vessel

Cond'l Independence in our Story



If you knew E, C, A, or B, your assessment of P(H) would change.

- E.g., if any of these are seen to be true, you would increase P(H) and decrease P(~H).
- This means H is not independent of E, or C, or A, or B.

If you knew B, you'd be in good shape to evaluate P(H). You would not need to know the values of E, C, or A. The influence these factors have on H is mediated by B.

- Craig doesn't get sent to the hospital because he's angry, he gets sent because he's had an aneurysm.
- So H is independent of E, and C, and A, given B

Cond'l Independence in our Story

Similarly:

- B is independent of E, and C, given A
- A is independent of E, given C

This means that:

- P(H | B, {A,C,E}) = P(H|B)
  - i.e., for any subset of {A,C,E}, this relation holds
- P(B | A, {C,E}) = P(B | A)
- P(A | C, {E}) = P(A | C)
- P(C | E) and P(E) don't "simplify"

Cond'l Independence in our Story

By the chain rule (for any instantiation of H…E):

$$P(H,B,A,C,E) = P(H|B,A,C,E)\, P(B|A,C,E)\, P(A|C,E)\, P(C|E)\, P(E)$$

By our independence assumptions:

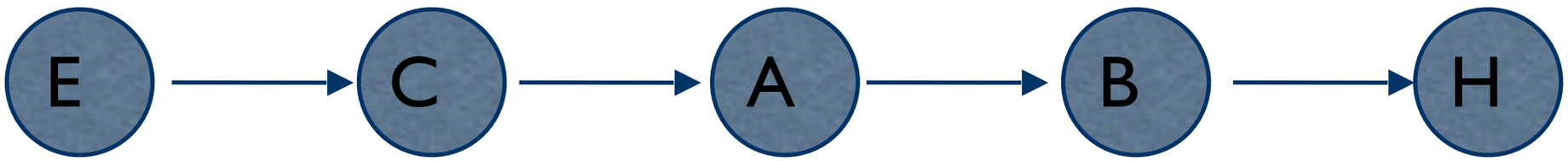
$$P(H,B,A,C,E) = P(H|B)\, P(B|A)\, P(A|C)\, P(C|E)\, P(E)$$

We can specify the full joint by specifying five local conditional distributions (joints): P(H|B); P(B|A); P(A|C); P(C|E); and P(E)

Adding Numbers

P(C|E)	= 0.9
P(~C|E)	= 0.1
P(C|~E)	= 0.5
P(~C|~E)	= 0.5

P(B|A)	= 0.2
P(~B|A)	= 0.8
P(B|~A)	= 0.1
P(~B|~A)	= 0.9



P(E)	= 0.7
P(~E)	= 0.3

P(A|C)	= 0.7
P(~A|C)	= 0.3
P(A|~C)	= 0.0
P(~A|~C)	= 1.0

P(H|B)	= 0.9
P(~H|B)	= 0.1
P(H|~B)	= 0.1
P(~H|~B)	= 0.9

Specifying the joint requires only 9 parameters (if we note that half of these are "1 minus" the others), instead of 31 for explicit representation

- That means inference is linear in the number of variables instead of exponential!
- Moreover, inference is linear generally if dependence has a chain structure

Making Inferences

Want to know P(A)? Proceed as follows:

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

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

These are all terms specified in our local distributions!

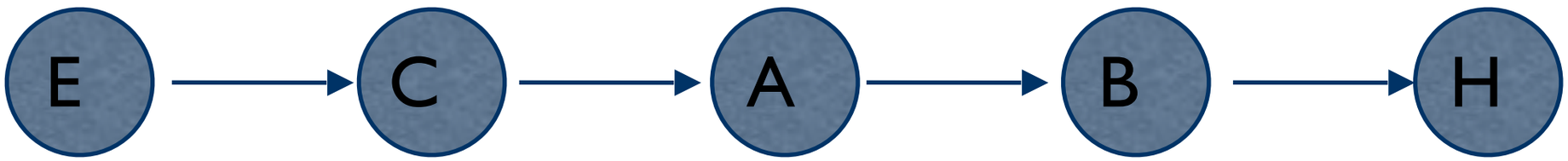
Making Inferences

P(E)	= 0.7
P(~E)	= 0.3

P(C|E)	= 0.9
P(~C|E)	= 0.1
P(C|~E)	= 0.5
P(~C|~E)	= 0.5

P(B|A)	= 0.2
P(~B|A)	= 0.8
P(B|~A)	= 0.1
P(~B|~A)	= 0.9

P(H|B)	= 0.9
P(~H|B)	= 0.1
P(H|~B)	= 0.1
P(~H|~B)	= 0.9



Computing P(A) in more concrete terms:

P(A|C)	= 0.7
P(~A|C)	= 0.3
P(A|~C)	= 0.0
P(~A|~C)	= 1.0

P(C) = P(C|E)P(E) + P(C|~E)P(~E) = 0.9 * 0.7 + 0.5 * 0.3 = 0.78

P(~C) = P(~C|E)P(E) + P(~C|~E)P(~E) = 0.22

P(~C) = 1 - P(C), as well

P(A) = P(A|C)P(C) + P(A|~C)P(~C) = 0.7 * 0.78 + 0.0 * 0.22 = 0.546

P(~A) = 1 – P(A) = 0.454

Bayesian Networks

- The structure we just described is a Bayesian network. A BN is a graphical representation of the direct dependencies over a set of variables, together with a set of conditional probability tables quantifying the strength of those influences.

- Bayes nets generalize the above ideas in very interesting ways, leading to effective means of representation and inference under uncertainty.

Bayesian Networks

A BN over variables $\{X_1, X_2, \ldots, X_n\}$ consists of:

- a directed acyclic graph (DAG) whose nodes are the variables
- a set of conditional probability tables (CPTs) that specify $P(X_i \mid \text{Parents}(X_i))$ for each X_i

Key notions (see text for defn's, all are intuitive):

- parents of a node: $\text{Par}(X_i)$
- children of node
- descendents of a node
- ancestors of a node
- family: set of nodes consisting of X_i and its parents

CPTs are defined over families in the BN

Another Bayesian Network

A: Aameri gives the lecture

S: It is sunny out

L: The lecturer arrives late

Assume that all instructors may arrive late in bad weather. Some instructors may be more likely to be late than others.

Another Bayesian Network

A: Aameri gives the lecture

S: It is sunny out

L: The lecturer arrives late

Assume that all instructors may arrive late in bad weather.
Some instructors may be more likely to be late than others.

We'll start by writing down what we know:
$P(S|A) = P(S)$, $P(S) = 0.3$, $P(A) = 0.5$

Lateness is not independent of the weather or the lecturer.

Another Bayesian Network

A: Aameri gives the lecture

S: It is sunny out

L: The lecturer arrives late

What does this mean? All instructors may arrive late in bad weather. Some instructors may be more likely to be late than others.

$P(S|A) = P(S), P(S) = 0.3, P(A) = 0.5$

We need to formulate $P(L|S,A)$ for all of the values of S and A.

Another Bayesian Network

A: Aameri gives the lecture

S: It is sunny out

L: The lecturer arrives late

$P(S|A) = P(S)$, $P(S) = 0.3$, $P(A) = 0.5$

$P(L|S,A) = 0.05$, $P(L|S,\sim A) = 0.1$, $P(L|\sim S,A) = 0.1$, $P(L|\sim S,\sim A) = 0.2$

Because of conditional independence, we only need 6 values to specify the full joint instead of 7. Conditional independence leads to computational savings!

Another Bayesian Network

A: Aameri gives the lecture

S: It is sunny out

L: The lecturer arrives late

P(S|A) = P(S), P(S) = 0.3, P(A) = 0.5

P(L|S,A) = 0.05, P(L|S,~A) = 0.1, P(L|~S,A) = 0.1, P(L|~S,~A) = 0.2

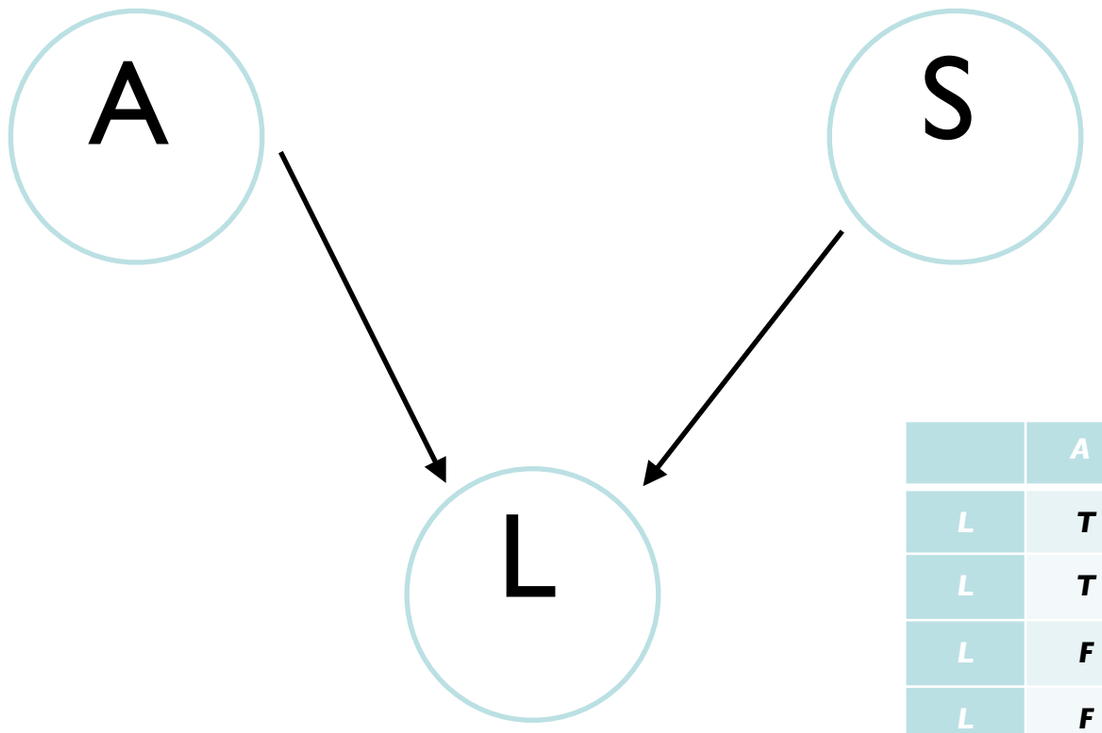
How can we calculate P(L,S,A)? Or P(L,~S,A)?

Drawing the Network

$$P(S|A) = P(S),\ P(S) = 0.3,\ P(A) = 0.5$$

$$P(L|S,A) = 0.05,\ P(L|S,\sim A) = 0.1,\ P(L|\sim S,A) = 0.1,\ P(L|\sim S,\sim A) = 0.2$$

A	T	0.5



S	T	0.3

	A	S	P(L=T|A,S)
L	T	T	0.05
L	T	F	0.1
L	F	T	0.1
L	F	F	0.2

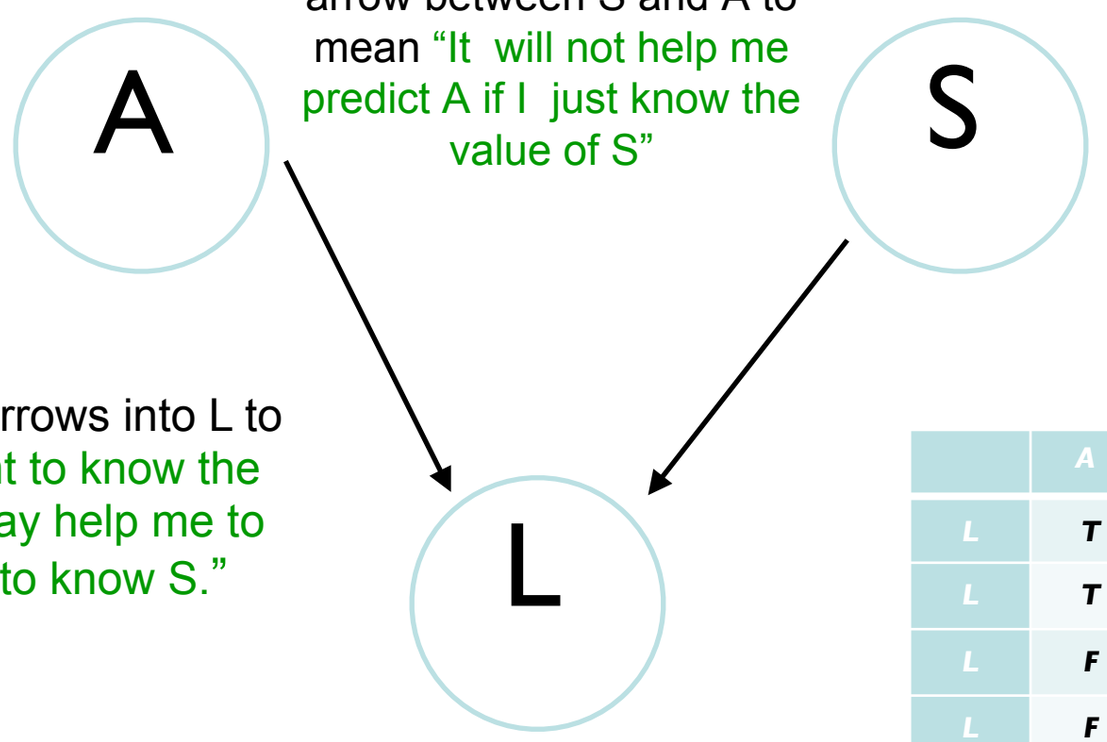
Drawing the Network

$P(S \mid M) = P(S)$
$P(S) = 0.3$
$P(M) = 0.6$

$P(L \mid M \wedge S) = 0.05$
$P(L \mid M \wedge \neg S) = 0.1$
$P(L \mid \neg M \wedge S) = 0.1$
$P(L \mid \neg M \wedge \neg S) = 0.2$

Read the absence of an arrow between S and A to mean "It will not help me predict A if I just know the value of S"

A	T	0.5



S	T	0.3

Read the two arrows into L to mean "If I want to know the value of L it may help me to know A and to know S."

	A	S	P(L=T|A,S)
L	T	T	0.05
L	T	F	0.1
L	F	T	0.1
L	F	F	0.2

Back to the network

Now let's suppose we have these three events:

A: Aameri gives the lecture

L: The lecturer arrives late

R : The lecturer concerns Reasoning with Bayes' Nets

And we know:

- Allin has a higher chance of being late than Aameri.
- Allin has a higher chance of giving lectures about reasoning with BNs

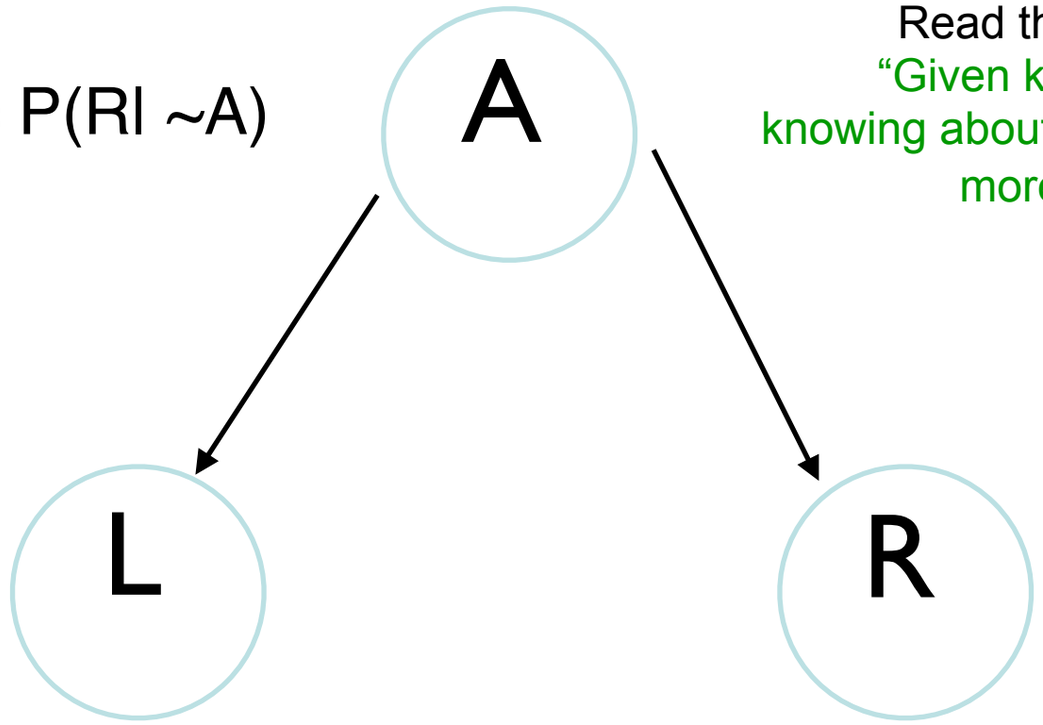
What kind of independences exist in our graph?

Back to the network

Once you know who the lecturer is, then whether they arrive late doesn't affect whether the lecture concerns Reasoning with Bayes' Nets, i.e.:

P(R|A,L) = P(R|A)

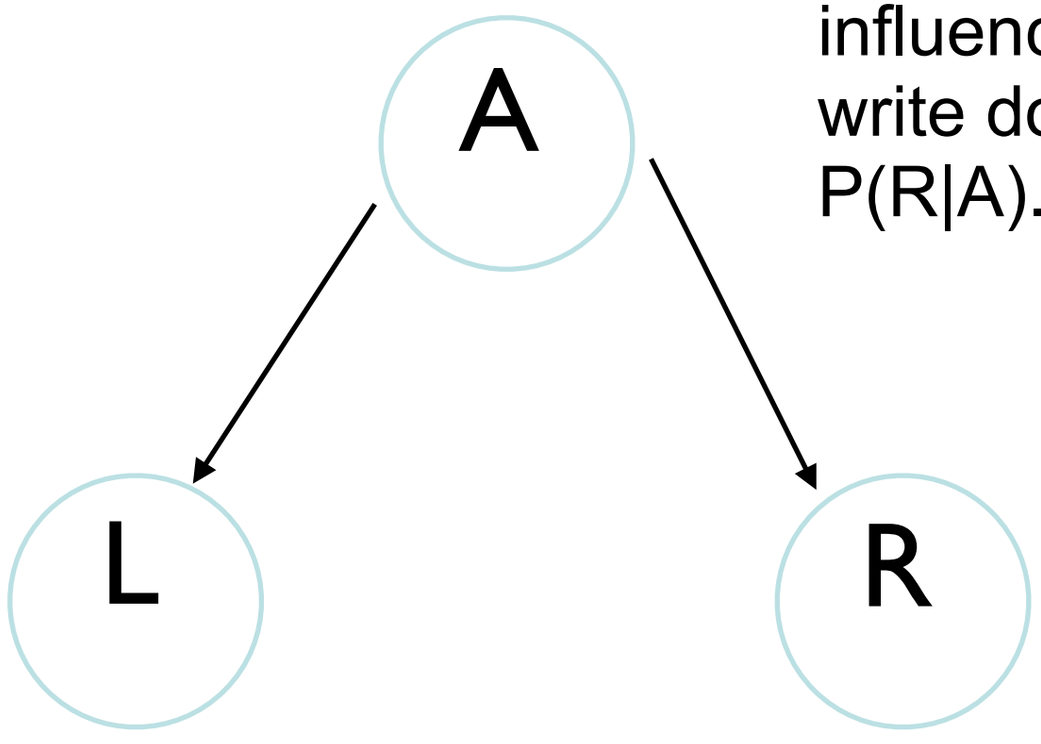
P(R|~A,L) = P(R|~A)



Read this diagram as "Given knowledge of A, knowing about L won't tell anything more about R."

The network reflects conditional independences

A	T	0.5



To specify CPTs, we first write down P(A). Then, because L and R are only directly influenced by M, we write down P(L|A) and P(R|A).

