

# Reasoning Under Uncertainty



Uncertainty material is covered in chapters 13 and 14.



Chapter 13 gives some basic background on probability from the point of view of A.I.

Chapter 14 talks about Bayesian Networks, exact reasoning in Bayes Nets as well as approximate reasoning, which will be main topics for us.

*Note: Slides in this section draw from Faheim Bacchus, Craig Boutillier, Andrew Moore, Sheila McIlraith ...*

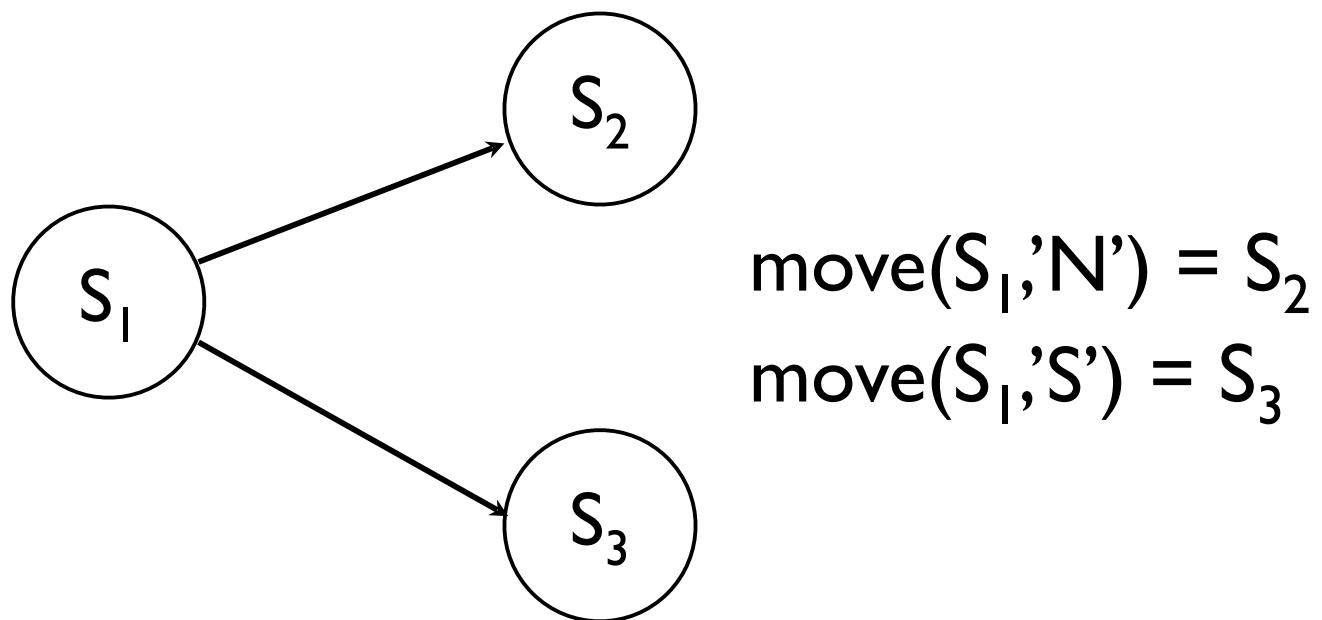


# Reasoning Under Uncertainty

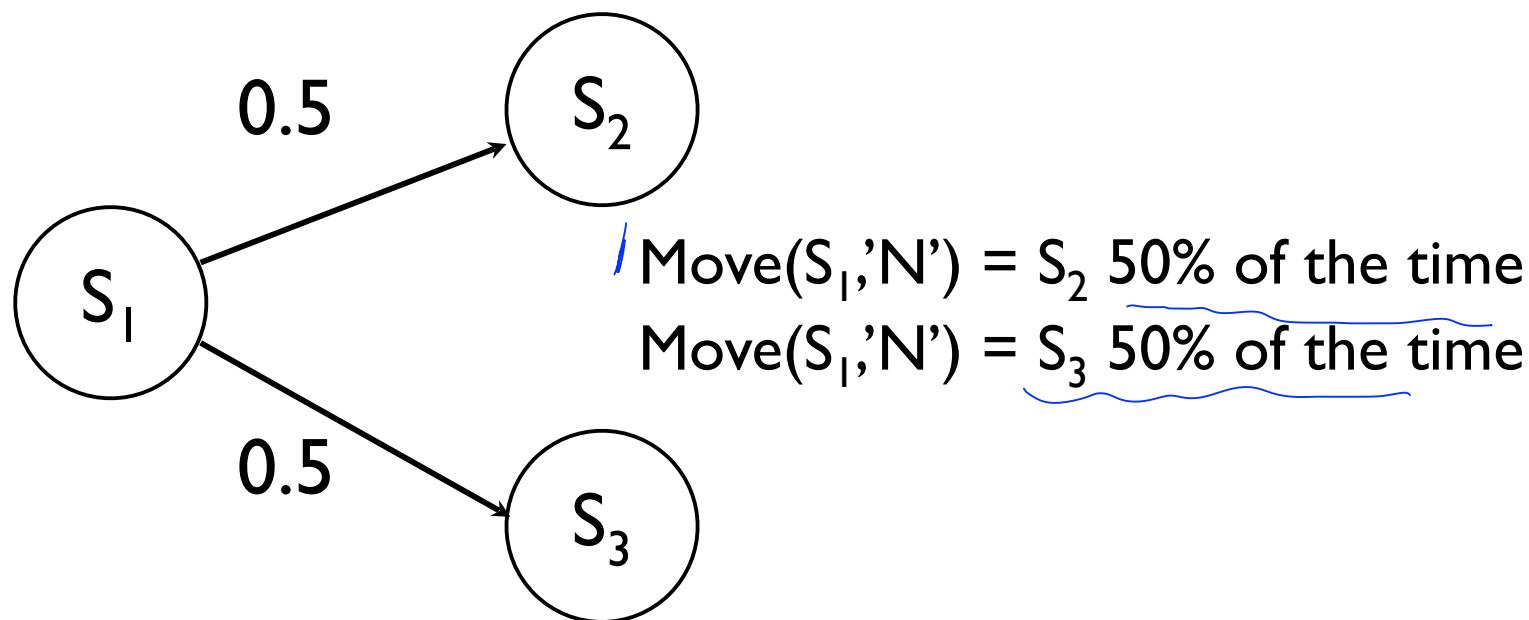
- The world is a very uncertain place.
- As of this point, we've basically danced around that fact. We've assumed that what we see in the world is really there, what we do in the world has predictable outcomes, etc.
- i.e., if you are in state  $S_1$  and you execute action A you arrive at state  $S_2$ .

//

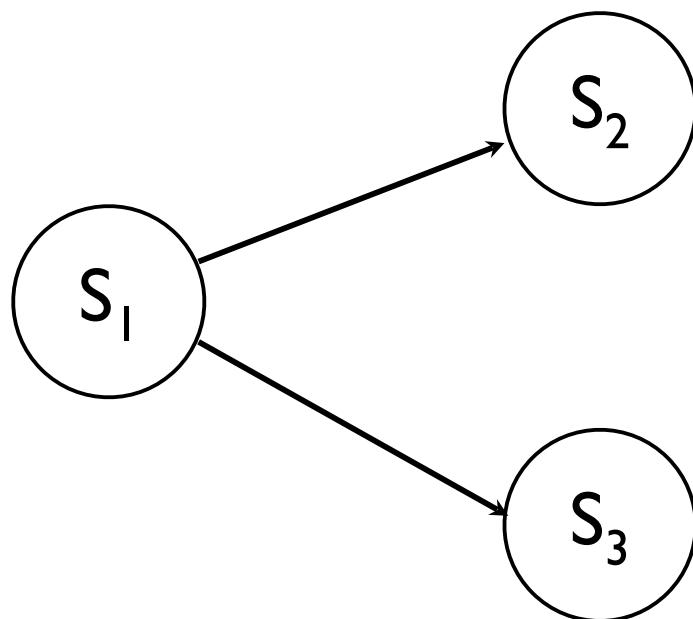
# Example: Sokoban



# Probabilistic Sokoban is a Very Different Game



# Probabilistic Sokoban is a Very Different Game



Based on what we can see,  
there's a 30% chance we're in  
cell  $S_1$ , 30% in  $S_s$  and 40% in  
 $S_3$ ....

# Life in an Uncertain World

We might not know the effects of an action

- The action might have a random component, like rolling dice.
- We might not know the long term effects of a drug.
- We might not know the status of a road when we choose to drive down it.

We may not know exactly what state we are in

- E.g., we can't see our opponents cards in a poker game.
- We don't know what a patient's ailment is.

 - We may still need to act, but we can't act solely on the basis of facts. We have to “gamble”.

# Uncertainty

But how do we gamble rationally?

- If we must arrive at the airport at 9pm on a week night we could “safely” leave for the airport  $\frac{1}{2}$  hour before.

Some probability of the trip taking longer, but the probability is low.

- If we must arrive at the airport at 4:30pm on Friday we most likely need 1 hour or more to get to the airport.

Relatively high probability of it taking 1.5 hours.

- Acting rationally under uncertainty means we must parameterize the likelihood of our actions’ outcomes and the relative value of these outcomes. We will do this with probability.

Events, Event Spaces

$$(X, Y, Z)$$

Probability of an Event, Axioms of Probability

$$P(X = \text{low}, Y = \text{morning}, Z = \text{me}) = 0.1$$

Conditional Probability

$$P(Z = \text{me} | Y = \text{morning}) = \frac{P(Z = \text{me}, Y = \text{morning})}{P(Y = \text{morning})}$$

Independence of Probabilities

$$P(Z = \text{me} | Y = \text{morning}) = P(Z = \text{me}) \quad Z \perp Y$$

Conditional Independence of Probabilities

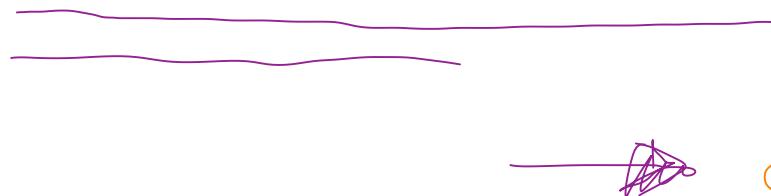
$$P(Z = \text{me} | Y = \text{morning}, X = \text{low}) = P(Z = \text{me} | Y = \text{morning}) \quad Z \perp X | Y$$

## The Chain Rule

$$P(\underline{\underline{X}}, \underline{\underline{Y}}, \underline{\underline{Z}}) = P(\underline{X} | \underline{Y} \underline{Z}) \cdot P(\underline{Y} | \underline{Z}) \cdot P(\underline{Z}) =$$

$$P(\underline{\underline{Z}} | \underline{\underline{Y}} \underline{\underline{X}}) \cdot P(\underline{\underline{Y}} | \underline{\underline{X}}) \cdot P(\underline{\underline{X}})$$

Joint Probability Table, Joint Distribution



Marginalization of a Joint Probability

$$\underline{\underline{P(X)}} = \sum_{\underline{\underline{Y}}} \sum_{\underline{\underline{Z}}} P(\underline{\underline{X}}, \underline{\underline{Y}}, \underline{\underline{Z}})$$

Bayes' Rule

$$P(\underline{\underline{X}} | \underline{\underline{Y}}) = \frac{P(\underline{\underline{Y}} | \underline{\underline{X}}) \cdot P(\underline{\underline{X}})}{P(\underline{\underline{Y}})}$$

Normalization of Probabilities

$$P(\underline{\underline{X}} = \underline{\underline{\text{low}}} | \underline{\underline{Y}} = \underline{\underline{\text{pm}}}) =$$

$$\frac{P(\underline{\underline{X}} = \underline{\underline{\text{low}}}, \underline{\underline{Y}} = \underline{\underline{\text{pm}}})}{P(\underline{\underline{Y}} = \underline{\underline{\text{pm}}})}$$

$$\underline{P(X|Y) = P(X)}$$

$$P(\underline{X}, \underline{Y}) = \underline{P(X)} \cdot \underline{P(Y)}$$

| X | Y | Z |  | $P(X, Y, Z)$ |
|---|---|---|--|--------------|
| 1 | 0 | 0 |  | 0,1          |
| 0 | 1 | 0 |  | 0,2          |
|   |   |   |  |              |

N binary variables

$\underline{2^N}$

$\underline{2^h}$

$$P(\underline{X}, \underline{Y}, \underline{Z}) =$$

$$2 \rightarrow \underline{\underline{P(X)}} \cdot$$

$$2 \rightarrow \underline{\underline{P(Y)}} \cdot$$

$$2 \rightarrow \underline{\underline{P(Z)}} \cdot$$

- 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.

# Bayesian Networks

... are tools to both  
represent uncertainty in event  
spaces and to reason about the  
likelihood of events.

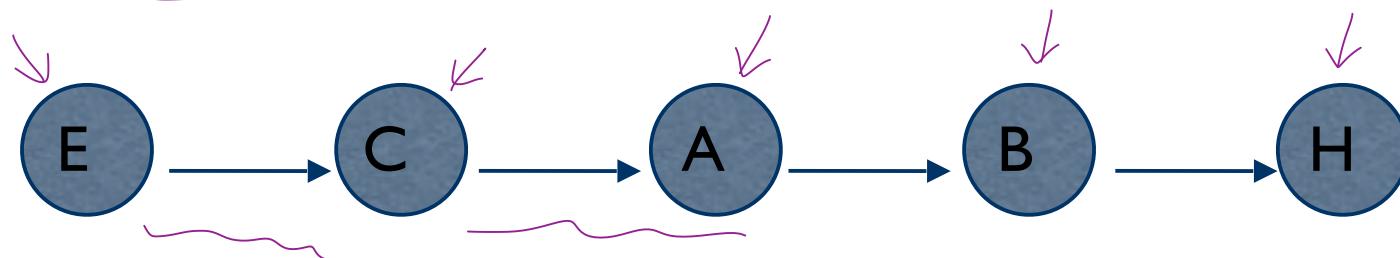
# Exploiting Conditional Independence

Let's see what conditional independence buys us, computationally

Consider a story:

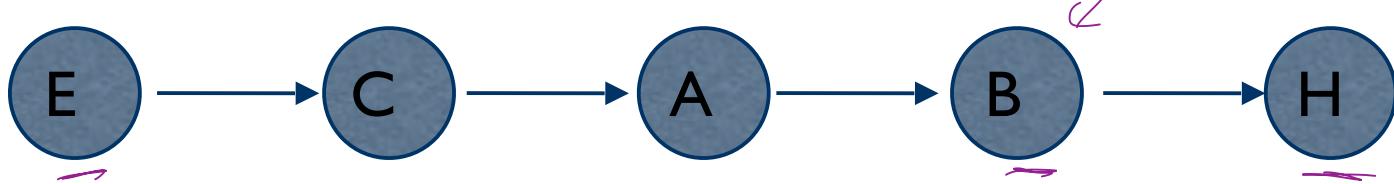
$$P(c | E) = P(c)$$

"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.

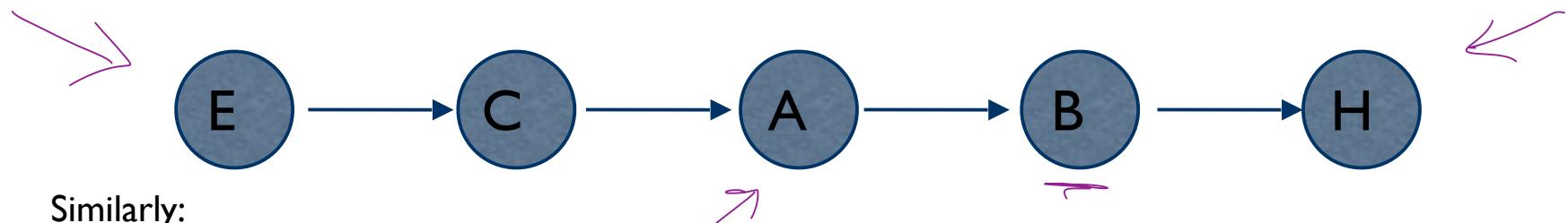
- E.g., if any of these are seen to be true, you would increase  $P(H)$  and decrease  $P(\sim H)$ .
- This means H is not independent of E, or C, or A, or B.

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

$$H \perp E, C, A \mid B$$

# Cond'l Independence in our Story



Similarly:

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

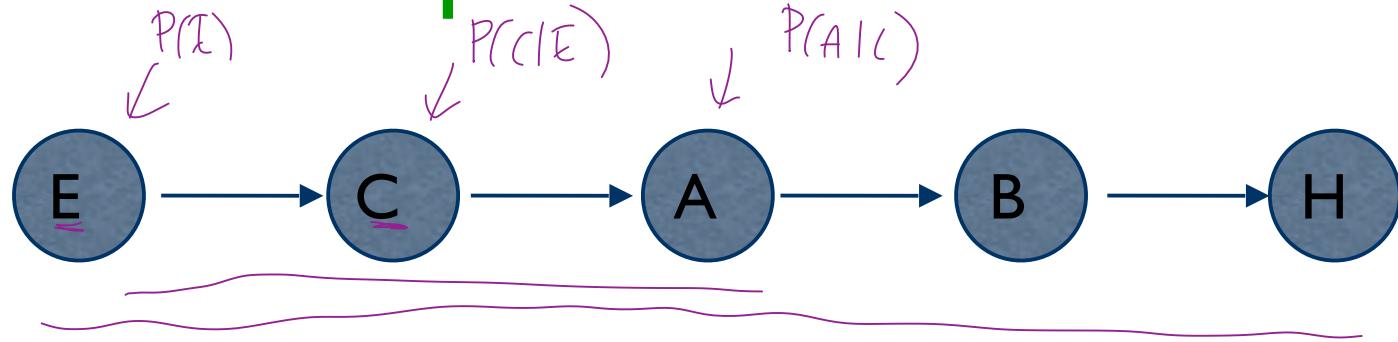
This means that:

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

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  $H \dots E$ ):

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

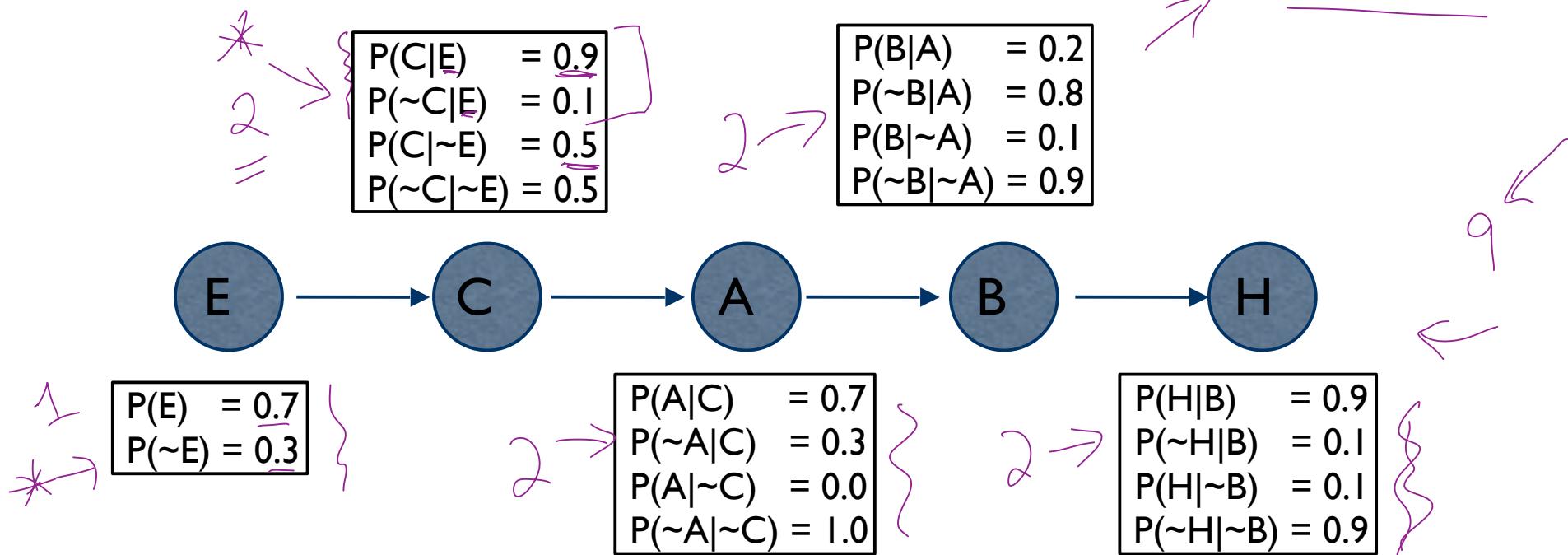
By our independence assumptions:

$$\underbrace{P(\underline{H}, \underline{B}, \underline{A}, \underline{C}, \underline{E})}_{=} = \underbrace{P(\underline{H}|\underline{B})}_{=} \underbrace{P(\underline{B}|\underline{A})}_{=} \underbrace{P(\underline{A}|\underline{C})}_{=} \underbrace{P(\underline{C}|\underline{E})}_{=} \underbrace{P(\underline{E})}_{=}$$

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  $P(E)$

32  
=

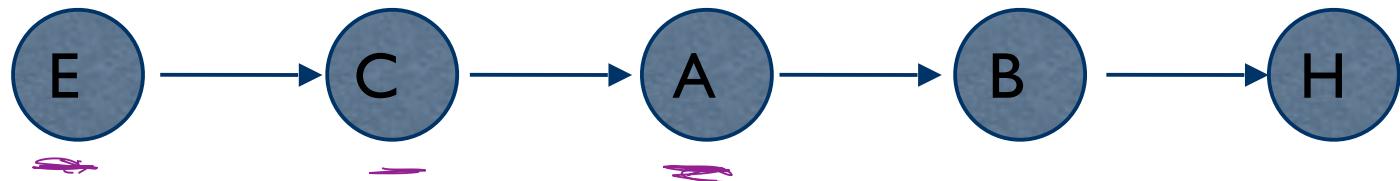
$P(E, C, B, A, H)$



Specifying the joint requires only 9 parameters (if we note that half of these are "I 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  $\underline{P(A)}$ ? Proceed as follows:

$$\begin{aligned} \underline{P(a)} &= \sum_{c_i \in Dom(C)} \Pr(a | \underline{c_i}) \Pr(\underline{c_i}) \quad \leftarrow \begin{array}{l} \sum_{c_i} P(a, c_i) \\ \sum_{c_i} P(c_i) \end{array} \\ &= \sum_{c_i \in Dom(C)} \Pr(a | \underline{c_i}) \sum_{e_i \in Dom(E)} \Pr(\underline{c_i | e_i}) \Pr(\underline{e_i}) \end{aligned}$$

These are all terms specified in our local distributions!

