CSC384h: Intro to Artificial Intelligence

Reasoning Under Uncertainty

- This material is covered in chapters 13, 14. Chapter 13 gives some basic background on probability from the point of view of AI. Chapter 14 talks about Bayesian Networks, exact reasoning in Bayes Nets as well as approximate reasoning, which will be main topics for us.

Uncertainty

- For the most part we have dealt with deterministic actions.
  - If you are in state S_1 and you execute action A you will always arrive at a particular state S_2.
- When there is a fixed initial state S_0, we will know exactly what state we are in after executing a sequence of deterministic actions (yours and the actions of the other agents).
- These assumptions are sensible in some domains
- But in many domains they are not true.
  - We have already seen some modeling of uncertainty in Expectimax search where we were not sure what our opponent would do.
  - But the actions were still deterministic-—we just didn't know which action was executed.

Uncertainty

- We might not know exactly what state we start off 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 might not know all of the effects of an action
  - The action might have a random component, like rolling dice.
  - We might not know all of the long term effects of a drug.
  - We might not know the status of a road when we choose the action of driving down it.
- In many applications we cannot ignore this uncertainty.
  - In some domains we can (e.g., build a schedule with some slack in it to account for delays).

Uncertainty

- In domains with uncertainty that we cannot ignore we still need to act, but we can't act solely on the basis of known true facts. We have to "gamble".
- E.g., we don't know for certain what the traffic will be like on a trip to the airport.

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 ½ 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 typically corresponds to maximizing one's expected utility.
  - various reason for doing this.

Maximizing Expected Utility

- Don't know what state arises from your actions due to uncertainty. But if you know (or can estimate) the probability you are in each of these different states (i.e., the probability distribution) you can compute the expected utility and take the actions that leads to a distribution with highest expected utility.
- We saw this idea before in Expectimax search.

Maximizing Expected Utility

- Probabilities of different outcomes.

Event	Go to Bloor St.	Go to Queen Street
Find Ice Cream	0.5	0.2
Find donuts	0.4	0.1
Find live music	0.1	0.7

- Your utility of different outcomes.

Event	Utility
Ice Cream	10
Donuts	5
Music	20

Maximizing Expected Utility

- Expected utility of different actions

Event	Go to Bloor St.	Go to Queen Street
Ice Cream	0.5 * 10	0.2 *10
Donuts	0.4 * 5	0.1 * 5
Music	0.1 * 20	0.7 * 20
Utility	**9.0**	**16.5**

- Maximize Expected Utility would say that you should "Go to Queen Street"
- But it would recommend going to Bloor if you liked ice cream and donuts more than live music.

Uncertainty

- To use the principle of maximizing expected utility we must have the probabilities
- So we need mechanisms for representing and reason about probabilities.

- We also need mechanisms for finding out utilities or preferences. This also is an active area of research.

Probability over Finite Sets.

- Probability is a function defined over a set of **atomic events U (the universe of events)**.
- It assigns a real number Pr(e) to each event $e \in U$, in the range [0,1].

- It assigns a value to every **set** of atomic events **F** by summing the probabilities of the members of that set.

$$\Pr(\mathbf{F}) = \sum_{e \in F} \Pr(e)$$

- Therefore: Pr({}) = 0
- Require Pr(**U**) = 1, i.e., sum over all events is 1.

Probability in General **(Review)**

- Given a set **U** (universe), a probability distribution over **U** is a function that maps ever subset of **U** to a real number and that satisfies the Axioms of Probability

1. **Pr(U) = 1**
2. **Pr(A) $\in$ [0,1]**
3. **Pr(A $\cup$ B) = Pr(A) + Pr(B) – Pr(A $\cap$ B)**
   - Count every element in A and in B—leads to counting elements in both A and B twice. So we have to subtract their sum once.

Probability over Feature Vectors

- Like CSPs, we have
  1. a set of variables $V_1, V_2, \ldots, V_n$
  2. a finite domain of values for each variable, $Dom[V_1]$, $Dom[V_2]$, …, $Dom[V_n]$.
- Each variable represents a different feature of the world that we might be interested in knowing.
- Each different total assignment to these variables will be an atomic event $e \in U$.
- So there are $\prod_i |Dom[V_i]|$ different atomic events.

Probability over Feature Vectors

- E.g., 3 variables V1, V2, V3, each with domain {1, 2, 3}. The set of atomic events are

(V1 = 1, V2 = 1, V3 = 1)	(V1 = 2, V2 = 1, V3 = 1)	(V1 = 3, V2 = 1, V3 = 1)
(V1 = 1, V2 = 1, V3 = 2)	(V1 = 2, V2 = 1, V3 = 2)	(V1 = 3, V2 = 1, V3 = 2)
(V1 = 1, V2 = 1, V3 = 3)	(V1 = 2, V2 = 1, V3 = 3)	(V1 = 3, V2 = 1, V3 = 3)
(V1 = 1, V2 = 2, V3 = 1)	(V1 = 2, V2 = 2, V3 = 1)	(V1 = 3, V2 = 2, V3 = 1)
(V1 = 1, V2 = 2, V3 = 2)	(V1 = 2, V2 = 2, V3 = 2)	(V1 = 3, V2 = 2, V3 = 2)
(V1 = 1, V2 = 2, V3 = 3)	(V1 = 2, V2 = 2, V3 = 3)	(V1 = 3, V2 = 2, V3 = 3)
(V1 = 1, V2 = 3, V3 = 1)	(V1 = 2, V2 = 3, V3 = 1)	(V1 = 3, V2 = 3, V3 = 1)
(V1 = 1, V2 = 3, V3 = 2)	(V1 = 2, V2 = 3, V3 = 2)	(V1 = 3, V2 = 3, V3 = 2)
(V1 = 1, V2 = 3, V3 = 3)	(V1 = 2, V2 = 3, V3 = 3)	(V1 = 3, V2 = 3, V3 = 3)

- There are 3*3*3 = 27 atomic events in this feature vector space.
- #of atomic events grows exponentially with the number of variables. n variables each with domain {0,1} → 2^n atomic events

Probability over Feature Vectors

- Often use probabilities of sets specified by assigning some variables.
  - **$V_1 = 1$ used to indicate the set of all atomic events (assignments) where $V_1 = 1$**

Pr($V_1 = 1$) =

$$\sum_{d_2 \in \mathrm{Dom}[V_2]} \sum_{d_3 \in \mathrm{Dom}[V_3]} \cdots \sum_{d_n \in \mathrm{Dom}[V_n]} \Pr(V_1=1, V_2=d_2, V_3=d_3, \ldots, V_n=d_n)$$

(V1 = 1, V2 = 1, V3 = 1)	(V1 = 2, V2 = 1, V3 = 1)	(V1 = 3, V2 = 1, V3 = 1)
(V1 = 1, V2 = 1, V3 = 2)	(V1 = 2, V2 = 1, V3 = 2)	(V1 = 3, V2 = 1, V3 = 2)
(V1 = 1, V2 = 1, V3 = 3)	(V1 = 2, V2 = 1, V3 = 3)	(V1 = 3, V2 = 1, V3 = 3)
(V1 = 1, V2 = 2, V3 = 1)	(V1 = 2, V2 = 2, V3 = 1)	(V1 = 3, V2 = 2, V3 = 1)
(V1 = 1, V2 = 2, V3 = 2)	(V1 = 2, V2 = 2, V3 = 2)	(V1 = 3, V2 = 2, V3 = 2)
(V1 = 1, V2 = 2, V3 = 3)	(V1 = 2, V2 = 2, V3 = 3)	(V1 = 3, V2 = 2, V3 = 3)
(V1 = 1, V2 = 3, V3 = 1)	(V1 = 2, V2 = 3, V3 = 1)	(V1 = 3, V2 = 3, V3 = 1)
(V1 = 1, V2 = 3, V3 = 2)	(V1 = 2, V2 = 3, V3 = 2)	(V1 = 3, V2 = 3, V3 = 2)
(V1 = 1, V2 = 3, V3 = 3)	(V1 = 2, V2 = 3, V3 = 3)	(V1 = 3, V2 = 3, V3 = 3)

Probability over Feature Vectors

- **$V_1 = 1$, $V_3 = 2$ used to indicate the set of all assignments where**
  $V_1 = 1$ and $V_3 = 2$.
  $Pr(V_1 = 1, V_3 = 2) =$

$$\sum_{d_2 \in \mathrm{Dom}[V_2]} \sum_{d_4 \in \mathrm{Dom}[V_4]} \cdots \sum_{d_n \in \mathrm{Dom}[V_n]} Pr(V_1=1, V_2=d_2, V_3=2, \ldots, V_n=d_n)$$

(V1 = 1, V2 = 1, V3 = 1)	(V1 = 2, V2 = 1, V3 = 1)	(V1 = 3, V2 = 1, V3 = 1)
(V1 = 1, V2 = 1, V3 = 2)	(V1 = 2, V2 = 1, V3 = 2)	(V1 = 3, V2 = 1, V3 = 2)
(V1 = 1, V2 = 1, V3 = 3)	(V1 = 2, V2 = 1, V3 = 3)	(V1 = 3, V2 = 1, V3 = 3)
(V1 = 1, V2 = 2, V3 = 1)	(V1 = 2, V2 = 2, V3 = 1)	(V1 = 3, V2 = 2, V3 = 1)
(V1 = 1, V2 = 2, V3 = 2)	(V1 = 2, V2 = 2, V3 = 2)	(V1 = 3, V2 = 2, V3 = 2)
(V1 = 1, V2 = 2, V3 = 3)	(V1 = 2, V2 = 2, V3 = 3)	(V1 = 3, V2 = 2, V3 = 3)
(V1 = 1, V2 = 3, V3 = 1)	(V1 = 2, V2 = 3, V3 = 1)	(V1 = 3, V2 = 3, V3 = 1)
(V1 = 1, V2 = 3, V3 = 2)	(V1 = 2, V2 = 3, V3 = 2)	(V1 = 3, V2 = 3, V3 = 2)
(V1 = 1, V2 = 3, V3 = 3)	(V1 = 2, V2 = 3, V3 = 3)	(V1 = 3, V2 = 3, V3 = 3)

Properties and Sets

- Any set of events A can be interpreted as a property: the set of events with property A. E.g., $V_1 = a$ is a property, all total assignments in which $V_1 = a$ have this property.
- Hence, we often write
  - $A \vee B$ — to represent the set of events with either property A or B: the set $A \cup B$

    $V_1 = a \vee V_1 = b$ — (Set of assignments were one of these is true)
  - $A \wedge B$ — to represent the set of events both property A and B: the set $A \cap B$

    $V_1 = a, V_2 = c$ — (Set of assignments where both of these are true)
  - $\neg A$ — to represent the set of events that do not have property A: the set U-A (i.e., the complement of A wrt the universe of events U)

    $V_1 \neq a$ — (set of assignments where V_1 has some value different from a)

Probability over Feature Vectors

- Problem:
  - There are an exponential number of atomic probabilities to specify. (can’t get all that data)
  - Computing, e.g, **Pr(V_1 = a)** would requires summing up an exponential number of items. (even if we have the data, we can compute efficiently with it)

- AI techniques for dealing with these two problems involve:
  - Using knowledge of conditional independence to simplify the problem and reduce the data and computational requirements.
  - Using approximation techniques after we have simplified with conditional independence. (Many approximation methods rely on having distributions structured by independence)

Key Probability Facts. **(Review)**

Conditional Probability **DEFINITION**	$\Pr(A \mid B) = \Pr(A \wedge B)\,/\Pr(B)$
Summing out Rule	$\Pr(A) = \sum_{C_i} \Pr(A \wedge C_i)$ when $\sum_{C_i} \Pr(C_i) = 1$ and $\Pr(C_i \wedge C_j) = 0\ (i \neq j)$
	$\Pr(A) = \sum_{C_i} \Pr(A \mid C_i) \mathrm{P}(C_i)$
Summing out Rule	$\Pr(A \mid B) = \sum_{C_i} \Pr(A \wedge C_i \mid B)$ when $\sum_{C_i} \Pr(C_i \mid B) = 1$ and $\Pr(C_i \wedge C_j \mid B) = 0\ (i \neq j)$
	$\Pr(A \mid B) = \sum_{C_i} \Pr(A \mid B \wedge C_i) \Pr(C_i \mid B)$

$\sum_{d \in Dom[V_i]} \Pr(V_i = d) = 1$ and $\Pr(V_i = d_k \wedge V_i = d_m) = 0\ (k \neq m)$

$\sum_{d \in Dom[V_i]} \Pr(V_i = d \mid V_j = c) = 1$ and $\Pr(V_i = d_k \wedge V_i = d_j \mid V_j = c) = 0\ (k \neq j)$

So summing out over the values of a feature is frequently used.