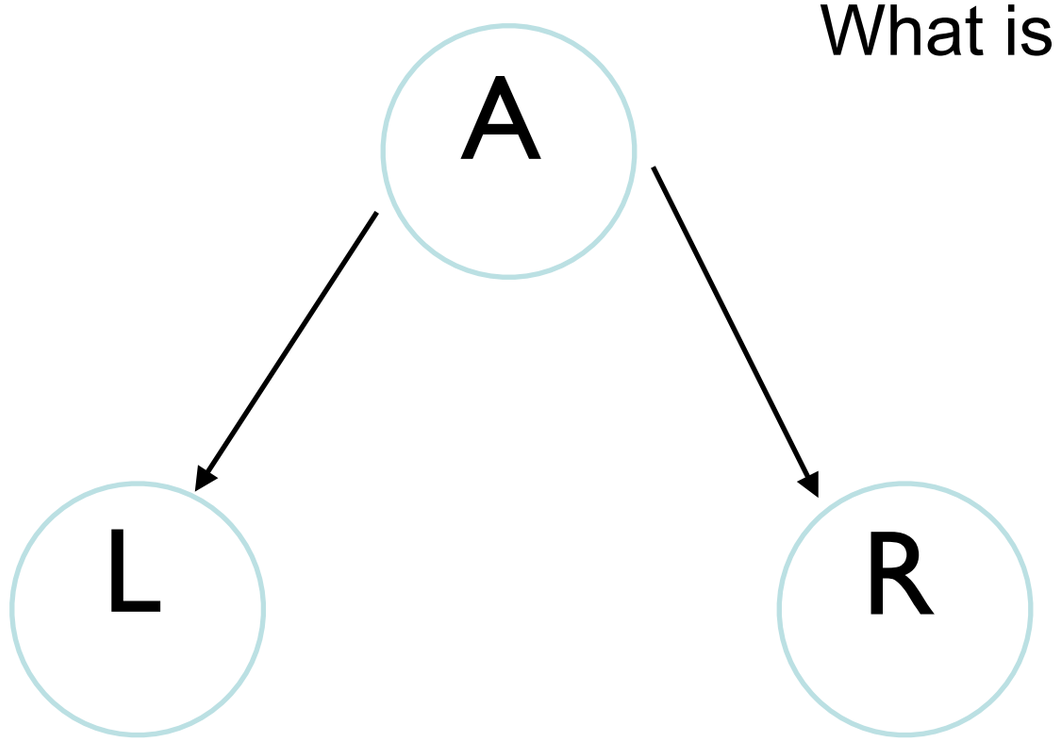
	A	P(L=T|A)
L	T	0.085
L	F	0.17

	A	P(R=T|A)
R	T	0.3
R	F	0.6

R is conditionally independent of L given A (and vice versa)

The network reflects conditional independences

A	T	0.6



	A	P(L=T|A)
L	T	0.085
L	F	0.17

	A	P(R=T|A)
R	T	0.3
R	F	0.6

What is P(L,R,A)?

What is P(~L,R,A)?

R is conditionally independent of L given A (and vice versa)

Building a Bayes Net

A: Aameri gives the lecture

L: The lecturer arrives late

R : The lecturer concerns Reasoning with Bayes' Nets

S: It is sunny out

T: The lecture starts before 15 minutes past the hour.

- T is only directly influenced by L (i.e. T is conditionally independent of R,A,S given L)
- L is only directly influenced by A and S (i.e. L is conditionally independent of R given A & S)
- R is only directly influenced by A (i.e. R is conditionally independent of L,S, given A)
- A and S are independent

Building a Bayes Net

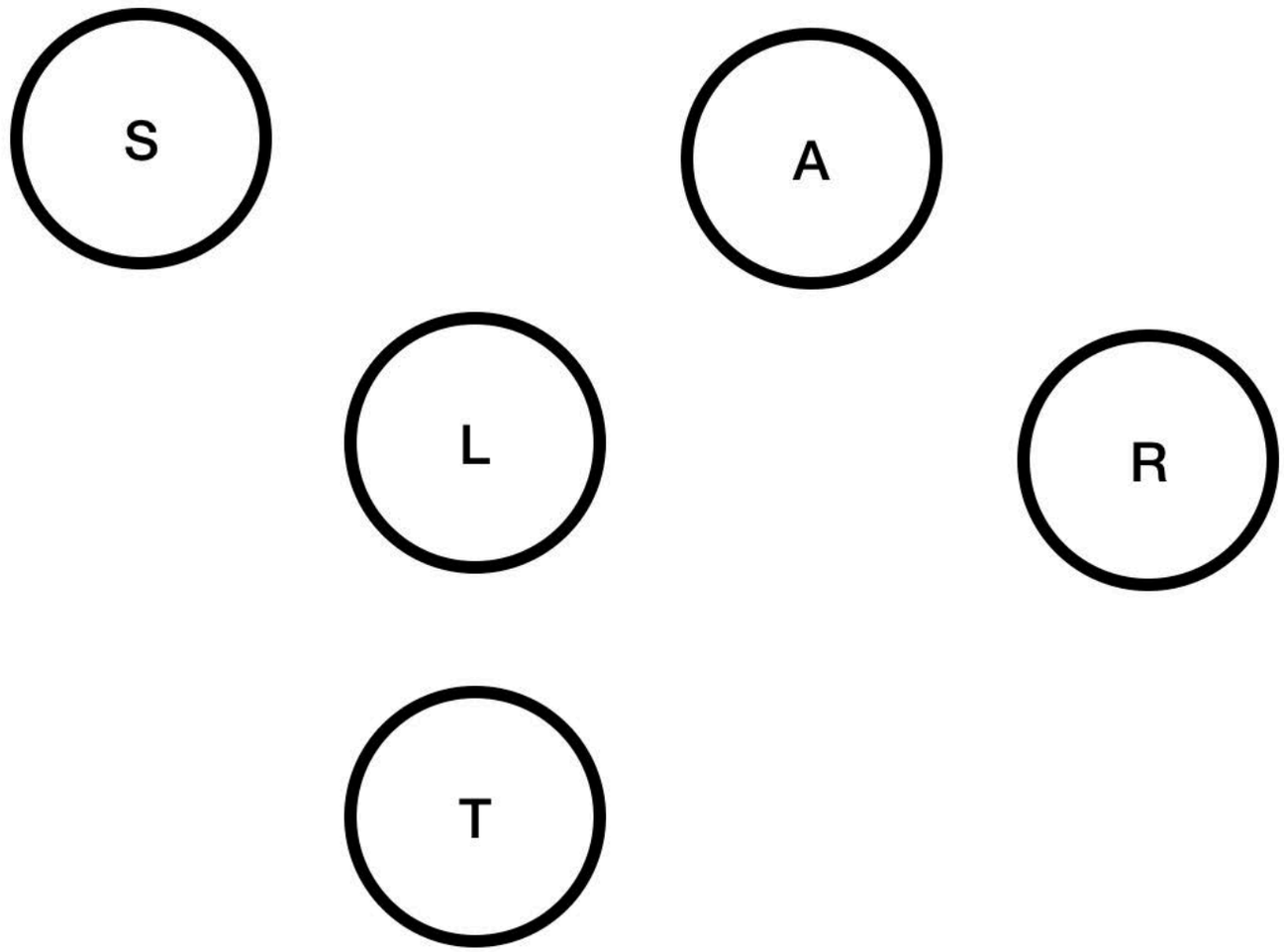
A: Aameri gives the lecture

L: The lecturer arrives late

R : The lecturer concerns Reasoning with Bayes' Nets

S: It is sunny out

T: Lecture starts < 15 minutes past.



Step One: Add variables

Building a Bayes Net

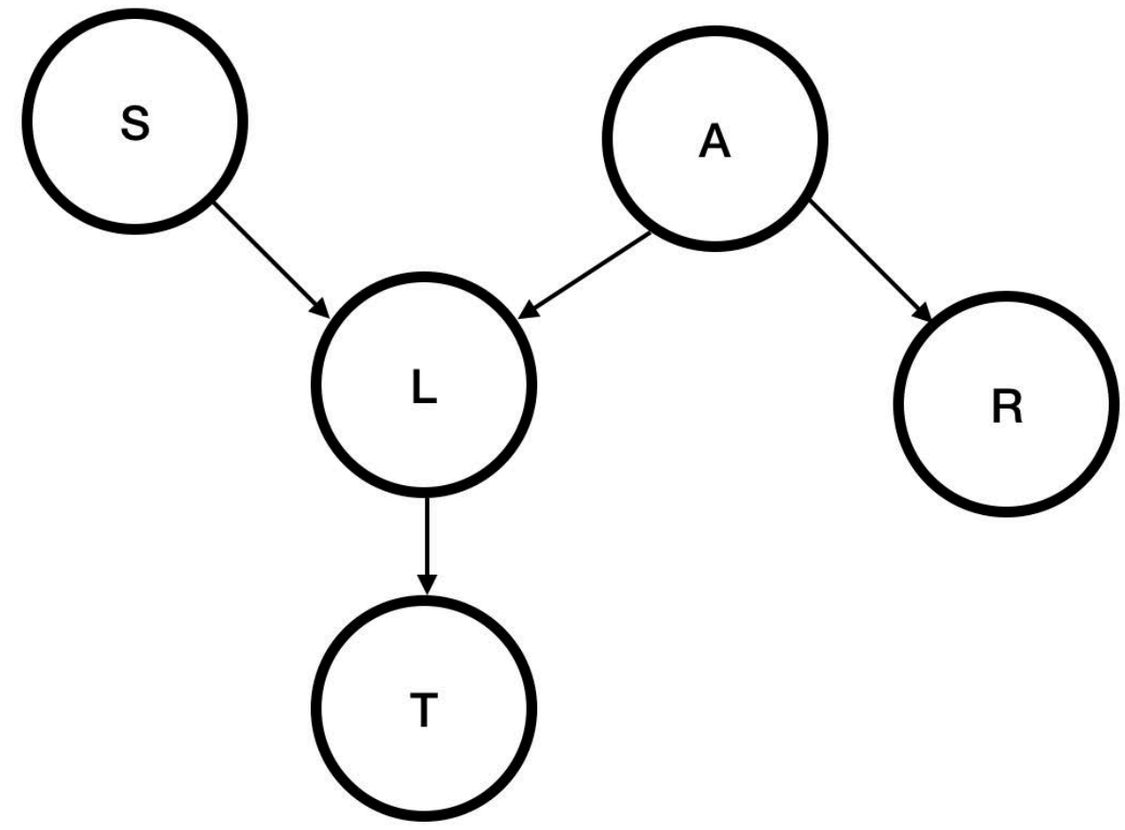
A: Aameri gives the lecture

L: The lecturer arrives late

R : The lecturer concerns Reasoning with Bayes' Nets

S: It is sunny out

T: Lecture starts < 15 minutes past.

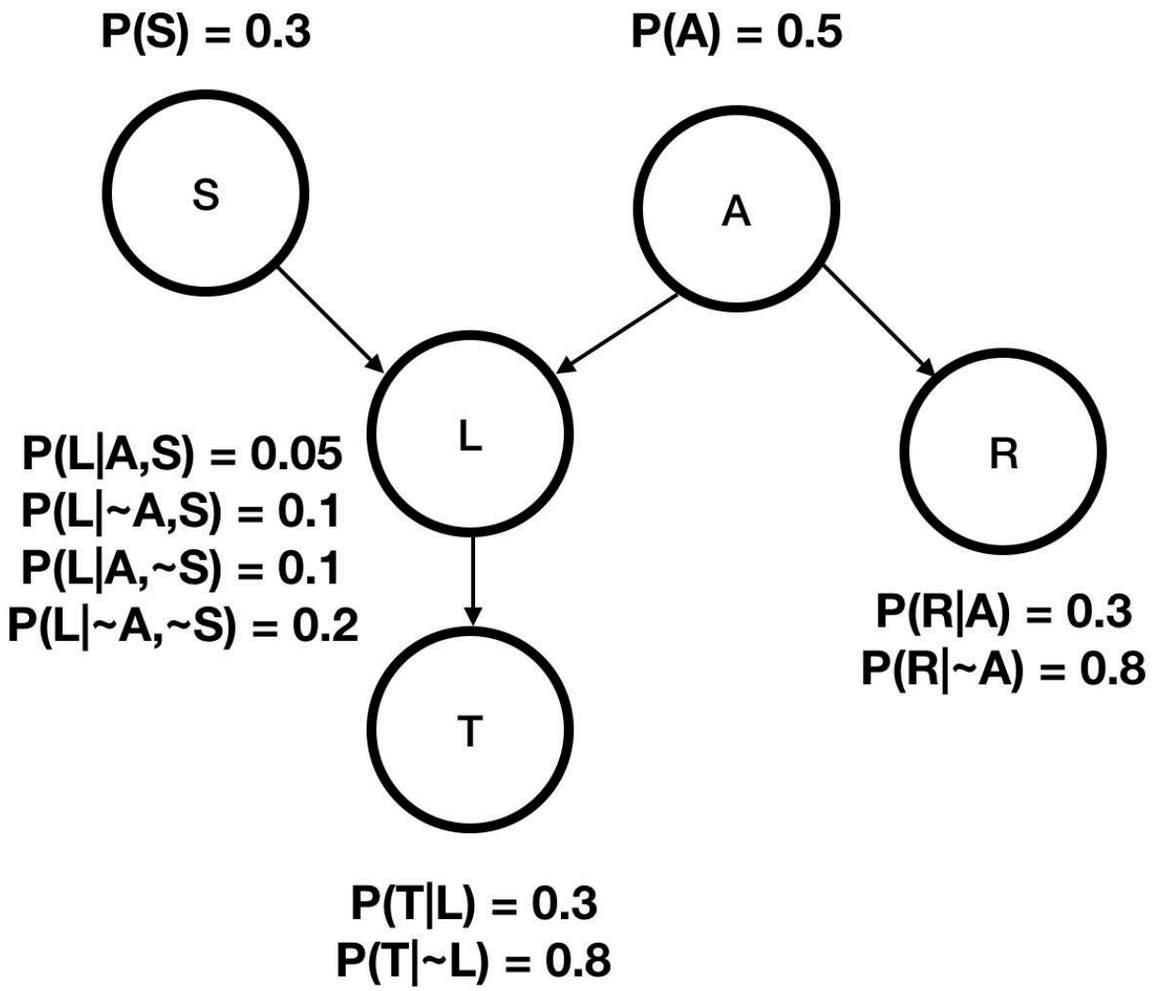


Step Two: add links.

The link structure must be acyclic.

If you assign node Y the parents $X_1, X_2, \ldots, X_n$, you are promising that, given $\{X_1, X_2, \ldots, X_n\}$, Y is conditionally independent of any other variable that's not a descendent of Y

Building a Bayes Net



A: Aameri gives the lecture

L: The lecturer arrives late

R : The lecturer concerns Reasoning with Bayes' Nets

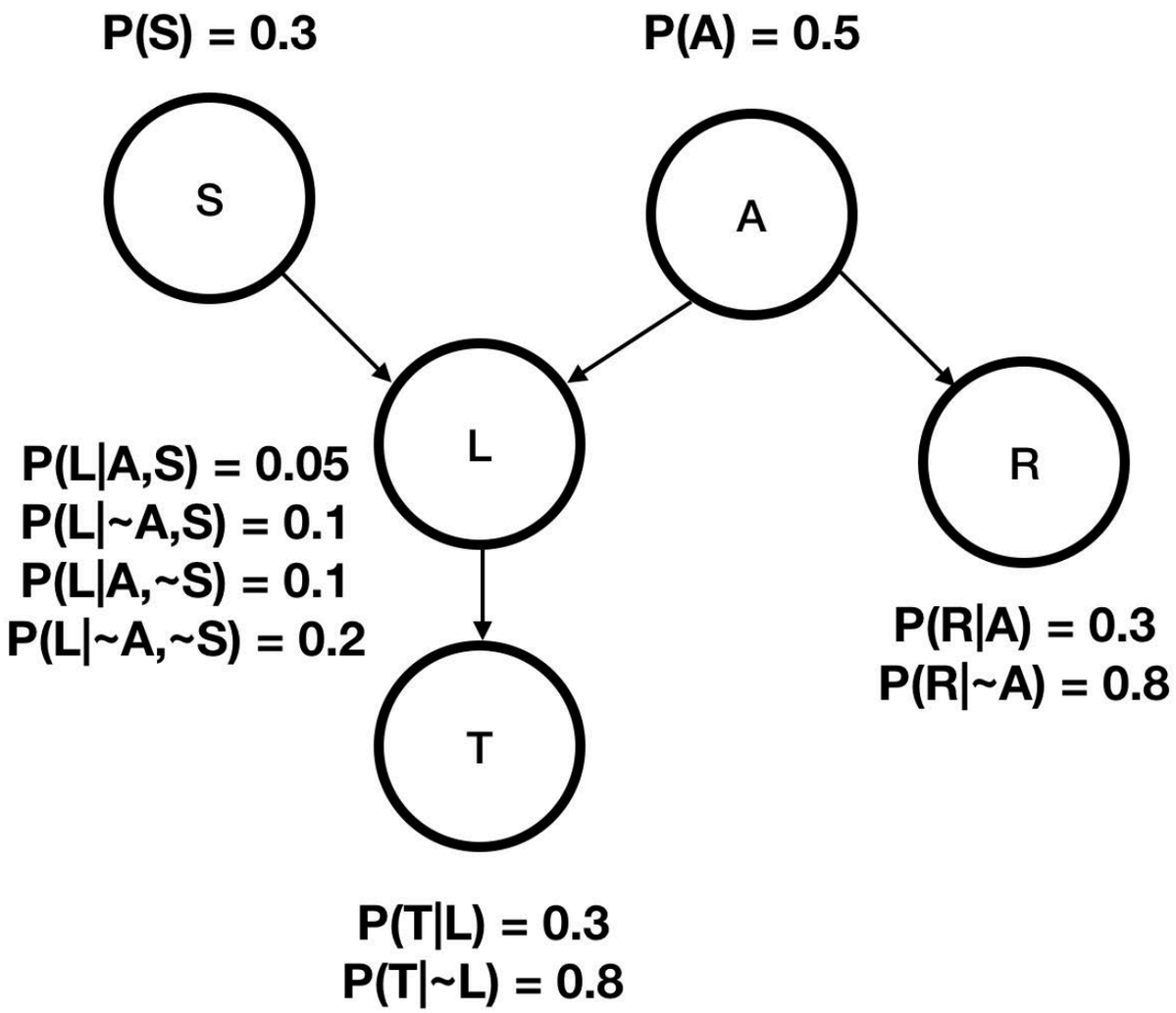
S: It is sunny out

T: Lecture starts < 15 minutes past.

Step Three: add a conditional probability table (CPT) for each node.

The table for X must define P(X|Parents) and for all combinations of the possible parent values.

Building a Bayes Net



A: Aameri gives the lecture

L: The lecturer arrives late

R : The lecturer concerns Reasoning with Bayes' Nets

S: It is sunny out

T: Lecture starts < 15 minutes past.

You can deduce many probability relations from a Bayes Net.

Note that variables that are not directly connected may still be correlated.

Building a Bayes Net

It is always possible to construct a Bayes net to represent any distribution over the variables X_1, $X_2,\ldots, X_n$, using any ordering of the variables.

Take any ordering of the variables (say, the order given). From the chain rule we obtain.

$$\Pr(X_1,\ldots,X_n) = \Pr(X_n|X_1,\ldots,X_{n-1})\Pr(X_{n-1}|X_1,\ldots,X_{n-2})\ldots\Pr(X_1)$$

Now for each X_i go through its conditioning set $X_1,\ldots,X_{i-1}$, and iteratively remove all variables X_j such that X_i is conditionally independent of X_j given the remaining variables. Do this until no more variables can be removed.

The final product specifies a Bayes net.

Causal Intuitions

- The BN can be constructed using an arbitrary ordering of the variables.
- However, some orderings will yield BN's with very large parent sets. This requires exponential space, and (as we will see later) exponential time to perform inference.
- Empirically, and conceptually, a good way to construct a BN is to use an ordering based on causality. This often yields a more natural and compact BN.

Causal Intuitions

Malaria, the flu and a cold all "cause" aches. So use the ordering that places causes before effects. Variables are Malaria (M), Flu (F), Cold (C), Aches (A):

$$P(M,F,C,A) = P(A|M,F,C)\ P(C|M,F)\ P(F|M)\ Pr(M)$$

Each of these disease affects the probability of aches, so the first conditional probability does not change.

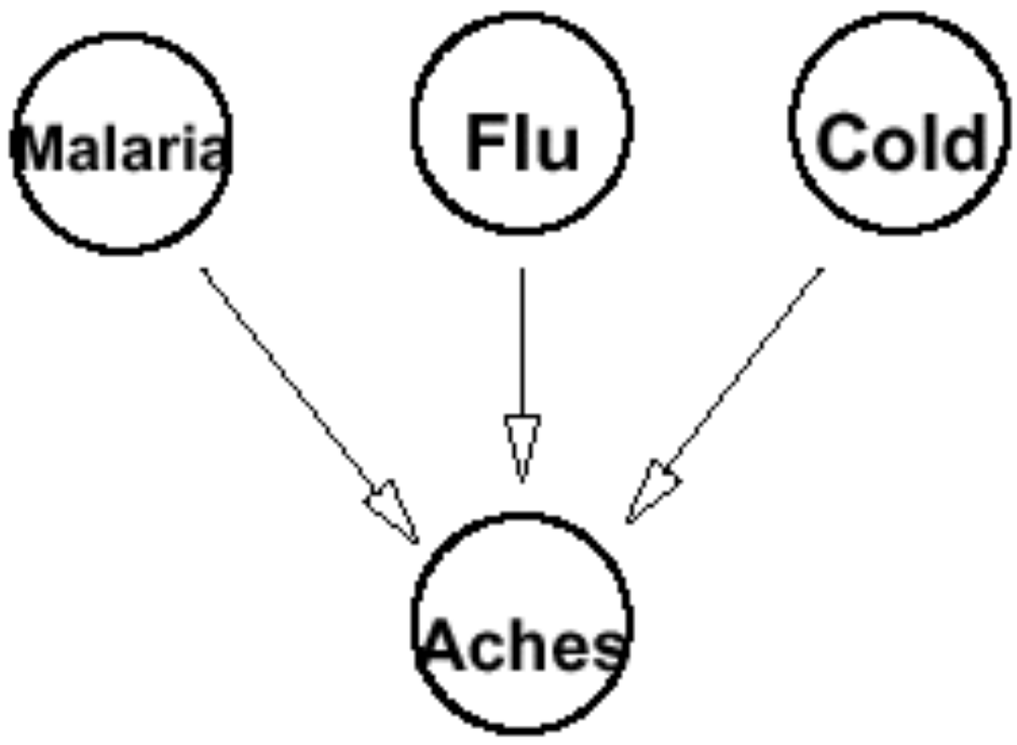
It is however reasonable to assume that these diseases are independent of each other: having or not having one does not change the probability of having the others.