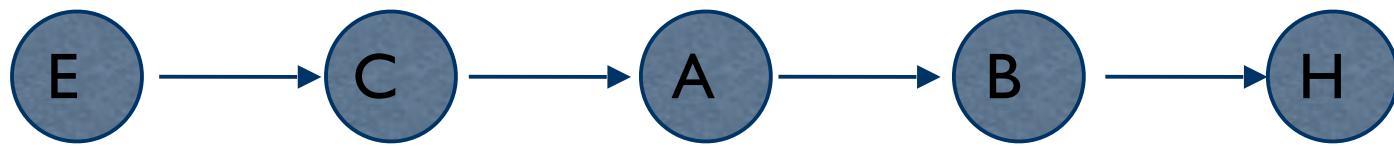
# Making Inferences

$$\begin{aligned} P(E) &= 0.7 \\ P(\sim E) &= 0.3 \end{aligned}$$

$$\begin{aligned} P(C|E) &= 0.9 \\ P(\sim C|E) &= 0.1 \\ P(C|\sim E) &= 0.5 \\ P(\sim C|\sim E) &= 0.5 \end{aligned}$$

$$\begin{aligned} P(B|A) &= 0.2 \\ P(\sim B|A) &= 0.8 \\ P(B|\sim A) &= 0.1 \\ P(\sim B|\sim A) &= 0.9 \end{aligned}$$

$$\begin{aligned} P(H|B) &= 0.9 \\ P(\sim H|B) &= 0.1 \\ P(H|\sim B) &= 0.1 \\ P(\sim H|\sim B) &= 0.9 \end{aligned}$$



Computing  $\underline{P(A)}$  in more concrete terms:  $\sum_{\text{e}} P(\text{c}, \text{e})$

✓  $P(C) = P(C|E)P(E) + P(C|\sim E)P(\sim E) = 0.9 * 0.7 + 0.5 * 0.3 = 0.78$

✓  $P(\sim C) = P(\sim C|E)P(E) + P(\sim C|\sim E)P(\sim E) = 0.22$

$P(\sim C) = 1 - P(C)$ , as well ✓

$$\begin{aligned} P(A|C) &= 0.7 \\ P(\sim A|C) &= 0.3 \\ P(A|\sim C) &= 0.0 \\ P(\sim A|\sim C) &= 1.0 \end{aligned}$$

✓  $P(A) = P(A|C)P(C) + P(A|\sim C)P(\sim C) = 0.7 * 0.78 + 0.0 * 0.22 = 0.546$

$P(\sim A) = 1 - P(A) = 0.454$

# 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  $\{X_1, X_2, \dots, X_n\}$  consists of:

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

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

parents of a node:  $\text{Par}(X_i)$

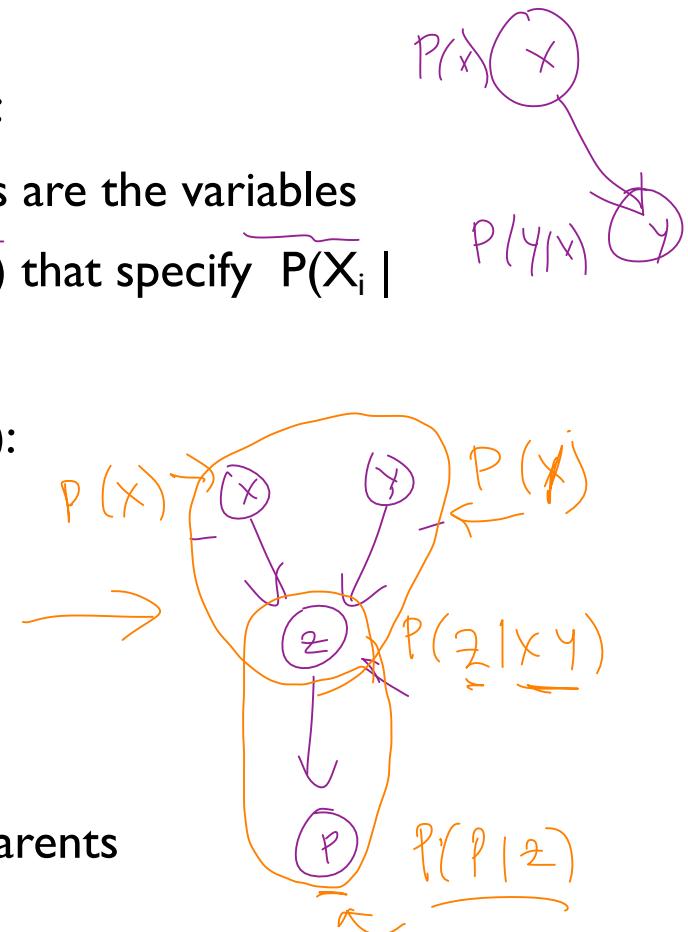
children of node

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

$$P(A, S, L)$$

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

| S | A | L |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | F |
| T | F | T |
| F | F | T |
| T | F | T |
| F | F | T |
| T | F | T |

P(L|S,A)  
P(L|S,A)  
0,1  
0,2  
0,6  
0,8

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  $P(\underline{L|S,A})$  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

$$P(S, A, L)$$

$$\nabla P(S|A) = P(S), \underline{P(S)} = 0.3, \underline{P(A)} = 0.5$$

$$\nabla \underline{P(L|S,A)} = 0.05, \underline{P(L|S,\sim A)} = 0.1, \underline{P(L|\sim S,A)} = 0.1, \underline{P(L|\sim S,\sim A)} = 0.2$$

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

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

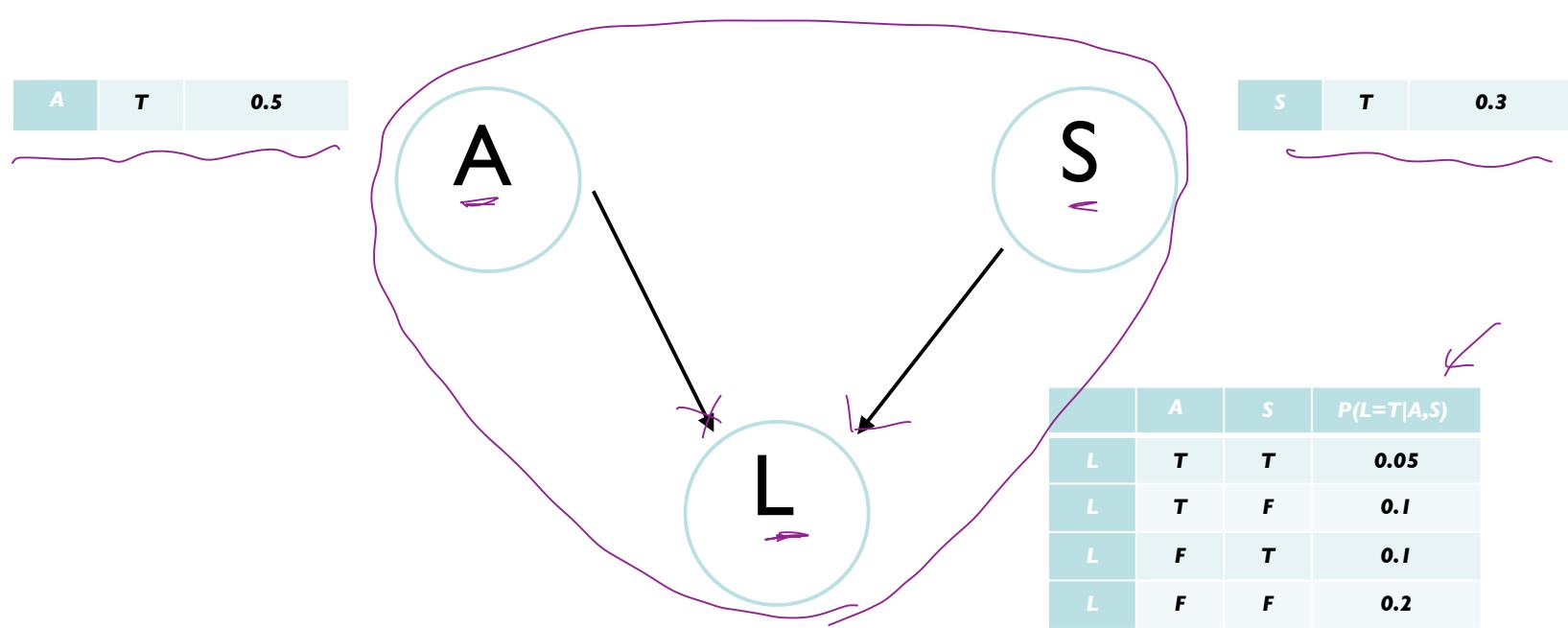
$$P(L|S,A) = 0.05, P(L|S,\sim A) = 0.1, P(L|\sim S,A) = 0.1, P(L|\sim S,\sim A) = 0.2$$

How can we calculate  $\underbrace{P(L,S,A)}$ ? Or  $\underbrace{P(L,\sim S,A)}$ ?

# Drawing the Network

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

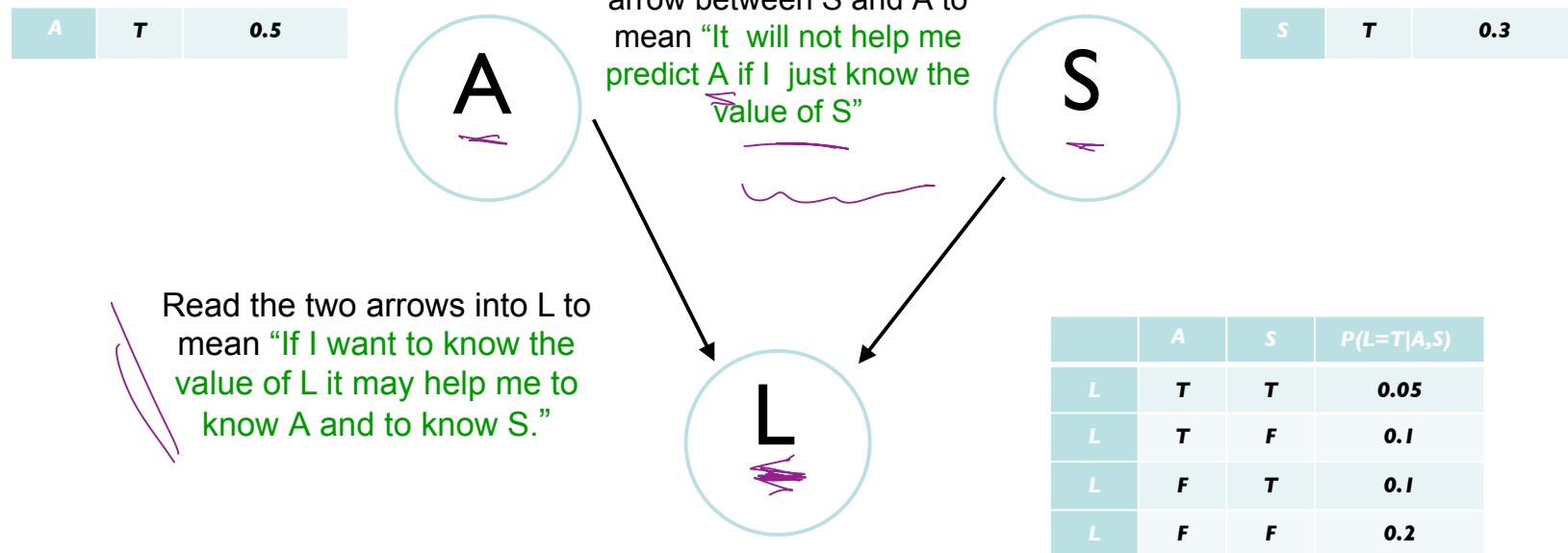
$$P(L|S,A) = 0.05, P(L|S,\sim A) = 0.1, P(L|\sim S,A) = 0.1, P(L|\sim S,\sim A) = 0.2$$



# Drawing the Network

|                   |                                     |
|-------------------|-------------------------------------|
| $P(S   M) = P(S)$ | $P(L   M \wedge S) = 0.05$          |
| $P(S) = 0.3$      | $P(L   M \wedge \neg S) = 0.1$      |
| $P(M) = 0.6$      | $P(L   \neg M \wedge S) = 0.1$      |
|                   | $P(L   \neg M \wedge \neg S) = 0.2$ |

$$P(A|S) = P(A)$$



# 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?

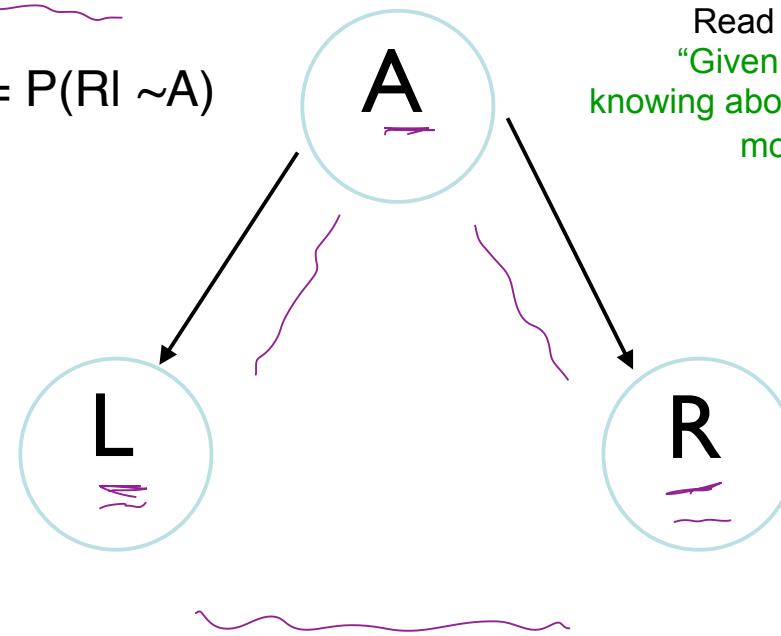
---

# 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.:

$$\underbrace{P(RIA, L)}_{\sim} = \underbrace{P(RIA)}_{\sim}$$

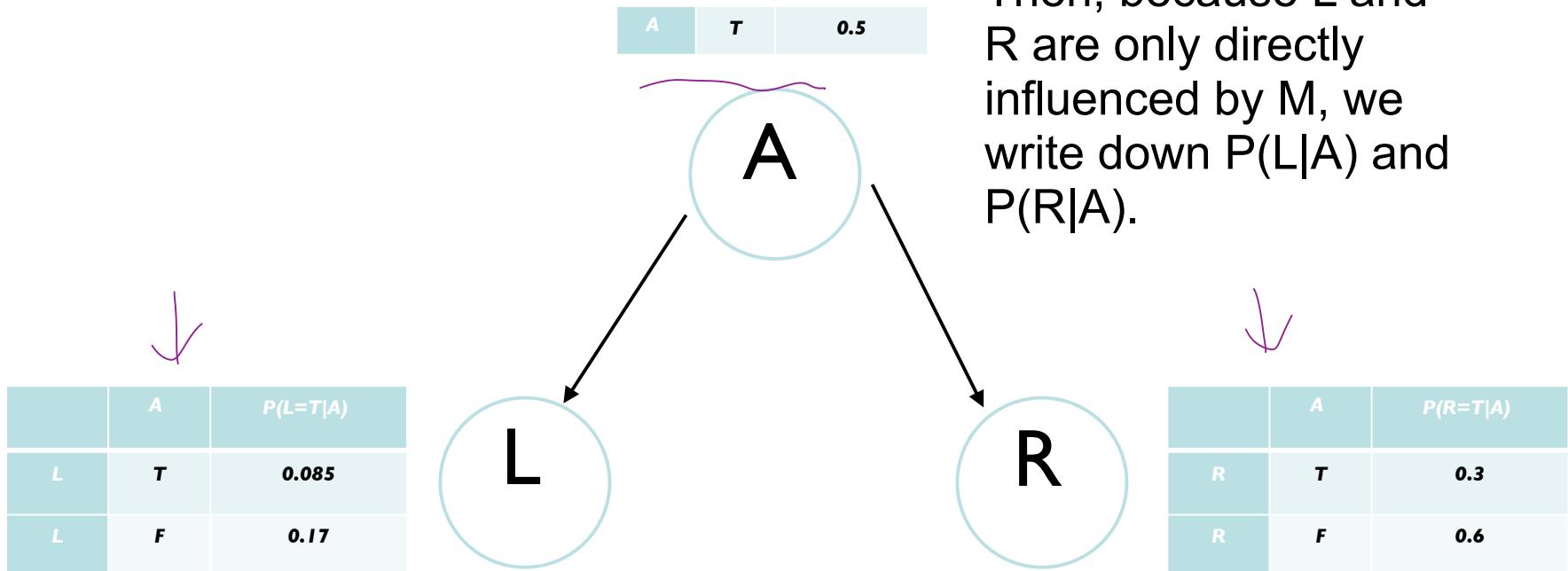
$$P(RI \sim A, L) = P(RI \sim A)$$



Read this diagram as  
“Given knowledge of A,  
knowing about L won’t tell anything  
more about R.”

# The network reflects conditional independences

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  $P(R|A)$ .



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

# The network reflects conditional independences

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

$$P(L|R,A) \cdot P(R|A) \cdot P(A) = \begin{array}{c} A \\ T \\ 0.6 \end{array}$$

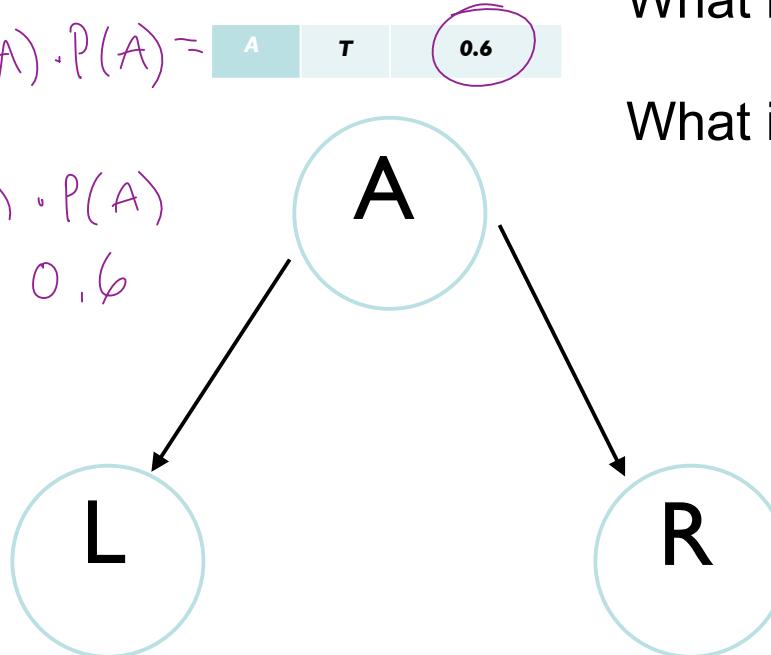
$$P(L|A) \cdot P(R|A) \cdot P(A)$$

$$0.085 \times 0.3 = 0.6$$

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

What is  $P(\sim L,R,A)$ ?

|   | 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)

# 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

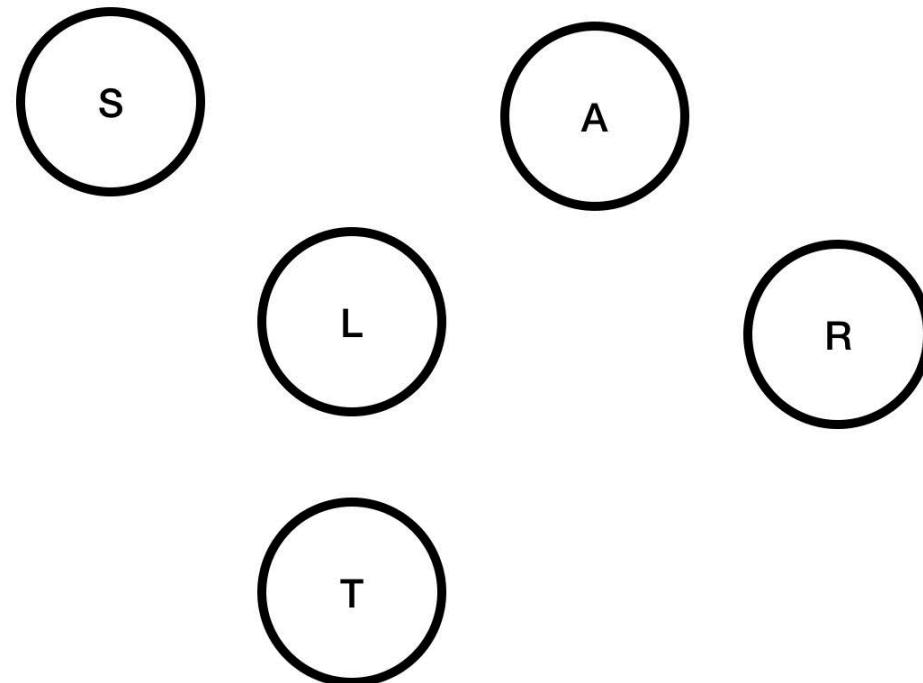
L: The lecturer arrives late

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

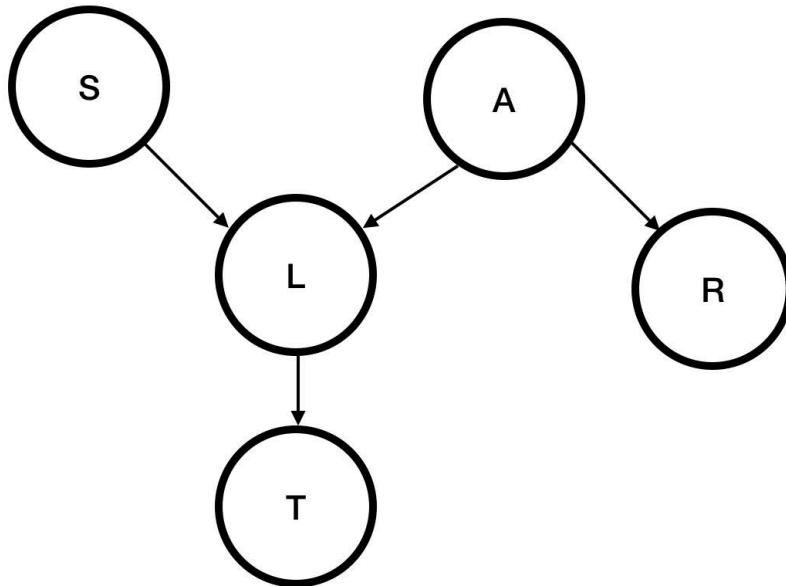
S: It is sunny out

T: Lecture starts < 15 minutes past.

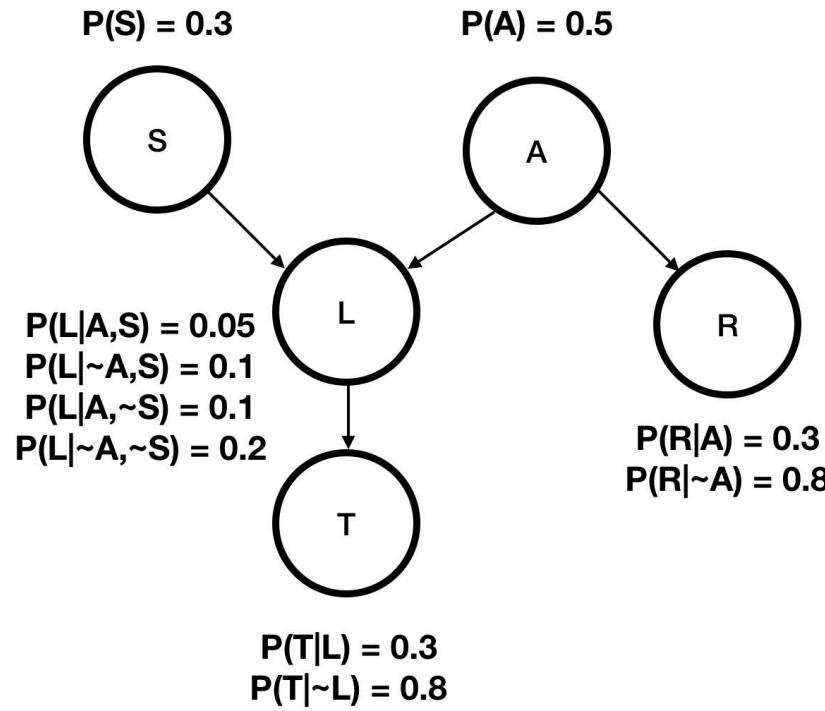
Step Two: add links.

The link structure must be acyclic.

If you assign node Y the parents  $X_1, X_2, \dots, X_n$ , you are promising that, given  $\{X_1, X_2, \dots, X_n\}$ , 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

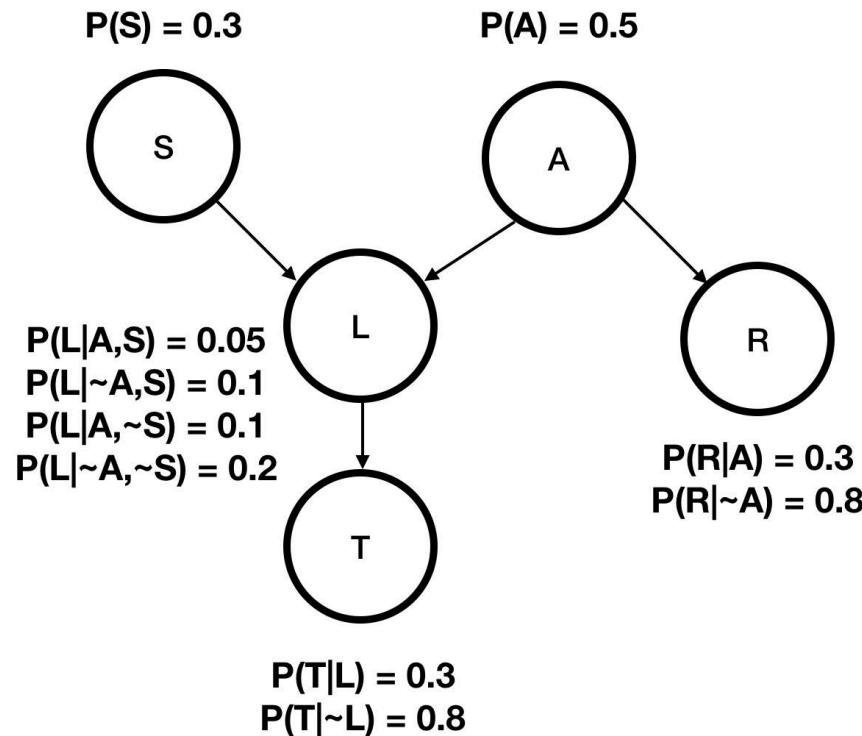
S: It is sunny out

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  $X_1, X_2, \dots, X_n$ , using any ordering of the variables.