Key Probability Facts. **(Review)**

Bayes Rule	$\Pr(A\|B) = \Pr(B\|A)\Pr(A)/\Pr(B)$
Chain Rule	$\Pr(A_1 \wedge A_2 \cdots \wedge A_n)$ $= \Pr(A_n\|A_1 \cdots \wedge A_{n-1}) \Pr(A_{n-1}\|A_1 \cdots \wedge A_{n-2})$ $\cdots \Pr(A_2\|A_1) \Pr(A_1)$
A and B are independent DEFINITION	$\Pr(A\|B) = \Pr(A)$
	$\Pr(A \wedge B) = \Pr(A)\Pr(B)$
A and B are conditionally independent given C DEFINITION	$\Pr(A\|B \wedge C) = \Pr(A\|C)$
	$\Pr(A \wedge B\|C) = \Pr(A\|C)\Pr(B\|C)$

Normalizing

- If we have a vector of k numbers, e.g., [3, 4, 2.5, 1, 10, 21.5] we can **normalize** these numbers by dividing each number by the sum of the numbers:
  - 3 + 4 + 2.5 +1 +10 + 21.5 = 42
  - Normalized vector
    = normalize([3, 4, 2.5, 1, 10, 21.5]) =
    [3/42, 4/42, 2.5/42, 1/42, 10/42, 21.5/42]
    = [0.071, 0.095, 0.060, 0.024, 0.238, 0.512]
- After normalizing the vector of numbers sums to 1
  - Exactly what is needed for these numbers to specify a probability distribution.

Normalize Some useful Facts

- normalize([x1,x2, …, xk]) = [x1/α, x2/α, …., xk/α]
  where $\alpha = \sum_i x_i$
- normalize([x1, x2 …, xk]) = normalize([β *x1, β *x2, …., β *xk])
  for any constant β.
  Multiplying the vector by a constant does not change the normalized vector
- normalize(normalize([x1,x2 …, xk])) = normalize([x1,x2, …, xk])
  multiple normalizations don't do anything more.
- [x1, x2, …, xk] = normalize([x1,x2,…,xk]) * α
  the original vector can be recovered by multiplying by some constant α.
  (we divide by α to normalize, multiply by α to recover).

Variable Independence

- With feature vectors we often want to state collections of independencies or conditional independencies
- V1 = 1 is independent of V2 = 1
  V1 = 1 is independent of V2 = 2
  V1 = 2 is independent of V2 = 1
  V1 = 2 is independent of V2 = 2
  …
- (Different features are independent irrespective of the specific values they take).
- So we often use statements of **variable independence**

Variable Independence

- **Pr(V1|V2) = Pr(V1) (V1 and V2 are independent)**
- **Pr(V1|V2,V3) = Pr(V1|V3) (V1 is conditionally independent of V2 given V3)**

- **It means that the independence holds no matter what value the variable takes:**

  - $\forall d_1 \in Dom[V_1], d_2 \in Dom[V_2]$:
    $\Pr(V_1 = d_1 \mid V_2 = d_2) = \Pr(V_1 = d_1)$

  - $\forall d_1 \in Dom[V_1], d_2 \in Dom[V_2], d_3 \in Dom[V_3]$:
    $\Pr(V_1 = d_1 \mid V_2 = d_2, V_3 = d_3) = \Pr(V_1 = d_1 \mid V_3 = d_3)$

Probabilities over Variables

- Pr(V1,V2) for variable V1 and V2 refers to a set of probabilities, one probability for each pair of values value of V1 and V2
  - It specifies Pr(V1=d1, V2=d2) for all d1$\in$Dom[V1] and d2$\in$Dom[V1]
  - E.g., if Dom[V1] = Dom[V2] = {1, 2, 3}; Pr(V1,V2) will be a vector of 9 numbers
    [Pr(V1=1,V2=1), Pr(V1=1, V2=2), Pr(V1=1,V2=3),
    Pr(V1=2,V2=1), Pr(V1=2, V2=2), Pr(V1=2,V2=3),
    Pr(V1=3,V2=1), Pr(V1=3, V2=2), Pr(V1=3,V2=3))]
  - This vector of probabilities specifies the joint distribution of V1 and V2

Conditional Probabilities over Variables

- Pr(V1|V2,V3) specifies a collection of distributions over V1, one for each d2∈Dom(V2) and d3∈Dom(V3)
- E.g., if Dom[V1] = Dom[V2] = Dom[V3] = {1, 2, 3} then
  Pr(V1|V2, V3) will specify 27 values:

Pr(V1=1|V2=1,V3=1)	Pr(V1=2|V2=1, V3=1)	Pr(V1=3|V2=1 V3=1)
Pr(V1=1|V2=1,V3=2)	Pr(V1=2|V2=1, V3=2)	Pr(V1=3|V2=1 V3=2)
Pr(V1=1|V2=1,V3=3)	Pr(V1=2|V2=1, V3=3)	Pr(V1=3|V2=1 V3=3)
Pr(V1=1|V2=2,V3=1)	Pr(V1=2|V2=2, V3=1)	Pr(V1=3|V2=2 V3=1)
Pr(V1=1|V2=2,V3=2)	Pr(V1=2|V2=2, V3=2)	Pr(V1=3|V2=2 V3=2)
Pr(V1=1|V2=2,V3=3)	Pr(V1=2|V2=2, V3=3)	Pr(V1=3|V2=2 V3=3)
Pr(V1=1|V2=3,V3=1)	Pr(V1=2|V2=3, V3=1)	Pr(V1=3|V2=3 V3=1)
Pr(V1=1|V2=3,V3=2)	Pr(V1=2|V2=3, V3=2)	Pr(V1=3|V2=3 V3=2)
Pr(V1=1|V2=3,V3=3)	Pr(V1=2|V2=3, V3=3)	Pr(V1=3|V2=3 V3=3)

- The values in each row form a different probability distribution.

Conditional Probabilities over Variables

- Useful to think of Pr(Vi) as a function. Give it a value for Vi it returns a number (a probability). These numbers form a probability distribution. The numbers can be stored in a table.
- Similarly Pr(V1|V2,V3) is also a function. Give it three values, one for V1, V2 and V3, it will return a number (a conditional probability). Note that for each fixed value of V2 and V3 this function specifies a probability distribution over the values of V1

Pr(V1| V2=1, V3=1) — a vector of probabilities, one for each assignment to V1

Pr(V1 |V2=1, V3=2) —another distribution over V1

Unconditional Independence

- K coin flips
- X1, …, Xk variables representing outcome of i'th coin flip
- Dom[X1] = Dom[X2] = … = Dom[Xk] = {Heads, Tails}

- Pr(X1=Heads, X2=Tails) = Pr(X1=Heads) Pr(X2=Tails)

  equivalently

  Pr(X2=Tails|X1 = Heads) = Pr(X2=Tails)

- This holds for all values of X1 and X2 and X3 …, and Xk so we can write

  Pr(X1|X2) = P(X1); P(X2|X3) = P(X3) …

  These variable independencies represent independency for all specific values.

Conditional Independence

- **E.g., Traffic, Umbrella, Raining**
  Dom[Traffic] = {Heavy, Normal}
  Dom[Umbrella] = {Used, Not used}
  Dom[Raining] = {Yes, No}

- Unconditional Independence is quite rare

  - Pr(Umbrella|Raining)
    Definitely not---Raining is main reason for using the Umbrella

  - Pr(Raining|Traffic) = Pr(Raining)
    No---heavy traffic is evidence for rain.

  - Pr(Umbrella|Traffic) = Pr(Umbrella)
    No---heavy traffic is evidence for rain which would influence Umbrella usage

- Conditional Independence quite common.
  - Pr(Traffic, Umbrella | Raining) =
    Pr(Traffic|Raining)*P(Umbrella|Raining)
    Yes, once we know the status of Raining, heavy traffic and umbrella usage are independent of each other

Conditional Probabilities over Variables

- AI techniques for dealing with these two problems involve using knowledge of conditional independence to simplify the problem

- We have a great deal of knowledge about conditional independencies in the real world

Exploiting Conditional Independence

- Consider a story:
  - If Craig woke up too early E, Craig probably needs coffee C; if C, Craig needs coffee, he's likely angry A. If A, there is an increased chance of an aneurysm (burst blood vessel) B. If B, 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

Exploiting Conditional Independence

E – Craig woke too early   A – Craig is angry   H – Craig hospitalized

C – Craig needs coffee   B – Craig burst a blood vessel

All these variables have domain [True, False] (they are Boolean variables), so we write lower case "e" to indicate that E = True, and ~e to indicate E = False, etc.