So $P(C|M,F) = P(C)$ and $P(F|M) = P(F)$

Causal Intuitions

This yields a fairly simple Bayes net.

We only need one big CPT, involving the family of "Aches".



Causal Intuitions

Suppose we build the BN for distribution P using the opposite ordering, i.e., we use ordering Aches, Cold, Flu, Malaria

$$P(A,C,F,M) = P(M|A,C,F)\ P(F|A,C)\ P(C|A)\ P(A)$$

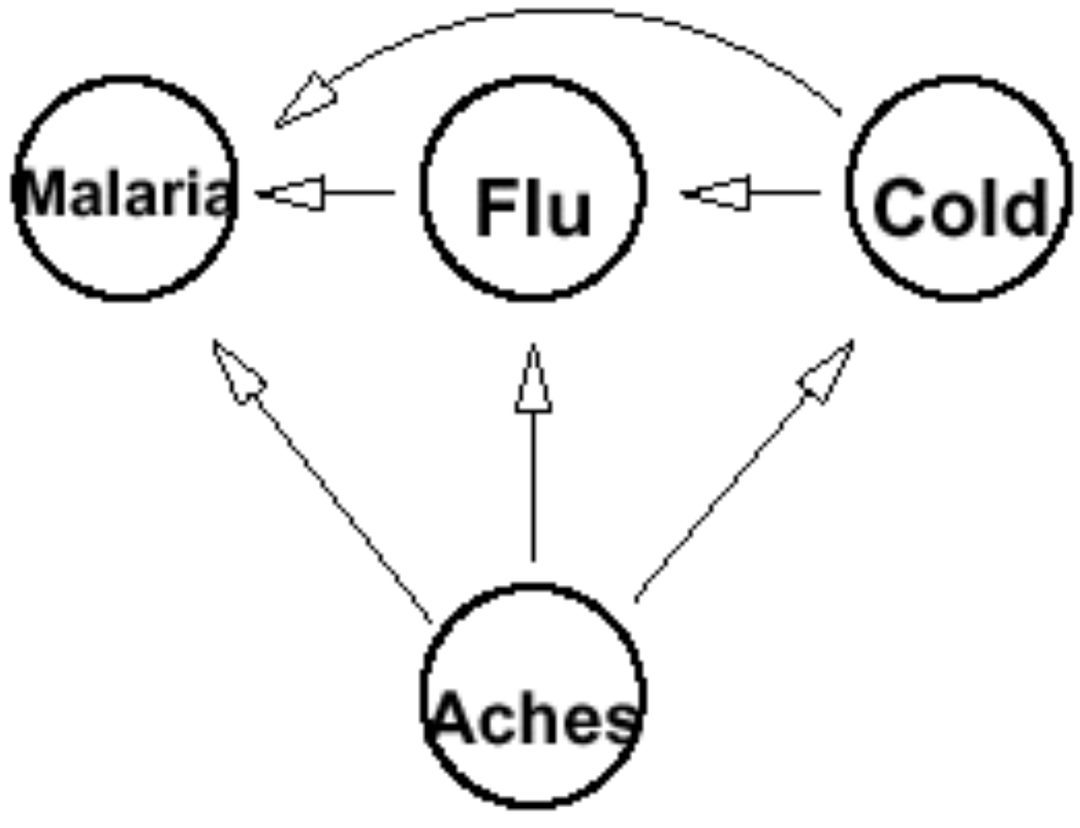
We can't reduce P(M|A,C,F). The probability of Malaria is clearly affected by knowing Aches. What about knowing Aches and Cold, or Aches and Cold and Flu?

Probability of Malaria is affected by both of these additional pieces of knowledge.

Knowing Cold and Flu lowers the probability that Aches are related to Malaria since they "explain away" the Aches!

Causal Intuitions

We obtain a much more complex Bayes net. In fact, we obtain no savings over explicitly representing the full joint distribution (i.e., representing the probability of every atomic event).



Bayes Net Example

I'm at work, neighbour John calls to say my alarm is ringing, but neighbour Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

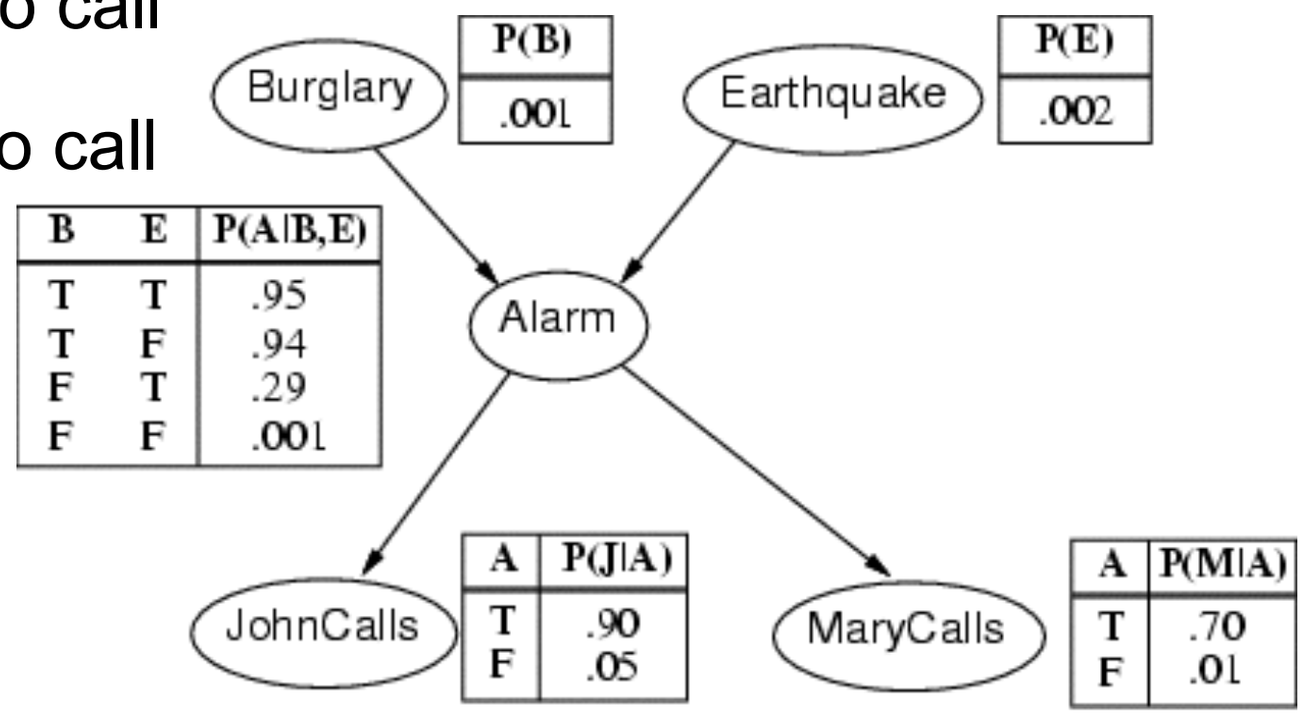
Variables: *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

The network topology reflects "causal" knowledge:

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

Burglary Example

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call



# of Params:1 + 1 + 4 + 2 + 2 = 10 (vs. 2^5-1 = 31)

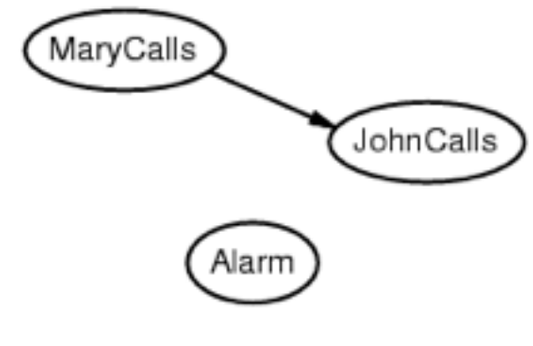
Burglary Example

Suppose we choose the ordering M, J, A, B, E

P(J | M) = P(J)?

Burglary Example

Suppose we choose the ordering M, J, A, B, E

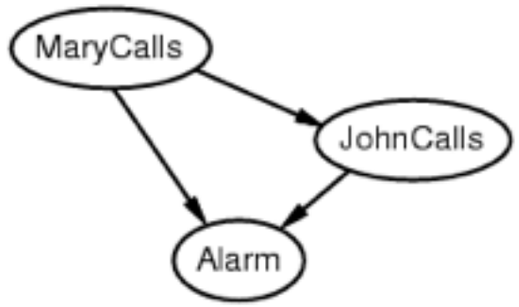


P(J | M) = P(J)? No

P(A | J, M) = P(A | J)? P(A | J, M) = P(A)?

Burglary Example

Suppose we choose the ordering M, J, A, B, E



P(J | M) = P(J)? No

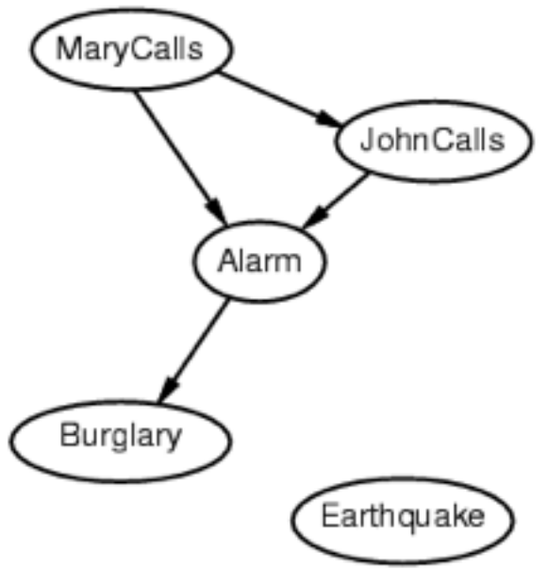
P(A | J, M) = P(A | J)? P(A | J, M) = P(A)? No

P(B | A, J, M) = P(B | A)?

P(B | A, J, M) = P(B)?

Burglary Example

Suppose we choose the ordering M, J, A, B, E



P(J | M) = P(J)? No

P(A | J, M) = P(A | J)? P(A | J, M) = P(A)? No

P(B | A, J, M) = P(B | A)? Yes

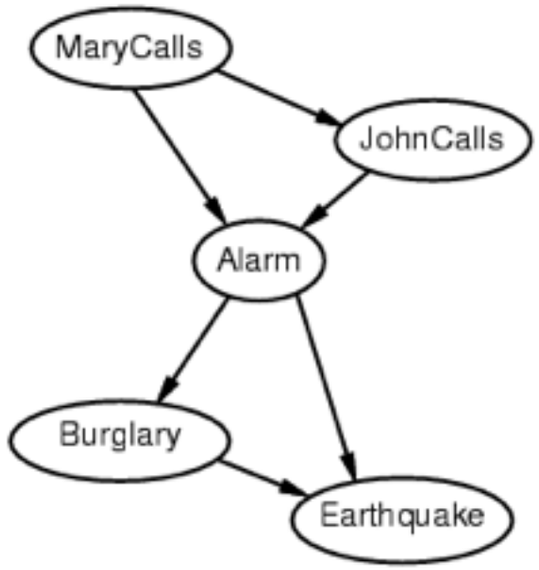
P(B | A, J, M) = P(B)? No

P(E | B, A ,J, M) = P(E | A)?

P(E | B, A, J, M) = P(E | A, B)?

Burglary Example

Suppose we choose the ordering M, J, A, B, E



P(J | M) = P(J)? No

P(A | J, M) = P(A | J)? P(A | J, M) = P(A)? No

P(B | A, J, M) = P(B | A)? Yes

P(B | A, J, M) = P(B)? No

P(E | B, A ,J, M) = P(E | A)? No

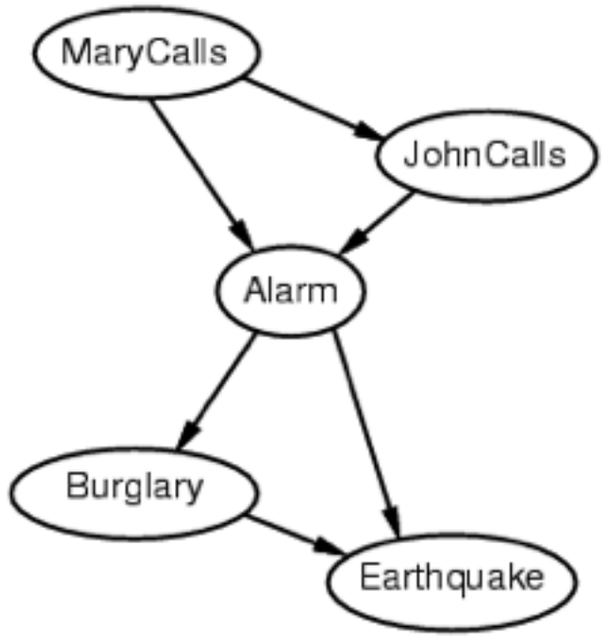
P(E | B, A, J, M) = P(E | A, B)? Yes

Burglary Example

Deciding conditional independence **is hard** in non-causal directions!

(Causal models and conditional independence seem hardwired in humans!)

Network is **less compact**: 1 + 2 + 4 + 2 + 4 = 13 numbers needed



Inference in Bayes Nets

Given a Bayes net

$P(X_1, X_2, \ldots, X_n) = P(X_n | P(Parents(X_n))) *$

$P(X_{n-1} | P(Parents(X_{n-1}))) * \ldots * P(X_1 | P(Parents(X_1)))$

And some evidence

$E = \{$a set of values for some of the variables$\}$

we want to compute the new probability distribution

$P(X_k | E)$

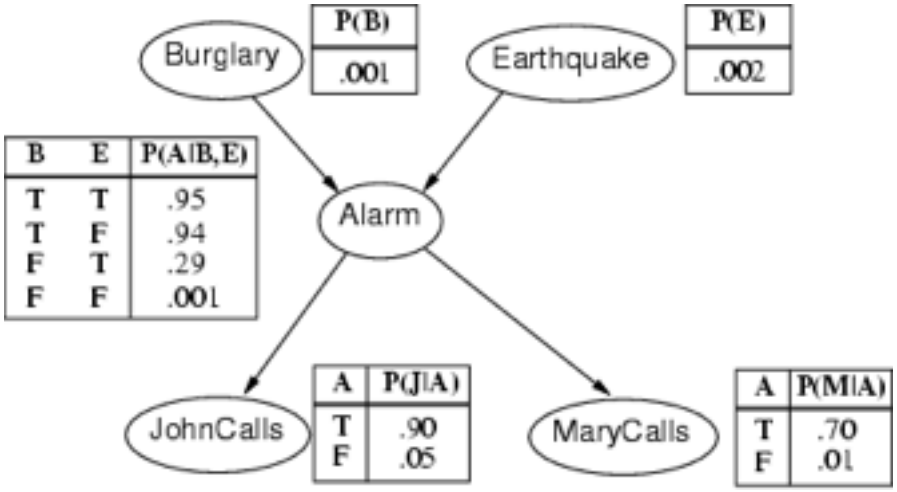
That is, we want to figure our

$P(X_k = d | E)$ for all $d \in Dom[X_k]$

Inference in Bayes Nets

- We might infer probability of different diseases given symptoms, or probability of hail storms given different metrological evidence, etc.
- In such cases getting a good estimate of the probability of the unknown event allows us to respond more effectively (gamble rationally)

Inference in Bayes Nets



In the Alarm example:

- P(Burglary,Earthquake, Alarm, JohnCalls, MaryCalls) =
  P(Earthquake) * P(Burglary) *
  P(Alarm|Earthquake,Burglary) *
  P(JohnCalls|Alarm) * P(MaryCalls|Alarm)
- We may want to infer things like
  P(Burglary=true| MaryCalls=false, JohnCalls=true)

Variable Elimination

If the network has a lot of variables, making exact inferences can be computationally hairy.

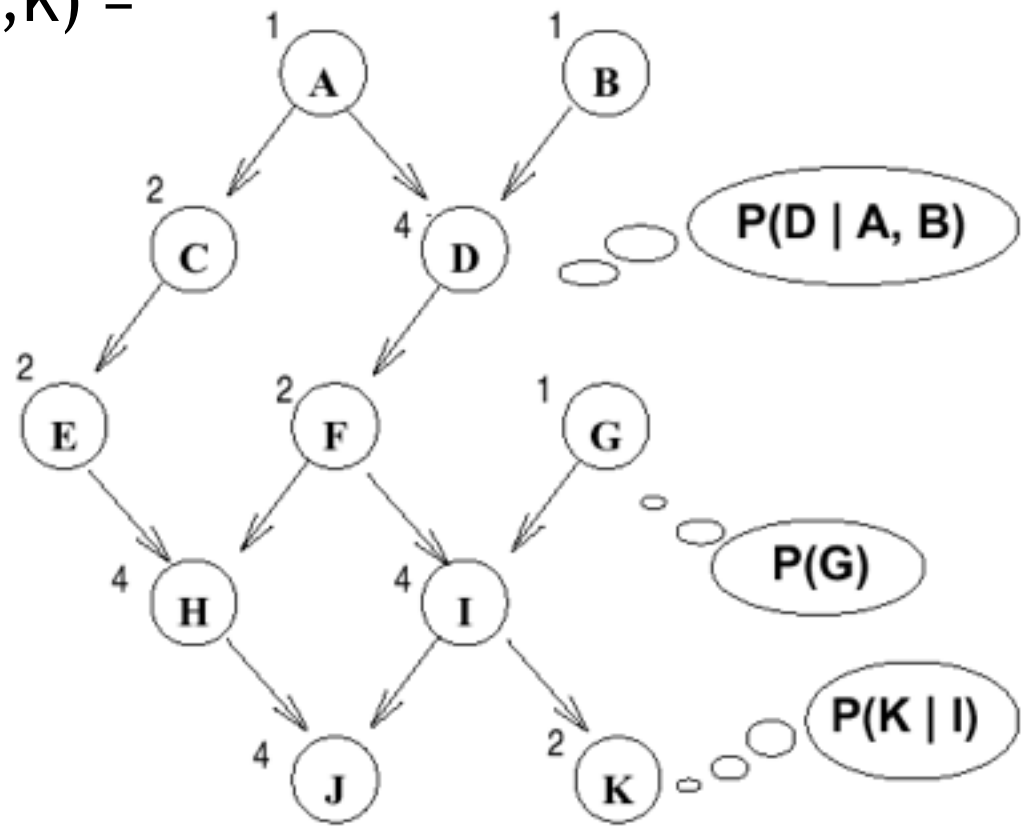
Variable elimination uses the product decomposition that defines a Bayes Net and the summing out rule to compute probabilities conditioned on evidence from information in the network (CPTs).

VE helps to reduce some of computation required to make exact inferences.

Example (Binary valued Variables)

P(A,B,C,D,E,F,G,H,I,J,K) =

P(A)
x P(B)
x P(C|A)
x P(D|A,B)
x P(E|C)
x P(F|D)
x P(G)
x P(H|E,F)
x P(I|F,G)
x P(J|H,I)
x P(K|I)



Example

P(A,B,C,D,E,F,G,H,I,J,K) =
P(A)P(B)P(C|A)P(D|A,B)P(E|C)P(F|D)P(G)P(H|E,F)
P(I|F,G)P(J|H,I)P(K|I)

Say that E (our evidence) = {H=true, I=false}, and we want to know

P(D|h,-i) (h: H is true, -h: H is false)

= P(d,h,-i)/N and P(-d,h,-i)/N where

N = P(h,-i) = P(d,h,-i) + P(-d,h,-i)

N can be considered a "normalizing" constant.