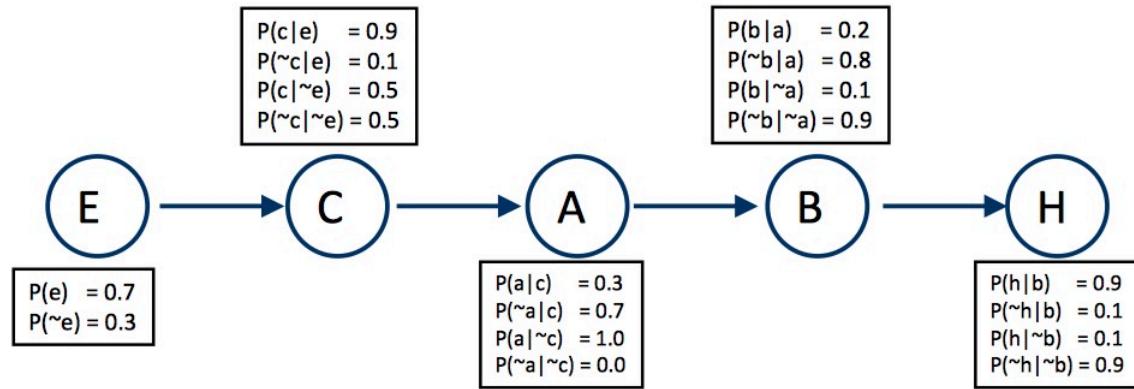
Take any ordering of the variables (say, the order given). From the chain rule we obtain.

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

Now for each  $X_i$  go through its conditioning set  $X_1, \dots, X_{i-1}$ , 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.

# Different kinds of Bayes Nets reflect different conditional independence relationships.



This one models a sequence of conditional independence relationships.

# Sequential Models

- ▶ sequences of observations occur in many domains:
  - ▶ Speech recognition
  - ▶ Robot localization
  - ▶ User attention
  - ▶ Medical monitoring



# Markov Models capture sequences

- Say we have one variable  $X$  (perhaps with a very large number of possible value assignments).
- We want to track the probability of different values of  $X$  (i.e. the probability distribution over  $X$ ) as its values change over time.
- Possible solution: Make multiple copies of  $X$ , one for each time point (we assume a discrete model of time):  $X_1, X_2, X_3 \dots X_t$
- A Markov Model is specified by the two following assumptions:
  - The current state  $X_t$  is conditionally independent of the earlier states given the previous state.

$$P(X_t | X_{t-1}, X_{t-2}, \dots, X_1) = P(X_t | X_{t-1})$$

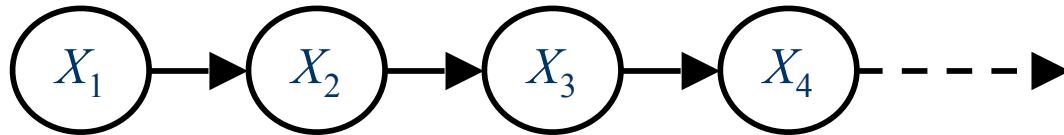
- The transitions between  $X_{t-1}$  and  $X_t$  are determined by probabilities that do not change over time (they are stationary probabilities).

$$P(X_t | X_{t-1})$$

---

# Markov Models

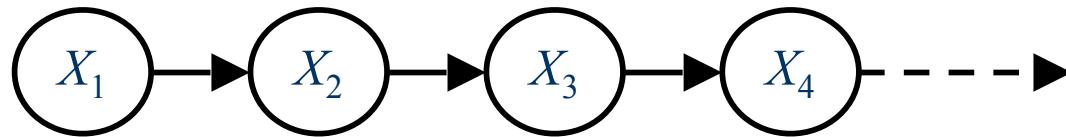
- ▶ These assumptions give rise to a Bayesian Network that looks like this:



- ▶  $P(X_1, X_2, X_3, \dots) = P(X_1)P(X_2|X_1)P(X_3|X_2) \dots$  (Assumption 1)
- ▶ All the CPTs (except  $P(X_1)$ ) are the same (Assumption 2)

---

# Markov Models



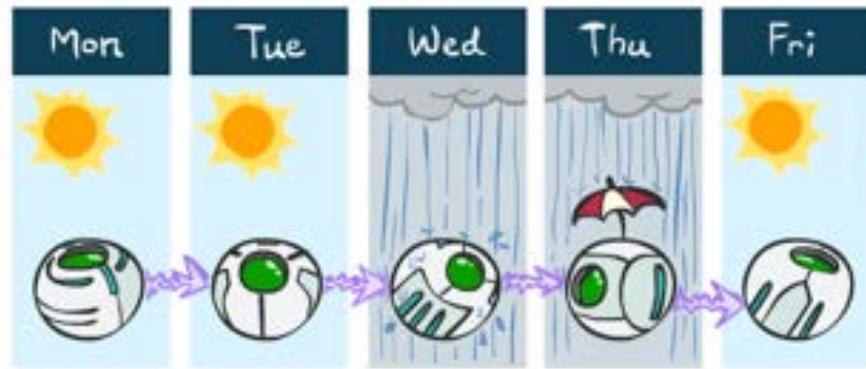
This model reflects a specific set of conditional independence relationships:

$X_{t-1}$  is conditionally independent of  $X_{t+1}, X_{t+2}, \dots$  given  $X_t$

The current state separates the past from the future.

---

# Example Markov Chain Weather



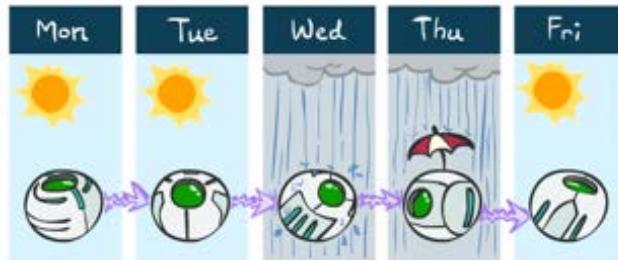
- ▶ States:  $X = \{\text{rain}, \text{sun}\}$
- Initial distribution:  
 $P(X_1=\text{sun}) = 1.0$

CPT  $P(X_t | X_{t-1})$ :

| $X_{t-1}$ | $X_t$ | $P(X_t   X_{t-1})$ |
|-----------|-------|--------------------|
| sun       | sun   | 0.9                |
| sun       | rain  | 0.1                |
| rain      | sun   | 0.3                |
| rain      | rain  | 0.7                |

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley].

# Example Markov Chain Weather



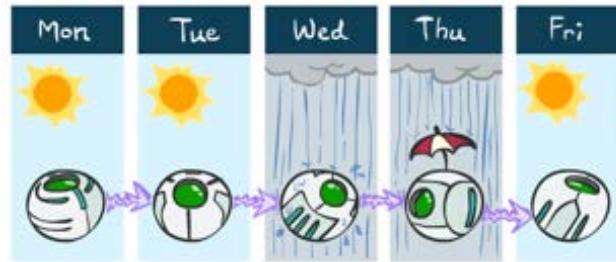
- ▶  $P(X_1 = \text{sun}) = 1.0$
- ▶ What is the probability distribution after one step,  $P(X_2)$ ?
- ▶ Use summing out rule with  $X_1$

$$P(X_2 = \text{sun}) = P(X_2 = \text{sun} | X_1 = \text{sun})P(X_1 = \text{sun}) + P(X_2 = \text{sun} | X_1 = \text{rain})P(X_1 = \text{rain})$$
$$0.9 \cdot 1.0 + 0.3 \cdot 0.0 = 0.9$$

CPT  $P(X_t | X_{t-1})$ :

| $X_{t-1}$ | $X_t$ | $P(X_t   X_{t-1})$ |
|-----------|-------|--------------------|
| sun       | sun   | 0.9                |
| sun       | rain  | 0.1                |
| rain      | sun   | 0.3                |
| rain      | rain  | 0.7                |

# Example Markov Chain Weather



- ▶ What is the probability distribution on day t ( $P(X_t)$ )?
- ▶ Sum out  $X_{t-1}$

$P(x_1)$  = known

$$\begin{aligned} P(x_t) &= \sum_{x_{t-1}} P(x_{t-1}, x_t) \\ &= \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1}) \end{aligned}$$

*Forward simulation*

Compute  $P(X_2)$  then  $P(X_3)$  then  $P(X_4)$  ...

CPT  $P(X_t | X_{t-1})$ :

| $X_{t-1}$ | $X_t$ | $P(X_t   X_{t-1})$ |
|-----------|-------|--------------------|
| sun       | sun   | 0.9                |
| sun       | rain  | 0.1                |
| rain      | sun   | 0.3                |
| rain      | rain  | 0.7                |

# Example Run of Forward Computation

- From initial observation of sun

$$\begin{array}{ccccc} \left\langle \begin{array}{c} 1.0 \\ 0.0 \end{array} \right\rangle & \left\langle \begin{array}{c} 0.9 \\ 0.1 \end{array} \right\rangle & \left\langle \begin{array}{c} 0.84 \\ 0.16 \end{array} \right\rangle & \left\langle \begin{array}{c} 0.804 \\ 0.196 \end{array} \right\rangle & \xrightarrow{\hspace{1cm}} \left\langle \begin{array}{c} 0.75 \\ 0.25 \end{array} \right\rangle \\ P(X_1) & P(X_2) & P(X_3) & P(X_4) & P(X_\infty) \end{array}$$

- From initial observation of rain

$$\begin{array}{ccccc} \left\langle \begin{array}{c} 0.0 \\ 1.0 \end{array} \right\rangle & \left\langle \begin{array}{c} 0.3 \\ 0.7 \end{array} \right\rangle & \left\langle \begin{array}{c} 0.48 \\ 0.52 \end{array} \right\rangle & \left\langle \begin{array}{c} 0.588 \\ 0.412 \end{array} \right\rangle & \xrightarrow{\hspace{1cm}} \left\langle \begin{array}{c} 0.75 \\ 0.25 \end{array} \right\rangle \\ P(X_1) & P(X_2) & P(X_3) & P(X_4) & P(X_\infty) \end{array}$$

- From yet another initial distribution  $\Pr(X_1)$ :

$$\begin{array}{ccc} \left\langle \begin{array}{c} p \\ 1 - p \end{array} \right\rangle & \dots & \xrightarrow{\hspace{1cm}} \left\langle \begin{array}{c} 0.75 \\ 0.25 \end{array} \right\rangle \\ P(X_1) & & P(X_\infty) \end{array}$$

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.]

# Stationary Distributions

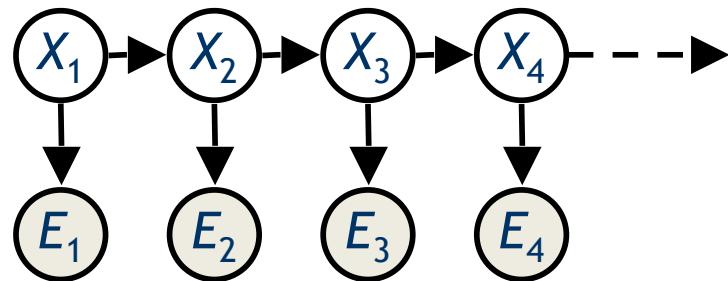
- ▶ For most Markov chains:
  - ▶ Influence of the initial distribution dissipates over time.
  - ▶ The distribution we end up in is independent of the initial distribution
- Stationary distribution
  - The distribution that we end up with is called the stationary distribution of the chain.
  - This satisfies:

$$P_\infty(X) = P_{\infty+1}(X) = \sum_x P(X|x)P_\infty(x)$$

- That is the stationary distribution does not change on a forward progression
  - We can compute it by solving simultaneous equations (or by forward simulating the system many times; forward simulation is generally computationally more effective)
-

# Hidden Markov Models

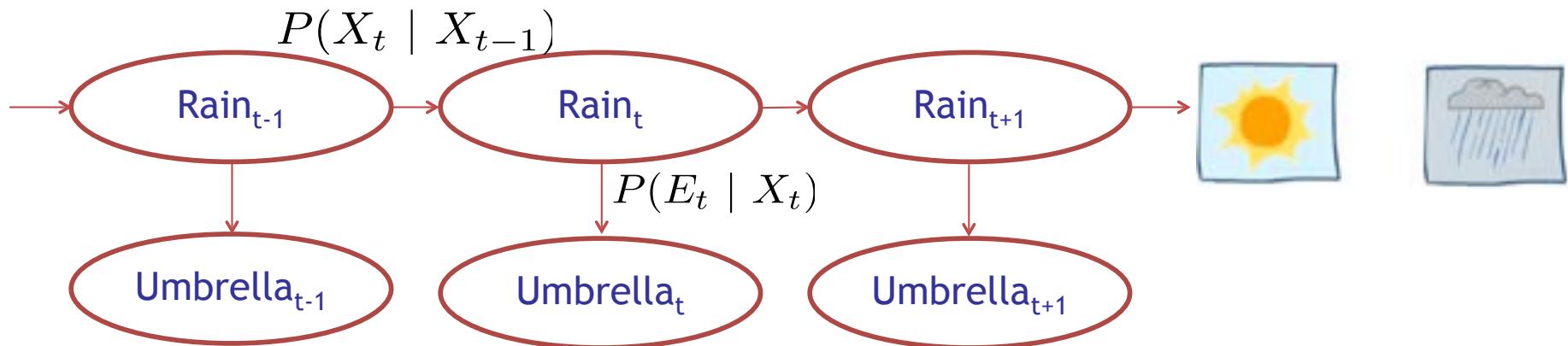
- ▶ Markov chains not so useful for most agents
  - ▶ Need observations to update your beliefs
- ▶ Hidden Markov models (HMMs)
  - ▶ Underlying Markov chain over states  $X$
  - ▶ But you also observe outputs (effects) at each time step



[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley].

---

# Example: Weather HMM



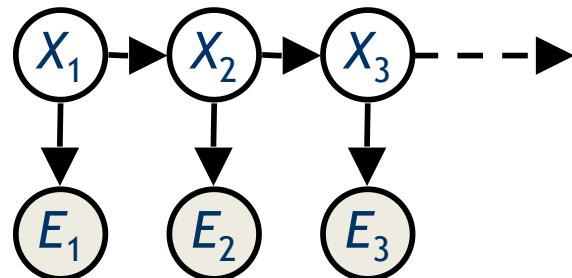
- ▶ An HMM is defined by:
  - ▶ Initial distribution:  $P(X_1)$
  - ▶ Transitions:  $P(X_t | X_{t-1})$
  - ▶ Emissions:  $P(E_t | X_t)$

| $R_t$ | $R_{t+1}$ | $P(R_{t+1}   R_t)$ |
|-------|-----------|--------------------|
| $+r$  | $+r$      | 0.7                |
| $+r$  | $-r$      | 0.3                |
| $-r$  | $+r$      | 0.3                |
| $-r$  | $-r$      | 0.7                |

| $R_t$ | $U_t$ | $P(U_t   R_t)$ |
|-------|-------|----------------|
| $+r$  | $+u$  | 0.9            |
| $+r$  | $-u$  | 0.1            |
| $-r$  | $+u$  | 0.2            |
| $-r$  | $-u$  | 0.8            |

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley].

# Joint Distribution of an HMM



- Assumptions:

- $P(X_t | X_{t-1} \dots X_1, E_{t-1}, \dots, E_1) = P(X_t | X_{t-1})$

Current state is conditionally independent of early states + evidence given previous state

- $P(X_t | X_{t-1})$  is the same for all time points  $t$

Probabilities are stationary

- $P(E_t | X_t, X_{t-1}, \dots, X_1, E_{t-1}, \dots, E_1) = P(E_t | X_t)$

Current evidence is conditionally independent of early states + early evidence given current state

*Note that two evidence items are not independent, unless one of the intermediate states is known.*

---

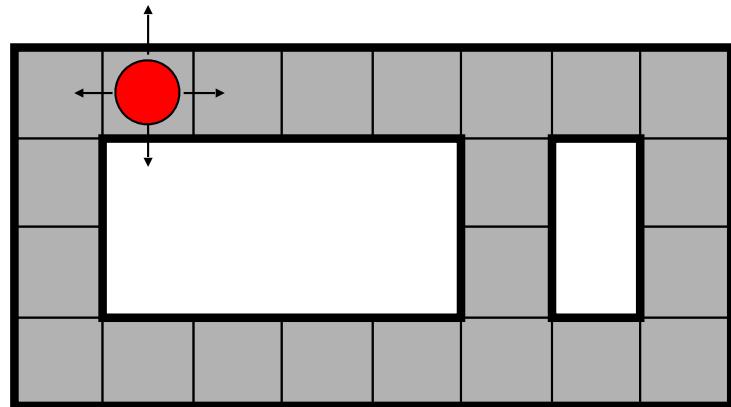
# Real HMM Examples

- ▶ Speech recognition HMMs:
  - ▶ Observations are acoustic signals (continuous valued)
  - ▶ States are specific positions in specific words (so, tens of thousands)
- ▶ Machine translation HMMs:
  - ▶ Observations are words (tens of thousands)
  - ▶ States are translation options
- ▶ Robot tracking:
  - ▶ Observations are range readings (continuous)
  - ▶ States are positions on a map (continuous)

# Tracking/Monitoring

- Monitoring is the task of tracking  $P(X_t | e_t \dots e_1)$  over time. i.e. determining state given current and previous observations.
  - $P(X_1)$  is the initial distribution over variable (or feature) X. Usually start with a uniform distribution over all values of X.
  - As time elapses and we make observations and must update our distribution over X, i.e. move from  $P(X_{t-1} | e_{t-1} \dots e_1)$  to  $P(X_t | e_t \dots e_1)$ .
  - This means updating HMM equations. Tools to do this existed before Bayes Nets, but we can relate inference tools to Variable Elimination.
-

# Example: Robot Localization



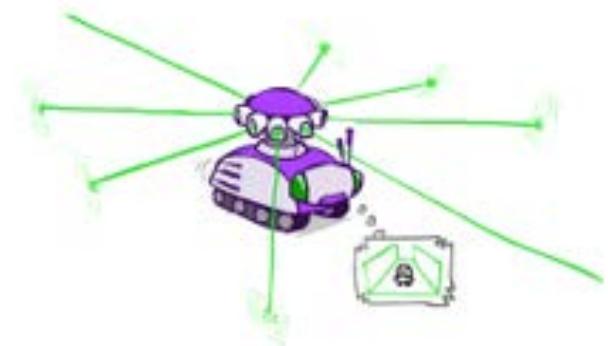
t=0

Sensor model: Can read in which directions there is a wall, never more than 1 mistake

Motion model: Either executes the move, or the robot with low probability does not move at all. Cannot move in wrong direction.

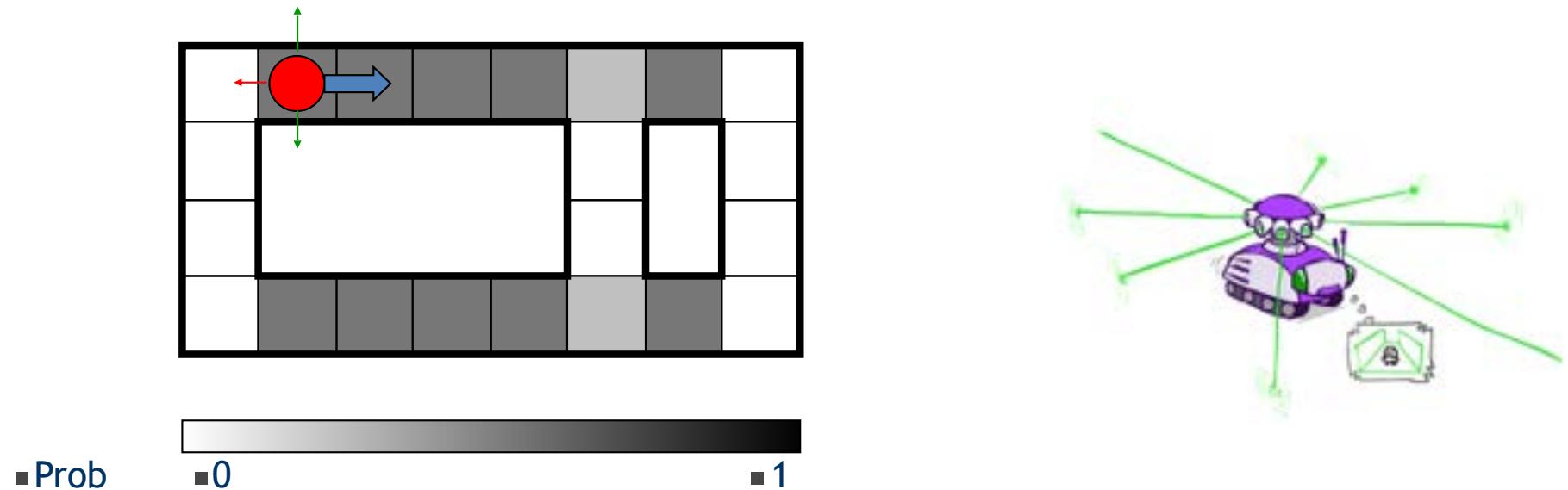
Initially uniform distribution over where robot is located—equally likely to be anywhere.

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. ]



*Example from  
Michael Pfeiffer*

# Example: Robot Localization



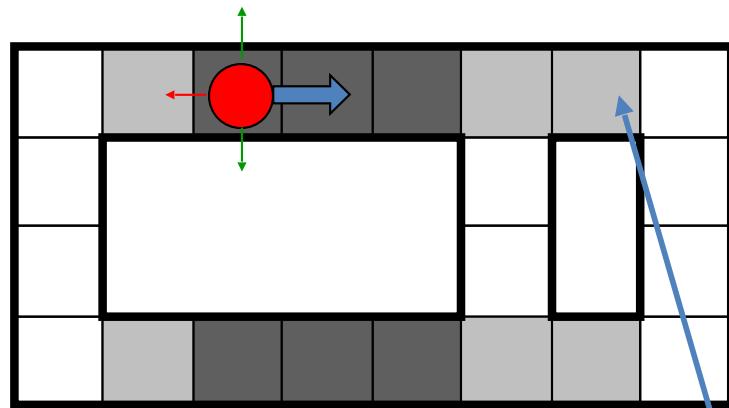
Initially don't know where you are. Observe a wall above and below, no wall to the left or right. Low probability of 1 mistake, 2 mistakes not possible

White: impossible to get this reading (more than one mistake)

Lighter grey: was possible to get the reading, but less likely because it required 1 mistake

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. ]

# Example: Robot Localization

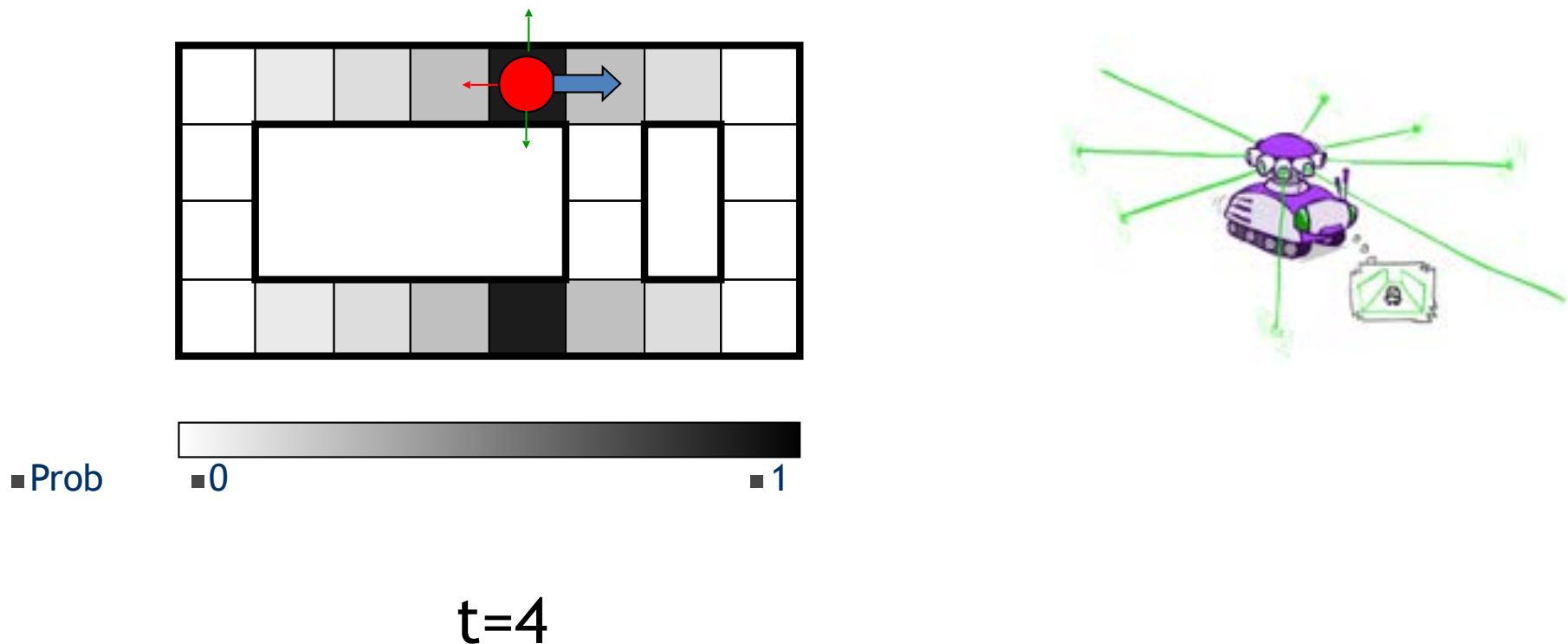


t=2: Move right. Low probability didn't move, else must have moved right.