Cond’l Independence in our Story

- If you learned any of E, C, A, or B, your assessment of Pr(H) would change.
  - E.g., if any of these are seen to be true, you would increase Pr(h) and decrease Pr(~h).
  - So H is *not independent* of E, or C, or A, or B.
- But if you knew value of B (true or false), learning the value of E, C, or A, would not influence Pr(H). Influence these factors have on H is mediated by their influence on B.
  - Craig doesn't get sent to the hospital because he's angry, he gets sent because he's had an aneurysm.
  - So H is *independent* of E, and C, and A, *given* B

Cond'l Independence in our Story

- Similarly:

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

- This means that:

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

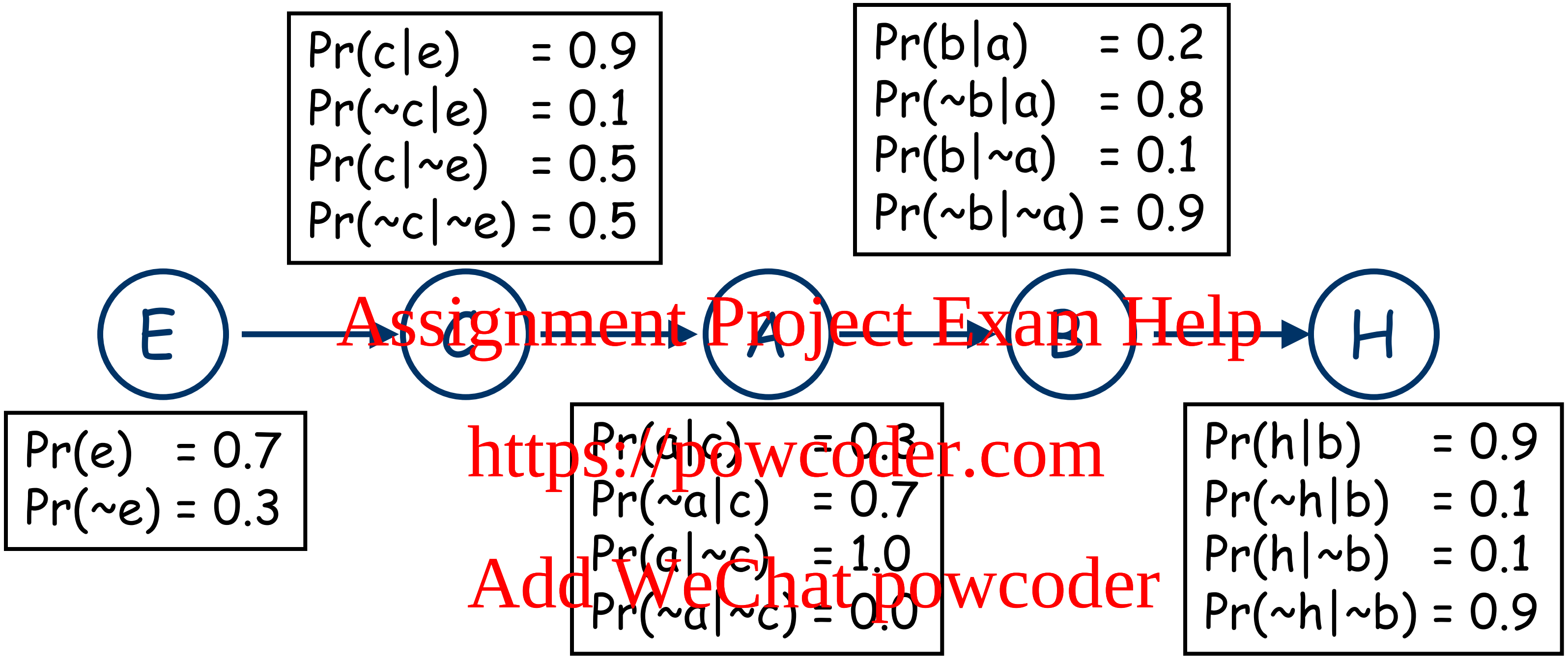
Cond'l Independence in our Story

- By the chain rule (for any instantiation of H…E):
  - Pr(H,B,A,C,E) =

    Pr(H|B,A,C,E) Pr(B|A,C,E) Pr(A|C,E) Pr(C|E) Pr(E)
- By our independence assumptions:
  - Pr(H,B,A,C,E) =

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

Example Quantification



- Specifying the joint distribution over E,C,A,B,H requires only 16 parameters (actually only 9 numbers since half the numbers are not needed since, e.g., P(~a|c) + P(a|c) = 1), instead of 32 for the explicit representation
  - linear in number of vars instead of exponential!
  - linear generally if dependence has a chain structure

Inference is Easy

$$E \rightarrow C \rightarrow A \rightarrow B \rightarrow H$$

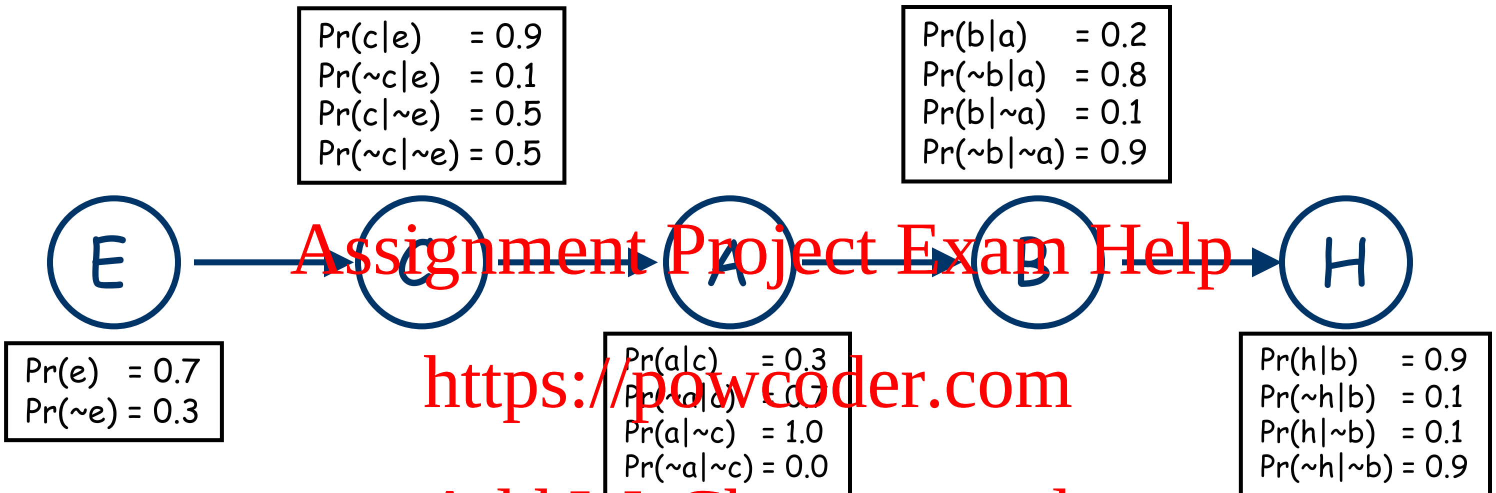
- Want to know P(a)? Use summing out rule:

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

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

These are all terms specified in our local distributions!

Inference is Easy



- Computing P(a) in more concrete terms:
  - P(c) = P(c|e)P(e) + P(c|~e)P(~e)
    = 0.9 * 0.7 + 0.5 * 0.3 = 0.78
  - P(~c) = P(~c|e)P(e) + P(~c|~e)P(~e)
    = 0.1 * 0.7 + 0.5 * 0.3 = 0.22 = 1 – P(c)
  - P(a) = P(a|c)P(c) + P(a|~c)P(~c)
    = 0.3 * 0.78 + 1.0 * 0.22 = 0.454
  - P(~a) = P(~a|c)P(c) + P(~a|~c)
    = 0.7 * 0.78 + 0.0 * 0.22 = 0.546 = 1 – P(a)

Bayesian Networks

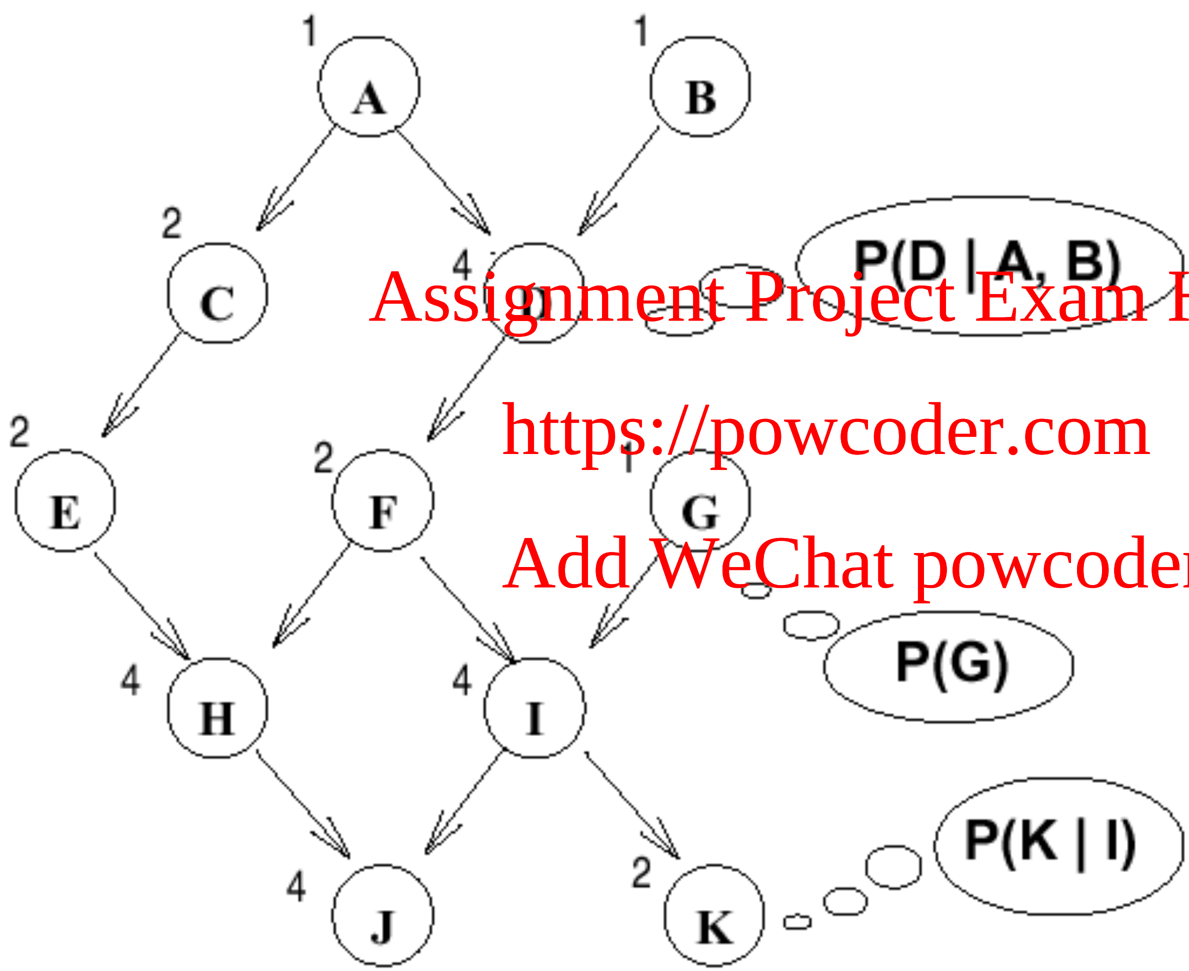
- The structure above is a *Bayesian network*. A BN is a *graphical representation* of the direct dependencies over a set of variables, together with a set of *conditional probability tables* quantifying the strength of those influences.

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

Bayesian Networks

- A BN over variables $\{X_1, X_2, \ldots, X_n\}$ consists of:
  - a DAG (directed acyclic graph) whose nodes are the variables
  - a set of **CPTs** (conditional probability tables) $\Pr(X_i \mid Par(X_i))$ for each X_i
- Key notions:
  - parents of a node: $Par(X_i)$
  - 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

Example (Binary valued Variables)



- A couple of the CPTS are "shown"

Semantics of Bayes Nets.

- A Bayes net specifies that the joint distribution over all of the variables in the net can be written as the following product decomposition.
- $Pr(X_1, X_2, \ldots, X_n)$
  $= Pr(X_n \mid Par(X_n)) * Pr(X_{n-1} \mid Par(X_{n-1}))$
  $* \cdots * Pr(X_1 \mid Par(X_1))$
- Like other equations over variables this decomposition holds for any set of values $d_1, d_2, \ldots, d_n$ for the variables $X_1, X_2, \ldots, X_n$.

Semantics of Bayes Nets.