Example

First, we write as a sum for each value of D (i.e. D = d and D = -d)

$\sum_{A,B,C,E,F,G,J,K} P(A,B,C,d,E,F,h,-i,J,K) = P(d,h,-i)$

$\sum_{A,B,C,E,F,G,J,K} P(A,B,C,-d,E,F,h,-i,J,K) = P(-d,h,-i)$

We start by computing P(d,h,-i). Use Bayes Net product decomposition to rewrite the summation:

$$\sum_{A,B,C,E,F,G,J,K} P(A,B,C,d,E,F,h,-i,J,K)$$
$$= \sum_{A,B,C,E,F,G,J,K} P(A)P(B)P(C|A)P(d|A,B)P(E|C)$$
$$P(F|d)P(G)P(h|E,F)P(-i|F,G)P(J|h,-i)$$
$$P(K|-i)$$

Example

Next, rearrange summations so that we are not summing over variables that do not depend on the summed variable.

$$= \sum_A \sum_B \sum_C \sum_E \sum_F \sum_G \sum_J \sum_K P(A)P(B)P(C|A)P(d|A,B)P(E|C)\ P(F|d)P(G)P(h|E,F)P(-i|F,G)P(J|h,-i)\ P(K|-i)$$

$$= \sum_A P(A) \sum_B P(B) \sum_C P(C|A)P(d|A,B) \sum_E P(E|C) \sum_F P(F|d) \sum_G P(G)P(h|E,F)P(-i|F,G) \sum_J P(J|h,-i) \sum_K P(K|-i)$$

$$= \sum_A P(A) \sum_B P(B)\ P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d)\ P(h|E,F) \sum_G P(G)\ P(-i|F,G) \sum_J P(J|h,\ -i) \sum_K P(K|-i)$$

Example

Now start computing from the last summation to the first.

$$\sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d) P(h|E,F) \sum_G P(G) P(-i|F,G) \sum_J P(J|h,-i) \sum_K P(K|-i)$$

$$\sum_K P(K|-i) = P(k|-i) + P(-k|-i) = c_1$$

$$\sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d) P(h|E,F) \sum_G P(G) P(-i|F,G) \sum_J P(J|h,-i)\, c_1$$

Example

$$\sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C)$$
$$\sum_F P(F|d) P(h|E,F) \sum_G P(G) P(\text{-}i|F,G)$$
$$\sum_J P(J|h,\text{-}i)\ c_1$$

In this expression, variable K has been eliminated.
We can move c_1 to the front of the equation as it does not depend on other variables.

$$c_1 \sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C)$$
$$\sum_F P(F|d) P(h|E,F) \sum_G P(G) P(\text{-}i|F,G)$$
$$\sum_J P(J|h,\text{-}i)$$

Example

$$c_1 \sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d) P(h|E,F) \sum_G P(G) P(-i|F,G) \sum_J P(J|h,-i)$$

$$\sum_J P(J|h,-i) = \sum_J P(J|h,-i)$$
$$= (P(j|h,-i) + P(-j|h,-i))$$
$$= c_2$$

Now we've eliminated variable J.

$$c_1 c_2 \sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d) P(h|E,F) \sum_G P(G) P(-i|F,G)$$

Example

$c_1\, c_2 \sum_A P(A) \sum_B P(B)\, P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d)\, P(h|E,F) \sum_G P(G)\, P(-i|F,G)$

$\sum_G P(G)\, P(-i|F,G)$
$= (P(g)P(-i|F,g) + P(-g)P(-i|F,-g))$

P(-i|F,g) depends on the value of F, so this is not a single number.

Instead, its value depends on the assignment to F. But once F is fixed as f or -f, the value of P(-i|F,g) is also fixed.

Example

$c_1 c_2 \sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d) P(h|E,F) \sum_G P(G) P(-i|F,G)$

$\sum_G P(G) P(-i|F,G)$
$= (P(g)P(-i|F,g) + P(-g)P(-i|F,-g))$

To manage, we introduce a function to represent this sum:

f1(F) = P(g)P(-i|F,g) + P(-g)P(-i|F,-g)

We must store these fixed numbers to represent the function:

f1(f) = P(g)P(-i|f,g) + P(-g)P(-i|f,-g)
f1(-f) = P(g)P(-i|-f,g) + P(-g)P(-i|-f,-g)

Example

$$c_1 c_2 \sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d) P(h|E,F) f1(F)$$

$$\sum_F P(F|d) P(h|E,F) f1(F) = P(f|d) P(h|E,f) f1(f) + P(-f|d) P(h|E,-f) f1(-f)$$

This sum depends on E, so we introduce a function of E

$$f2(E) = P(f|d) P(h|E,f) f1(f) + P(-f|d) P(h|E,-f) f1(-f)$$

Again store two fixed values, f2(e) and f2(-e), to represent this function.

Variable Elimination (VE)

- We can continue this way, eliminating one variable after another by introducing functions. At the end of the process we will sum out A and be left with P(d,h,-i).
- We can repeat the process to compute P(-d,h,-i).
- Remember the normalizing constant (N) is P(d,h,-i) + P(-d,h,-i). If we normalize the numbers (ensure they sum to one) we will have P(d|h,-i) and P(d|h,i).
- Alternately, we could consider D to be the last variable in the elimination process, and end our process when by computing f(D). The fixed values, f(d) and f(-d), will be the values we want as they correspond to P(d|h,-i) and P(d|h,i).

Variable Elimination (VE)

- This process is called variable elimination (VE)
- VE computes intermediate functions and stores values that we reuse many times during computations.
- In this way variable elimination is a form of dynamic programming. Dynamic programming is a technique that stores solutions to sub-computations in order to avoid repeating them many times over.

Relevant (return to this later)

Note that in the sum

$\sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d) P(h|E,F) \sum_G P(G) P(-i|F,G) \sum_J P(J|h,-i) \sum_K P(K|-i)$

$\sum_K P(K|-i) = 1$ (Why?)

This means $\sum_J P(J|h,-i) \sum_K P(K|-i) = \sum_J P(J|h,-i)$

By the same reasoning, $\sum_J P(J|h,-i) = 1$.

So we can, in theory, drop these last two terms from the computation. The variables J and K are not relevant given our query D and evidence (–i and –h). But for now we will keep the terms; we will revisit relevance later.

Variable Elimination (VE)

- In general, each stage VE sums out the innermost variable, creating a function over any variables in that sum.
- Each function is represented as a table of numbers: one number for each different instantiation of the variables in the sum.
- The size of the tables is exponential in the number of variables that appear in the sum, e.g.,

$$\sum_F P(F|D)\ P(h|E,F)\ f1(F)$$

depends on the value of D and E, thus we will obtain |Dom[D]|*|Dom[E]| different numbers in the resulting table.

Factors

- We call the tables of values computed by VE factors.
- Note that the original probabilities that appear in the summation, e.g., P(C|A), are also tables of values (one value for each instantiation of C and A).
- Thus we also call the original CPTs factors.
- Each factor is a function of some variables, e.g., P(C|A) = f(A,C): it maps each value of its arguments onto a number.
  - A tabular representation is exponential in the number of variables in the factor.

Operations on Factors

- If we examine the summation process we will see that various operations repeatedly occur on factors.

- Notation: $f(\underline{X},\underline{Y})$ denotes a factor over the variables $\underline{X}$ and $\underline{Y}$ (where $\underline{X}$ and $\underline{Y}$ are sets of variables)

The Product of Two Factors

- Let f(X,Y) & g(Y,Z) be two factors with variables Y in common
- The *product* of f and g, denoted h = f x g (or sometimes just h = fg), is defined as:

$$h(X,Y,Z) = f(X,Y) \times g(Y,Z)$$

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 of a Factor

- Let **f(X,Y)** be a factor
- We can *sum out* variable X from f to produce a new factor h = Σ_X f, which is defined: **h(Y)** = $\Sigma_{x \in Dom(X)}$ **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		

Restricting a Factor

- Let **f(X,Y)** be a factor
- We can *restrict* factor f *to* X = a by setting X to the value x and "deleting" incompatible elements of f's domain .

Define $h = f_{X=a}$ as: **h(Y) = f(a,Y)**

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

Variable Elimination the Algorithm

Given query var **Q**, evidence vars **E** (variables observed to have values **e**), and remaining vars **Z**. Let **F** be factors in original CPTs.

1. Replace each factor $f \in \mathbf{F}$ that mentions a variable(s) in **E** with its restriction $f_{\mathbf{E}=\mathbf{e}}$ (this might yield a "constant" factor)
2. For each $\mathbf{Z}_j$ - in the order given - eliminate $\mathbf{Z}_j \in \mathbf{Z}$ as follows:
   (a) Compute new factor $g_j = \sum_{Z_j} f_1 \times f_2 \times \ldots \times f_k$, where the f_i are the factors in **F** that include $\mathbf{Z}_j$
   (b) Remove the factors f_i (that mention $\mathbf{Z}_j$) from **F** and add new factor g_j to **F**
3. The remaining factors at the end of this process will refer only to the query variable **Q**. Take their product and normalize to produce **P(Q|E)**.

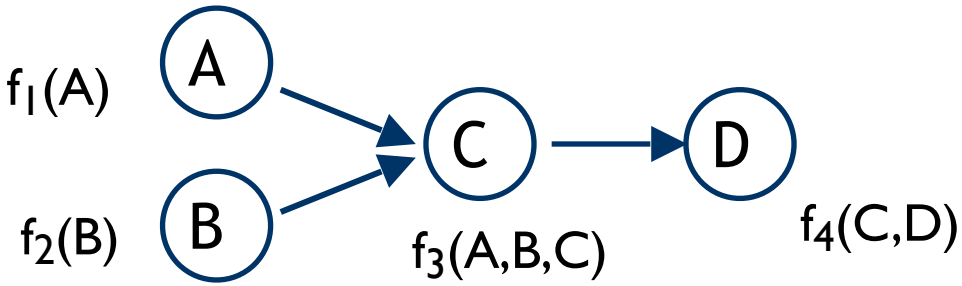
VE: Example

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

Query: P(A)?

Evidence: D = d

Elimination Order: C, B



Step 1 (Restriction): Replace $f_4(C,D)$ with $f_5(C) = f_4(C,d)$

Step 2 (Eliminate C): Compute & add $f_6(A,B) = \Sigma_C\ f_5(C)\ f_3(A,B,C)$ to list of factors.

Remove: $f_3(A,B,C)$, $f_5(C)$

Step 3 (Eliminate B): Compute & add $f_7(A) = \Sigma_B\ f_6(A,B)\ f_2(B)$ to list of factors.

Remove: $f_6(A,B)$, $f_2(B)$

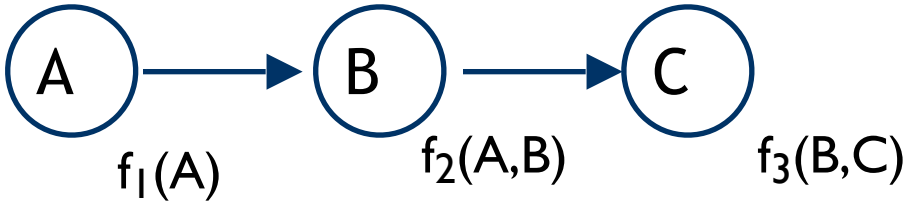
Step 4 (Normalize Final Factors): $f_7(A)$, $f_1(A)$. The product $f_1(A) \times f_7(A)$ is (un-normalized) posterior for A. So we normalize:

$P(A|d) = \alpha\ f_1(A) \times f_7(A)$

where $\alpha = 1/\sum_A f_1(A)f_7(A)$

Numeric Example

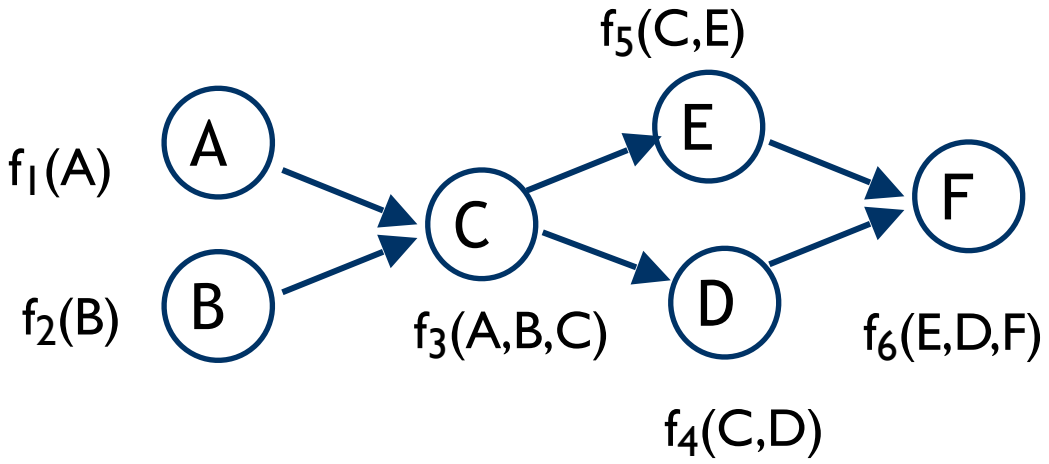
Here's an example with some . We want P(C) given no evidence.



$f_1(A)$		$f_2(A,B)$		$f_3(B,C)$		$f_4(B)$ Σ_A $f_2(A,B)f_1(A)$		$f_5(C)$ Σ_B $f_3(B,C)$ $f_4(B)$	
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				

VE: Buckets as a Notational Device

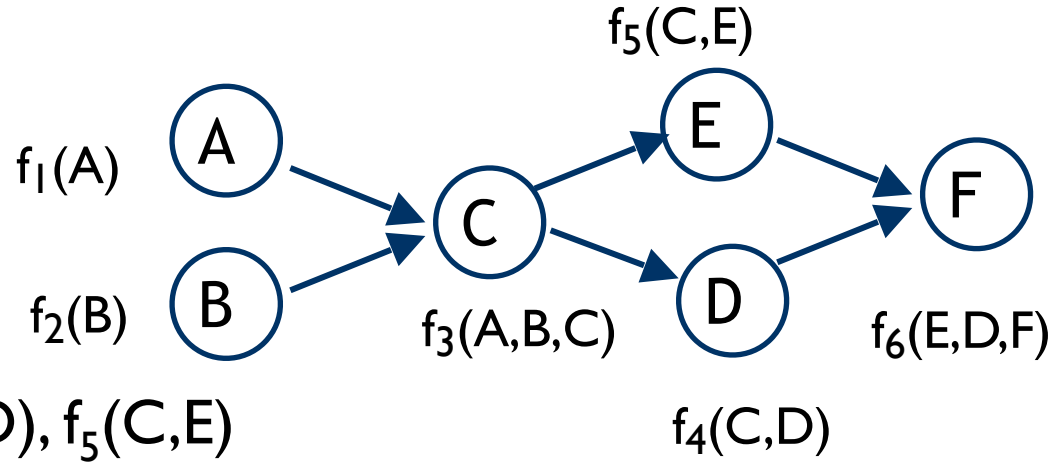
Ordering:
C,F,A,B,E,D



1. C:
2. F:
3. A:
4. B:
5. E:
6. D:

VE: Buckets—Place Original Factors in first applicable bucket.

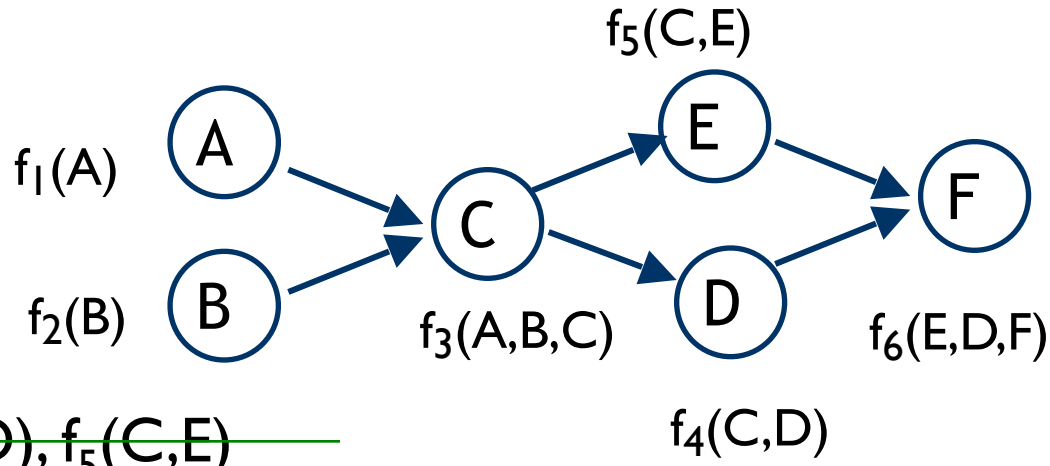
Ordering:
C,F,A,B,E,D



1. C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$
4. B: $f_2(B)$
5. E:
6. D:

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

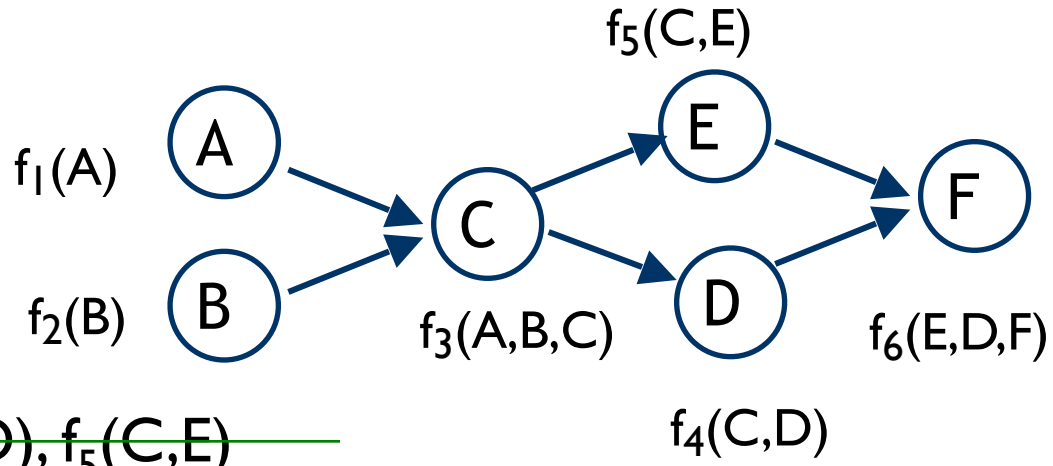


1. ~~C: $f_3(A,B,C), f_4(C,D), f_5(C,E)$~~
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E:
6. D:

1. $\Sigma_C f_3(A,B,C) \times f_4(C,D) \times f_5(C,E)$
   $= f_7(A,B,D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

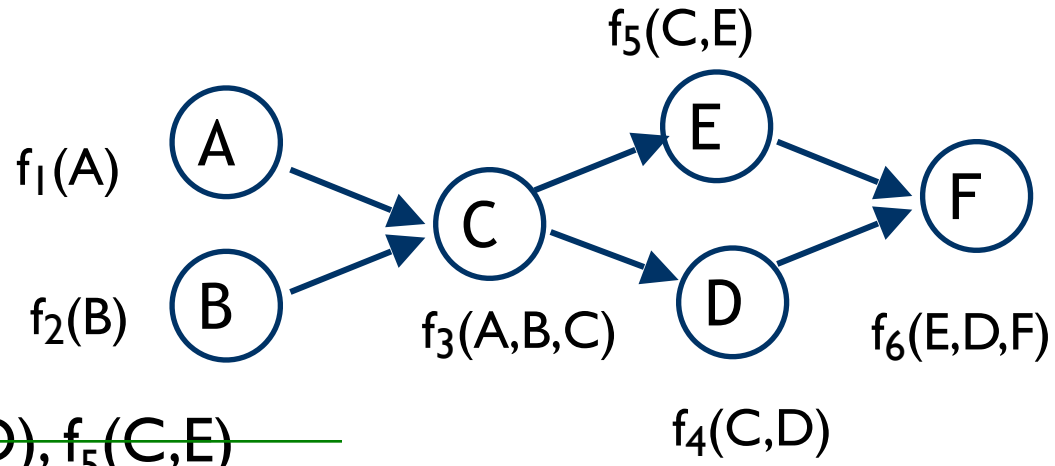


1. ~~C: $f_3(A,B,C), f_4(C,D), f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. A: $f_1(A), f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E: $f_8(E,D)$
6. D:

2. $\Sigma_F f_6(E,D,F) = f_8(E,D)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

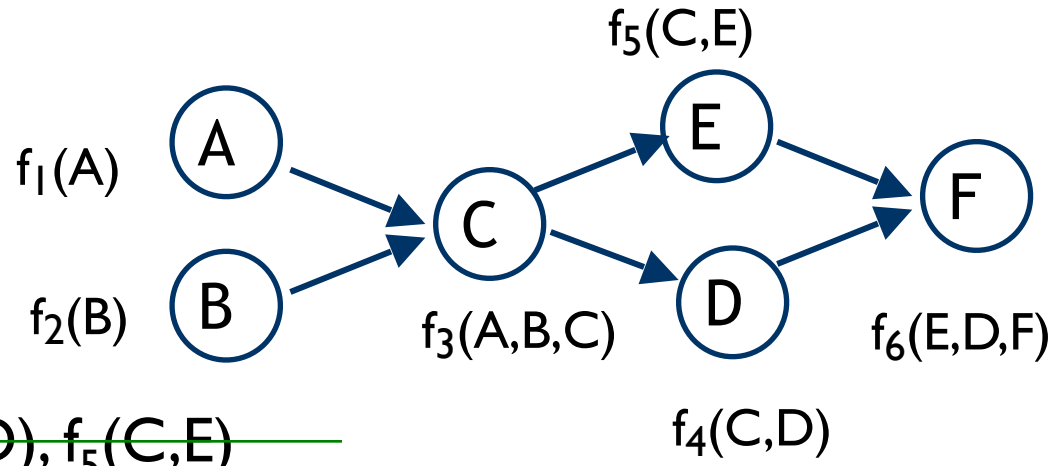


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. B: $f_2(B)$, $f_9(B,D,E)$
5. E: $f_8(E,D)$
6. D:

3. $\Sigma_A f_1(A) \times f_7(A,B,D,E)$
   $= f_9(B,D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D

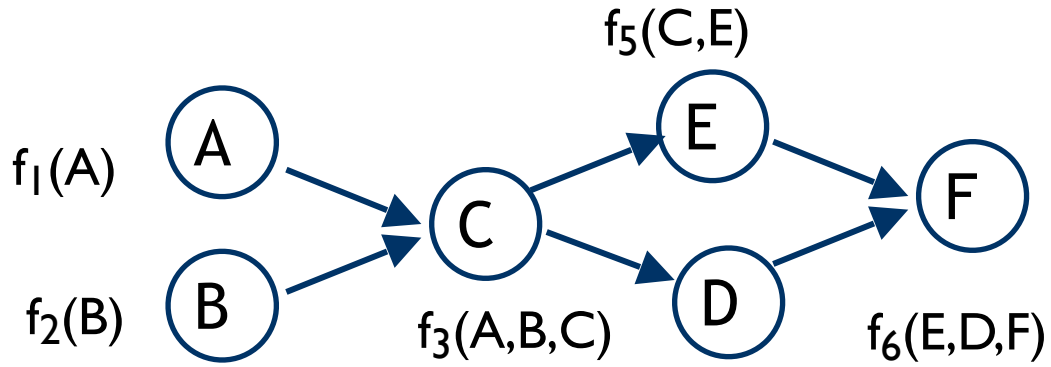


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. E: $f_8(E,D)$, $f_{10}(D,E)$
6. D:

4. $\Sigma_B f_2(B) \times f_9(B,D,E)$
   $= f_{10}(D,E)$

VE: Eliminate the variables in order, placing new factor in first applicable bucket.