Still observing wall above and below

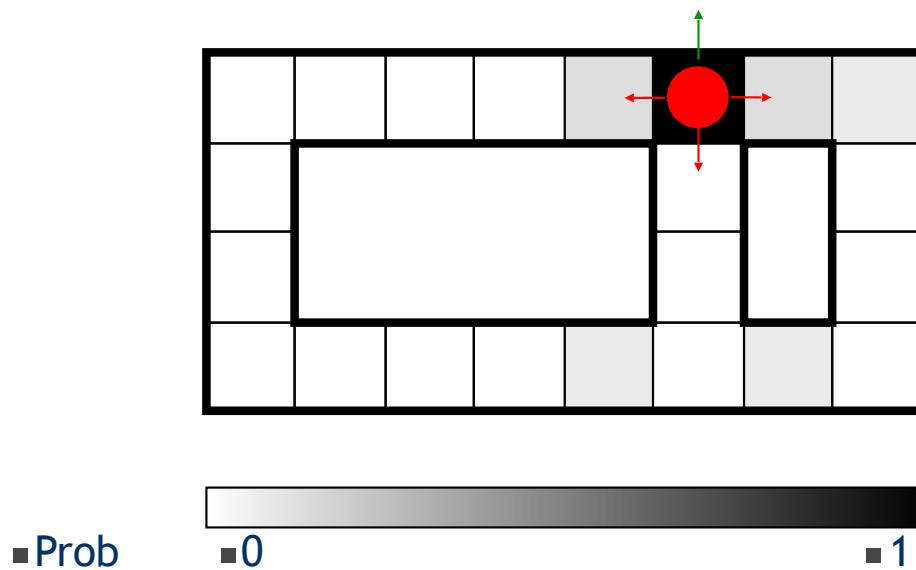
- can only be here if
- (a) was at low probability square to the left
  - (b) was at this square and action didn't work.

# Example: Robot Localization

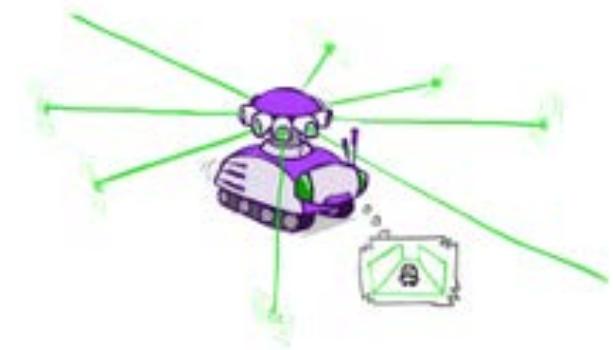


[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.]

# Example: Robot Localization

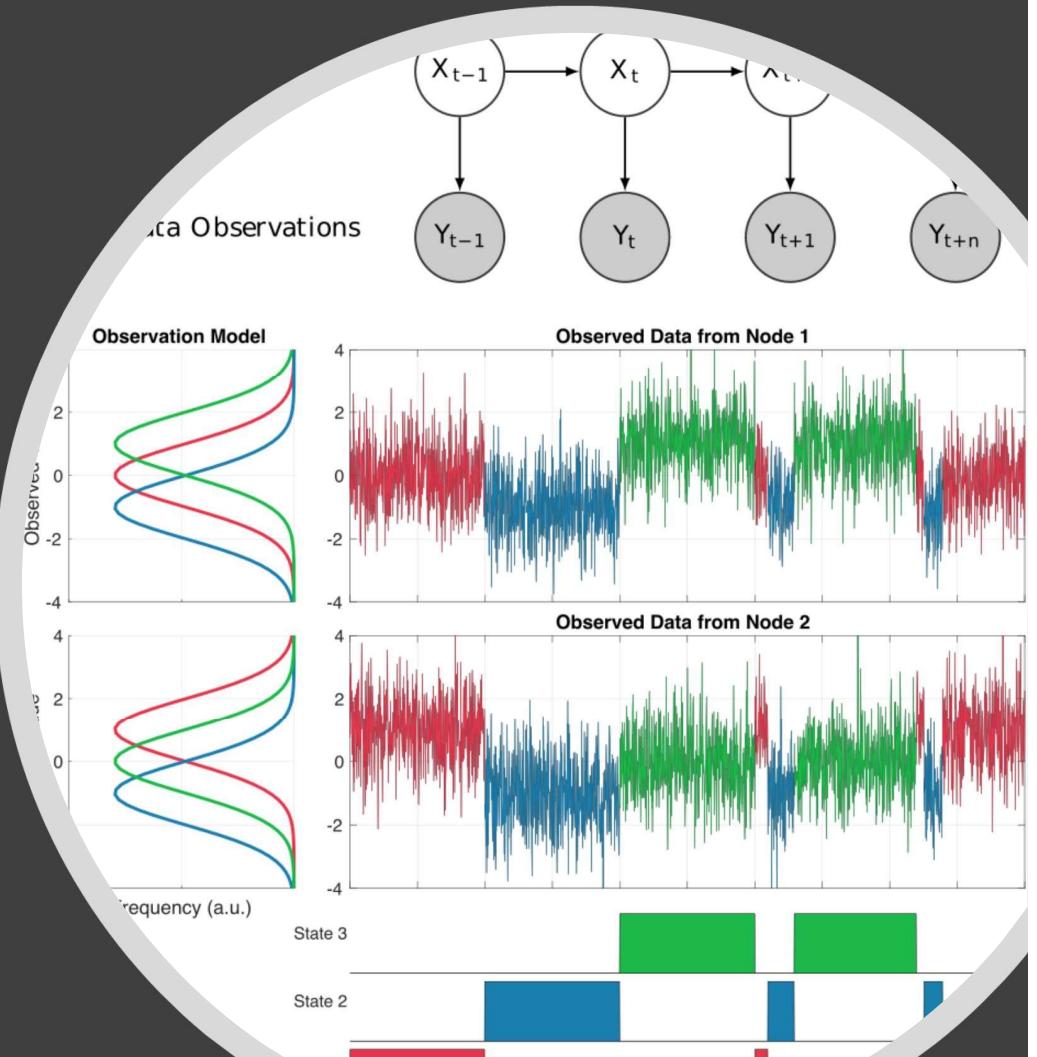


$t=5$



[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.]

# Update Rules for HMMs



# Forward Algorithm for HMMs

- Given a set of observations, we want to create a **belief** that tells us what state we think we're in.
- With HMMs, we can use the **Forward Algorithm** to calculate beliefs that are conditioned on prior observations.

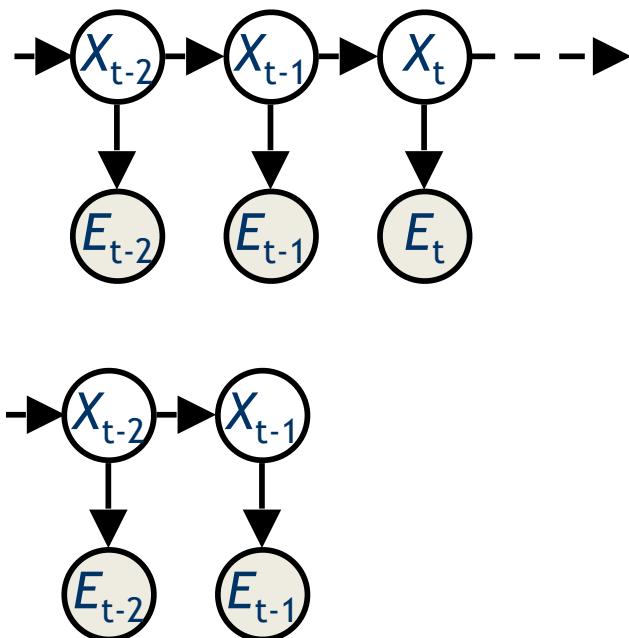
This algorithm is a special version of **Variable Elimination (VE)**, which we will cover in more depth shortly. VE can be applied to any Bayes Net, not just Markov Models.

- The structure of the HMM allows for simple update rules:

We will store beliefs about state at time  $t$ , and use these to calculate beliefs at time  $(t+1)$ .

# Forward Algorithm for HMMs

- To create a distribution of beliefs about  $X_{t-1}$ , we need only look at the variables that are ancestors of  $X_{t-1}$



# Forward Algorithm for HMMs

We want

$$P(X_{t-1} | e_{t-1}, e_{t-2} \dots e_1) = P(X_{t-1}, e_{t-1}, e_{t-2} \dots e_1) / P(e_{t-1}, e_{t-2} \dots e_1).$$

We calculate the numerator by marginalizing  $X_1, X_2 \dots X_{t-2}$  from  $P(X_{t-1}, X_{t-2}, X_{t-3} \dots X_1, e_{t-1}, e_{t-2} \dots e_1)$ .

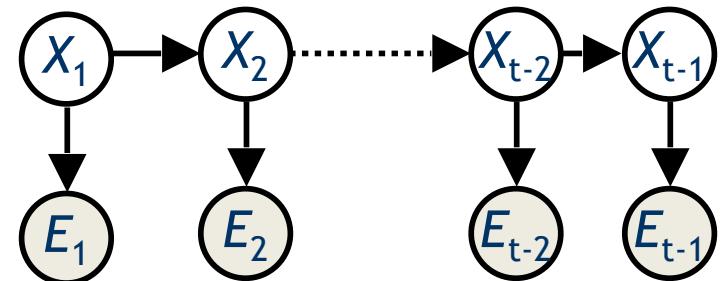
$$X_1: P(X_1) P(e_1 | X_1) P(X_2 | X_1)$$

$$X_2: P(e_2 | X_2) P(X_3 | X_2)$$

...

$$X_{t-2}: P(e_{t-2} | X_{t-2}) P(X_{t-1} | X_{t-2})$$

$$X_{t-1}: P(e_{t-1} | X_{t-1})$$



# Forward Algorithm for HMMs

Summing out  $X_1$  we get a factor of  $X_2$ ; summing out  $X_2$  we get a factor of  $X_3$  and so on:

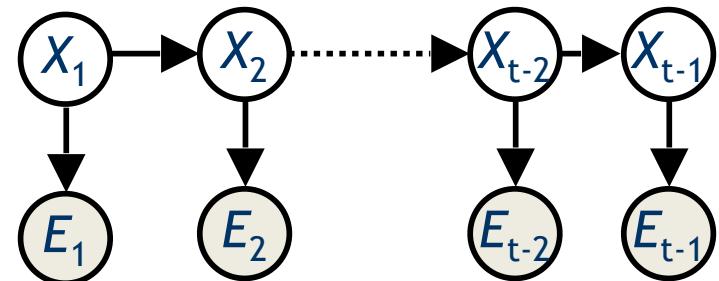
$$X_1: P(X_1) P(e_1 | X_1) P(X_2 | X_1)$$

$$X_2: P(e_2 | X_2) P(X_3 | X_2) F_2(X_2)$$

...

$$X_{t-2}: P(e_{t-2} | X_{t-2}) P(X_{t-1} | X_{t-2}) F_{t-2}(X_{t-2})$$

$$X_{t-1}: P(e_{t-1} | X_{t-1}) F_{t-1}(X_{t-1})$$



# Forward Algorithm for HMMs

$$X_1: P(X_1) P(e_1 | X_1) P(X_2 | X_1)$$

$$X_2: P(e_2 | X_2) P(X_3 | X_2) F_2(X_2)$$

...

$$X_{t-2}: P(e_{t-2} | X_{t-2}) P(X_{t-1} | X_{t-2}) F_{t-2}(X_{t-2})$$

$$X_{t-1}: P(e_{t-1} | X_{t-1}) F_{t-1}(X_{t-1})$$

So:

$$P(X_{t-1} | e_{t-1}, e_{t-2}, \dots, e_1) = \text{normalize}(P(e_{t-1} | X_{t-1}) F_{t-1}(X_{t-1}))$$

This is a table with one value for each  $X_{t-1}$

---

# Forward Algorithm for HMMs

Now say time has passed but no observation has been made yet.

$$X_1: P(X_1) P(e_1 | X_1) P(X_2 | X_1)$$

$$X_2: P(e_2 | X_2) P(X_3 | X_2)$$

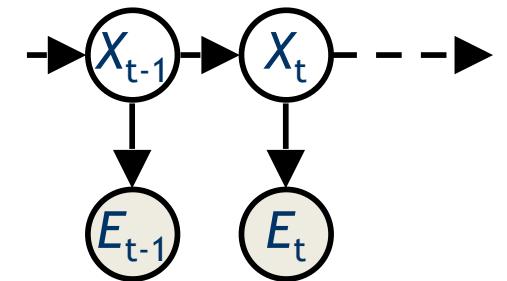
...

$$X_{t-2}: P(e_{t-2} | X_{t-2}) P(X_{t-1} | X_{t-2})$$

$$X_{t-1}: P(e_{t-1} | X_{t-1}) \text{P}(X_t | X_{t-1})$$

$$X_t:$$

Same buckets with one new one ( $X_t$ ) and one new factor ( $P(X_t | X_{t-1})$ ).



# Forward Algorithm for HMMs

Sum out variables, as before:

$$X_1: P(X_1) P(e_1 | X_1) P(X_2 | X_1)$$

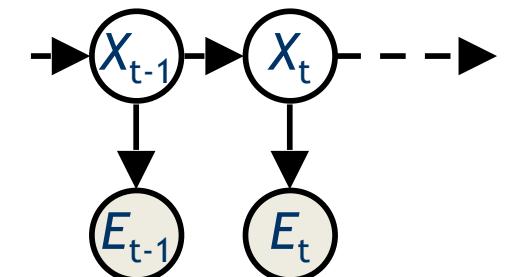
$$X_2: P(e_2 | X_2) P(X_3 | X_2) F_2(X_2)$$

...

$$X_{t-2}: P(e_{t-2} | X_{t-2}) P(X_{t-1} | X_{t-2}) F_{t-2}(X_{t-2})$$

$$X_{t-1}: P(e_{t-1} | X_{t-1}) P(X_t | X_{t-1}) F_{t-1}(X_{t-1})$$

$$X_t: F_t(X_t)$$



$$F_t(X_t) = \sum_{d \in Dom[X_{t-1}]} P(e_{t-1} | X_{t-1}) P(X_t | X_{t-1}) F_{t-1}(X_{t-1})$$

# Forward Algorithm for HMMs

We saw  $P(X_{t-1} | e_{t-1}, e_{t-2}, \dots, e_1) = \text{normalize}(P(e_{t-1} | X_{t-1}) F_{t-1}(X_{t-1}))$

Means

$$F_t(X_t) = \sum_{d \in Dom[X_{t-1}]} P(e_{t-1} | X_{t-1}) P(X_t | X_{t-1}) F_{t-1}(X_{t-1})$$

or

$$F_t(X_t) = c^* \sum_{d \in Dom[X_{t-1}]} P(X_t | X_{t-1}) P(X_{t-1} | e_{t-1}, e_{t-2}, \dots, e_1)$$

.... where  $c$  is the normalization constant.

$P(X_t | e_{t-1}, e_{t-2}, \dots, e_1) = \text{normalize}(F_t(X_t))$

$P(X_t | e_{t-1}, e_{t-2}, \dots, e_1) =$

$\text{normalize}(\sum_{d \in Dom[X_{t-1}]} P(X_t | X_{t-1}) P(X_{t-1} | e_{t-1}, e_{t-2}, \dots, e_1))$

... we drop  $c$  (because we are normalizing)

---

# Forward Algorithm for HMMs

How to incorporate the observation  $e_t$ ? The process looks similar:

$$X_1: P(X_1) P(e_1 | X_1) P(X_2 | X_1)$$

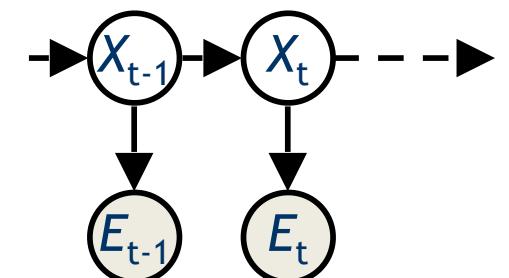
$$X_2: P(e_2 | X_2) P(X_3 | X_2) F_2(X_2)$$

...

$$X_{t-2}: P(e_{t-2} | X_{t-2}) P(X_{t-1} | X_{t-2}) F_{t-2}(X_{t-2})$$

$$X_{t-1}: P(e_{t-1} | X_{t-1}) P(X_t | X_{t-1}) F_{t-1}(X_{t-1})$$

$$X_t: F_t(X_t) \text{P}(e_t | X_t)$$



We add  $P(e_t | X_t)$  to the bucket for  $X_t$  and normalize.

---

# Forward Algorithm for HMMs

$$\text{So } P(X_t | e_t, e_{t-1}, \dots, e_1) = F_t(X_t)P(e_t | X_t)$$

We saw that

$$P(X_t | e_{t-1}, e_{t-2}, \dots, e_1) = \text{normalize}(F_t(X_t)) = c^* F_t(X_t)$$

So

$$\begin{aligned} P(X_t | e_t, e_{t-1}, e_{t-2}, \dots, e_1) &= \text{normalize}(c^* F_t(X_t) * P(e_t | X_t)) \\ &= \text{normalize}(F_t(X_t) * P(e_t | X_t)) \end{aligned}$$

... we again drop  $c$  (because we are normalizing)

---

## HMM Rules, Recap

1. Access initial distribution ( $P(X_1)$ )

2. Calculate state estimates over time:

$$P(X_t | e_{t-1}, e_{t-2} \dots, e_1) =$$

$$\text{normalize}(\sum_{d \in Dom[X_{t-1}]} P(X_t | X_{t-1}) P(X_{t-1} | e_{t-1}, e_{t-2} \dots, e_1))$$

3. Weight with observation:

$$P(X_t | e_t, e_{t-1}, e_{t-2} \dots, e_1) =$$

$$\text{normalize}(P(X_t | e_{t-1}, e_{t-2} \dots, e_1) * P(e_t | X_t))$$



# Example: Passage of Time

- As time passes, uncertainty “accumulates”

■ (Transition model: ghosts usually go clockwise)

|       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|
| <0.01 | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| <0.01 | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| <0.01 | <0.01 | 1.00  | <0.01 | <0.01 | <0.01 |
| <0.01 | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |

T = 1

|       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|
| <0.01 | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| <0.01 | <0.01 | 0.06  | <0.01 | <0.01 | <0.01 |
| <0.01 | 0.76  | 0.06  | 0.06  | <0.01 | <0.01 |
| <0.01 | <0.01 | 0.06  | <0.01 | <0.01 | <0.01 |

T = 2

|      |      |       |       |       |       |
|------|------|-------|-------|-------|-------|
| 0.05 | 0.01 | 0.05  | <0.01 | <0.01 | <0.01 |
| 0.02 | 0.14 | 0.11  | 0.35  | <0.01 | <0.01 |
| 0.07 | 0.03 | 0.05  | <0.01 | 0.03  | <0.01 |
| 0.03 | 0.03 | <0.01 | <0.01 | <0.01 | <0.01 |

T = 5



# Example: Observation

- As we get observations, beliefs get re-weighted, uncertainty “decreases”

|      |      |       |       |       |       |
|------|------|-------|-------|-------|-------|
| 0.05 | 0.01 | 0.05  | <0.01 | <0.01 | <0.01 |
| 0.02 | 0.14 | 0.11  | 0.35  | <0.01 | <0.01 |
| 0.07 | 0.03 | 0.05  | <0.01 | 0.03  | <0.01 |
| 0.03 | 0.03 | <0.01 | <0.01 | <0.01 | <0.01 |

Before observation

|       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|
| <0.01 | <0.01 | <0.01 | <0.01 | 0.02  | <0.01 |
| <0.01 | <0.01 | <0.01 | 0.83  | 0.02  | <0.01 |
| <0.01 | <0.01 | 0.11  | <0.01 | <0.01 | <0.01 |
| <0.01 | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |

After observation



$$P(X_t | e_t, e_{t-1}, \dots, e_1) = c^*(P(X_t | e_{t-1}, e_{t-2}, \dots, e_1) * P(e_t | X_t))$$

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.]

# Most Likely Explanation

---

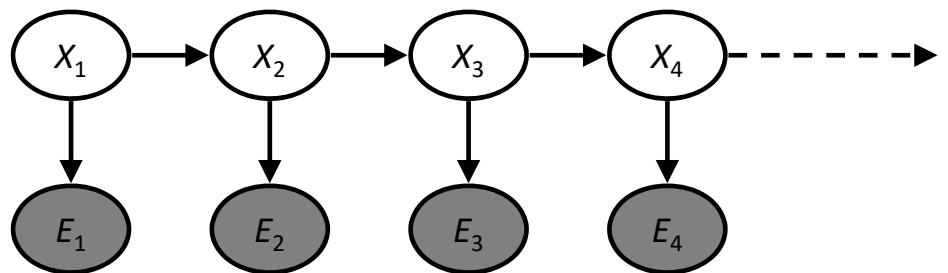


[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.]

# HMMs: MLE Queries

---

- HMMs defined by
  - States X
  - Observations E
  - Initial distribution:  $P(X_1)$
  - Transitions:  $P(X|X_{-1})$
  - Emissions:  $P(E|X)$



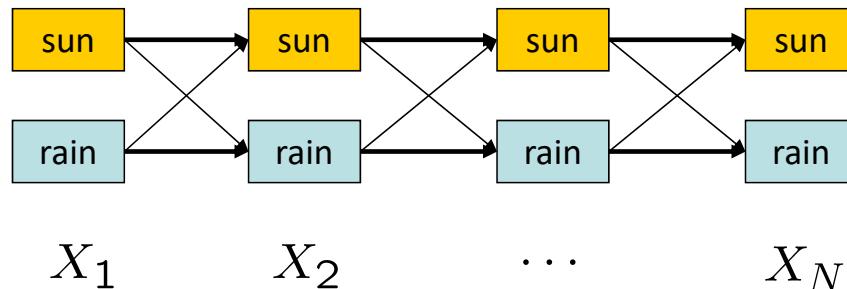
- New query: most likely explanation:  $\arg \max_{x_{1:t}} P(x_{1:t}|e_{1:t})$
- New method: the Viterbi algorithm

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.]

# State Trellis

---

- State trellis: graph of states and transitions over time

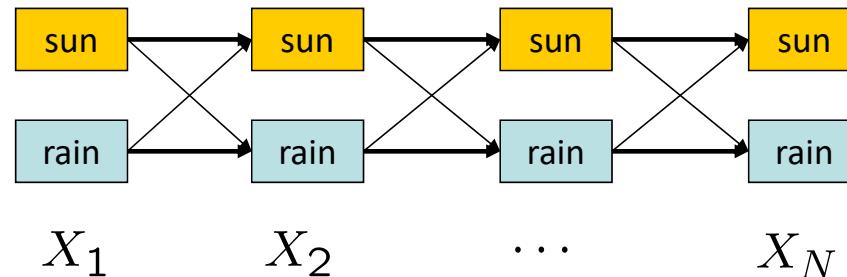


- Each arc represents some transition  $x_{t-1} \rightarrow x_t$
- Each arc has weight  $P(x_t|x_{t-1})P(e_t|x_t)$
- Each path is a sequence of states
- The product of weights on a path is that sequence's probability along with the evidence
- Forward algorithm computes sums of paths, Viterbi computes best paths

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.]