- E.g., say we have X_1, X_2, X_3 each with domain $Dom[X_i] = \{a, b, c\}$ and we have

  $Pr(X_1,X_2,X_3)$
  $= P(X_3|X_2)\ P(X_2)\ P(X_1)$

  Then

  $Pr(X_1=a,X_2=a,X_3=a)$
  $= P(X_3=a|X_2=a)\ P(X_2=a)\ P(X_1=a)$

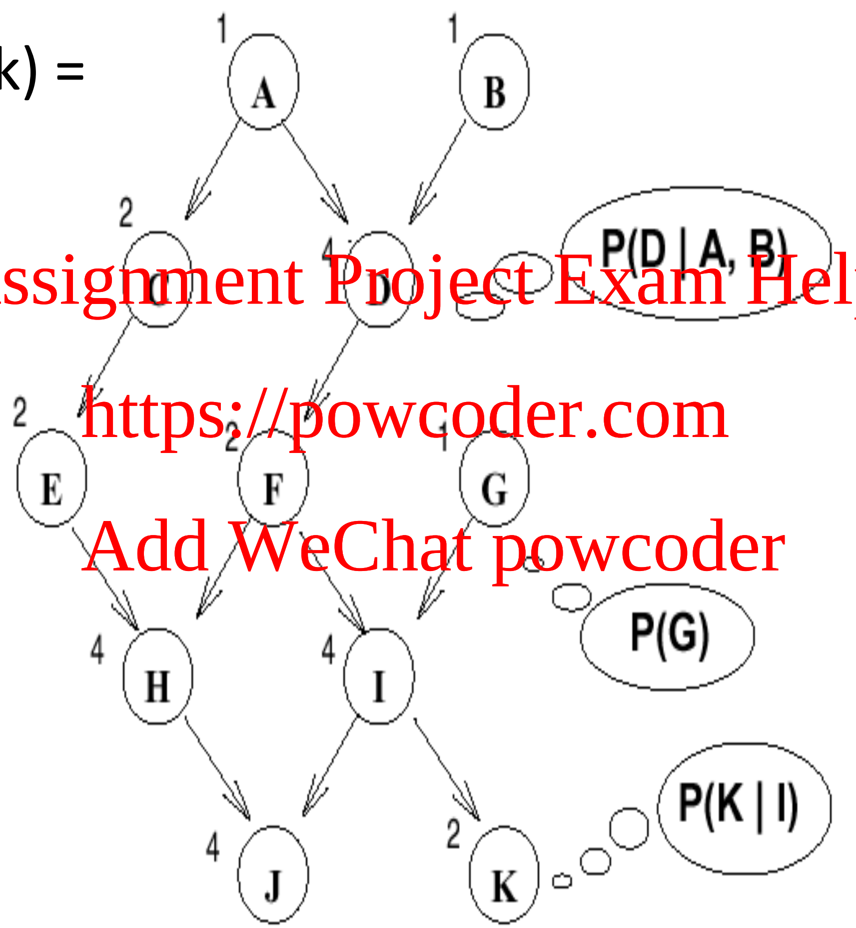
  $Pr(X_1=a,X_2=a,X_3=b)$
  $= P(X_3=b|X_2=a)\ P(X_2=a)\ P(X_1=a)$

  $Pr(X_1=a,X_2=a,X_3=c)$
  $= P(X_3=c|X_2=a)\ P(X_2=a)\ P(X_1=a)$
  $Pr(X_1=a,X_2=b,X_3=a)$
  $= P(X_3=a|X_2=b)\ P(X_2=b)\ P(X_1=a)$
  …

Example (Binary valued Variables)

$Pr(a,b,c,d,e,f,g,h,i,j,k) =$

$Pr(a)$
$\times Pr(b)$
$\times Pr(c|a)$
$\times Pr(d|a,b)$
$\times Pr(e|c)$
$\times Pr(f|d)$
$\times Pr(g)$
$\times Pr(h|e,f)$
$\times Pr(i|f,g)$
$\times Pr(j|h,i)$
$\times Pr(k|i)$



- Explicit joint requires $2^{11}-1 = 2047$ parmeters
- BN requires only 27 parmeters (the number of entries for each CPT is listed)

Semantics of Bayes Nets.

- Note that this means we can compute the probability of any setting of the variables using only the information contained in the CPTs of the network.

Constructing a Bayes Net

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

- Take any ordering of the variables. From the chain rule we obtain.

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

- Now for each Xi go through its conditioning set $X_1,\ldots,X_{i-1}$, and remove all variables X_j such that X_i is conditionally independent of X_j given the remaining variables.
- The final product will specify a Bayes net.

Constructing a Bayes Net

- The end result will be a product decomposition/Bayes net
  $\Pr(X_n \mid Par(X_n)) \Pr(X_{n-1} \mid Par(X_{n-1}))\ldots \Pr(X_1)$

- Now we specify the numeric values associated with each term $\Pr(X_i \mid Par(X_i))$ in a CPT.
- Typically we represent the CPT as a table mapping each setting of $\{X_i, Par(X_i)\}$ to the probability of X_i taking that particular value given that the variables in $Par(X_i)$ have their specified values.
- If each variable has d different values.
  - We will need a table of size $d^{|\{X_i, Par(X_i)\}|}$.
  - That is, exponential in the size of the parent set.

- Note that the original chain rule
  $\Pr(X_1,\ldots,X_n) = \Pr(X_n \mid X_1,\ldots,X_{n-1})\Pr(X_{n-1} \mid X_1,\ldots,X_{n-2})\ldots\Pr(X_1)$
  requires as much space to represent as representing the probability of each individual atomic event.

Causal Intuitions

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

Causal Intuitions

- Malaria, the flu and a cold all "cause" aches. So use the ordering that causes come before effects
  Malaria, Flu, Cold, Aches

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

- Each of these disease affects the probability of aches, so the first conditional probability cannot simplify.

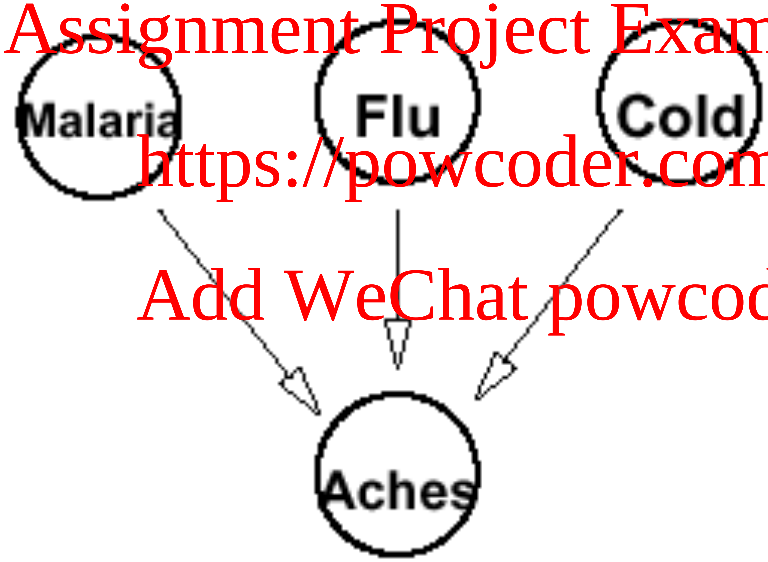
- It is reasonable to assume that these diseases are independent of each other: having or not having one does not change the probability of having the others. So $\Pr(C|M,F) = \Pr(C)$ $\Pr(F|M) = \Pr(F)$

- This gives us the simplified decomposition of the joint probablity

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

Causal Intuitions

- This yields a fairly simple Bayes net.
- Only need one big CPT, involving the family of "Aches".



Causal Intuitions

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

    Pr(A,C,F,M) = Pr(M | A,C,F) Pr(F | A,C) Pr(C | A) Pr(A)

  - We can't reduce Pr(M | A,C,F). 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 of Flu lowers the probability of Aches indicating Malaria since they "explain away" Aches!

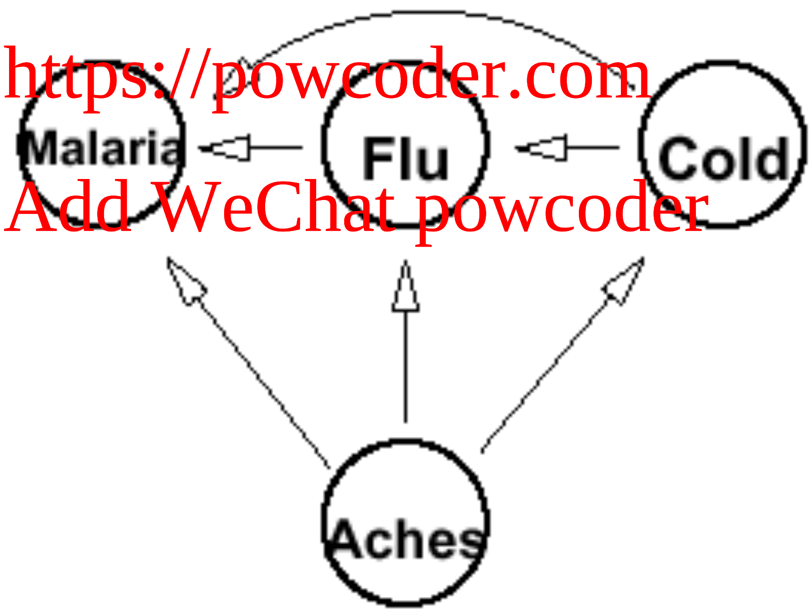
Causal Intuitions

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

- Similarly, we can't reduce $\Pr(F|A,C)$ – Cold explains away Aches
- $\Pr(C|A) \neq \Pr(C)$ — clearly probability of Cold goes up with Aches

Causal Intuitions

- 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 Examples

- You are at work, neighbor John calls to say your alarm is ringing, but neighbor Mary doesn't call. Sometimes alarm is set off by minor earthquakes. Is there a burglar?
- Variables: *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

- Network topology reflects "causal" knowledge:

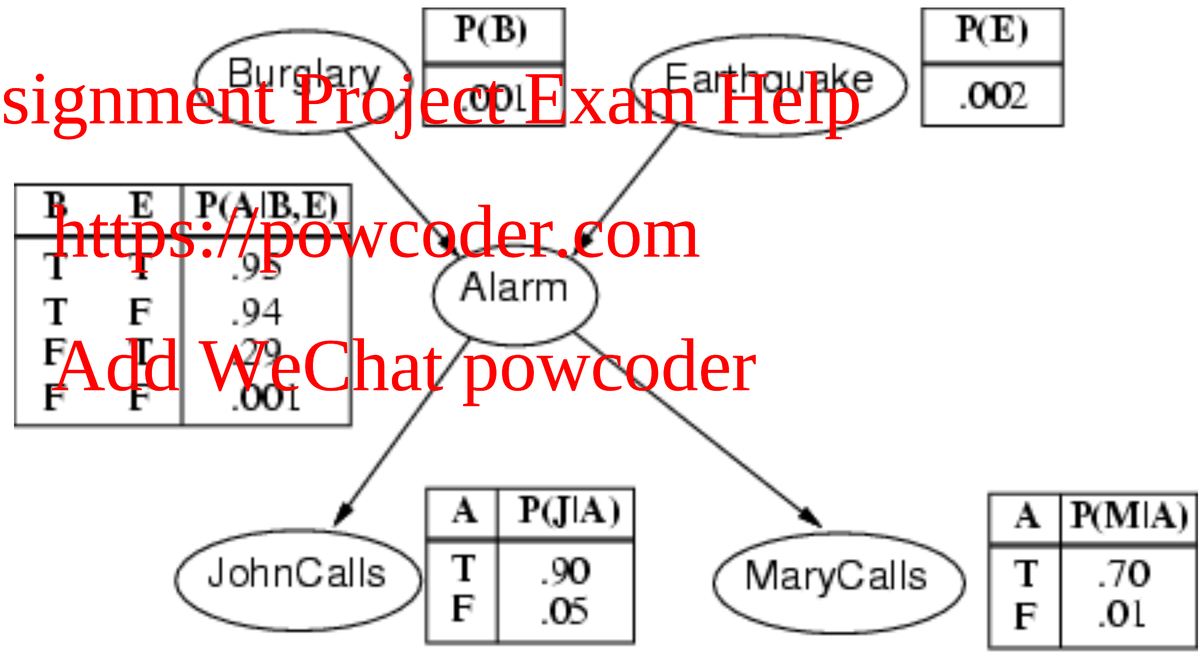
  - A burglar can cause the alarm to go off
  - An earthquake can cause the alarm to go off

  - The alarm can cause Mary to call
  - The alarm can cause John to call

  - But the alarm does not cause an earthquake, nor does Mary or John calling cause the alarm

Burglary Example

- A burglary 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. $2^5 - 1 = 31$)

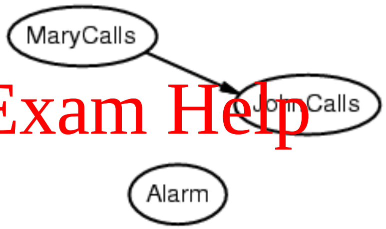
Example of Constructing Bayes Network

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

$\mathbf{P}(J \mid M) = \mathbf{P}(J)$?