Ordering:
C,F,A,B,E,D



1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. ~~E: $f_8(E,D)$, $f_{10}(D,E)$~~
6. D: $f_{11}(D)$

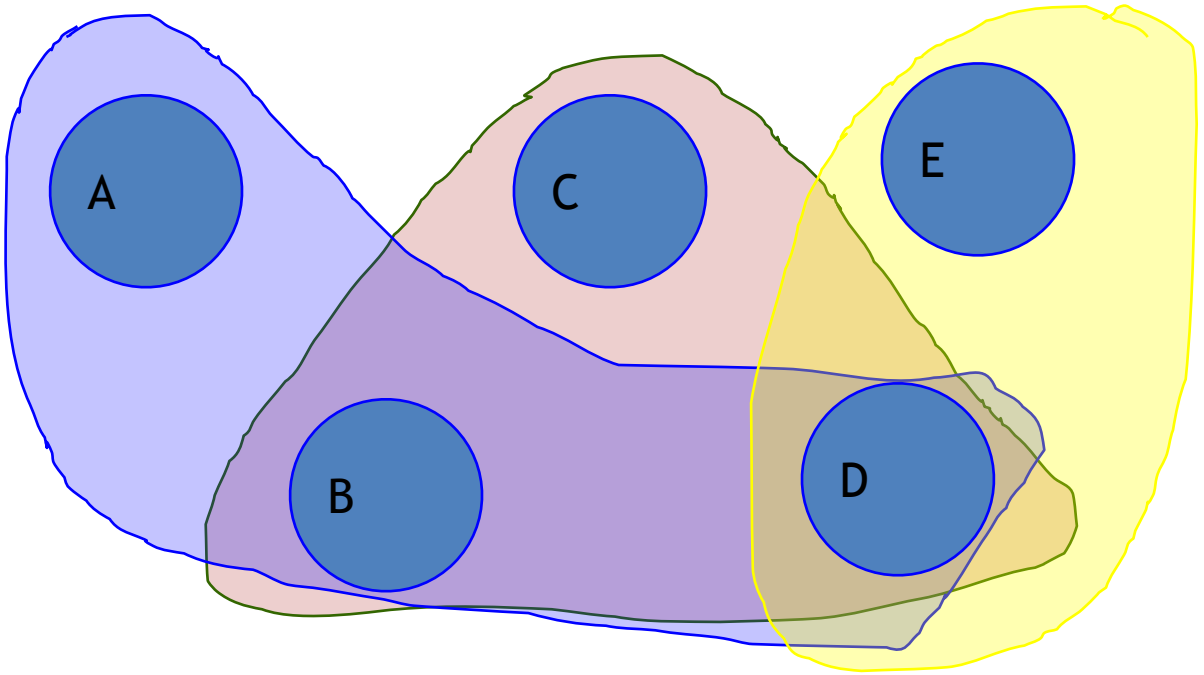
5. $\Sigma_E f_8(E,D) \times f_{10}(D,E) = f_{11}(D)$

f_{11} is the final answer, once we normalize it.

Hypergraphs

A **hypergraph** has vertices just like an ordinary graph, but instead of edges between two vertices X↔Y it contains **hyperedges**.

- A hyperedge is a set of vertices (i.e., potentially more than one)



{A,B,D}
{B,C,D}
{E,D}

Hypergraph

Hypergraph of Bayes Net includes:

- The set of vertices are the nodes of the Bayes net.
- The hyperedges are the families appearing in each CPT.

  $\{X_i\} \cup \text{Parents}(X_i)$

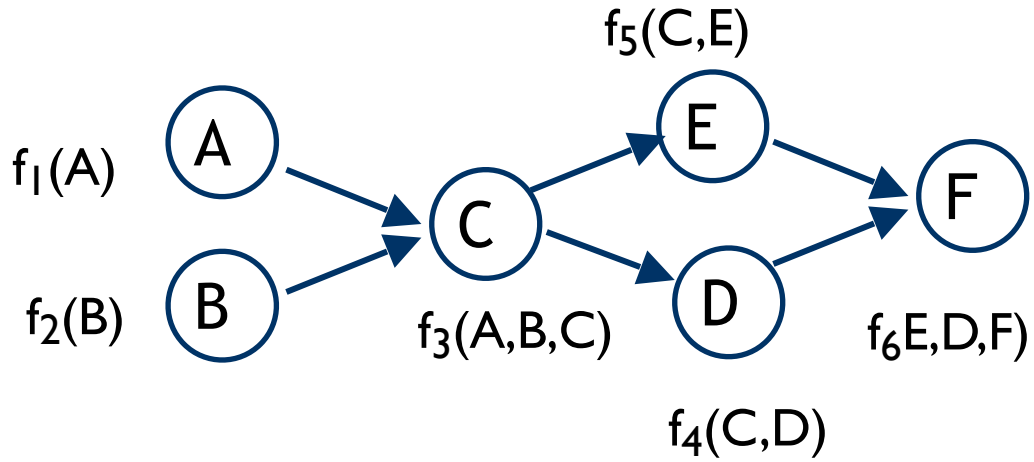
Variable Elimination in the HyperGraph

To eliminate variable X_i in the hypergraph we

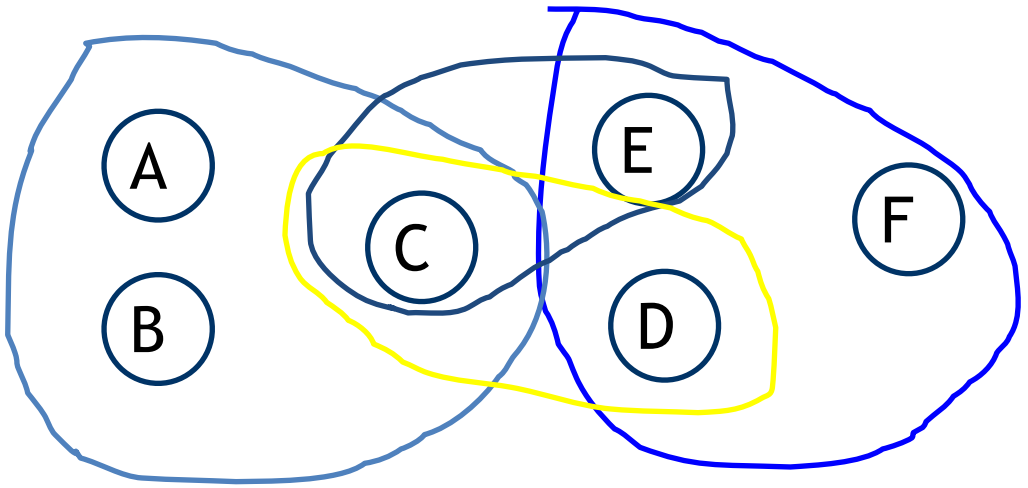
- Remove the vertex X_i
- Create a new hyperedge H_i equal to the union of all of the hyperedges that contain X_i minus X_i
- Remove all of the hyperedges containing X from the hypergraph.
- Add the new hyperedge H_i to the hypergraph.

VE Factors

Ordering:
C,F,A,B,E,D

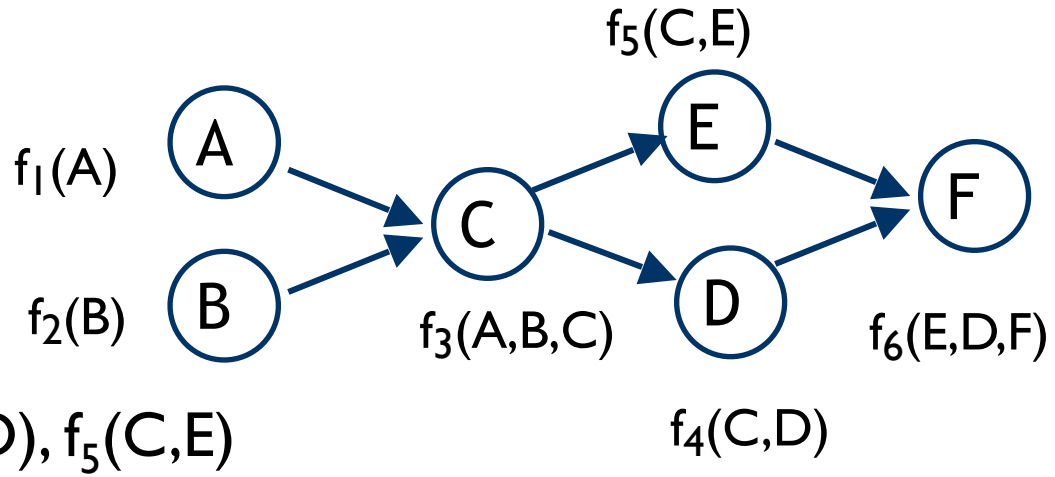


1. C:
2. F:
3. A:
4. B:
5. E:
6. D:



VE: Place Original Factors in first applicable bucket.

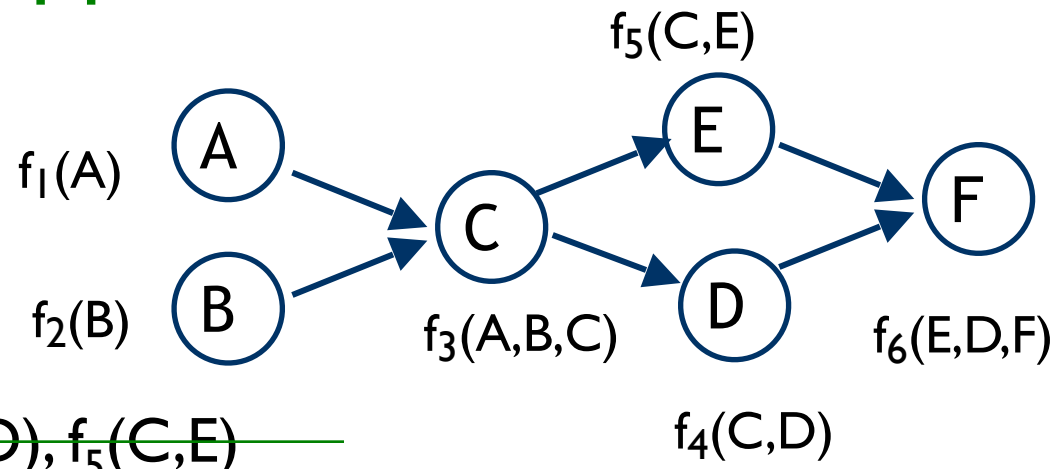
Ordering:
C,F,A,B,E,D



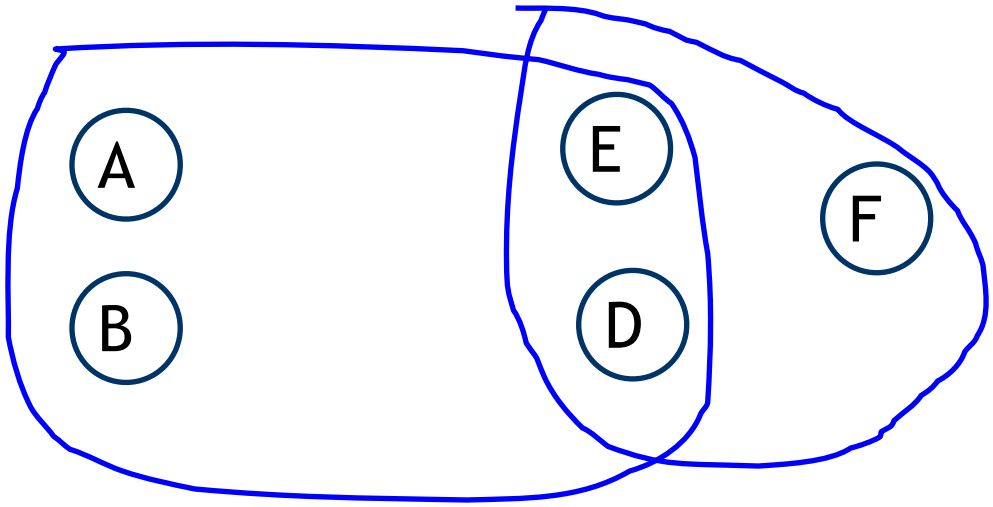
1. C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$
4. B: $f_2(B)$
5. E:
6. D:

VE: Eliminate C, placing new factor f7 in first applicable bucket.

Ordering:
C,F,A,B,E,D

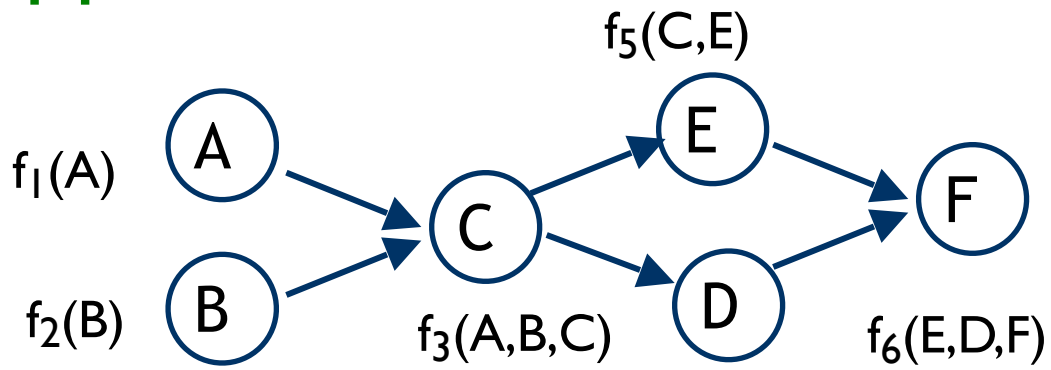


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E:
6. D:

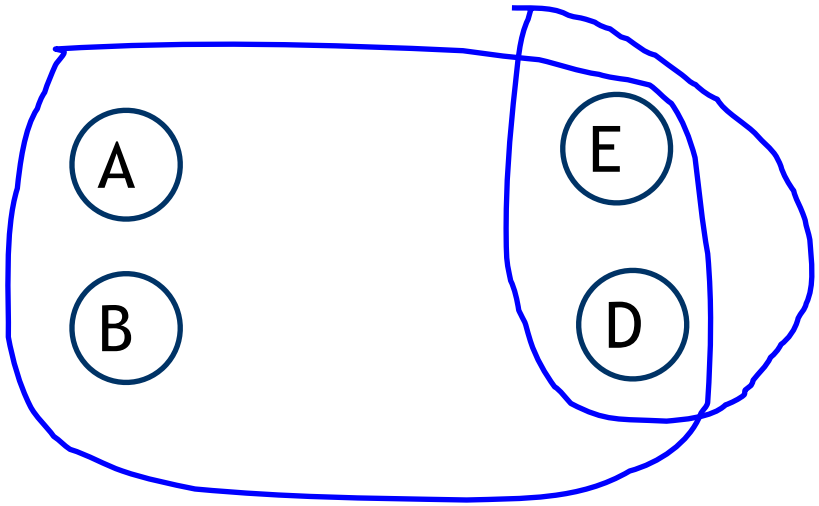


VE: Eliminate F, placing new factor f8 in first applicable bucket.

Ordering:
C,F,A,B,E,D

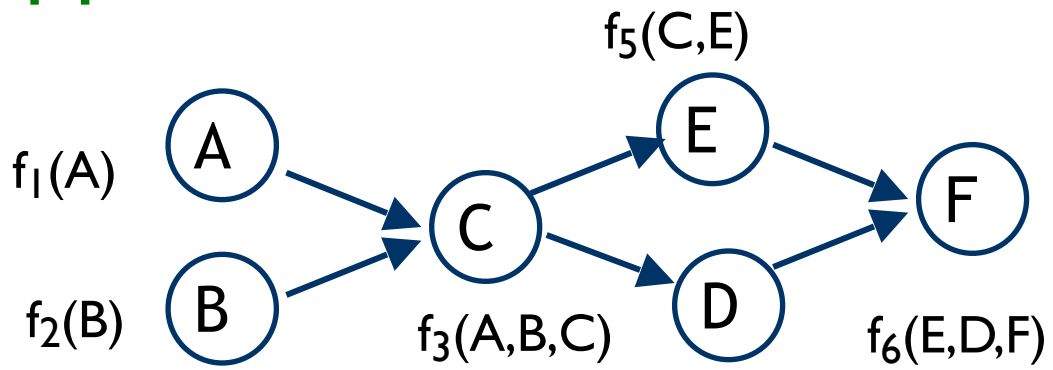


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E: $f_8(E,D)$
6. D:

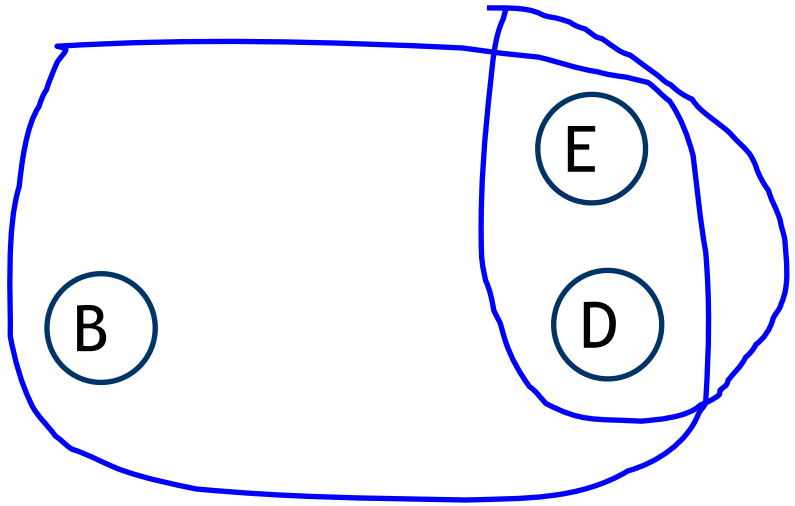


VE: Eliminate A, placing new factor f9 in first applicable bucket.

Ordering:
C,F,A,B,E,D

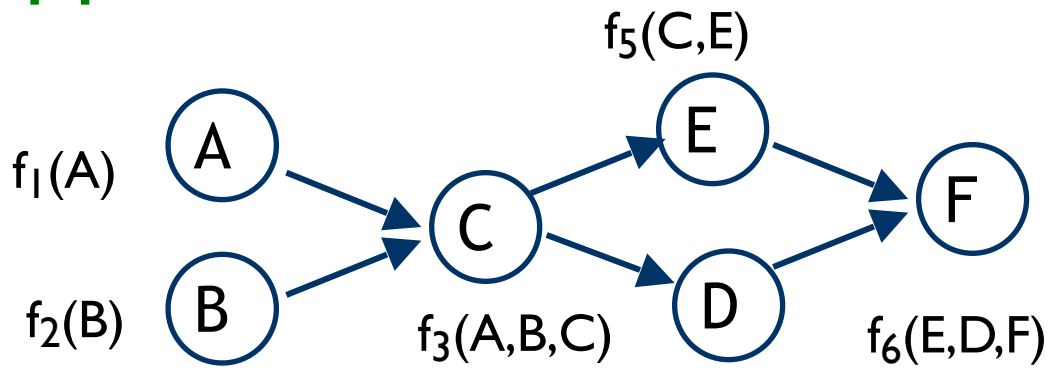


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. B: $f_2(B)$, $f_9(B,D,E)$
5. E: $f_8(E,D)$
6. D:

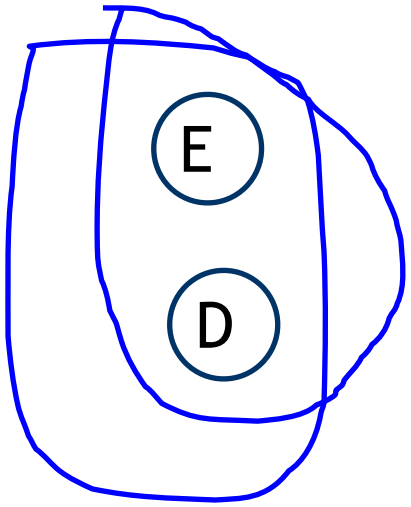


VE: Eliminate B, placing new factor f10 in first applicable bucket.