# Forward / Viterbi Algorithms

---



Forward Algorithm (Sum)

$$f_t[x_t] = P(x_t, e_{1:t})$$

$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]$$

Viterbi Algorithm (Max)

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

# Not all Bayes Nets will be Markov Chains. Causal Intuitions can help us organize these BNs.

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.

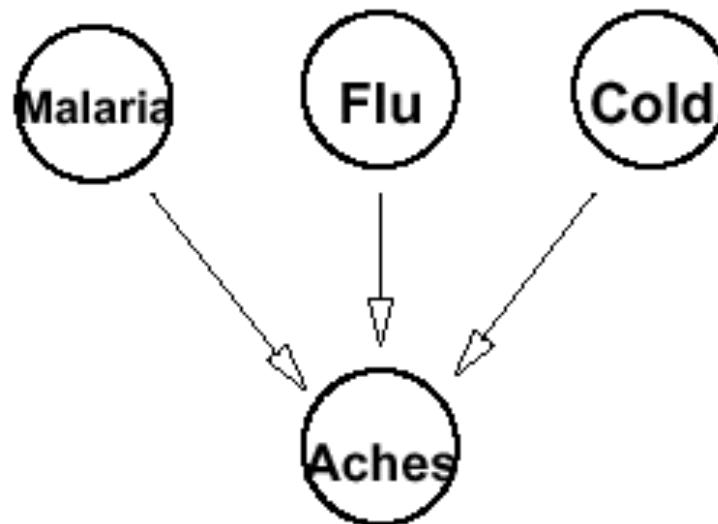
So  $P(C|M,F) = P(C)$  and  $P(F|M) = P(F)$

---

# 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!

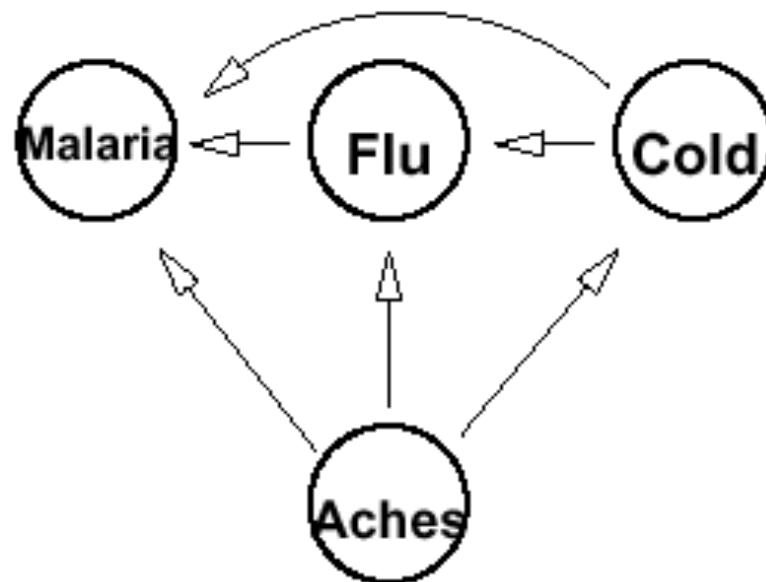
---

# Causal Intuitions

- 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

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?

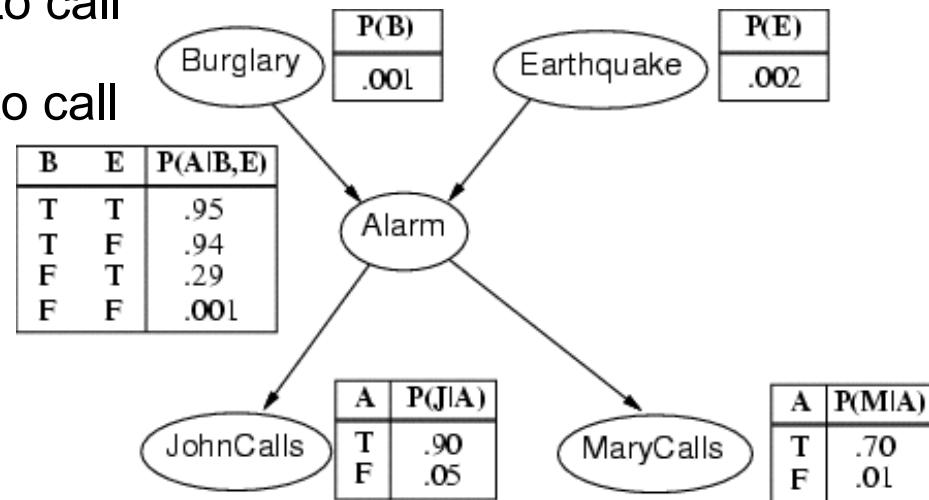
Variables: *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

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

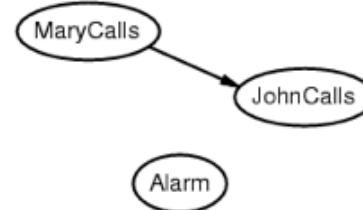


$$P(J | M) = P(J)?$$

---

# Burglary Example

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



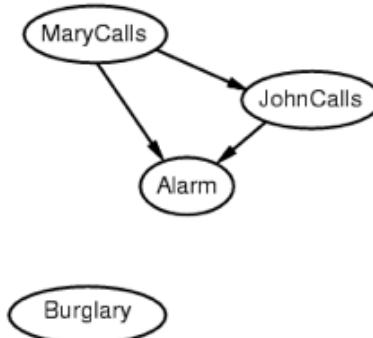
$P(J | M) = P(J)$ ? No

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

---

# 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

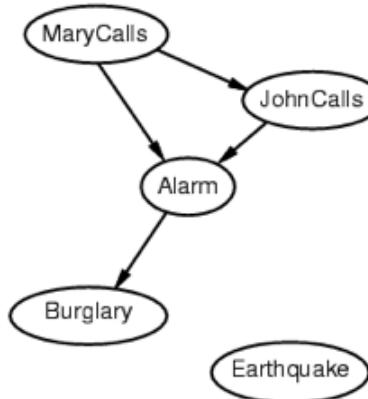
$P(A | J, M) = P(A | J)?$   $P(A | J, M) = P(A)?$  No

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

$P(J | M) = P(J)? \text{No}$

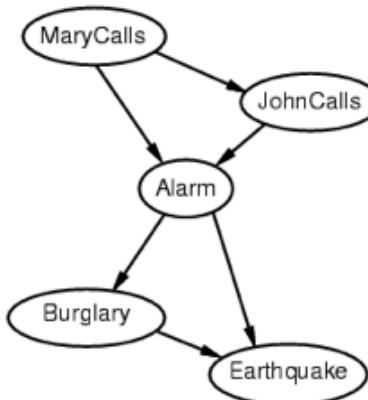
$P(A | J, M) = P(A | J)? P(A | J, M) = P(A)? \text{No}$

$P(B | A, J, M) = P(B | A)? \text{Yes}$

$P(B | A, J, M) = P(B)? \text{No}$

$P(E | B, A, J, M) = P(E | A)? \text{No}$

$P(E | B, A, J, M) = P(E | A, B)? \text{Yes}$



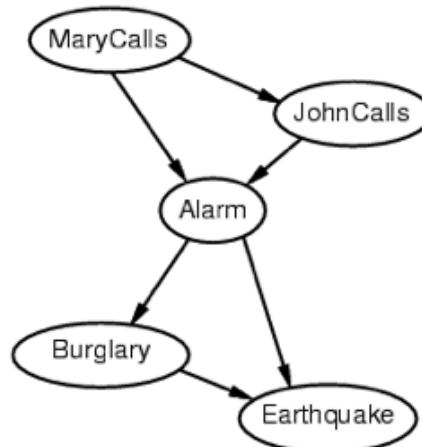
---

# 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, Generally

Given a Bayes net

$$P(X_1, X_2, \dots, X_n) = P(X_n | P(\text{Parents}(X_n))) *$$

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

And some evidence

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

we want to compute the new probability distribution

$$P(X_k | E)$$

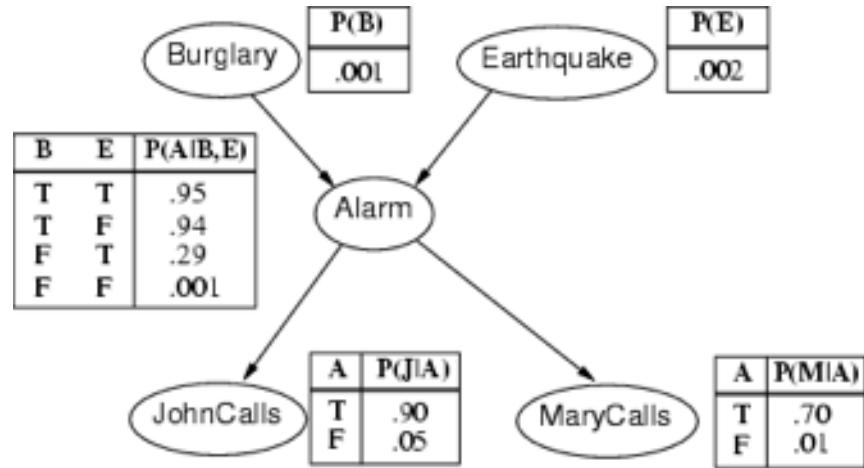
That is, we want to figure our

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

---

# Inference in Bayes Nets

In the Alarm example:



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

---

# Variable Elimination

Variable elimination is a tool to help with inference. We saw it in action when we looked at the Forward Algorithm.

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

VE helps to organize the calculations required to make exact inferences.

---

# Example (Binary valued Variables)

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

$$P(A)$$

$$\times P(B)$$

$$\times P(C | A)$$

$$\times P(D | A, B)$$

$$\times P(E | C)$$

$$\times P(F | D)$$

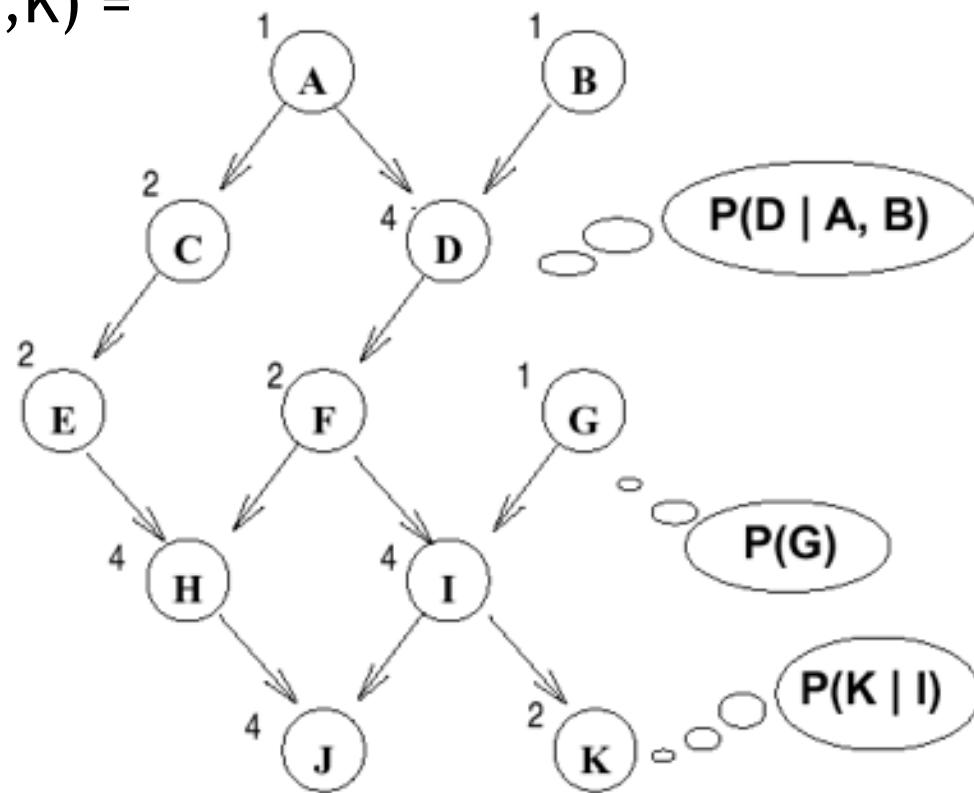
$$\times P(G)$$

$$\times P(H | E, F)$$

$$\times P(I | F, G)$$

$$\times P(J | H, I)$$

$$\times P(K | I)$$



---

# Example

$$\begin{aligned} P(A, B, C, D, E, F, G, H, I, J, K) = \\ P(A)P(B)P(C|A)P(D|A, B)P(E|C)P(F|D)P(G)P(H|E, F) \\ P(I|F, G)P(J|H, I)P(K|I) \end{aligned}$$

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

$$P(D|h, -i) \quad (h: H \text{ is true}, -h: H \text{ is false})$$

$$= P(d, h, -i)/N \text{ and } P(-d, h, -i)/N \text{ where}$$

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

$N$  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:

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

---

# 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.

$$\begin{aligned} & \sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \\ & \quad \sum_F P(F|d) P(h|E,F) \sum_G P(G) P(-i|F,G) \\ & \quad \sum_J P(J|h,-i) \\ & \quad \sum_K P(K|-i) \end{aligned}$$

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

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

---

# Example

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

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.

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

---

# Example

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

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

Now we've eliminated variable J.

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

---

# Example

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

$$\begin{aligned} & \sum_G P(G) P(-i|F,G) \\ &= (P(g)P(-i|F,g) + P(-g)P(-i|F,-g)) \end{aligned}$$

$P(-i|F,g)$  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

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

$$\begin{aligned} & \sum_G P(G) P(\text{-i}|F,G) \\ &= (P(g)P(\text{-i}|F,g) + P(-g)P(\text{-i}|F,-g)) \end{aligned}$$

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

$$f1(F) = P(g)P(\text{-i}|F,g) + P(-g)P(\text{-i}|F,-g)$$

We must store these fixed numbers to represent the function:

$$f1(f) = P(g)P(\text{-i}|f,g) + P(-g)P(\text{-i}|f,-g)$$

$$f1(\text{-f}) = P(g)P(\text{-i}|\text{-f},g) + P(-g)P(\text{-i}|\text{-f},-g)$$

---

# Example

$$c_1 c_2 \sum_A P(A) \sum_B P(B) P(d|A,B) \sum_C P(C|A) \sum_E P(E|C) \sum_F P(F|d) P(h|E,F) f_1(F)$$

$$\sum_F P(F|d) P(h|E,F) f_1(F) = P(f|d) P(h|E,f) f_1(f) + P(-f|d) P(h|E,-f) f_1(-f)$$

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

$$f_2(E) = P(f|d) P(h|E,f) f_1(f) + P(-f|d) P(h|E,-f) f_1(-f)$$

Again store two fixed values,  $f_2(e)$  and  $f_2(-e)$ , 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

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

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

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

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

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.,

$$\sum_F P(F|D) P(h|E,F) f_1(F)$$

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  $h = f \times g$  (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  $h = \sum_X f$ , which is defined:  $h(Y) = \sum_{x \in \text{Dom}(X)} f(x, Y)$

| 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<br>$f_{A=a}$ |     |
|-----------|-----|-------------------------|-----|
| 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  $f \in F$  that mentions a variable(s) in  $E$  with its **restriction**  $f_{E=e}$  (this might yield a “constant” factor)
2. For each  $Z_j$  - in the order given - eliminate  $Z_j \in Z$  as follows:
  - (a) Compute **new factor**  $g_j = \sum_{Z_j} f_1 \times f_2 \times \dots \times f_k$ , 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 at the end of this process will refer only to the query variable  $Q$ . **Take their product and normalize** to produce  $P(Q|E)$ .

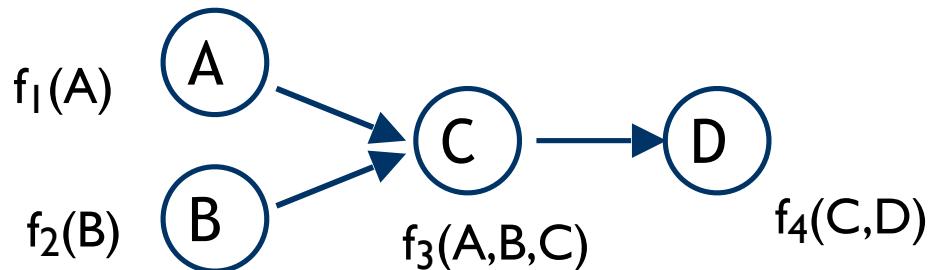
# VE: Example

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  $f_6(A,B) = \sum_C f_5(C) f_3(A,B,C)$  to list of factors.

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

**Step 3 (Eliminate B):** Compute & add  $f_7(A) = \sum_B f_6(A,B) f_2(B)$  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:

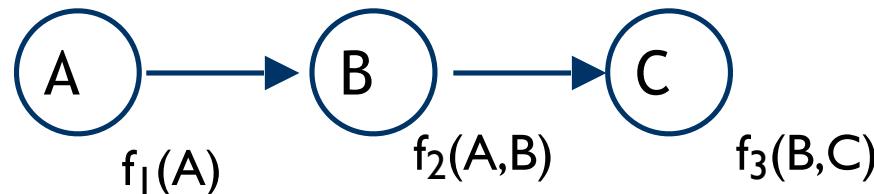
$$P(A|d) = \alpha f_1(A) \times f_7(A)$$

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

---

# Numeric Example

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

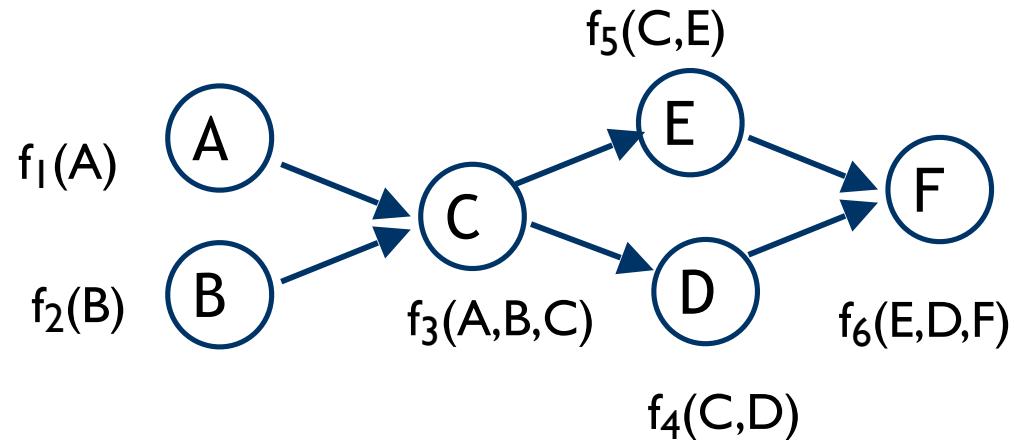


| $f_1(A)$ |     | $f_2(A,B)$     |     | $f_3(B,C)$     |     | $f_4(B)$<br>$\Sigma_A f_2(A,B)f_1(A)$ |      | $f_5(C)$<br>$\Sigma_B f_3(B,C) f_4(B)$ |       |
|----------|-----|----------------|-----|----------------|-----|---------------------------------------|------|--|-------|
| a        | 0.9 | ab             | 0.9 | bc             | 0.7 | b                                     | 0.85 | c                                      | 0.625 |
| $\sim a$ | 0.1 | $a\sim b$      | 0.1 | $b\sim c$      | 0.3 | $\sim b$                              | 0.15 | $\sim c$                               | 0.375 |
|          |     | $\sim ab$      | 0.4 | $\sim bc$      | 0.2 |                                       |      |  |       |
|          |     | $\sim a\sim b$ | 0.6 | $\sim b\sim c$ | 0.8 |                                       |      |  |       |

---

# VE: Buckets as a Notational Device

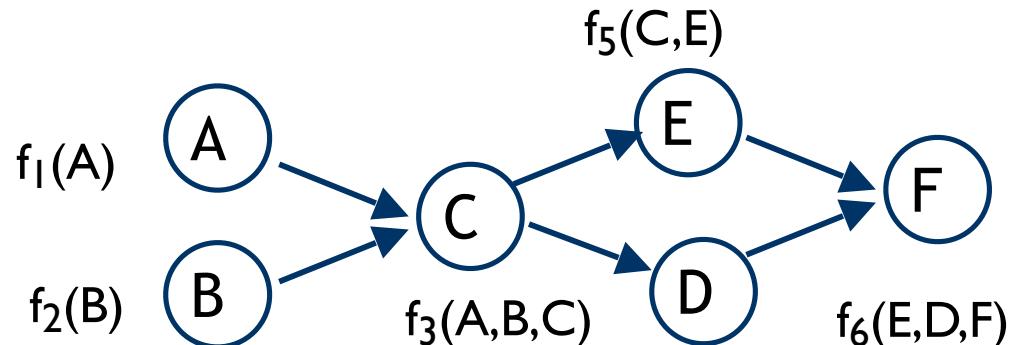
Ordering:  
C,F,A,B,E,D



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

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

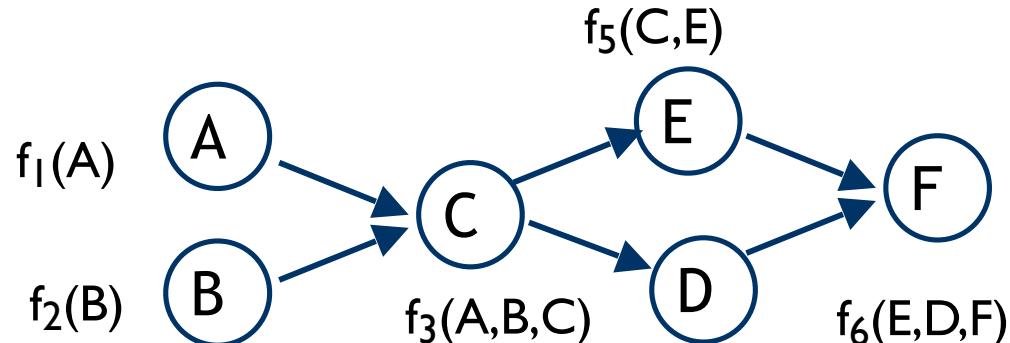
Ordering:  
C,F,A,B,E,D



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

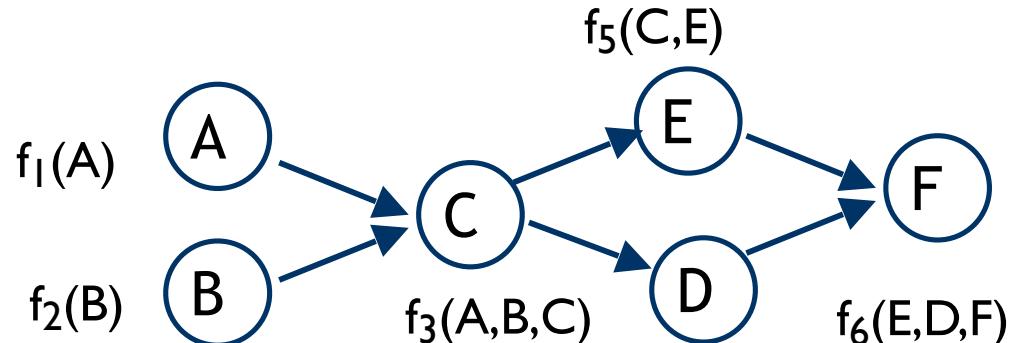


1. ~~C:  $f_3(A,B,C), f_4(C,D), f_5(C,E)$~~
  2. F:  $f_6(E,D,F)$
  3. A:  $f_1(A), f_7(A,B,D,E)$
  4. B:  $f_2(B)$
  5. E:
  6. D:
- 

$$\begin{aligned} \text{1. } & \sum_C f_3(A,B,C) \times f_4(C,D) \times f_5(C,E) \\ & = f_7(A,B,D,E) \end{aligned}$$