Example continue…

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



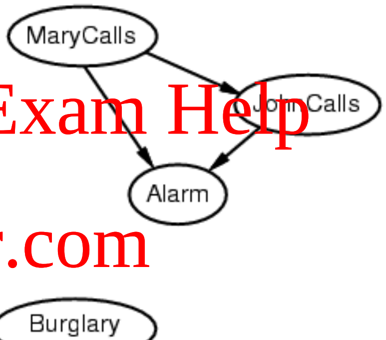
P(*J | M*) = ***P***(*J*)?

No

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

Example continue…

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



$\mathbf{P}(J \mid M) = \mathbf{P}(J)$?

No

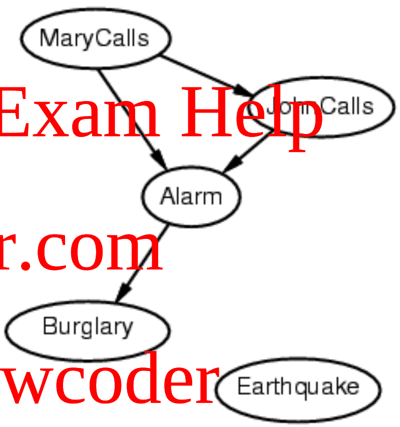
$\mathbf{P}(A \mid J, M) = \mathbf{P}(A \mid J)$? $\mathbf{P}(A \mid J, M) = \mathbf{P}(A)$? **No**

$\mathbf{P}(B \mid A, J, M) = \mathbf{P}(B \mid A)$?

$\mathbf{P}(B \mid A, J, M) = \mathbf{P}(B)$?

Example continue…

- 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**

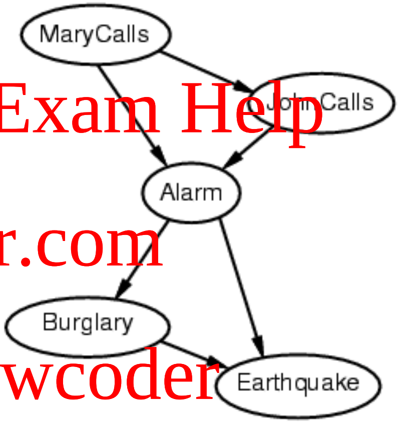
P(*B* | *A*, *J*, *M*) = ***P***(*B*)? **No**

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

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

Example continue…

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



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

No

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

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

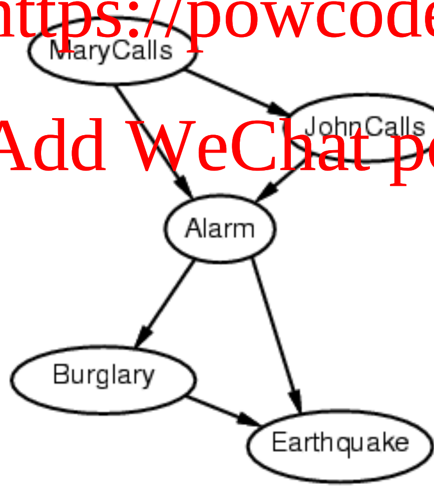
P(*B* | *A*, *J*, *M*) = ***P***(*B*)? **No**

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

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

Example continue…

- Deciding conditional independence **is hard** in non-causal directions!
- (Causal models and conditional independence seem hardwired for humans!)
- Network is **less compact**: 1 + 2 + 4 + 2 + 4 = 13 numbers needed



Inference in Bayes Nets

- Given a Bayes net

  $\Pr(X_1, X_2, \ldots, X_n)$
  $= \Pr(X_n \mid Par(X_n)) * \Pr(X_{n-1} \mid Par(X_{n-1}))$
  $* \cdots * \Pr(X_1 \mid Par(X_1))$

- And some evidence E = {specific known values for some of the variables} we want to compute the new probability distribution

  $\Pr(X_k \mid E)$

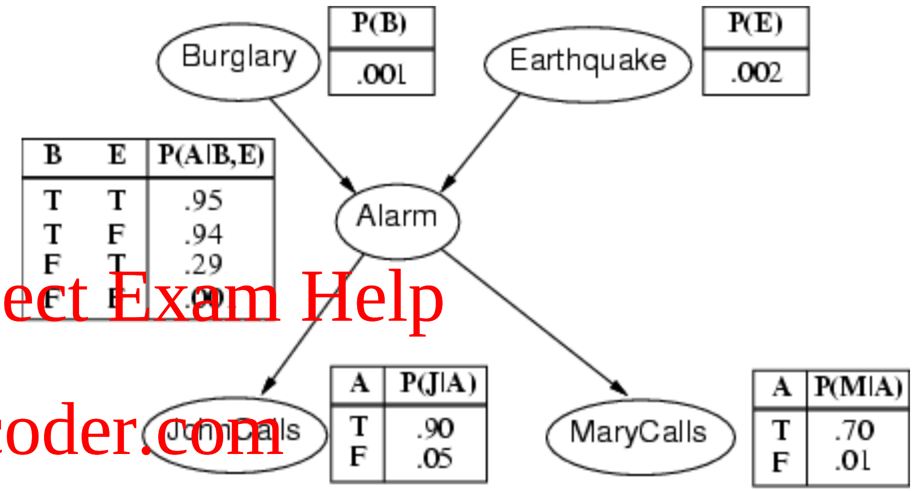
- That is, we want to figure out $\Pr(X_k = d \mid E)$ for all $d \in Dom[X_k]$

Inference in Bayes Nets

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

Inference in Bayes Nets

- In the Alarm example:



- Pr(Burglary,Earthquake, Alarm, JohnCalls, MaryCalls) =
  Pr(Earthquake) * Pr(Burglary) *
  Pr(Alarm|Earthquake,Burglary) *
  Pr(JohnCalls|Alarm) * Pr(MaryCalls|Alarm)
- And, e.g., we want to compute things like
  Pr(Burglary=True| MaryCalls=false, JohnCalls=true)

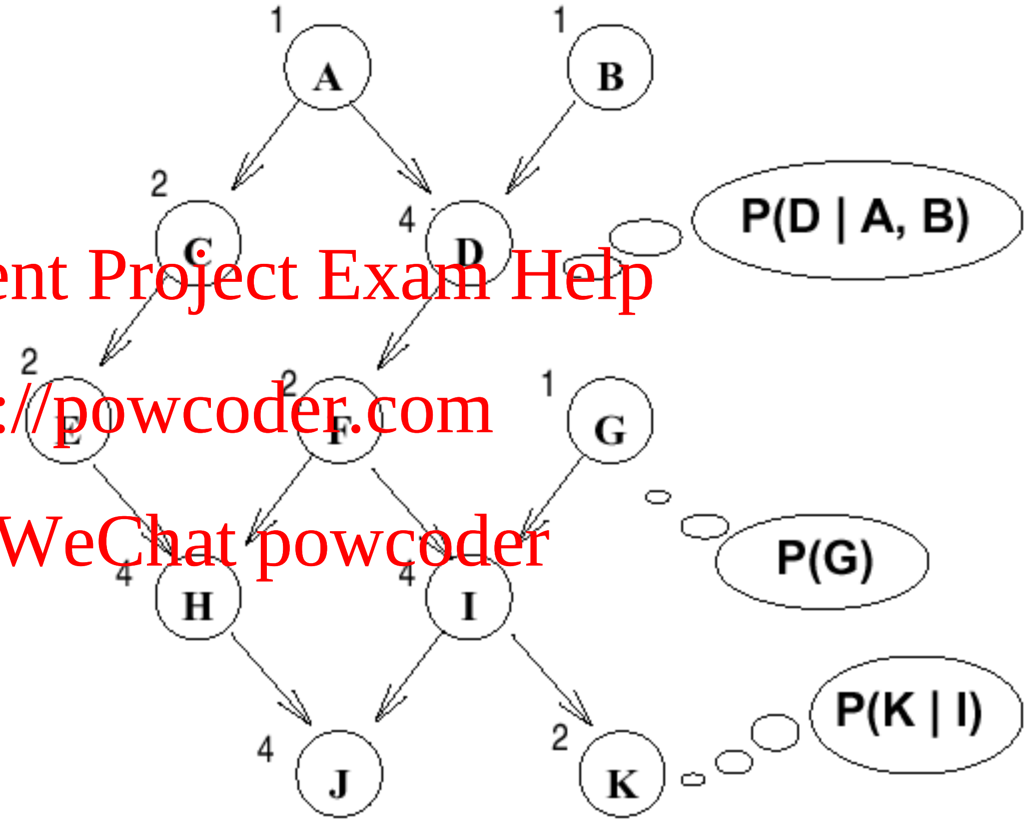
Variable Elimination

- Variable elimination uses the product decomposition that defines the Bayes Net and the summing out rule to compute posterior probabilities from the information (CPTs) already in the network.

Example (Binary valued Variables)

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

$\Pr(A)$
$\times \Pr(B)$
$\times \Pr(C|A)$
$\times \Pr(D|A,B)$
$\times \Pr(E|C)$
$\times \Pr(F|D)$
$\times \Pr(G)$
$\times \Pr(H|E,F)$
$\times \Pr(I|F,G)$
$\times \Pr(J|H,I)$
$\times \Pr(K|I)$



Example

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

$\Pr(A)\Pr(B)\Pr(C|A)\Pr(D|A,B)\Pr(E|C)\Pr(F|D)\Pr(G)$
$\Pr(H|E,F)\Pr(I|F,G)\Pr(J|H,I)\Pr(K|I)$

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

$\Pr(D|h,i)$ (h: H is true, -h: H is false)

1. Write as a sum for each value of D

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

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

Example

2. Pr(d,h,-i) + Pr(-d,h,-i) = Pr(h,-i)
3. Pr(d|h,-i) = Pr(d,h,-i)/Pr(h,-i)
   Pr(-d|h,-i) = Pr(-d,h,-i)/Pr(h,-i)

So we are computing Pr(d,h,-i) and Pr(-d,h,-i) and then dividing by their sum:

[Pr(d,h,-i), Pr(-d,h,-i)] / (Pr(d,h,-i) + Pr(-d,h,-i))

This the same as normalizing the vector [Pr(d,h,-i), Pr(-d,h,-i)].