Ordering:
C,F,A,B,E,D

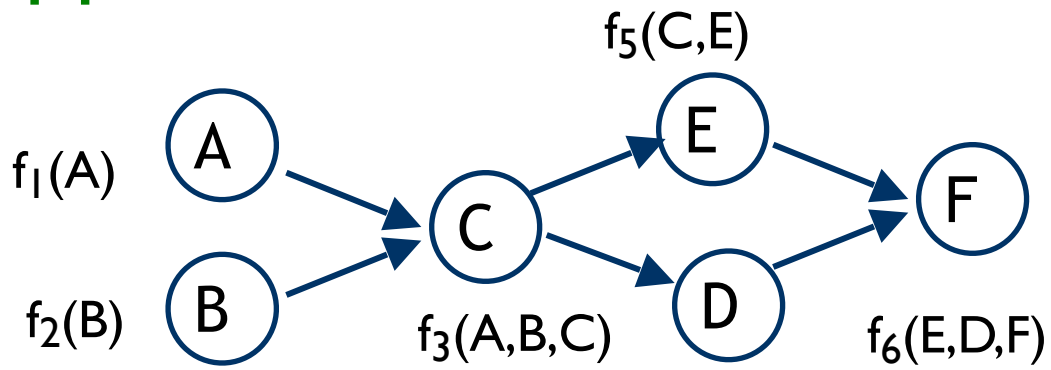


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. E: $f_8(E,D)$, $f_{10}(D,E)$
6. D:

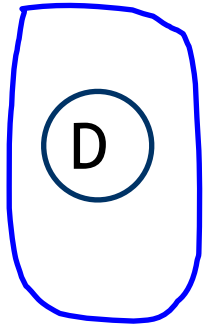


VE: Eliminate E, placing new factor f11 in first applicable bucket.

Ordering:
C,F,A,B,E,D

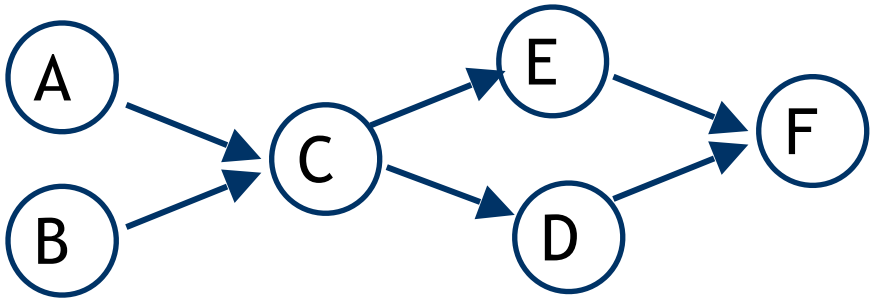


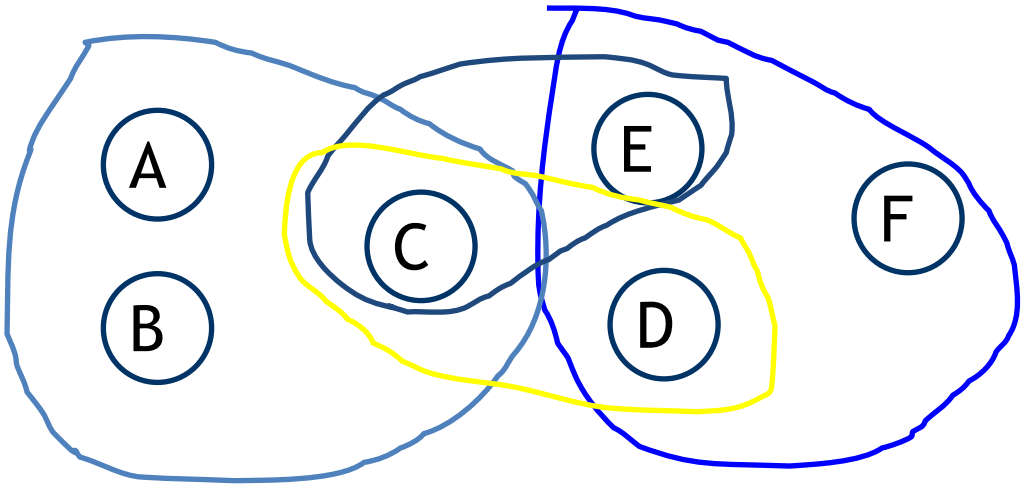
1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. ~~E: $f_8(E,D)$, $f_{10}(D,E)$~~
6. D: $f_{11}(D)$



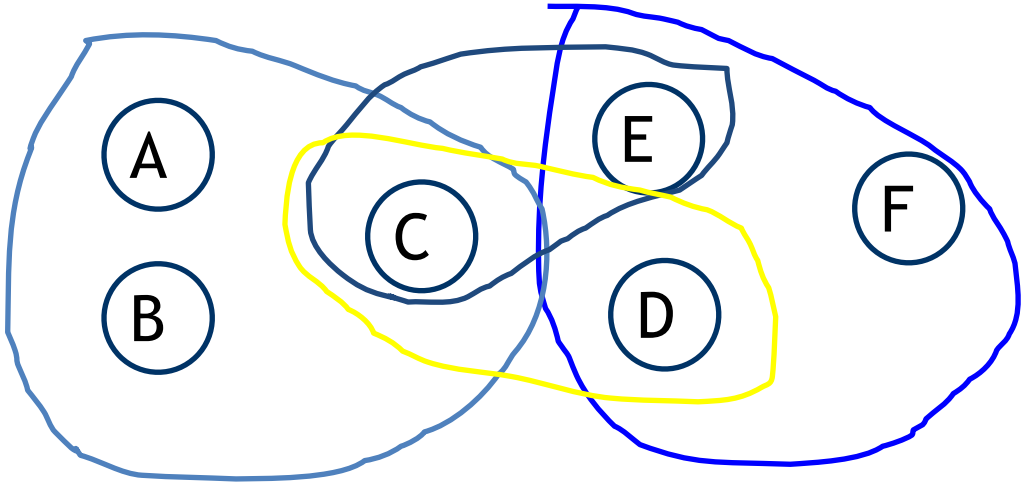
Complexity of Variable Elimination

P(A,B,C,D,E,F) =
    P(A)P(B)
X   P(C|A,B)
X   P(E|C)
X   P(D|C)
X   P(F|E,D).

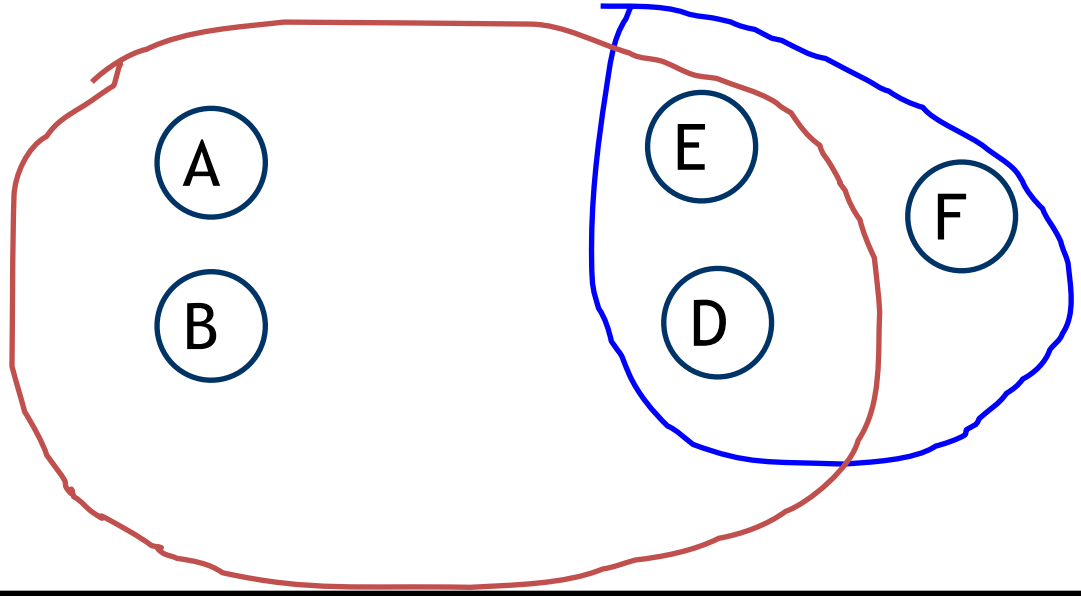




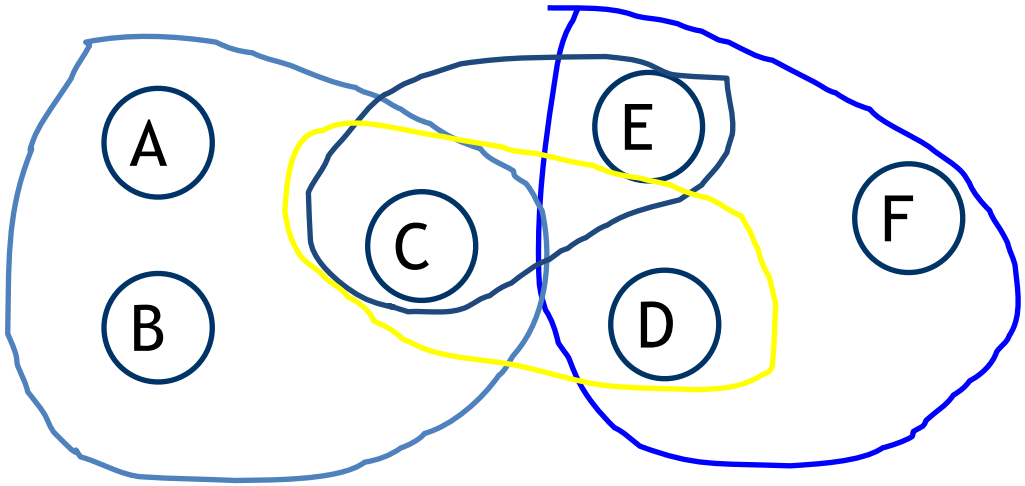
Complexity of Variable Elimination



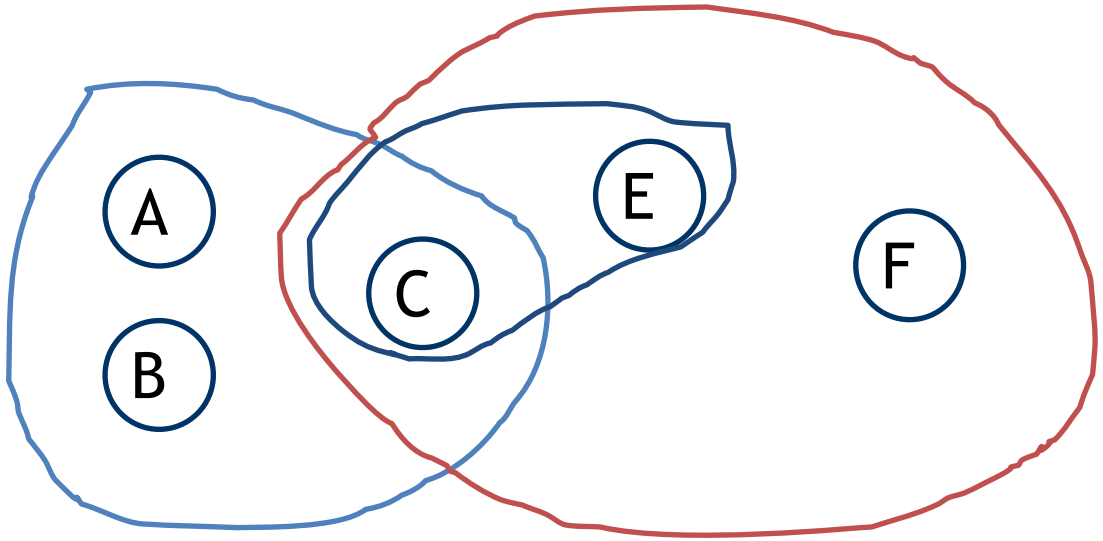
- Eliminate C



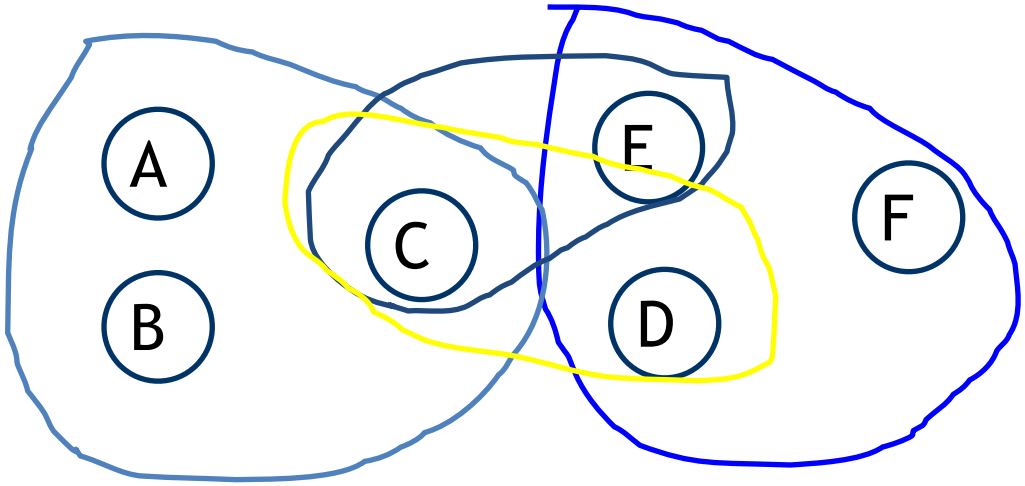
Complexity of Variable Elimination



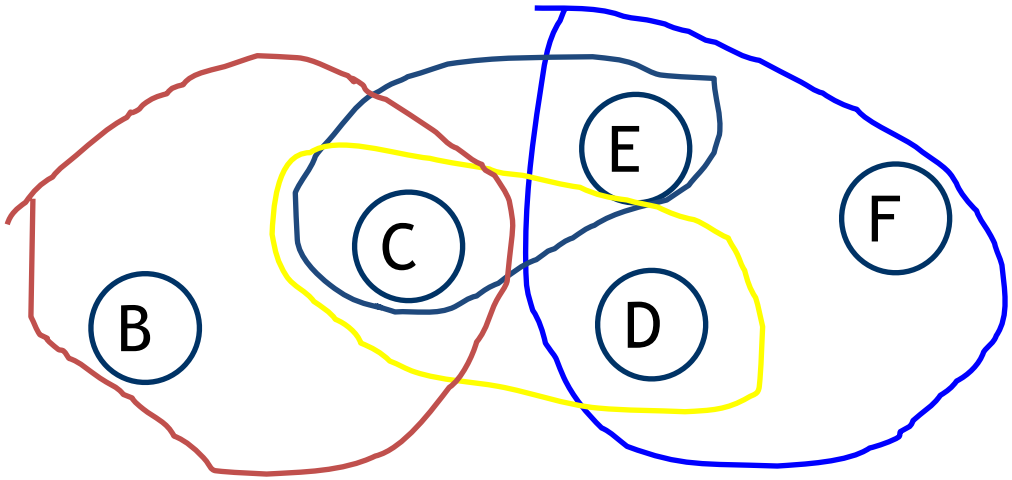
- Eliminate D



Complexity of Variable Elimination



- Eliminate A



Complexity of Variable Elimination

- Given an ordering π of the variables and an initial hypergraph $\mathscr{H}$ eliminating these variables yields a sequence of hypergraphs

  $\mathscr{H} = H_0, H_1, H_2, \ldots, H_n$

  where H_n contains only one vertex (the query variable).

- The elimination width of VE reflects the maximum size (number of variables) of any hyperedge in any of the hypergraphs $H_0, H_1, \ldots, H_n$.
- The largest size of any hyperedge in the previous example was 4 ({A,B,E,D} in H_1 and H_2).

Tree Width

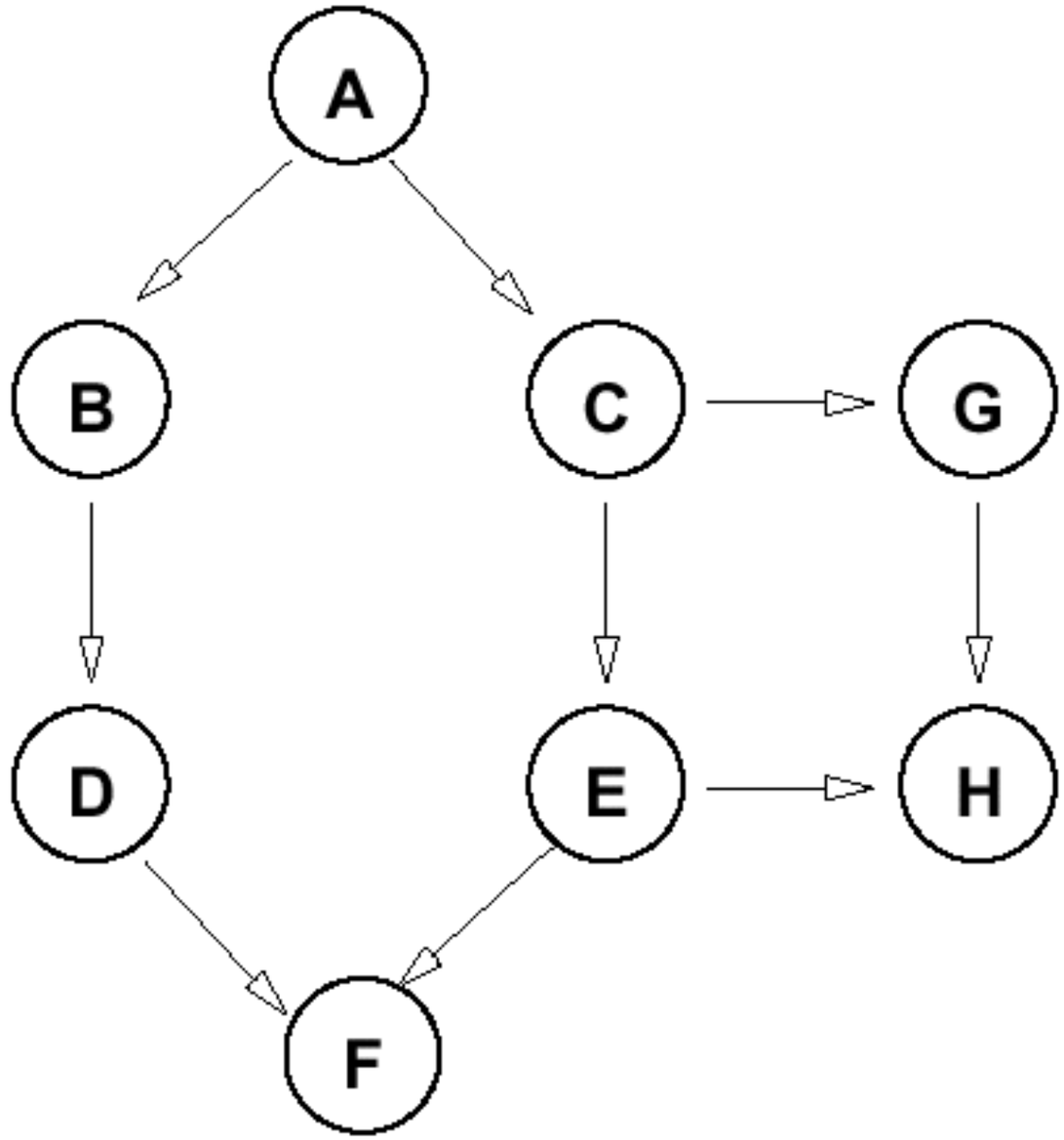
- Given a hypergraph $\mathscr{H}$ with vertices $\{X_1, X_2, \ldots, X_n\}$ the tree width (ω) of $\mathscr{H}$ is the MINIMUM elimination width of *any of the n!* different orderings of the X_i minus *1*.