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

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

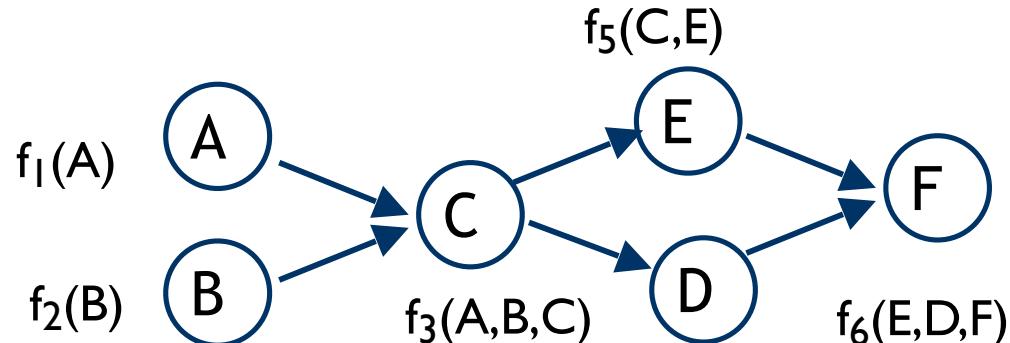


1. ~~C:  $f_3(A, B, C)$ ,  $f_4(C, D)$ ,  $f_5(C, E)$~~
  2. ~~F:  $f_6(E, D, F)$~~
  3. A:  $f_1(A)$ ,  $f_7(A, B, D, E)$
  4. B:  $f_2(B)$
  5. E:  $f_8(E, D)$
  6. D:
- 

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

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

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

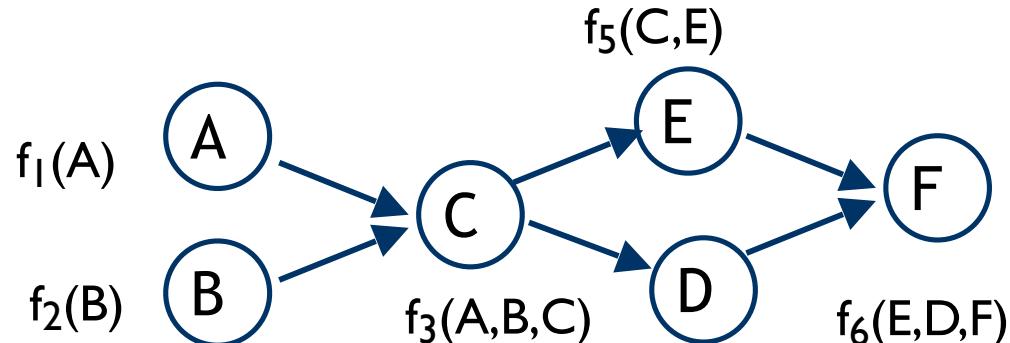


1. ~~C:  $f_3(A,B,C)$ ,  $f_4(C,D)$ ,  $f_5(C,E)$~~
  2. ~~F:  $f_6(E,D,F)$~~
  3. ~~A:  $f_1(A)$ ,  $f_7(A,B,D,E)$~~
  4. B:  $f_2(B)$ ,  ~~$f_9(B,D,E)$~~
  5. E:  $f_8(E,D)$
  6. D:
- 

$$\begin{aligned}3. \sum_A f_1(A) \times f_7(A,B,D,E) \\= \textcolor{green}{f_9(B,D,E)}\end{aligned}$$

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

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

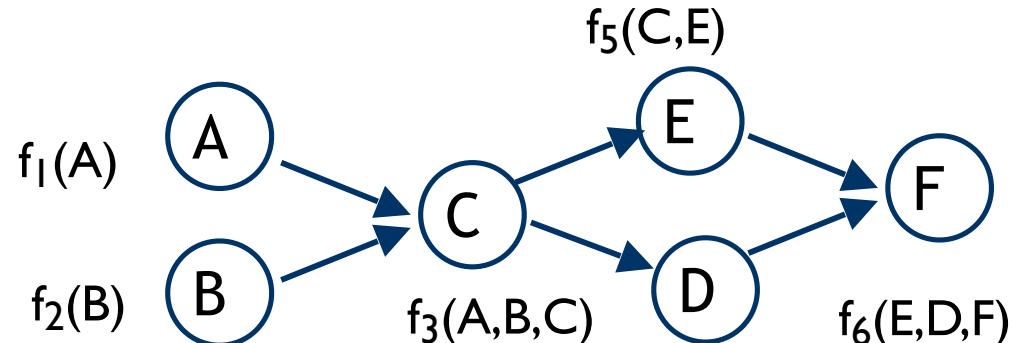


1. ~~C:  $f_3(A, B, C), f_4(C, D), f_5(C, E)$~~
  2. ~~F:  $f_6(E, D, F)$~~
  3. ~~A:  $f_1(A), f_7(A, B, D, E)$~~
  4. ~~B:  $f_2(B), f_9(B, D, E)$~~
  5. E:  $f_8(E, D), f_{10}(D, E)$
  6. D:
- 

$$\begin{aligned}4. \sum_B f_2(B) \times f_9(B, D, E) \\= f_{10}(D, E)\end{aligned}$$

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

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

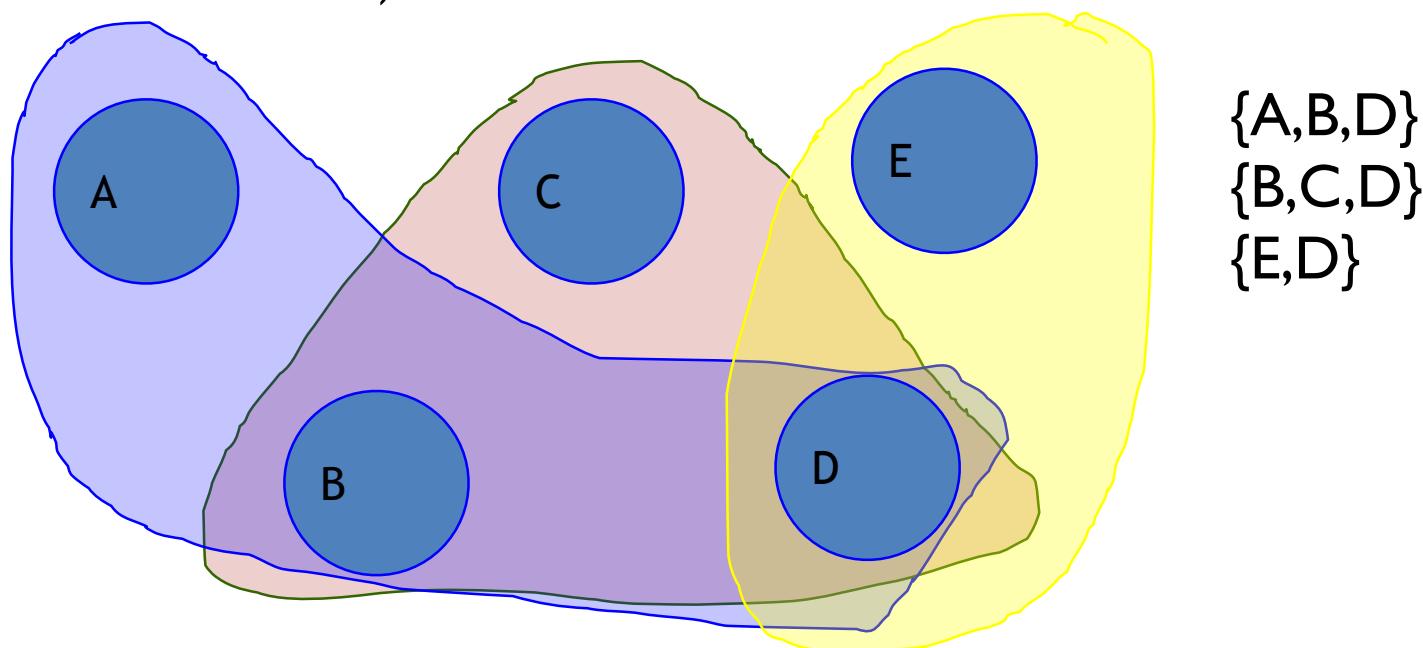


1. ~~C:  $f_3(A, B, C)$ ,  $f_4(C, D)$ ,  $f_5(C, E)$~~   $f_4(C, D)$
  2. ~~F:  $f_6(E, D, F)$~~   $\sum_E f_8(E, D) \times f_{10}(D, E)$   
 $= f_{11}(D)$
  3. ~~A:  $f_1(A)$ ,  $f_7(A, B, D, E)$~~   $f_{11}$  is the final answer, once we normalize it.
  4. ~~B:  $f_2(B)$ ,  $f_9(B, D, E)$~~
  5. ~~E:  $f_8(E, D)$ ,  $f_{10}(D, E)$~~
  6. ~~D:  $f_{11}(D)$~~
-

# Hypergraphs

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

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



# 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.

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



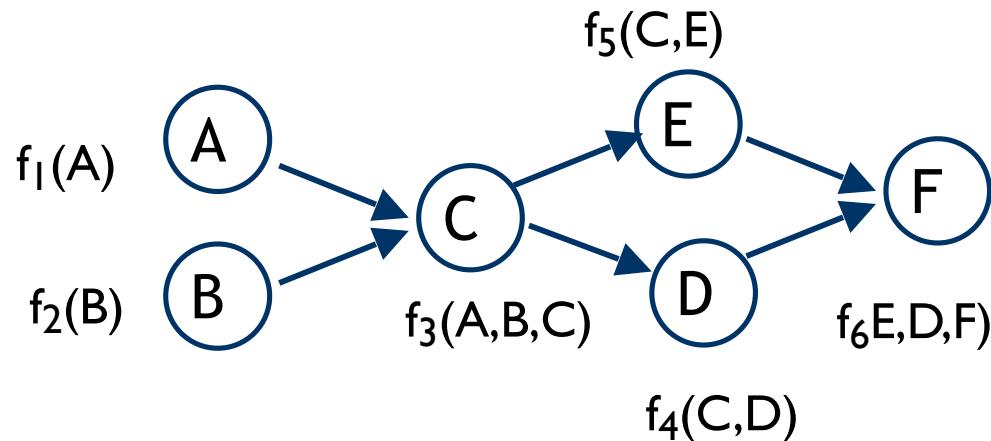
# Variable Elimination in the HyperGraph

To eliminate variable  $X_i$  in the hypergraph we

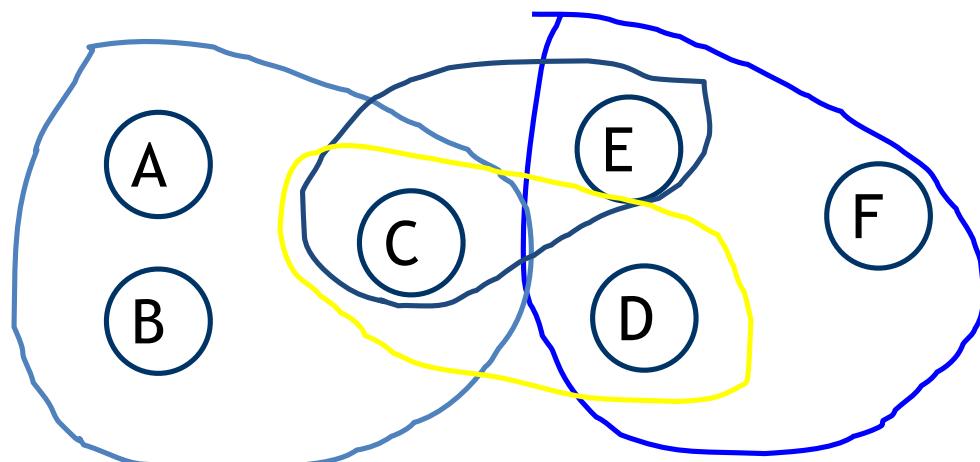
- Remove the vertex  $X_i$
  - Create a new hyperedge  $H_i$  equal to the union of all of the hyperedges that contain  $X_i$  minus  $X_i$
  - Remove all of the hyperedges containing  $X$  from the hypergraph.
  - Add the new hyperedge  $H_i$  to the hypergraph.
-

# VE Factors

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

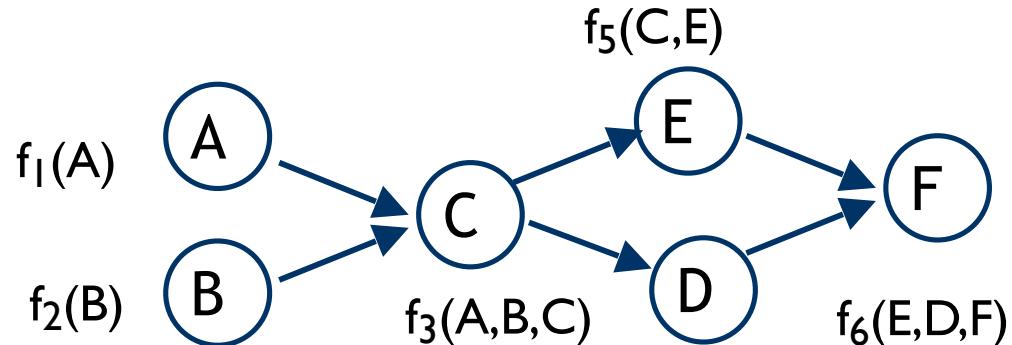


1. C:
2. F:
3. A:
4. B:
5. E:
6. D:



## VE: Place Original Factors in first applicable bucket.

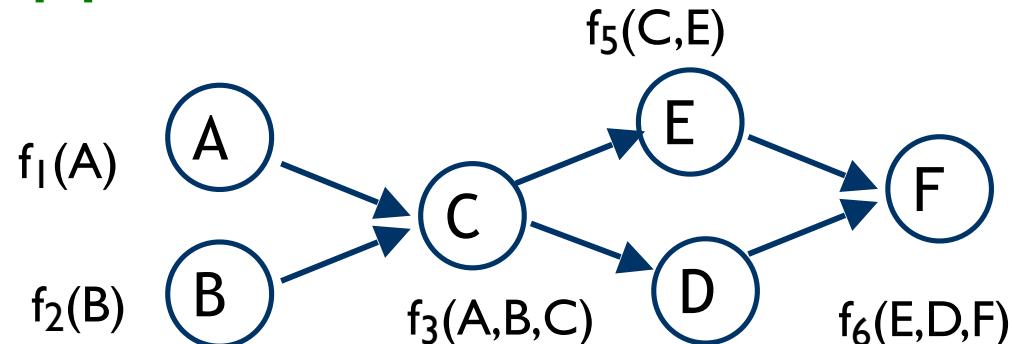
Ordering:  
C,F,A,B,E,D



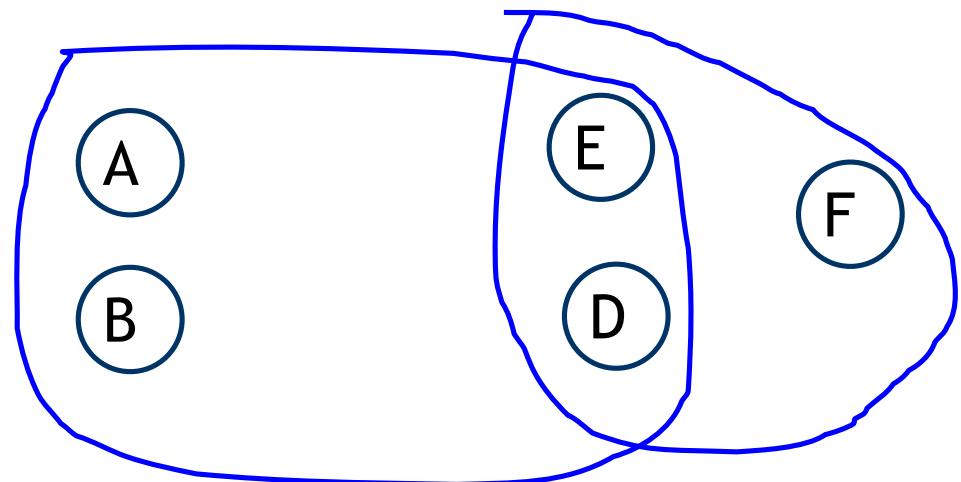
1. C:  $f_3(A, B, C)$ ,  $f_4(C, D)$ ,  $f_5(C, E)$
  2. F:  $f_6(E, D, F)$
  3. A:  $f_1(A)$
  4. B:  $f_2(B)$
  5. E:
  6. D:
-

VE: Eliminate C, placing new factor  $f_7$  in first applicable bucket.

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

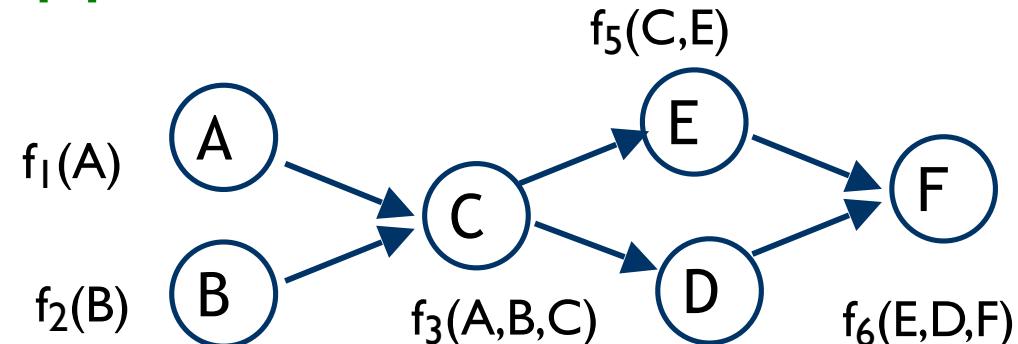


1. ~~C:  $f_3(A,B,C)$ ,  $f_4(C,D)$ ,  $f_5(C,E)$~~
2. F:  $f_6(E,D,F)$
3. A:  $f_1(A)$ ,  $f_7(A,B,D,E)$
4. B:  $f_2(B)$
5. E:
6. D:

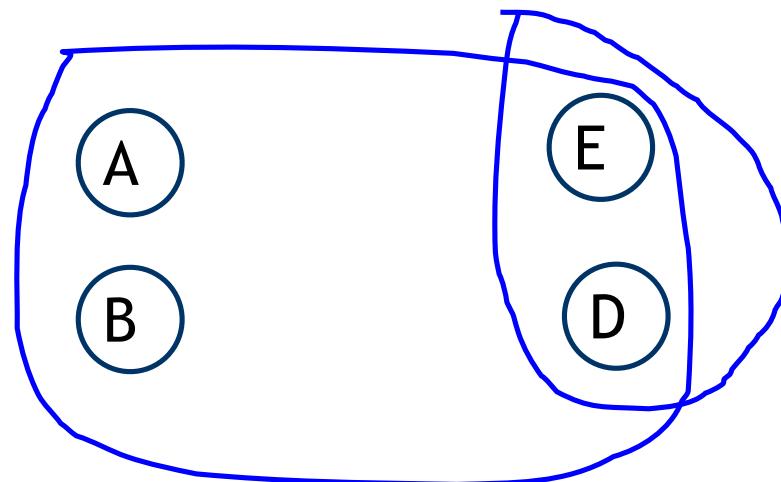


## VE: Eliminate F, placing new factor $f_8$ in first applicable bucket.

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

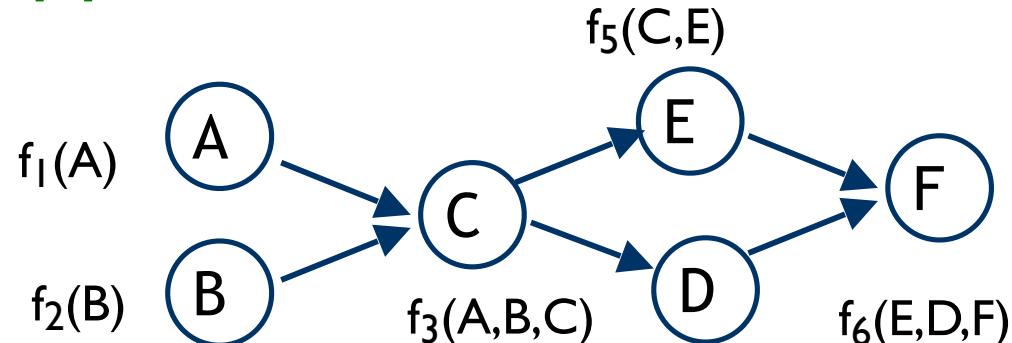


1. ~~C:  $f_3(A,B,C)$ ,  $f_4(C,D)$ ,  $f_5(C,E)$~~
2. ~~F:  $f_6(E,D,F)$~~
3. A:  $f_1(A)$ ,  $f_7(A,B,D,E)$
4. B:  $f_2(B)$
5. E:  $f_8(E,D)$
6. D:

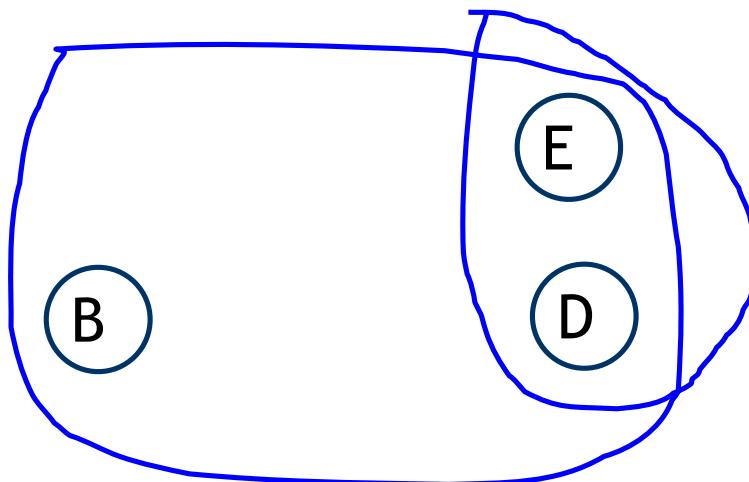


## VE: Eliminate A, placing new factor $f_9$ in first applicable bucket.

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

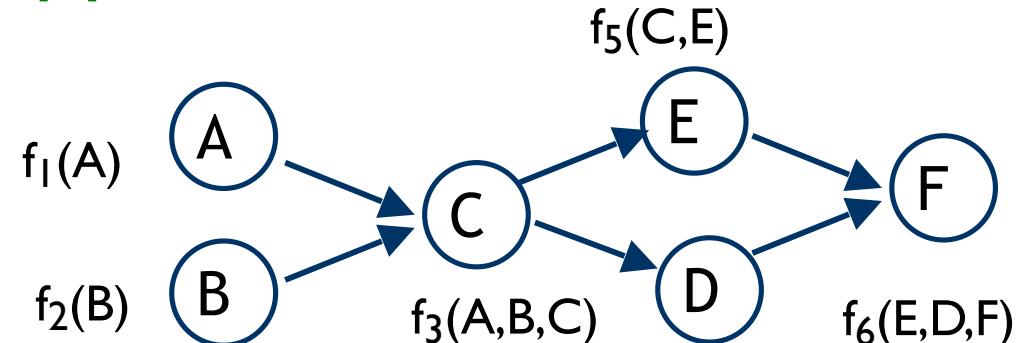


1. ~~C:  $f_3(A,B,C)$ ,  $f_4(C,D)$ ,  $f_5(C,E)$~~
2. ~~F:  $f_6(E,D,F)$~~
3. ~~A:  $f_1(A)$ ,  $f_7(A,B,D,E)$~~
4. B:  $f_2(B)$ ,  $f_9(B,D,E)$
5. E:  $f_8(E,D)$
6. D:



## VE: Eliminate B, placing new factor $f_{10}$ in first applicable bucket.

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



1. ~~C:  $f_3(A,B,C)$ ,  $f_4(C,D)$ ,  $f_5(C,E)$~~

$f_4(C,D)$

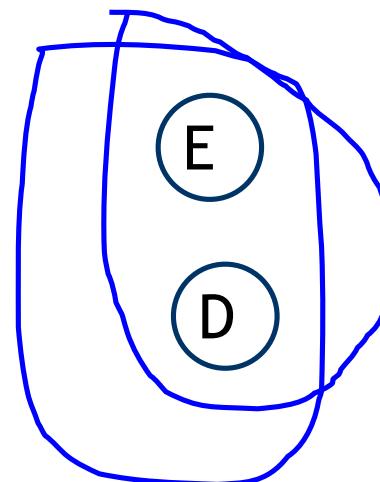
2. ~~F:  $f_6(E,D,F)$~~

3. ~~A:  $f_1(A)$ ,  $f_7(A,B,D,E)$~~

4. ~~B:  $f_2(B)$ ,  $f_9(B,D,E)$~~

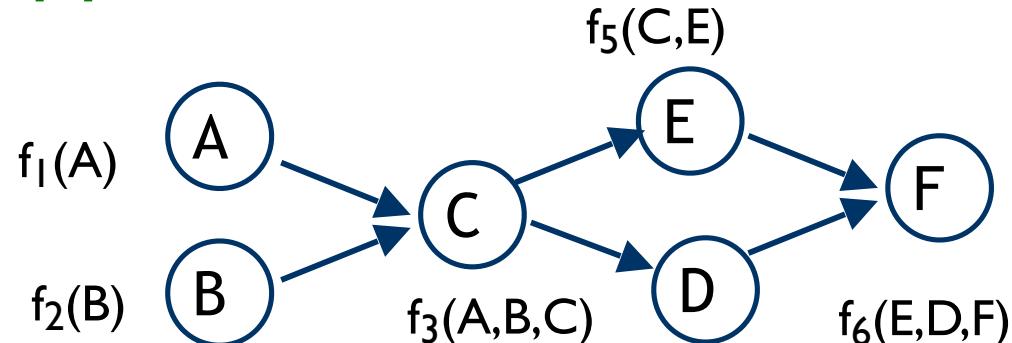
5. E:  $f_8(E,D)$ ,  $f_{10}(D,E)$

6. D:



VE: Eliminate E, placing new factor  $f_{11}$  in first applicable bucket.