We always normalize at the end to obtain the probability distribution.

Example

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

Use Bayes Net product decomposition to rewrite summation:

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

Now rearrange summations so that we are not summing over terms do not depend on the summed variable.

Example

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

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

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

Example

- Now we compute the sums innermost first.

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

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

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

Example

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

- Note:

1. We have a new expression that does not have the variable K. K has been eliminated.

2. c_1 does not depend on any of the variables so we can move it to the front

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

Example

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

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

Now we have eliminated J, again the sum does not depend on any variable

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

Example

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

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

Note: The terms involving G also contain the variable F. So when we sum out G we end up with a different number for every assignment to F. (In this case –f, and f).

So the sum depends on F. But once F is fixed to f or –f, the sum yields a fixed number.

Example

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

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

Let's introduce a new "function" to represent this sum. This new function depends on F

$f1(F) = \Pr(g)\Pr(-i|F,g) + \Pr(-g)\Pr(-i|F,-g)$

$f1(f) = \Pr(g)\Pr(-i|f,g) + \Pr(-g)\Pr(-i|f,g)$
$f1(-f) = \Pr(g)\Pr(-i|-f,g) + \Pr(-g)\Pr(-i|-f,g)$

are fixed numbers

Example

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

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

This sum depends on E. So we can once again introduce a new function to represent this sum. This new function depends on E

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

$f2(e)$ and $f2(-e)$ are fixed numbers

Example

- We can continue this way eliminating one variable after another. Until we sum out A to obtain a single number equal to Pr(d,h,-i)
- A similar computation produces Pr(-d,h,-i)

- Normalizing these two numbers gives us Pr(d|h,i) and Pr(-d|h,i)

- Or as we will see later, we can keep D as a variable, and compute a function of D, fk(D) whose two values fk(d) and fk(-d) are the values we want.

Variable Elimination (Dynamic Programming)

- This process is called variable elimination.

- By computing the intermediate functions f1(F), f2(E) etc. we are actually storing values that we can reuse many times during the computation.

- In this way variable elimination is a form of dynamic programming, where we save sub-computations to avoid re-computations.

Relevance (return to this later)

- Note that in the sum
  $\sum_A \Pr(A) \sum_B \Pr(B) \Pr(d|A,B) \sum_C \Pr(C|A) \sum_E \Pr(E|C)$
  $\quad \sum_F \Pr(F|d) \Pr(h|E,F) \sum_G \Pr(G) \Pr(-i|F,G)$
  $\quad \sum_J \Pr(J|h,-i)$
  $\quad \sum_K \Pr(K|-i)$

  we have that $\sum_K \Pr(K|-i) = 1$ (Why?)

  Furthermore $\sum_J \Pr(J|h,-i) = 1$.

  So we could drop these last two terms from the computation---J and K are not relevant given our query D and our evidence –i and –h. For now we keep these terms.

Variable Elimination (VE)

- In general, at each stage VE will sum out the innermost variable, computing a new function over the variables in that sum.
- The function specifies one number for each different instantiation of its variables.
- We store these functions as a table with one entry per instantiation of the variables
- The size of these tables is exponential in the number of variables appearing in the sum, e.g.,

$$\sum_F \Pr(F|D)\, \Pr(h|E,F)\, t(F)$$

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 these 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 to a number.

  - A tabular representation is exponential in the number of variables in the factor.

Operations on Factors

- If we examine the inside-out summation process we see that various operations occur on factors.
- We can specify the algorithm for Variable Elimination by precisely specifying these operations.

- Notation: f(**X**,**Y**) denotes a factor over the variables **X** $\cup$ **Y** (where **X** and **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 * g (or sometimes just h = fg), is defined:

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

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 with variable X (**Y** is a set)
- We *sum out* variable X from f to produce a new factor h = Σ_X f, which is defined:

$$h(\mathbf{Y}) = \Sigma_{x \in Dom(X)} f(x, \mathbf{Y})$$

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 with variable X (**Y** is a set)
- We *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) = $f_{A=a}$	
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** (set of variables observed to have values **e**), remaining vars **Z**. Let F be original CPTs.

1. Replace each factor $f \in F$ that mentions a variable(s) in **E** with its restriction $f_{\mathbf{E}=\mathbf{e}}$ (this might yield a factor over no variables, a constant)
2. For each Z_j—in the order given—eliminate $Z_j \in \mathbf{Z}$ as follows:
   (a) Compute new factor $g_j = \Sigma_{Z_j} f_1 \times f_2 \times \ldots \times f_k$,
   where the f_i are the factors in F that include Z_j
   (b) Remove the factors f_i that mention Z_j from F and add new factor g_j to F
3. The remaining factors refer only to the query variable Q. Take their product and normalize to produce Pr(Q|E)

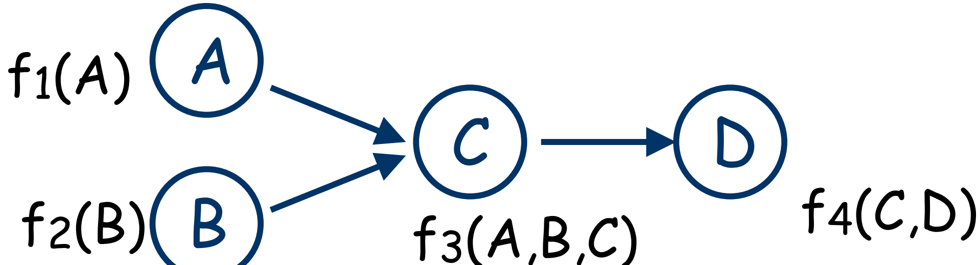
VE: Example

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

Query: P(A)?

Evidence: D = d

Elim. Order: C, B



Restriction: replace $f_4(C,D)$ with $f_5(C) = f_4(C,d)$

Step 1: Compute & Add $f_6(A,B) = \Sigma_C\, f_5(C)\, f_3(A,B,C)$

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

Step 2: Compute & Add $f_7(A) = \Sigma_B\, f_6(A,B)\, f_2(B)$

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

Last factors: $f_7(A)$, $f_1(A)$. The product $f_1(A) \times f_7(A)$ is (unnormalized) posterior. So…

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

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

Numeric Example

- Here's the example with some numbers

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

Numeric Example

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

$$f_4(b) = \Sigma_A f_2(A,b)f_1(A)$$
$$= f_2(a,b)f_1(a) + f_2(\sim a,b)f_1(\sim a)$$
$$= 0.9{*}0.9 + 0.4 * 0.1 = 0.85$$

Numeric Example

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

$f_4(\sim b) = \Sigma_A\ f_2(A,\sim b)f_1(A)$

$= f_2(a,\sim b)f_1(a) + f_2(\sim a,\sim b)f_1(\sim a)$

$= 0.1*0.9 + 0.6 * 0.1 = 0.15$

Note: $f_4(b) + f_4(\sim b) = 1$. But the intermediate factors are not always probabilities.

Numeric Example

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

$f_5(c) = \Sigma_B f_3(B,c)f_4(B)$

$= f_3(b,c)f_4(b) + f_3(\sim b,c)f_4(\sim b)$

$= 0.7*0.85 + 0.2*0.15 = 0.625$

Numeric Example

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

$f_5(\sim c) = \Sigma_B f_3(B,\sim c)f_4(B)$

$= f_3(b,\sim c)f_4(b) + f_3(\sim b,\sim c)f_4(\sim b)$

$= 0.3*0.85 + 0.8*0.15 = 0.375$

Numeric Example

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

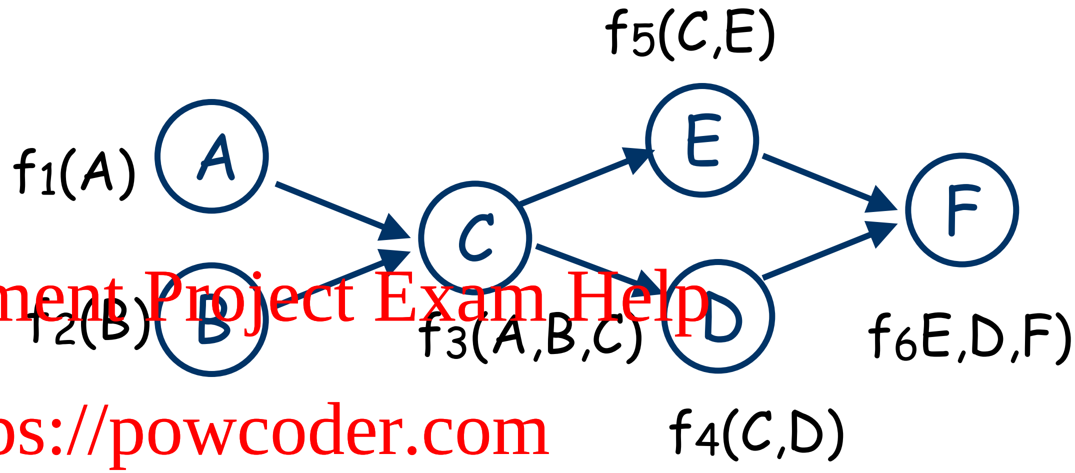
$f_5(C)$ is already normalized

$\Pr(c) = 0.625$

$\Pr(\sim c) = 0.375$

VE: Buckets as a Notational Device

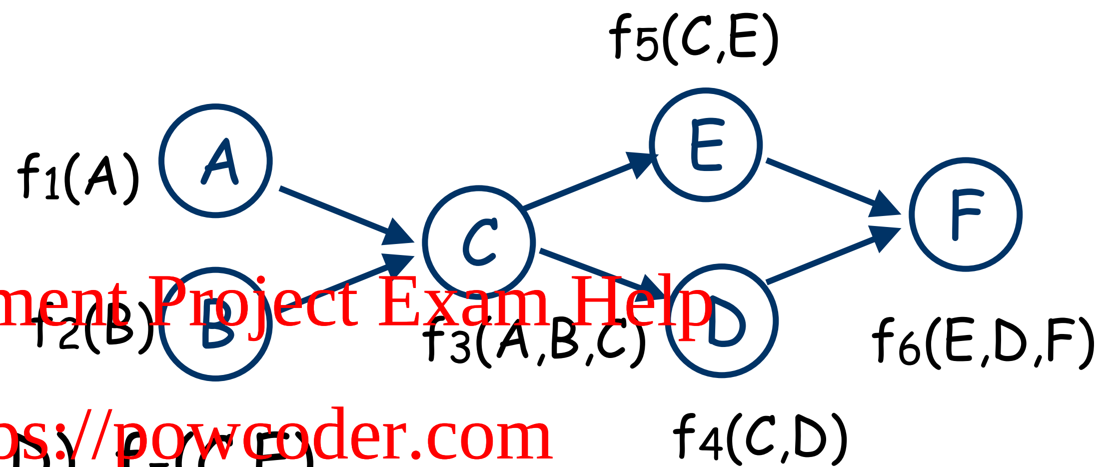
Ordering:
C,F,A,B,E,D



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

VE: Buckets—Place Original Factors in first bucket that contains one of its variables