- Thus VE has best case complexity of $2^{O(\omega)}$ where ω is the tree width of the initial Bayes Net.
  - Largest factor takes $2^{O(\omega)}$ space to store
  - Largest factor takes $2^{O(\omega)}$ time to process

- In the worst case, the tree width is equal to the number of variables (minus 1)

Different Orderings = Different Elimination Widths

Suppose query variable is D. Consider different orderings for this network



E,C,A,B,G,H,F :

- Bad

A,F,H,G,B,C,E:

- Good

Tree Width

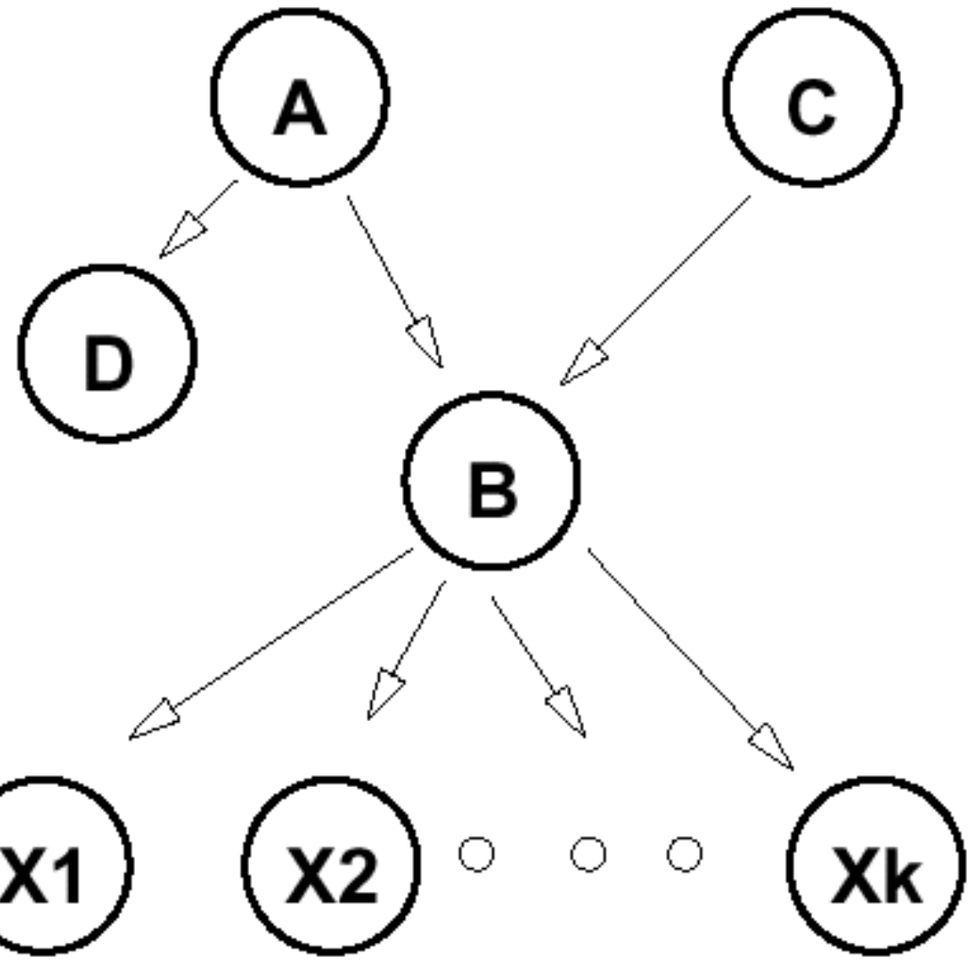
- Exponential in the tree width is the best that VE can do.
  - But, finding an ordering that has elimination width equal to tree width is NP-Hard.
    - so in practice there is no point in trying to speed up VE by finding the best possible elimination ordering.
  - Instead, heuristics are used to find orderings with good (low) elimination widths.
  - In practice, this can be very successful. Elimination widths can often be relatively small, 8-10 even when the network has 1000s of variables.
    - Thus VE can be *much* more efficient than simply summing the probability of all possible events (which is exponential in the number of variables).
    - Sometimes, however, the tree width is equal to the number of variables (minus 1).

Finding Good Orderings

- A *polytree* is a singly connected Bayes Net: in particular there is only one path between any two nodes.
- Good orderings are easy to find for polytrees
  - At each stage eliminate *a singly connected node*.
  - Because we have a polytree we are assured that a singly connected node will exist at each elimination stage.
  - The size of the factors in the tree never increase.

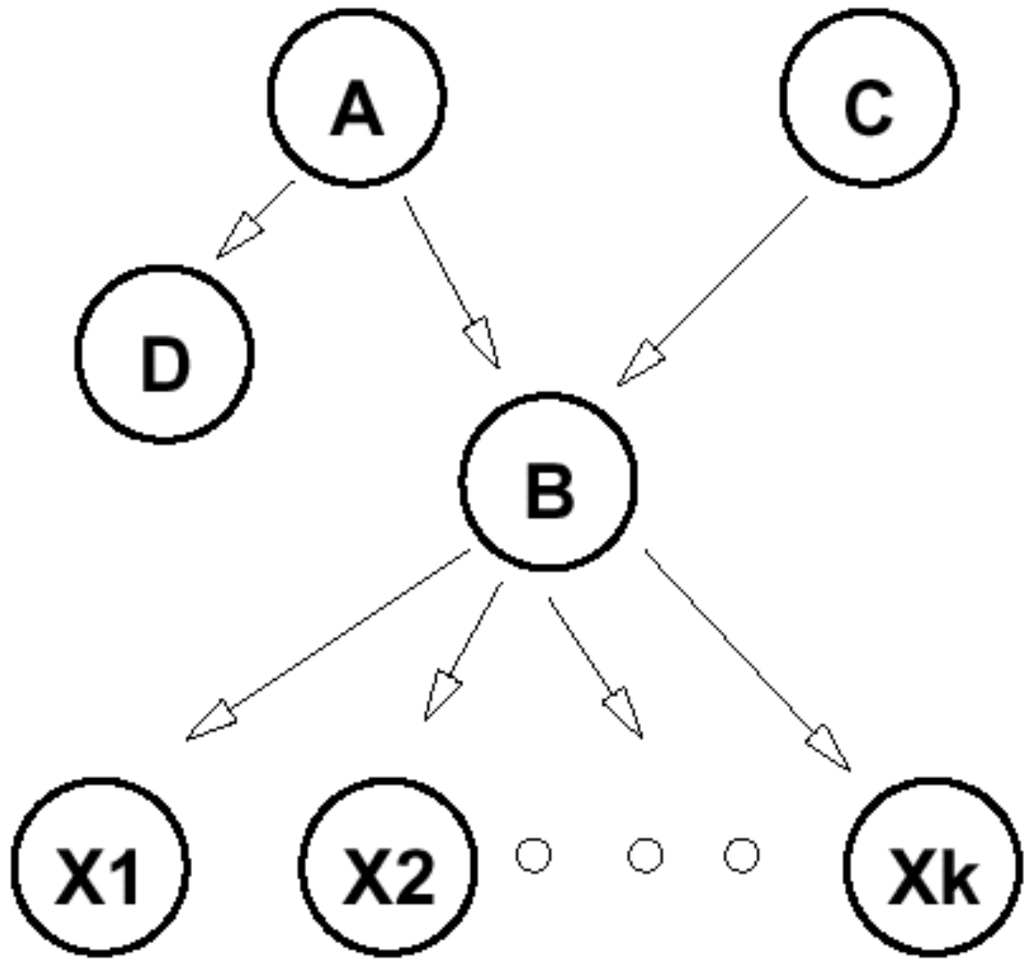
Elimination Ordering: Polytrees

- Eliminating singly connected nodes allows VE to run in linear time
  - e.g., in this network, eliminate D, A, C, X1,…; or eliminate X1,… Xk, D, A C; or mix it up.
- Result: no factor is ever larger than original CPTs
- Eliminating B before these gives factors that include A,C, X1,… and Xk!!!

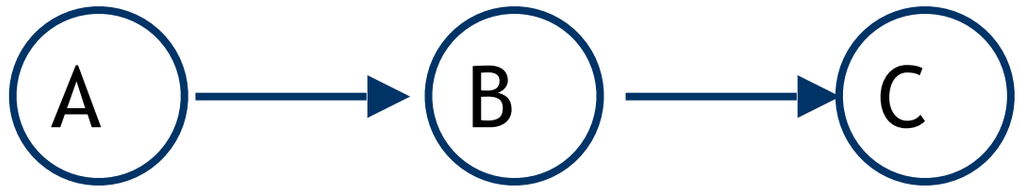


Min Fill Heuristic

- A fairly effective heuristic is to always eliminate next the variable that creates the smallest size factor.
- This is called the min-fill heuristic.
  - B creates a factor of size k+2
  - A creates a factor of size 2
  - D creates a factor of size 1
- This heuristic always solves polytrees in linear time.



Relevance



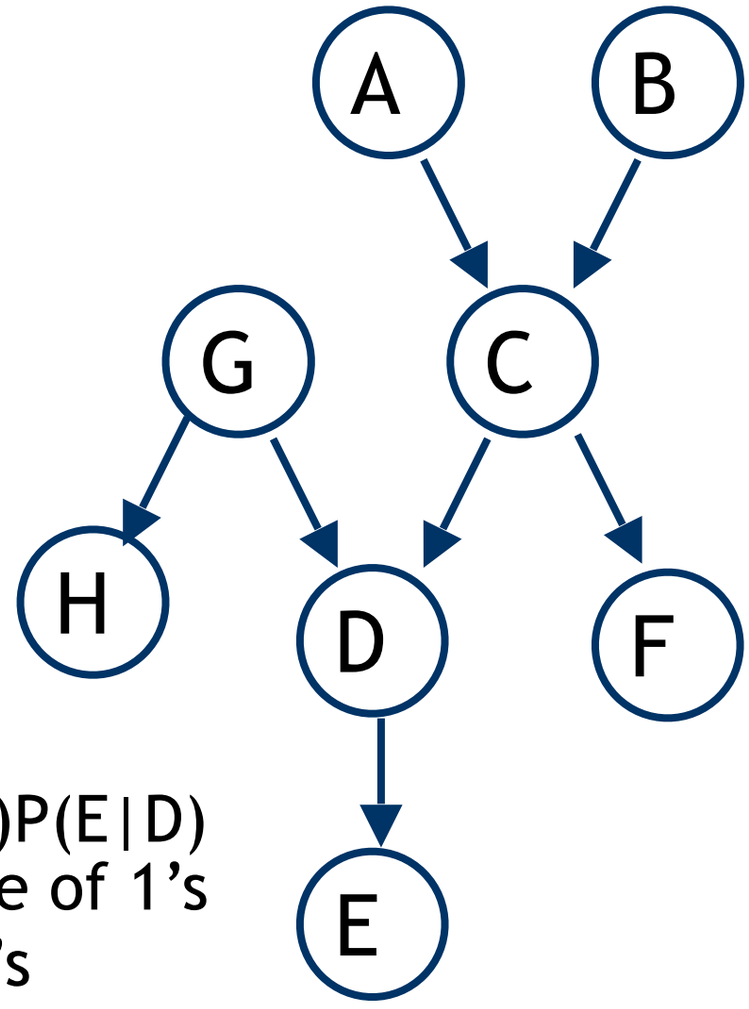
- Certain variables have no impact on the query. For example, in network ABC, computing P(A) with no evidence requires elimination of B and C.
  - But when you sum out these variables, you compute a trivial factor (that always evaluates to 1); for example:
  - Eliminating C: $f_4(B) = \Sigma_C f_3(B,C) = \Sigma_C P(C|B)$
  - This is 1 for any value of B (e.g., P(c|b) + P(~c|b) = 1)
- No need to think about B or C for this query

Relevance

- Can restrict attention to *relevant* variables. Given query **Q**, evidence **E**:
  - **Q** itself is relevant
  - if any node **Z** is relevant, its parents are relevant
  - if e∈**E** is a descendent of a relevant node, then E is relevant
- We can restrict our attention to the *sub-network comprising only relevant variables* when evaluating a query Q

Relevance: Examples

- Query: P(F)
  - relevant: F, C, B, A
- Query: P(F|E)
  - relevant: F, C, B, A
  - also: E, hence D, G
  - *intuitively, we need to compute P(C|E) to compute P(F|E)*
- Query: P(F|H)
  - relevant: F, C, B, A



$P(A)P(B)P(C|A,B)P(F|C)\ P(G)P(h|G)P(D|G,C)P(E|D)$

$= \ldots P(G)P(h|G)P(D|G,C)\ \sum_E P(E|D)$ = a table of 1's

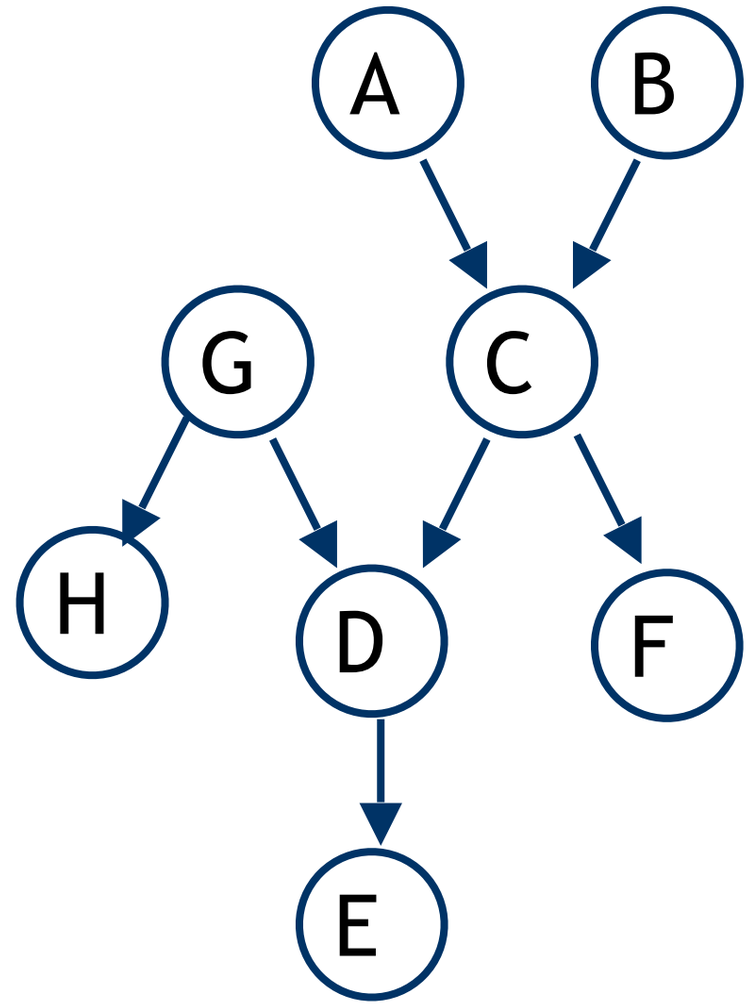
$= \ldots P(G)P(h|G)\ \sum_D P(D|G,C)$ = a table of 1's

$= [P(A)P(B)P(C|A,B)P(F|C)]\ [\sum_G P(G)P(h|G)]$

$[\sum_G P(G)P(h|G)] \neq 1$ but irrelevant

once we normalize, multiplies each value of F equally

Relevance: Examples



Query: P(F|E,C)

- algorithm says all variables except H are relevant; but really none except C, F (since C cuts of all influence of others)
- algorithm is overestimating relevant set

Independence in a Bayes Net

- Another piece of information we can obtain from a Bayes net is the "structure" of relationships in the domain.
- The structure of the BN means: every X_i is *conditionally independent of all of its non-descendants given it parents*:

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

for any subset $S \subseteq \text{NonDescendents}(X_i)$

More generally ….

Conditional independencies can be useful in computation, explanation, etc. related to a BN

How do we determine if two variables X, Y are independent given a set of variables E?

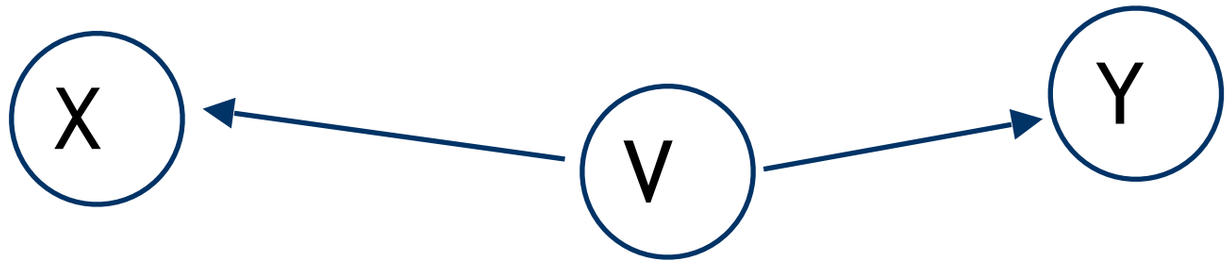
- We can use a (simple) graphical property called **d-separation**

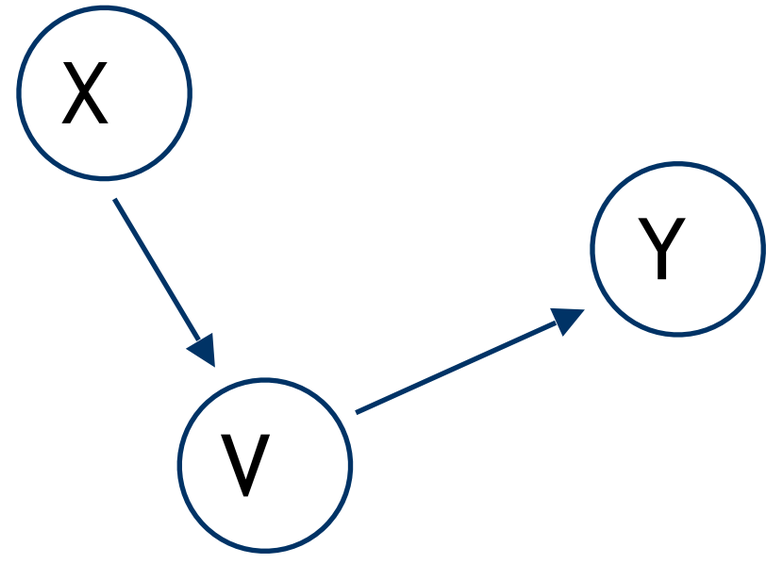
  D-separation: A set of variables E **d-separates** X and Y if it **blocks** every undirected path in the BN between X and Y (we'll define Blocks next.)

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

- thus BN gives us an easy way to tell if two variables are independent (set E = {}) or conditionally independent given E.

What does it mean to be blocked?

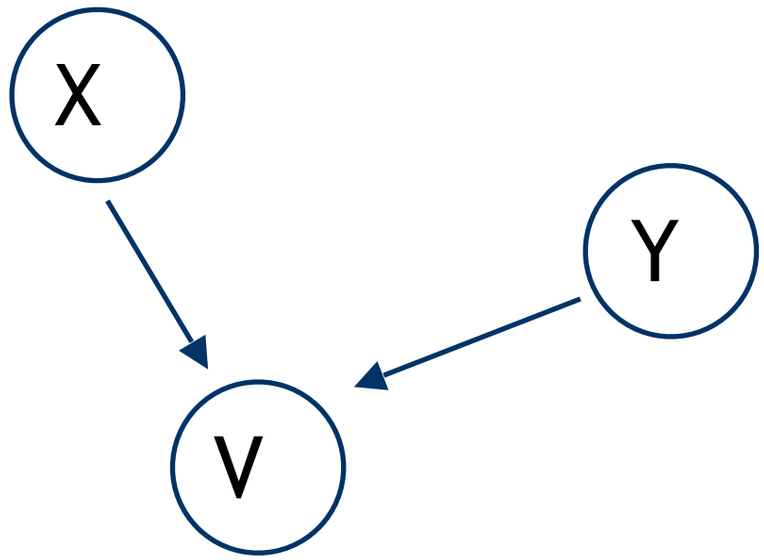




There exists a variable *V* on the path such that
- it **is** in the evidence set *E*
- the arcs putting *V* in the path are “tail-to-tail”

Or, there exists a variable *V* on the path such that
- it **is** in the evidence set *E*
- the arcs putting *V* in the path are “tail-to-head”

What does it mean to be blocked?