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



1. ~~C:  $f_3(A,B,C)$ ,  $f_4(C,D)$ ,  $f_5(C,E)$~~

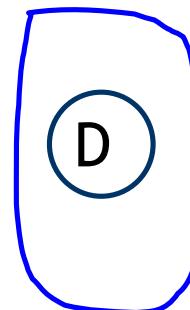
2. ~~F:  $f_6(E,D,F)$~~

3. ~~A:  $f_1(A)$ ,  $f_7(A,B,D,E)$~~

4. ~~B:  $f_2(B)$ ,  $f_9(B,D,E)$~~

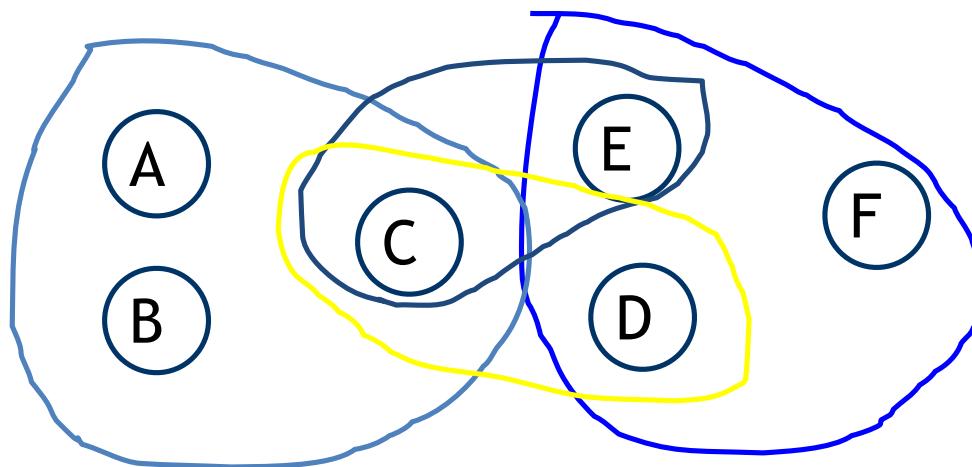
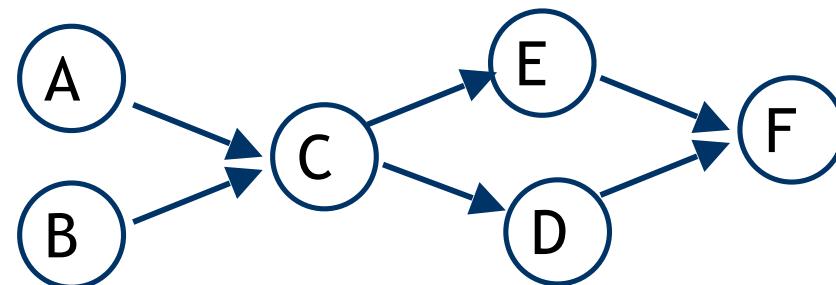
5. ~~E:  $f_8(E,D)$ ,  $f_{10}(D,E)$~~

6.  $D: f_{11}(D)$



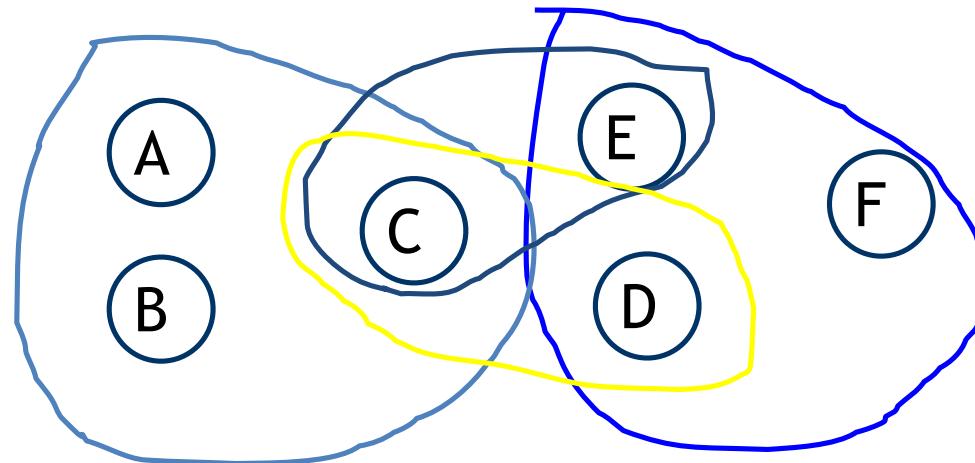
# Complexity of Variable Elimination

$$\begin{aligned} P(A, B, C, D, E, F) = \\ P(A)P(B) \\ \times P(C|A, B) \\ \times P(E|C) \\ \times P(D|C) \\ \times P(F|E, D). \end{aligned}$$

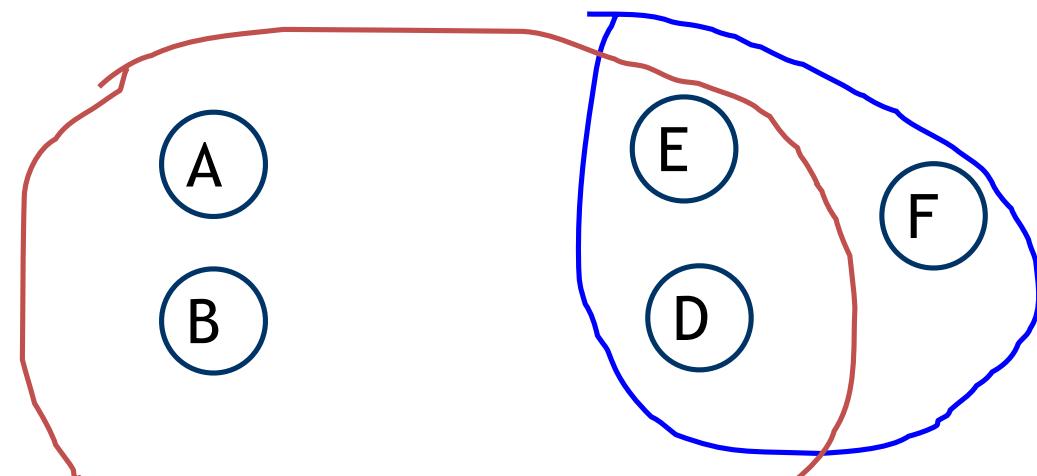


---

# Complexity of Variable Elimination

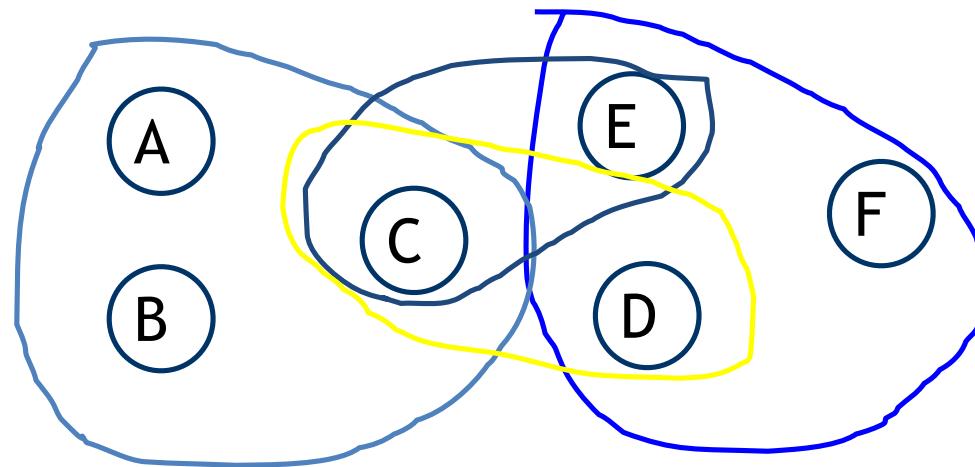


- Eliminate C

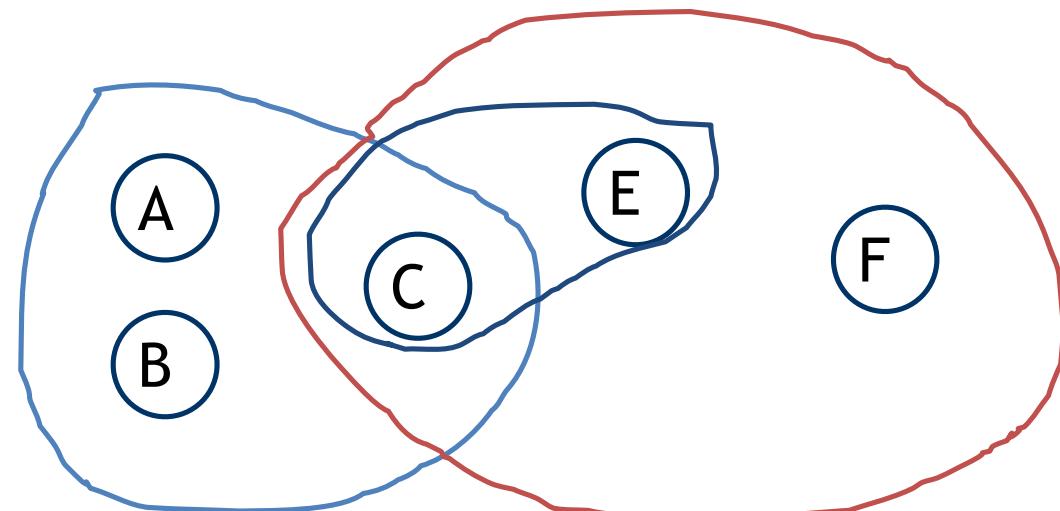


---

# Complexity of Variable Elimination

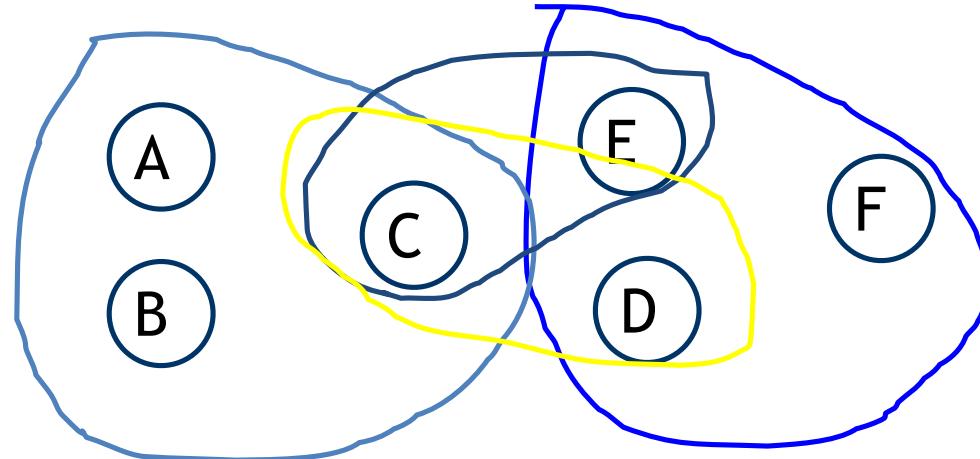


- Eliminate D

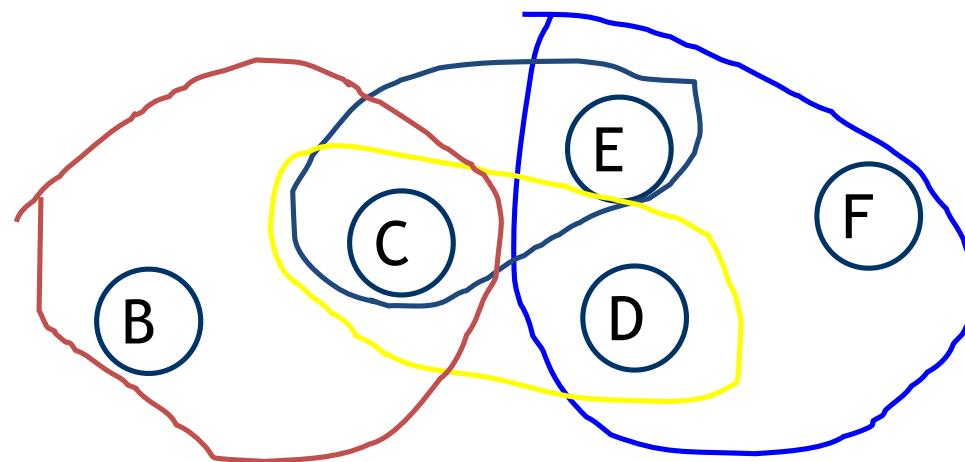


---

# Complexity of Variable Elimination



- Eliminate A



---

# Complexity of Variable Elimination

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

$$\mathcal{H} = H_0, H_1, H_2, \dots, H_n$$

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  $H_0, H_1, \dots, H_n$ .
- 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  $\mathcal{H}$  with vertices  $\{X_1, X_2, \dots, X_n\}$  the tree width ( $\omega$ ) of  $\mathcal{H}$  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  $\omega$  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

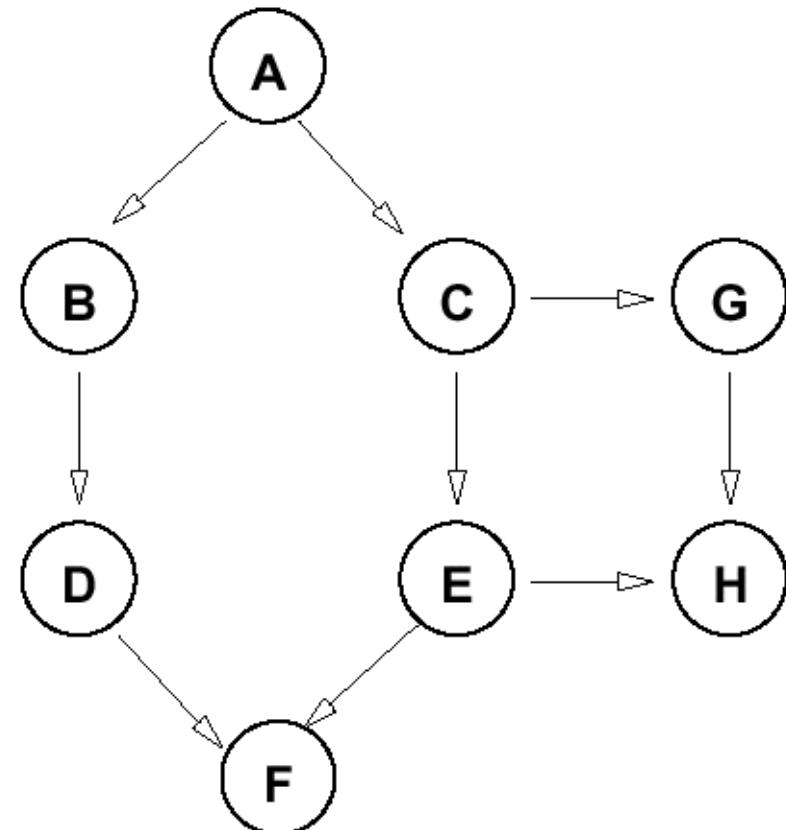
Suppose query variable is D. Consider different orderings for this network

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).

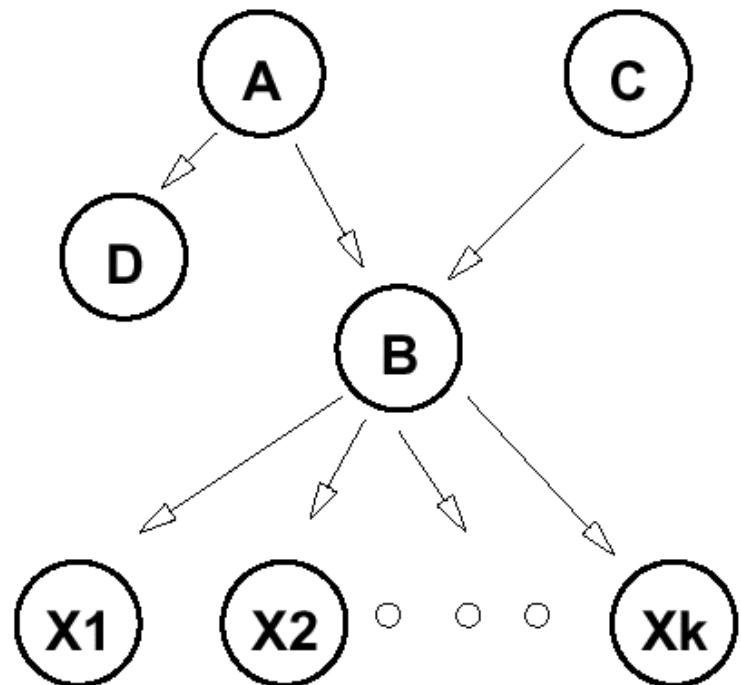
---

# 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.
-

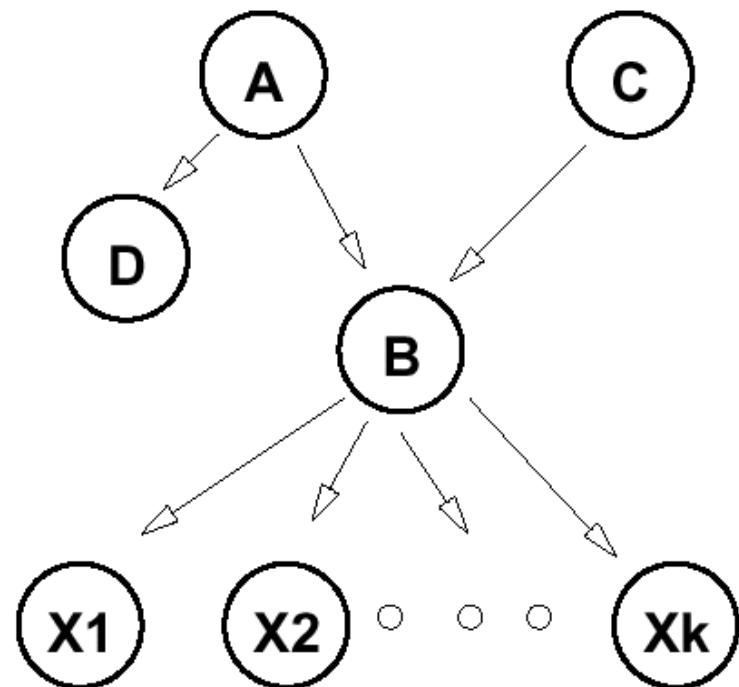
# Elimination Ordering: Polytrees

- Eliminating singly connected nodes allows VE to run in linear time
  - e.g., in this network, eliminate D, A, C, X<sub>1</sub>,...; or eliminate X<sub>1</sub>,... X<sub>k</sub>, 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, X<sub>1</sub>,... and X<sub>k</sub>!!!

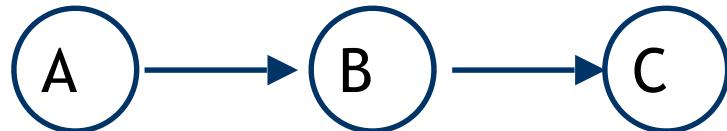


# Min Fill Heuristic

- 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



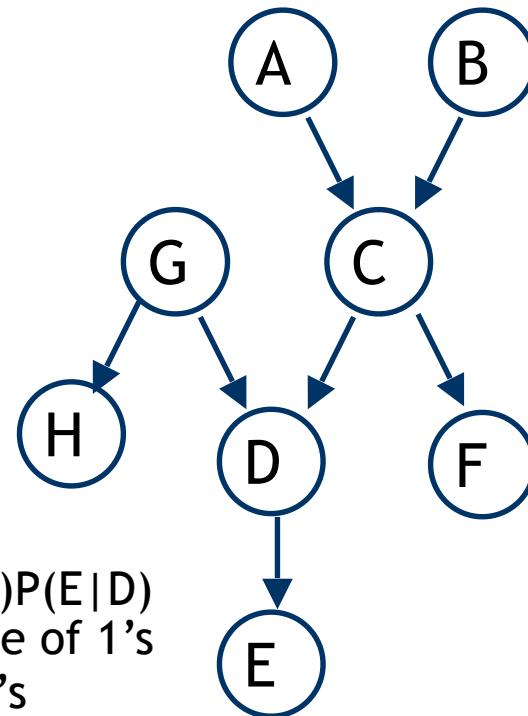
- Certain variables have no impact on the query.  
For example, in network ABC, computing  $P(A)$  with no evidence requires elimination of B and C.
    - But when you sum out these variables, you compute a trivial factor (that always evaluates to 1); for example:
    - Eliminating C:  $f_4(B) = \sum_C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

# Relevance

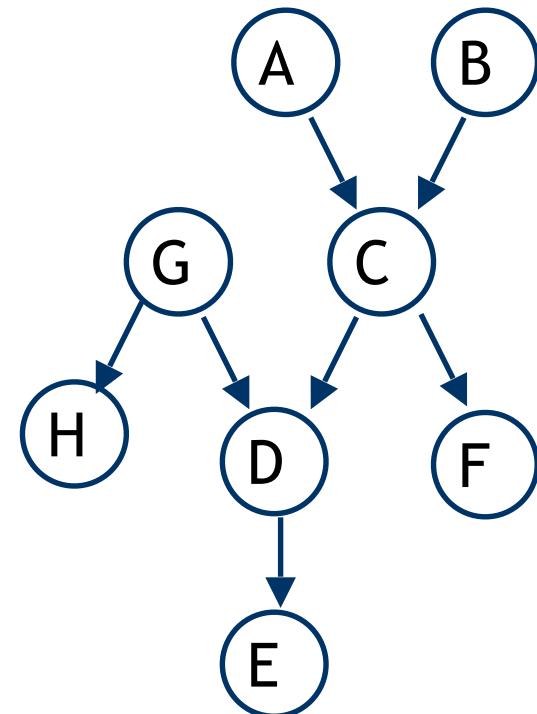
- Can restrict attention to *relevant* variables.  
Given query  $Q$ , evidence  $E$ :
    - $Q$  itself is relevant
    - if any node  $Z$  is relevant, its parents are relevant
    - if  $e \in E$  is a descendent of a relevant node, then  $E$  is relevant
  - We can restrict our attention to the *sub-network 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
  - $$\begin{aligned} P(A)P(B)P(C|A,B)P(F|C) & P(G)P(h|G)P(D|G,C)P(E|D) \\ = \dots P(G)P(h|G)P(D|G,C) \sum_E P(E|D) & = \text{a table of 1's} \\ = \dots P(G)P(h|G) \sum_D P(D|G,C) & = \text{a table of 1's} \\ = [P(A)P(B)P(C|A,B)P(F|C)] [\sum_G P(G)P(h|G)] & \\ [\sum_G P(G)P(h|G)] & \neq 1 \text{ but irrelevant} \end{aligned}$$
  - 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 off 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 its parents*:

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

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**

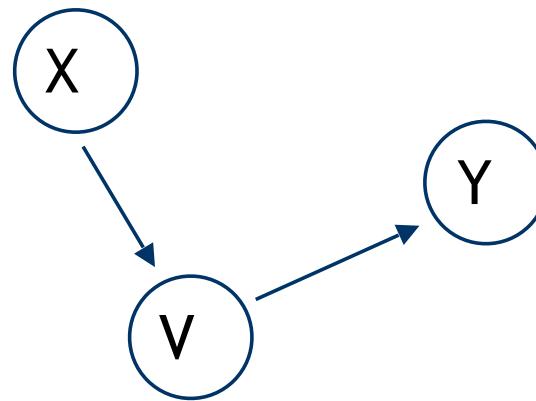
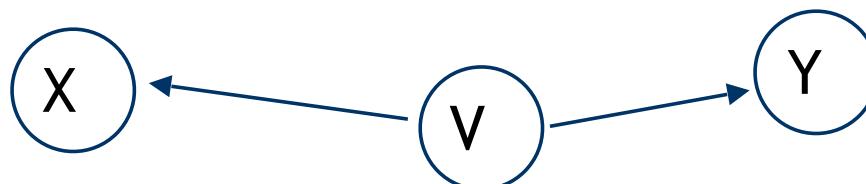
**D-separation:** A set of variables E **d-separates** X and Y if it **blocks** every undirected path in the BN between X and Y (we'll define Blocks next.)