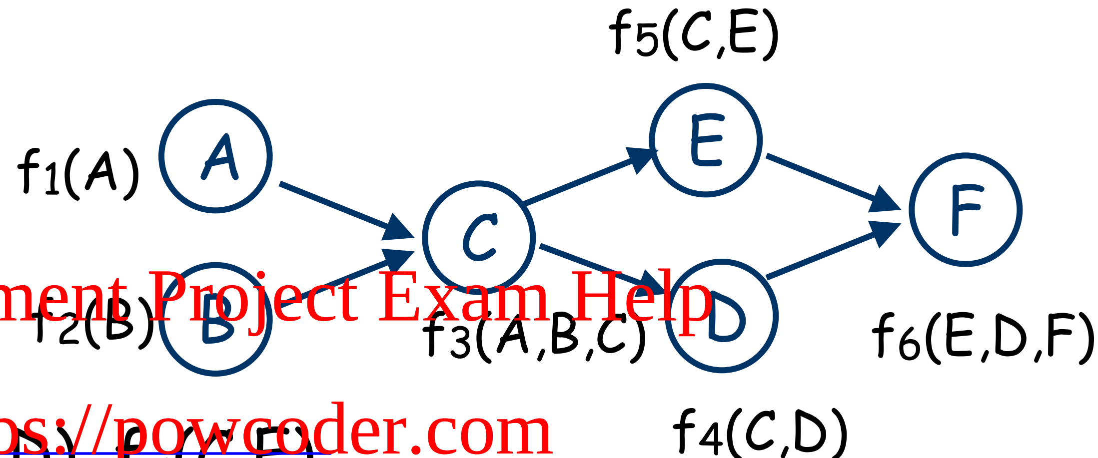
Ordering:
C,F,A,B,E,D



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: Buckets—Place Original Factors in first bucket that contains one of its variables

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

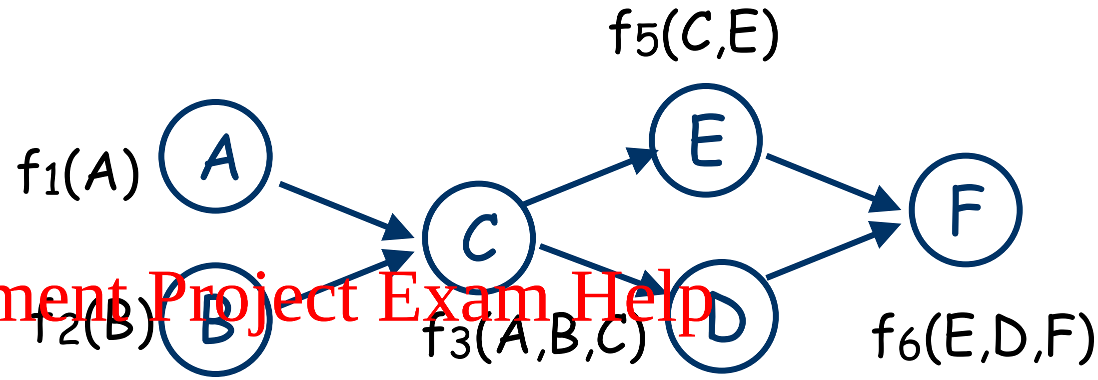


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E:
6. D:

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

VE: Buckets—Place Original Factors in first bucket that contains one of its variables

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



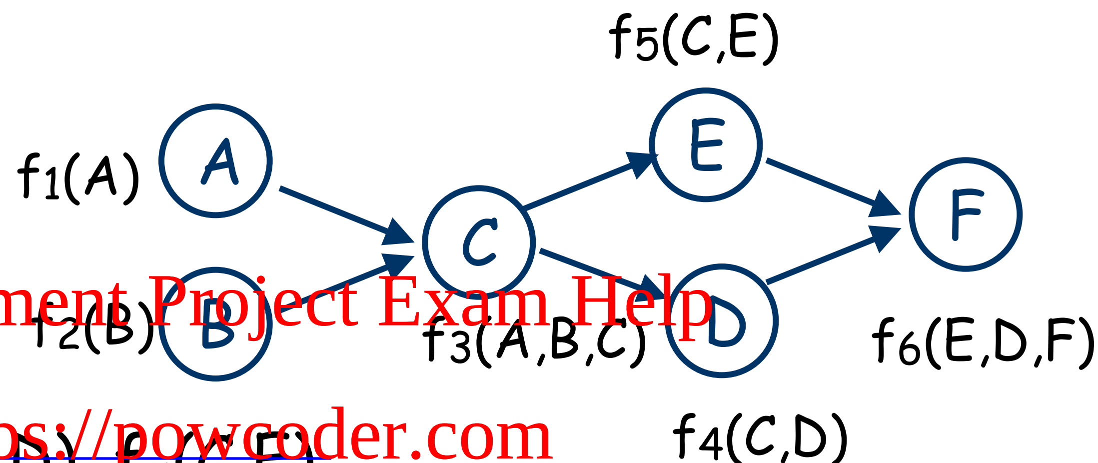
1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E: $f_8(E,D)$
6. D:

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

 Fahiem Bacchus, University of Toronto

VE: Buckets—Place Original Factors in first bucket that contains one of its variables

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



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

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

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

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

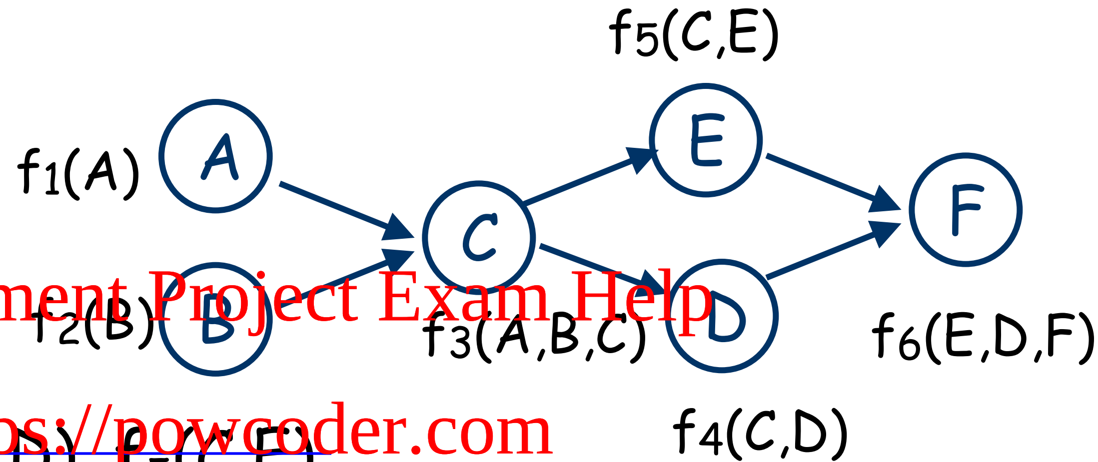
5. E: $f_8(E,D)$

6. D:

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

VE: Buckets—Place Original Factors in first bucket that contains one of its variables

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

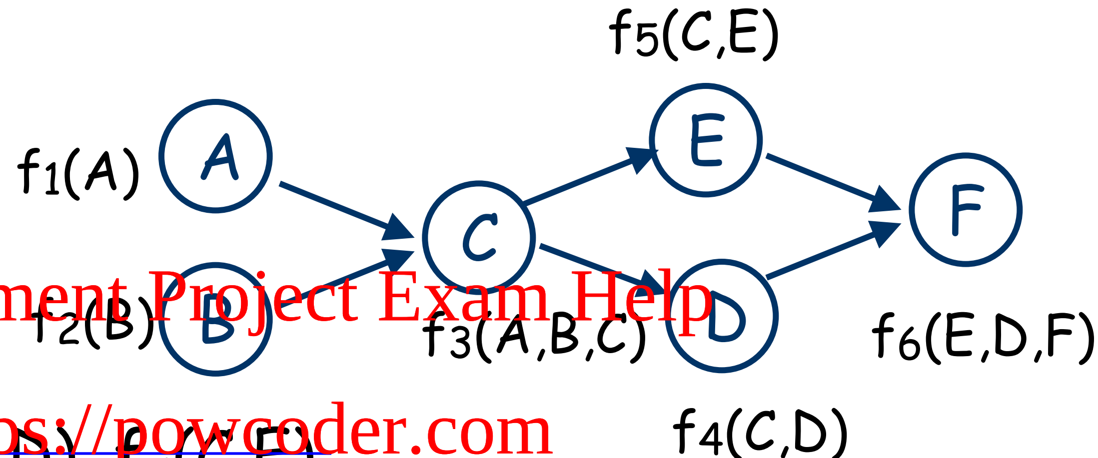


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. ~~A: $f_1(A)$, $f_7(A,B,D,E)$~~
4. ~~B: $f_2(B)$, $f_9(B,D,E)$~~
5. E: $f_8(E,D)$, $f_{10}(D,E)$
6. D:

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

VE: Buckets—Place Original Factors in first bucket that contains one of its variables

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



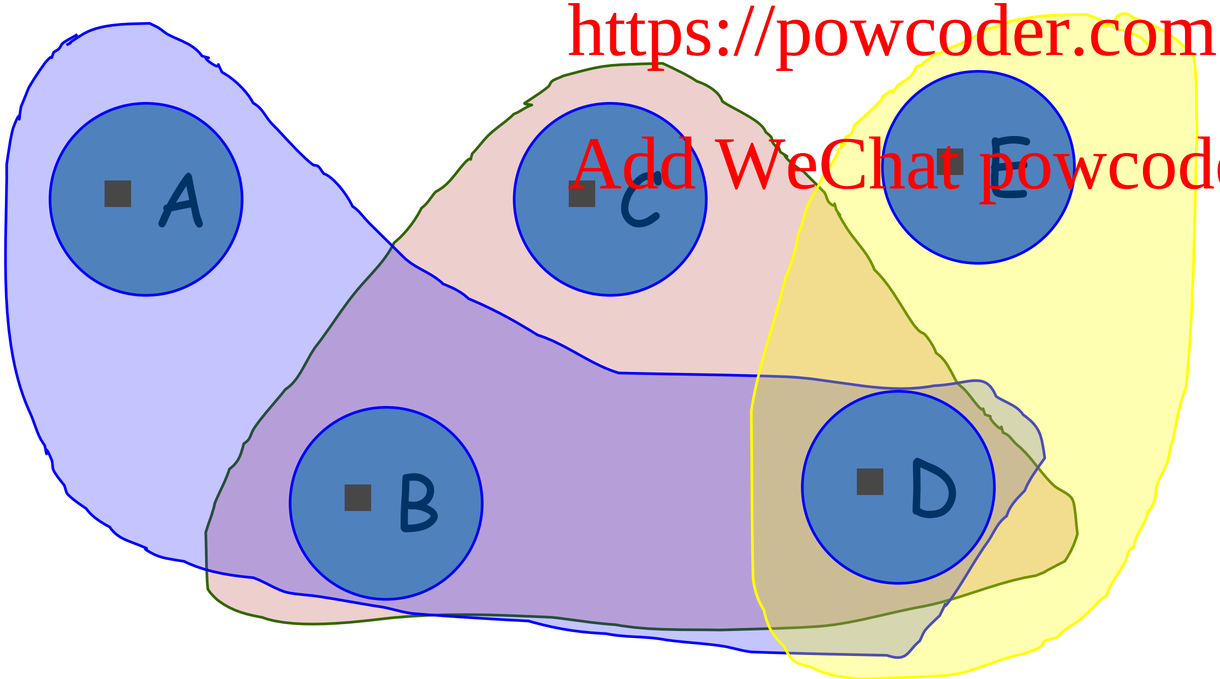
1. $\cancel{C: f_3(A,B,C), f_4(C,D), f_5(C,E)}$
2. $\cancel{F: f_6(E,D,F)}$
3. $\cancel{A: f_1(A), f_7(A,B,D,E)}$
4. $\cancel{B: f_2(B), f_9(B,D,E)}$
5. $\cancel{E: f_8(E,D), f_{10}(D,E)}$
6. D: $f_{11}(D)$