If the variable V is on the path such that the arcs putting V on the path are "head-to-head", the variables are still blocked .... so long as:

V is NOT in the evidence set E

neither are any of its descendants

Blocking: Graphical View

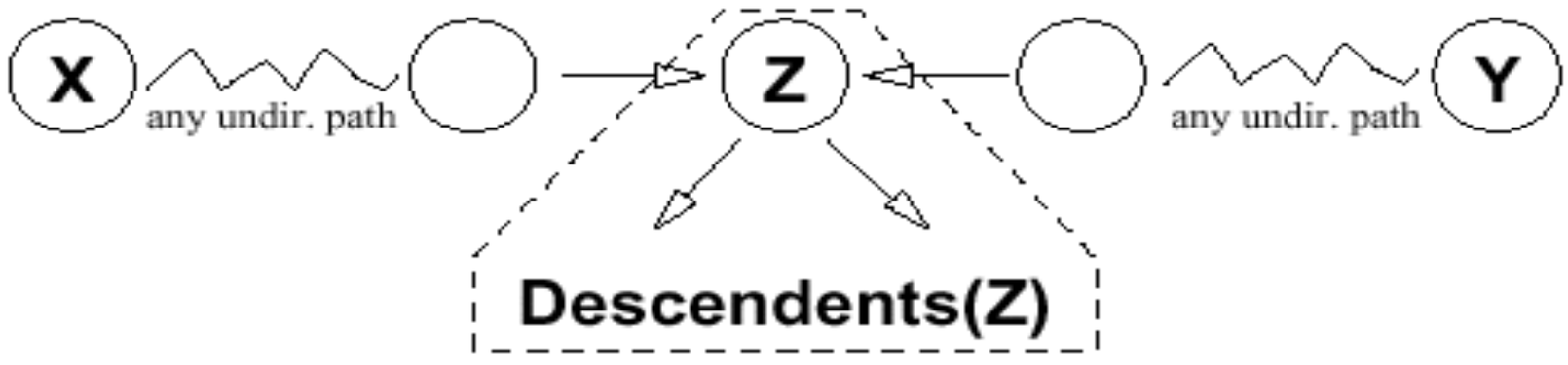
(1)

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

(2)

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

(3)

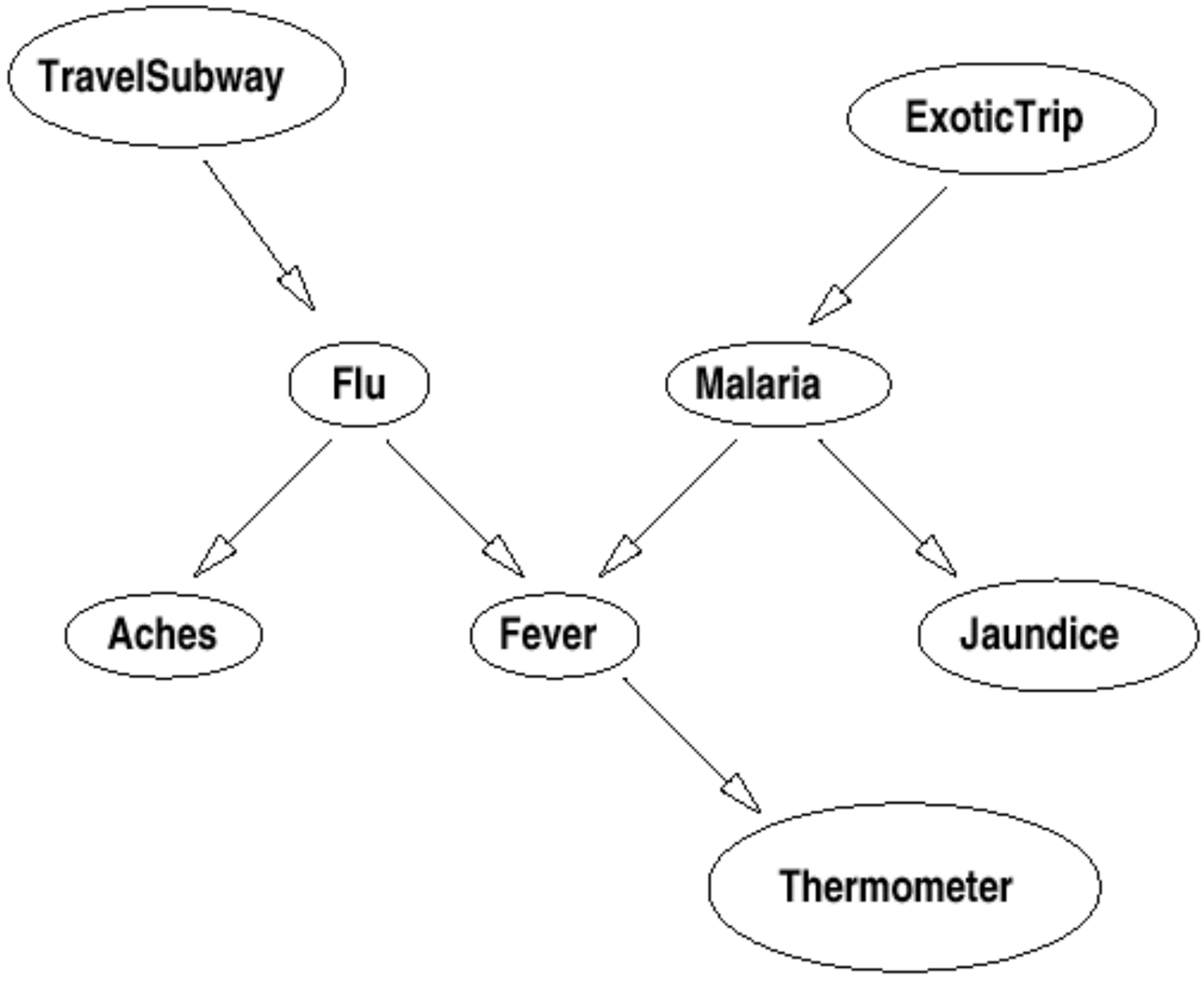


If Z is ***not*** in evidence and***no*** descendent of Z is in evidence, then the path between X and Y is blocked

D-separation implies conditional independence

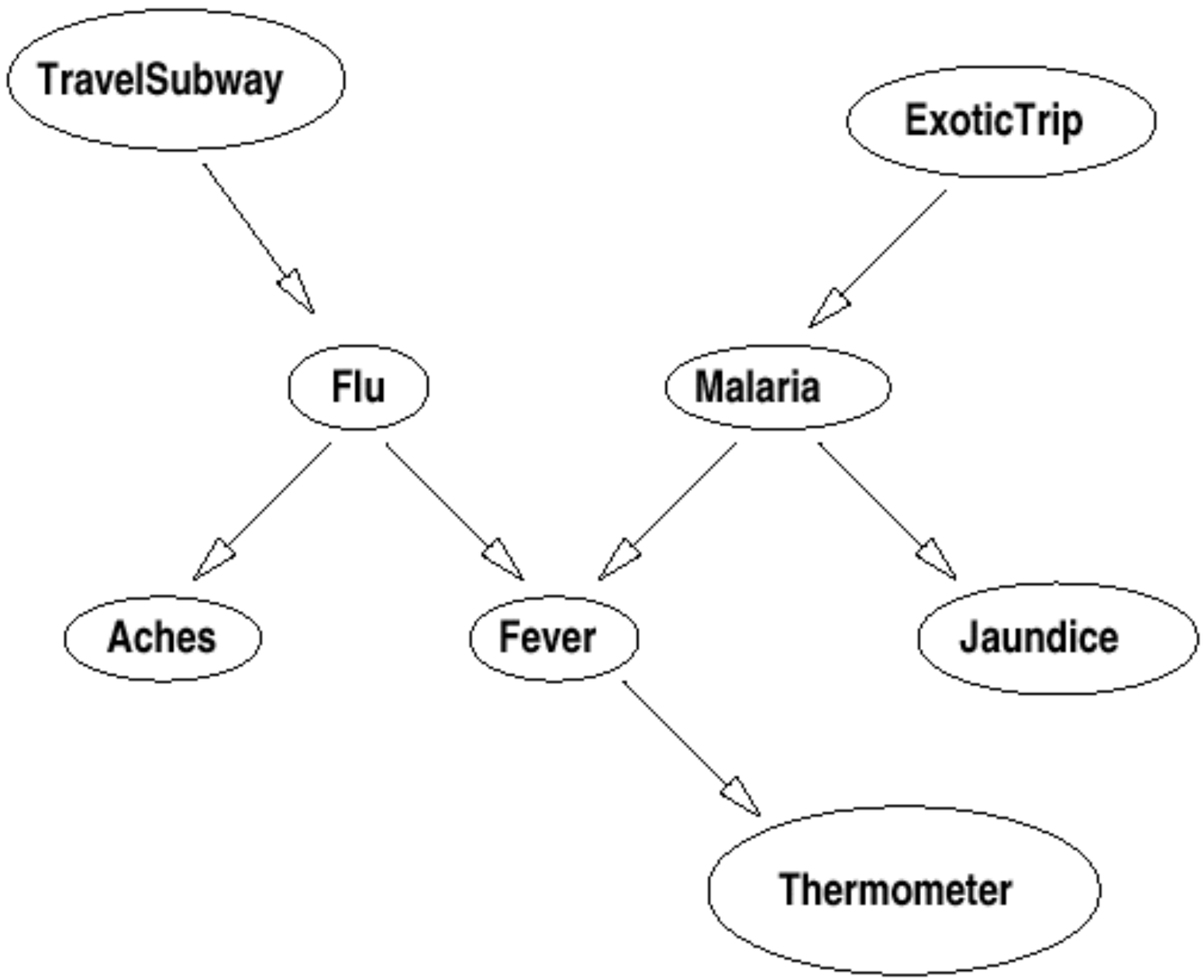
Theorem [Verma & Pearl, 1998]: If a set of evidence variables E d-separates X and Z in a Bayesian network's graph, then X is independent of Z given E.

D-Separation: Intuitions



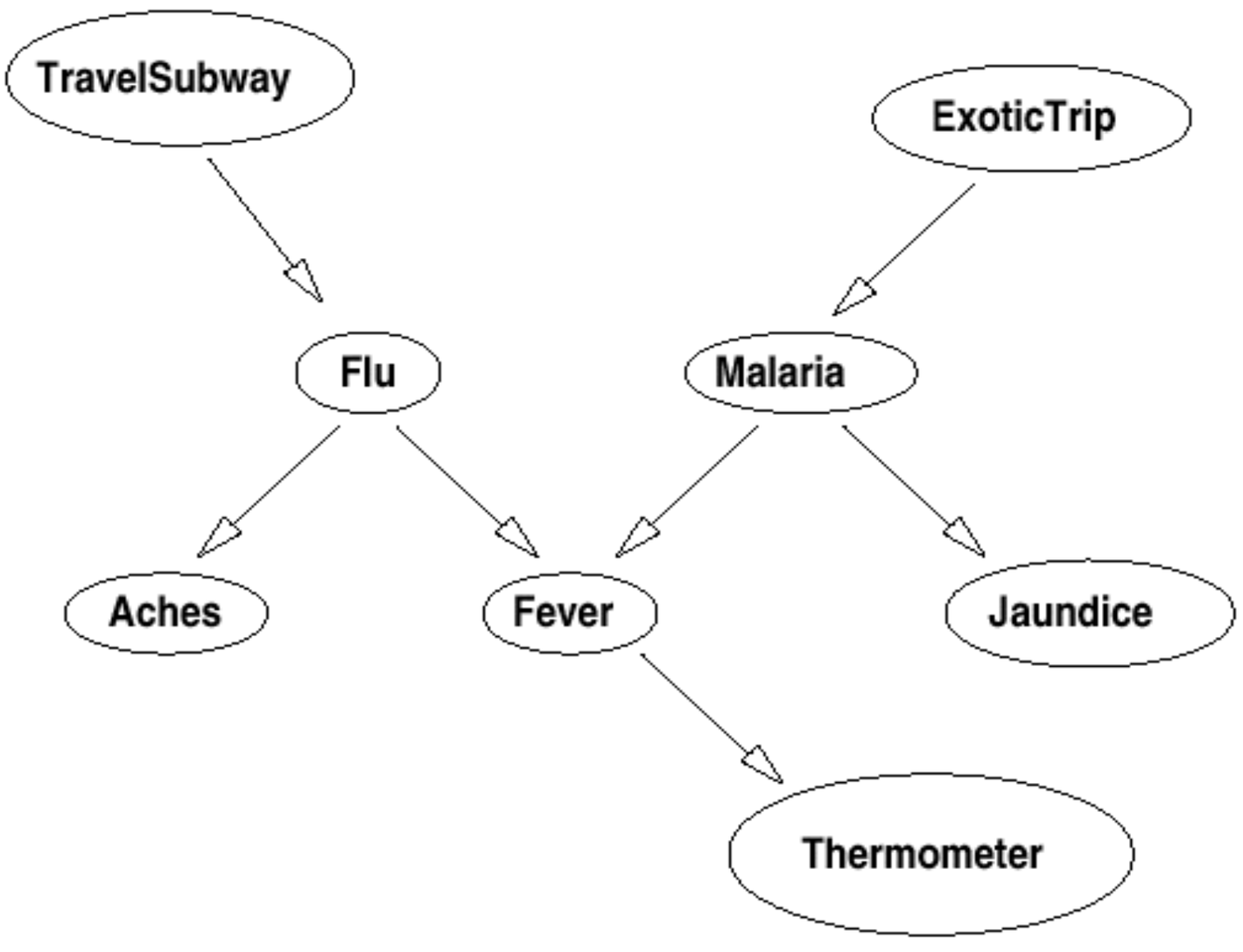
Subway and Therm are dependent; but are independent given Flu (since Flu blocks the only path)

D-Separation: Intuitions



Aches and Fever are dependent; but are independent given Flu (since Flu blocks the only path). Similarly for Aches and Therm (dependent, but indep. given Flu).

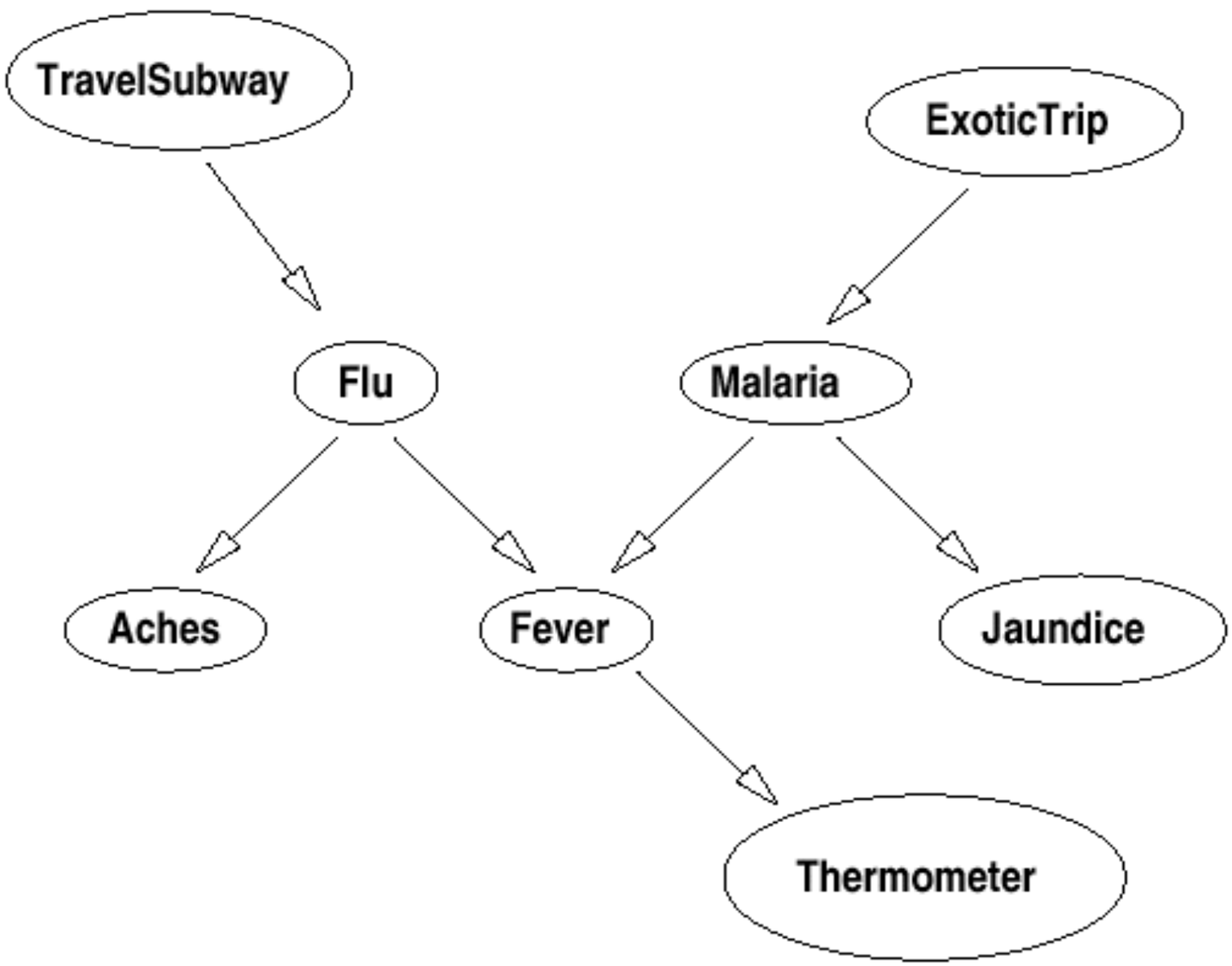
D-Separation: Intuitions



Flu and Mal are indep. (given no evidence): Fever blocks the path, since it is not in evidence, nor is its descendant Therm.

Flu and Mal are dependent given Fever (or given Therm): nothing blocks path now.

D-Separation: Intuitions



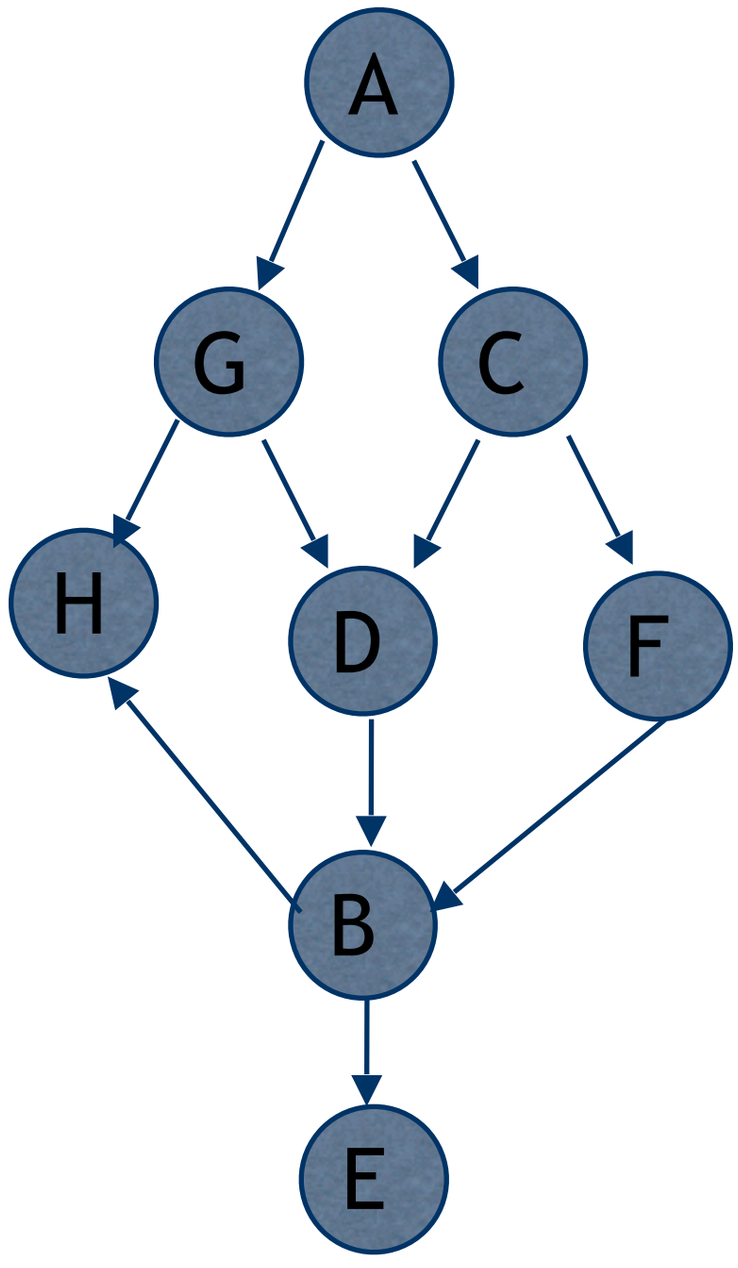
Subway, ExoticTrip are indep.;

they are dependent given Therm;

they are indep. given Therm and Malaria. This for exactly the same reasons for Flu/Mal above.

D-Separation Example

In the following network determine if A and E are independent given the evidence.



1. A and E given no evidence?
2. A and E given {C}?
3. A and E given {G,C}?
4. A and E given {G,C,H}?
5. A and E given {G,F}?
6. A and E given {F,D}?
7. A and E given {F,D,H}?
8. A and E given {B}?
9. A and E given {H,B}?
10. A and E given {G,C,D,H,D,F,B}?

D-Separation Example

In the following network determine if A and E are independent given the evidence.

1. A and E given no evidence? N
2. A and E given {C}? N
3. A and E given {G,C}? Y
4. A and E given {G,C,H}? Y
5. A and E given {G,F}? N
6. A and E given {F,D}? Y
7. A and E given {F,D,H}? N
8. A and E given {B}? Y
9. A and E given {H,B}? Y
10. A and E given {G,C,D,H,D,F,B}? Y