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”

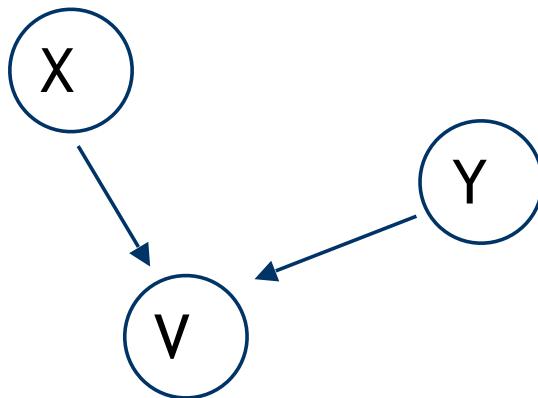
Or, there exists a variable  $V$  on the path such that

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?

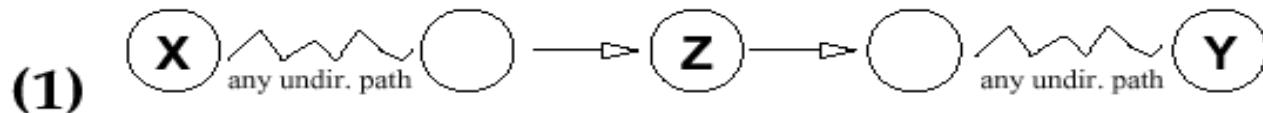


If the variable V is on the path such that the arcs putting V on the path are “head-to-head”, the variables are still blocked .... so long as:

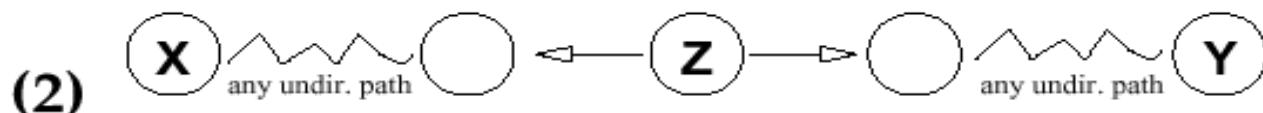
V is NOT in the evidence set E  
neither are any of its descendants

---

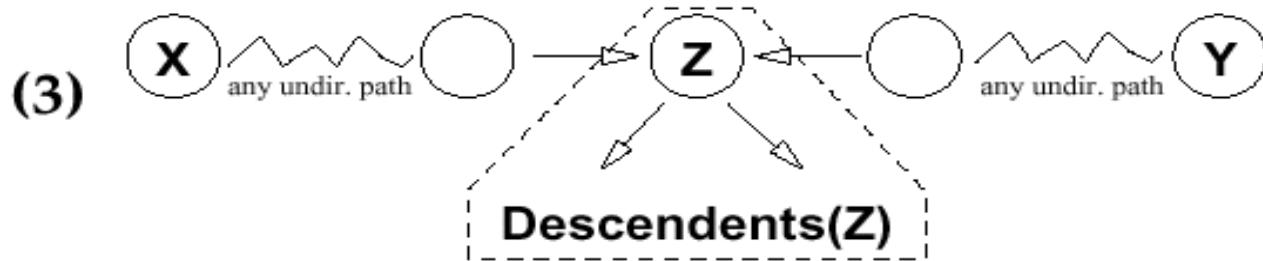
# Blocking: Graphical View



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



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



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

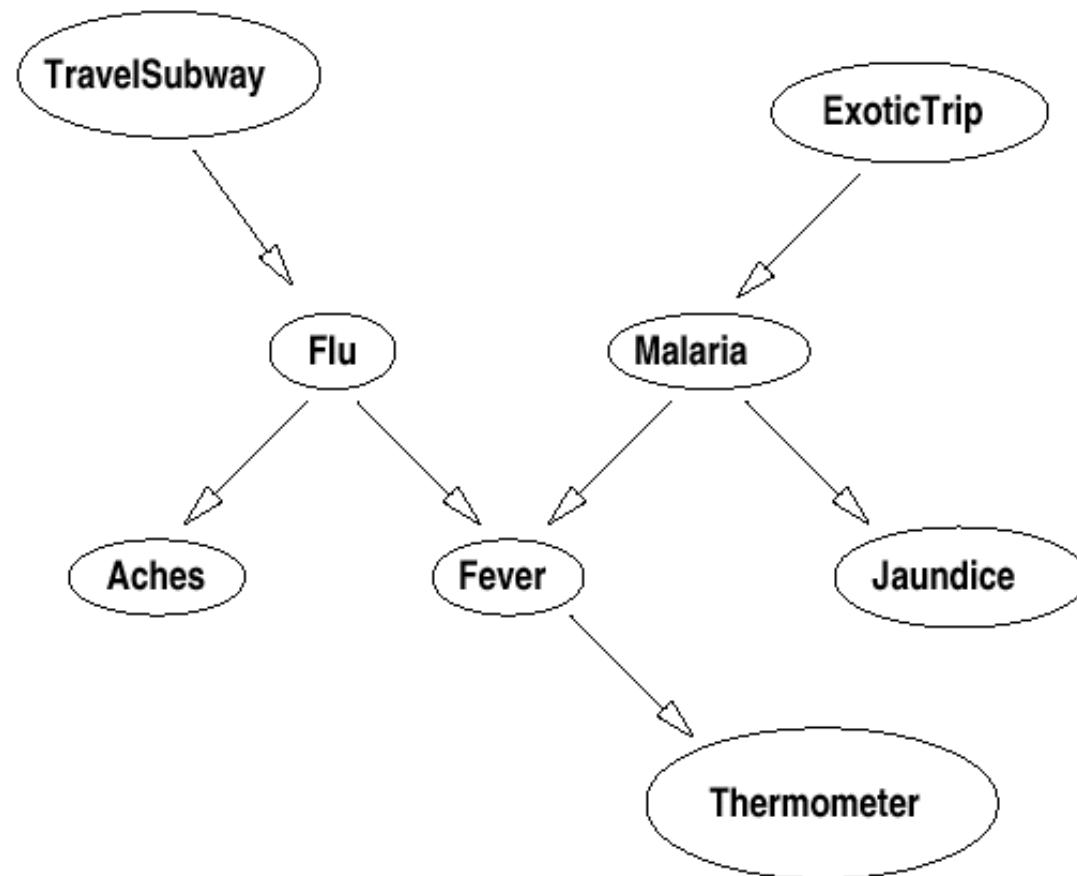
---

# 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  $E$ .

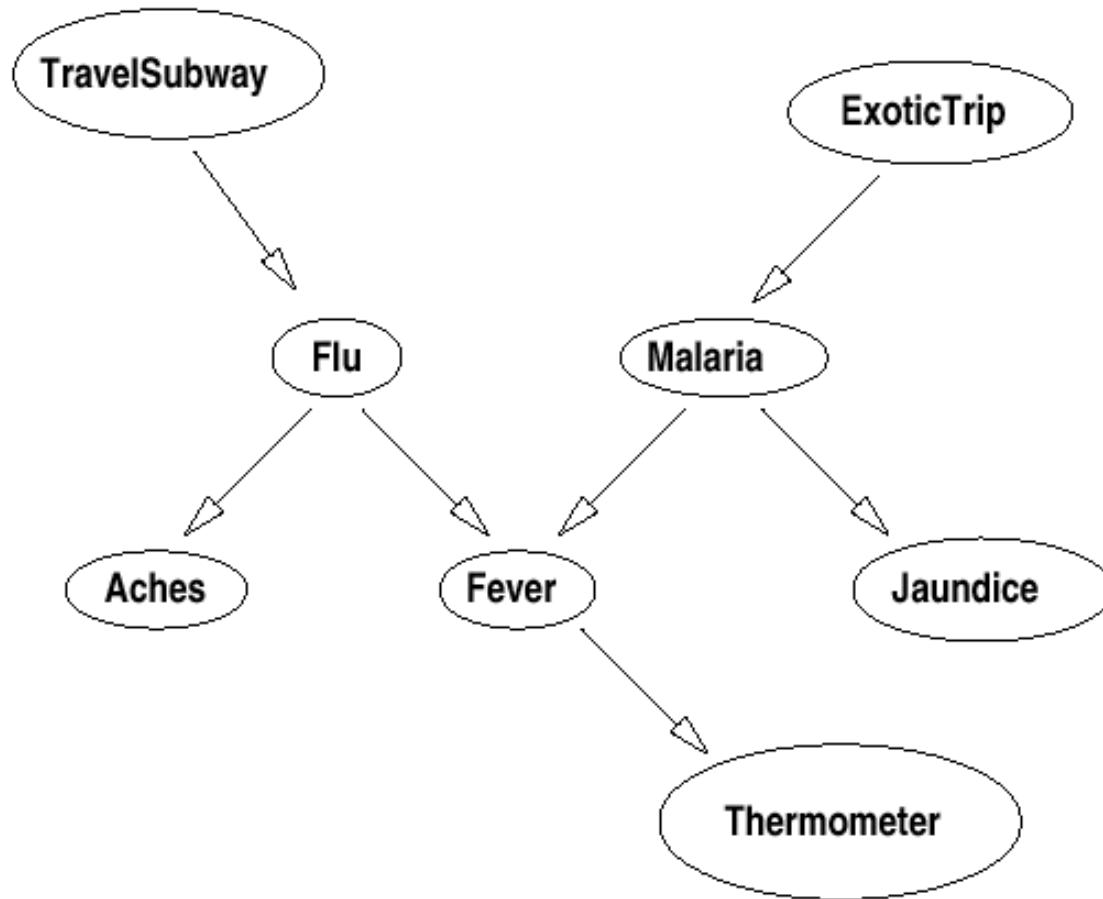


# D-Separation: Intuitions



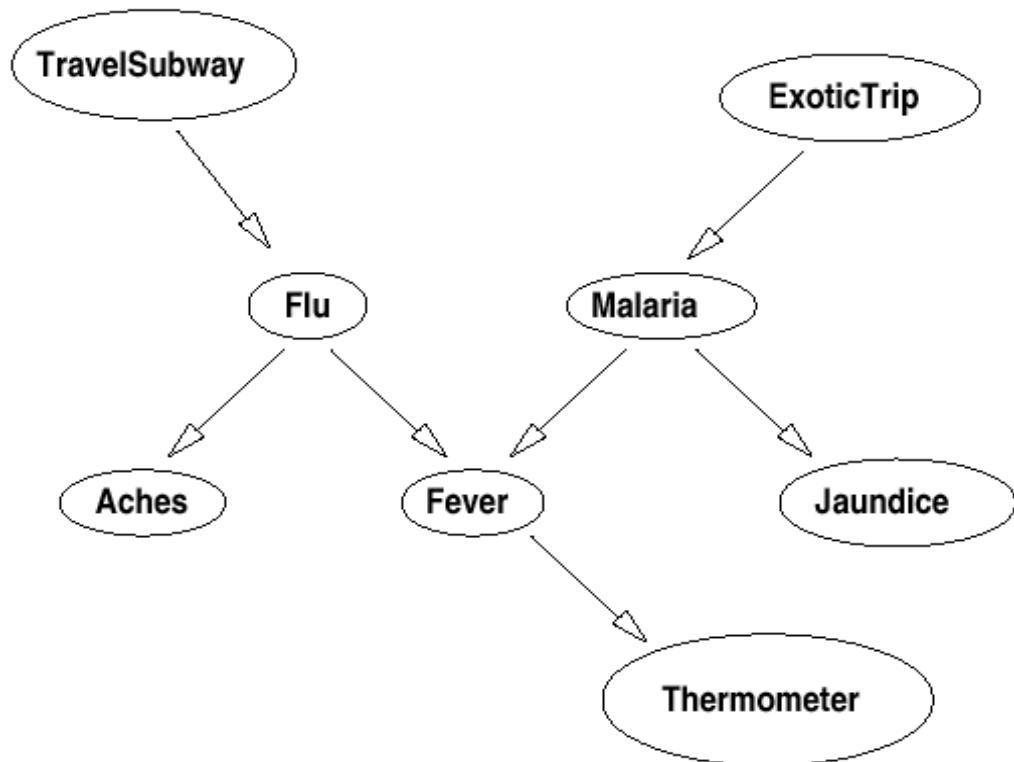
Subway and Therm are dependent; but are independent given Flu (since Flu blocks the only path)

# 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).

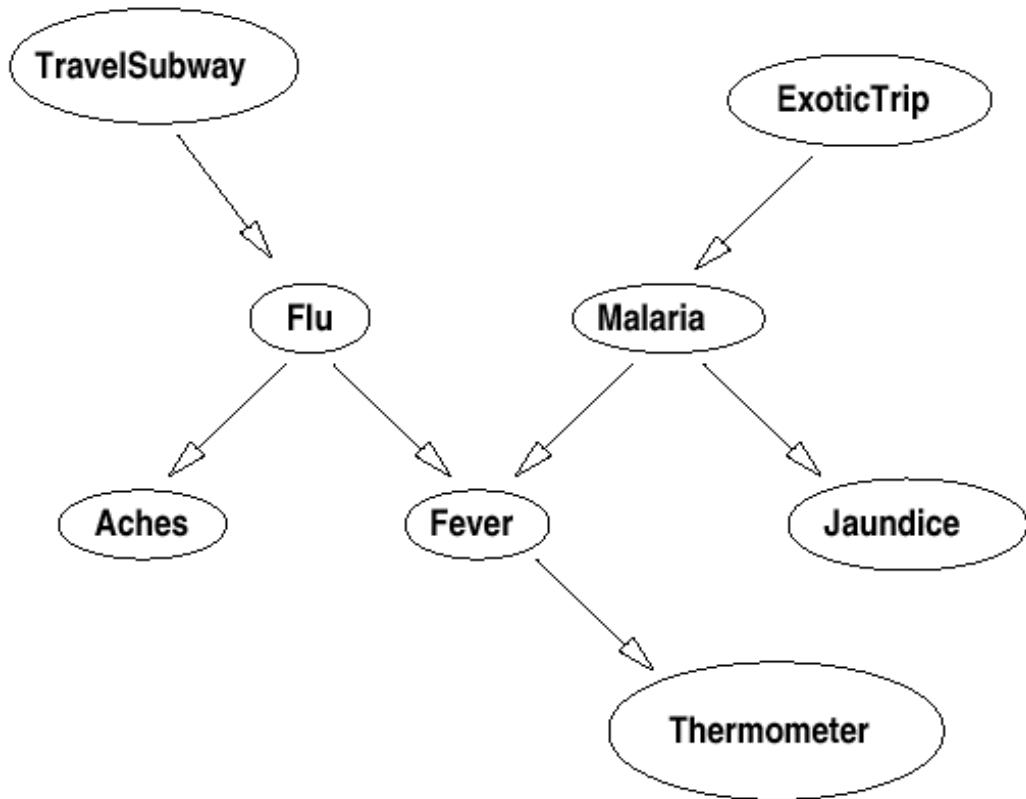
# D-Separation: Intuitions



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

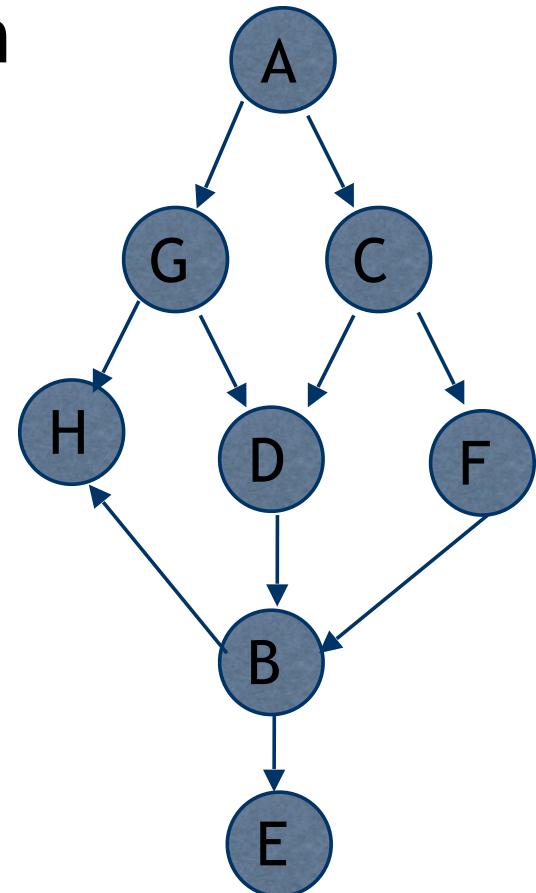


Subway, ExoticTrip are indep.;  
they are dependent given Therm;  
they are indep. given Therm and Malaria. This  
for exactly the same reasons for Flu/Mal above.

## 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}?

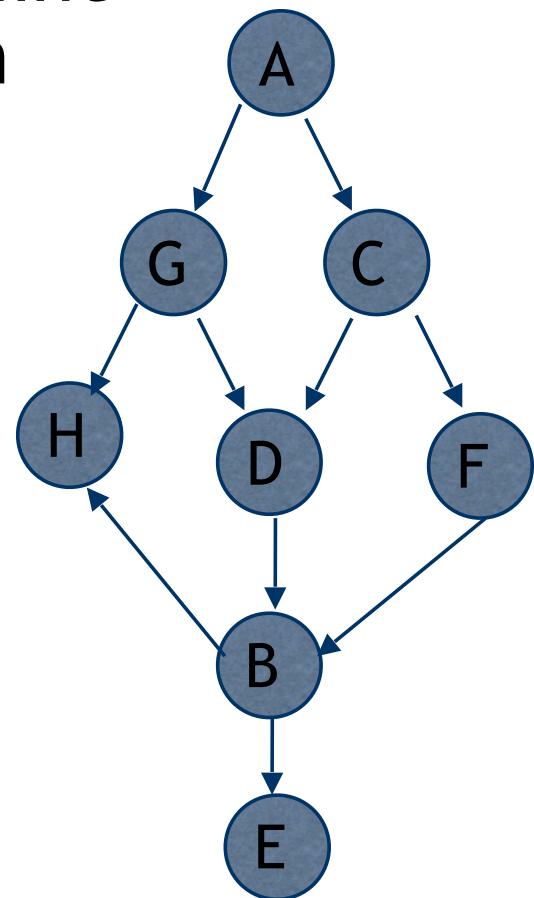


---

## D-Separation Example

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

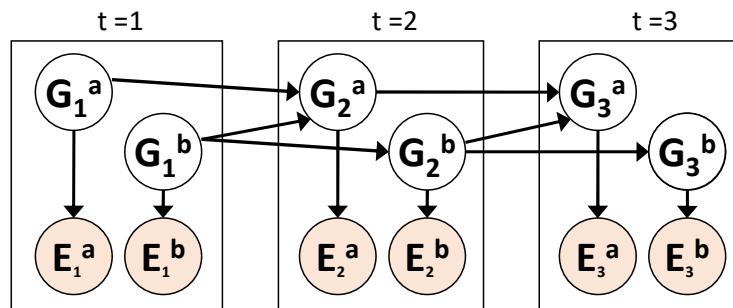
1. A and E given no evidence? N
2. A and E given {C}? N
3. A and E given {G,C}? Y
4. A and E given {G,C,H}? Y
5. A and E given {G,F}? N
6. A and E given {F,D}? Y
7. A and E given {F,D,H}? N
8. A and E given {B}? Y
9. A and E given {H,B}? Y
10. A and E given {G,C,D,H,F,B}? Y



---

# Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables from time  $t$  can condition on those from  $t-1$

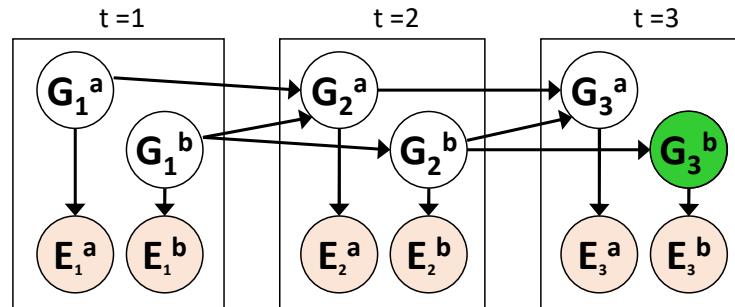


- Dynamic Bayes nets are a generalization of HMMs

[Slide created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

# Exact Inference in DBNs

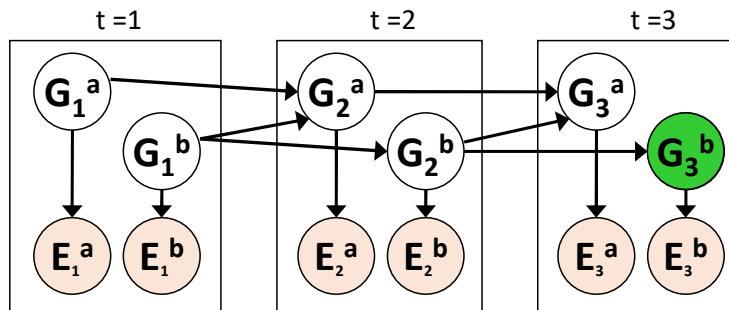
- Variable elimination applies to dynamic Bayes nets
- Procedure: “unroll” the network for  $T$  time steps, then eliminate variables until  $P(X_T | e_{1:T})$  is computed



- Online belief updates: Eliminate all variables from the previous time step; store factors for current time only (just like HMMs but the updates are not as simple)

# Particle Filtering in DBN

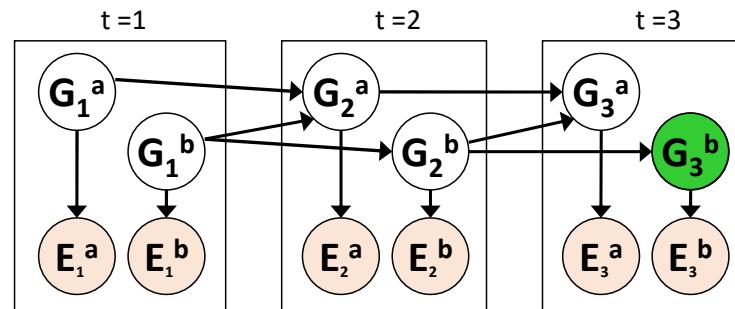
- As with HMMs we can have a set of particles. Now each particle must represent the state of all variables in the Bayes net at time step  $t$  except for the evidence variables (as these will become known)
- In this example, the particles will have a pair of values ( $G_t^a = p_a, G_t^b = p_b$ )



- To progress time for each particle we would sample a new value for  $G_{t+1}^a$  and  $G_{t+1}^b$  from the CPT given in the Bayes net:  $P(G_{t+1}^a | G_t^a = p_a, G_t^b = p_b)$  and  $P(G_{t+1}^b | G_t^a = p_a, G_t^b = p_b)$ .
- Note that we can sample each value  $p'_a$  and  $p'_b$  in the new particle  $p'$  separately since  $G_{t+1}^a$  and  $G_{t+1}^b$  are independent in this Bayes net given  $G_t^a = p_a$  and  $G_t^b = p_b$

# Particle Filtering in DBN

- Then to update by observation, we would weight each particle  $\mathbf{p}=(\mathbf{p}_a, \mathbf{p}_b)$  by the product of  $P(e_{t+1}^a | G_{t+1}^a = \mathbf{p}_a) * P(e_{t+1}^b | G_{t+1}^b = \mathbf{p}_b)$
- Again because  $E_{t+1}^a$  and  $E_{t+1}^b$  are independent of each other given  $G_{t+1}^a, G_{t+1}^b$  in this Bayes net.



- If the variables and evidence at each time step  $t$  are dependent on each other then the updates to the particles are a bit more complex.