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

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

Complexity of Variable Elimination

- Hypergraph of Bayes Net.
  - Hypergraph has vertices just like an ordinary graph, but instead of edges between two vertices X↔Y it contains **hyperedges**.
    - A hyperedge is a set of vertices (i.e., potentially more than one)



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

Fahiem Bacchus, University of Toronto

Complexity of Variable Elimination

- Hypergraph of Bayes Net.
  - The set of vertices are precisely the nodes of the Bayes net.
  - The hyperedges are the variables appearing in each CPT.
    - $\{X_i\} \cup Par(X_i)$

Complexity of Variable Elimination

- Pr(A,B,C,D,E,F) =

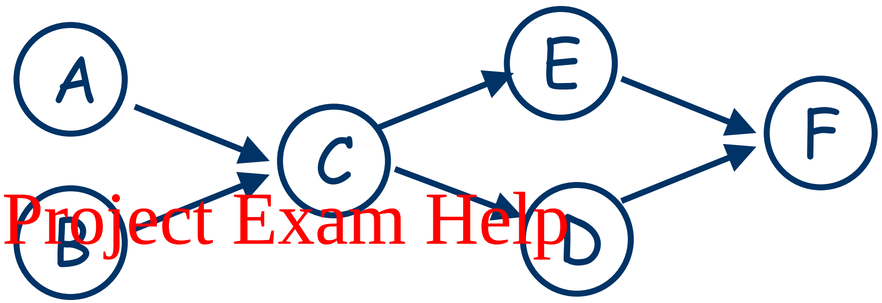
  Pr(A)Pr(B)

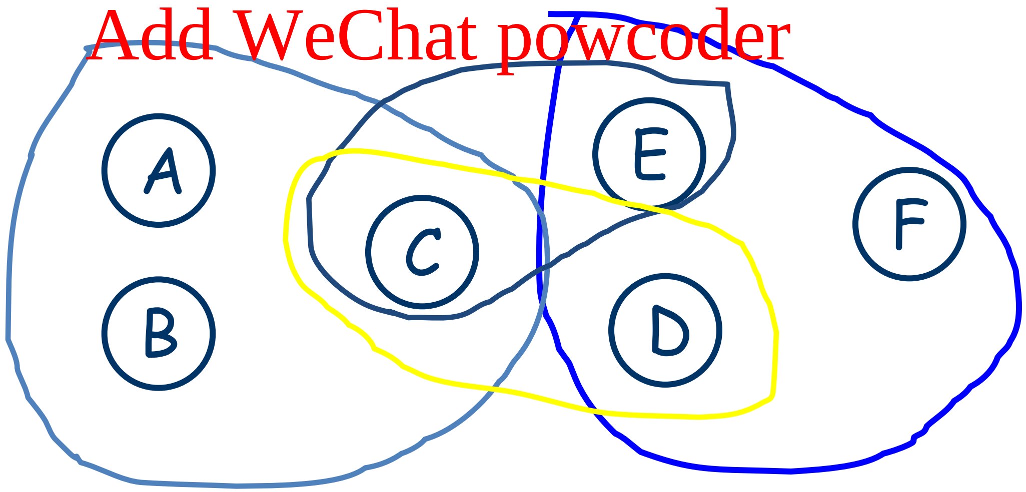
  X Pr(C|A,B)

  X Pr(E|C)

  X Pr(D|C)

  X Pr(F|E,D).

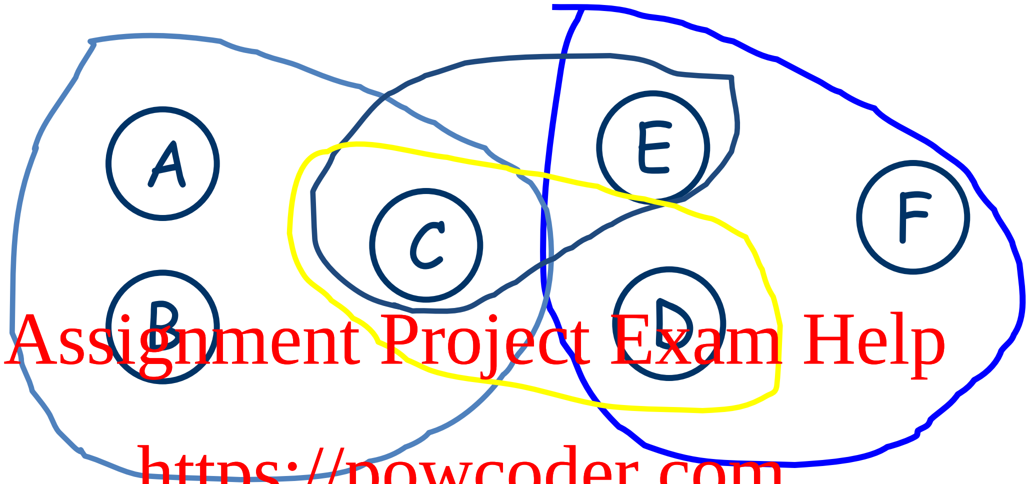




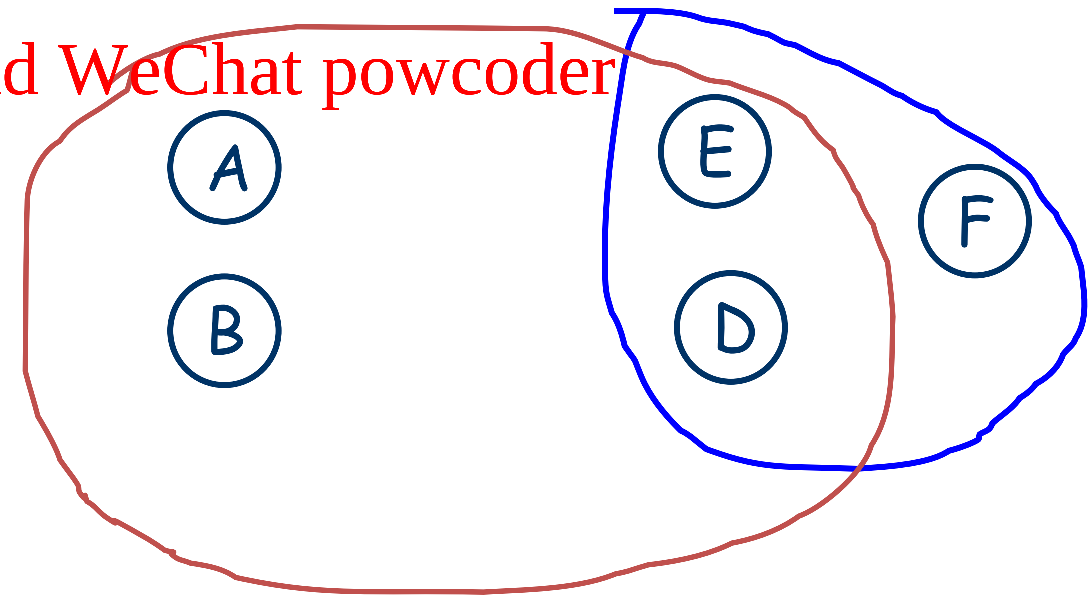
Variable Elimination in the HyperGraph

- To eliminate variable X_i in the hypergraph we
  - 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.

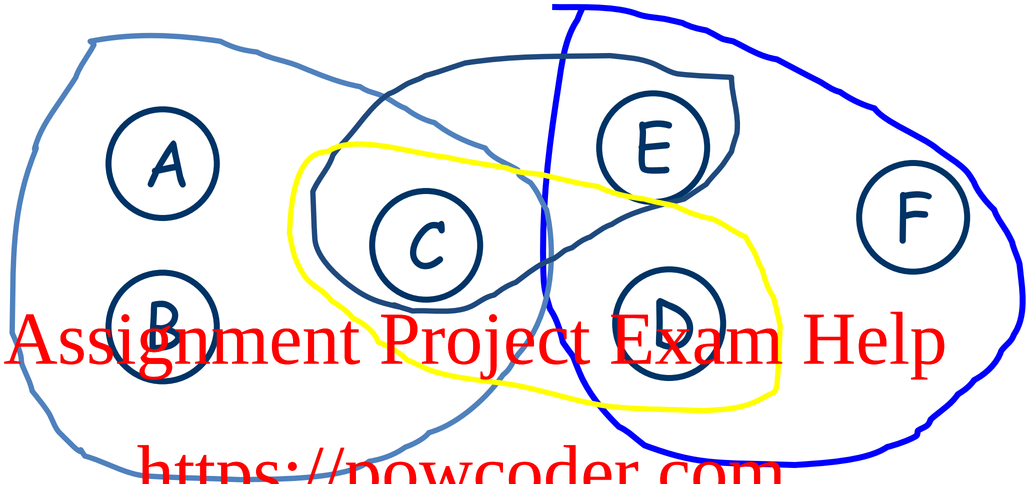
Complexity of Variable Elimination



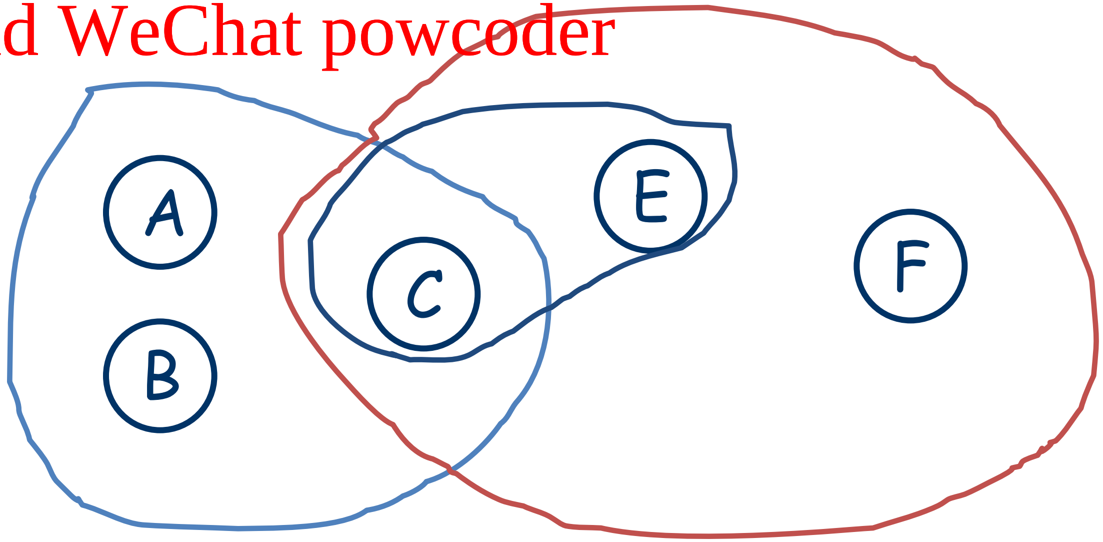
- Eliminate C



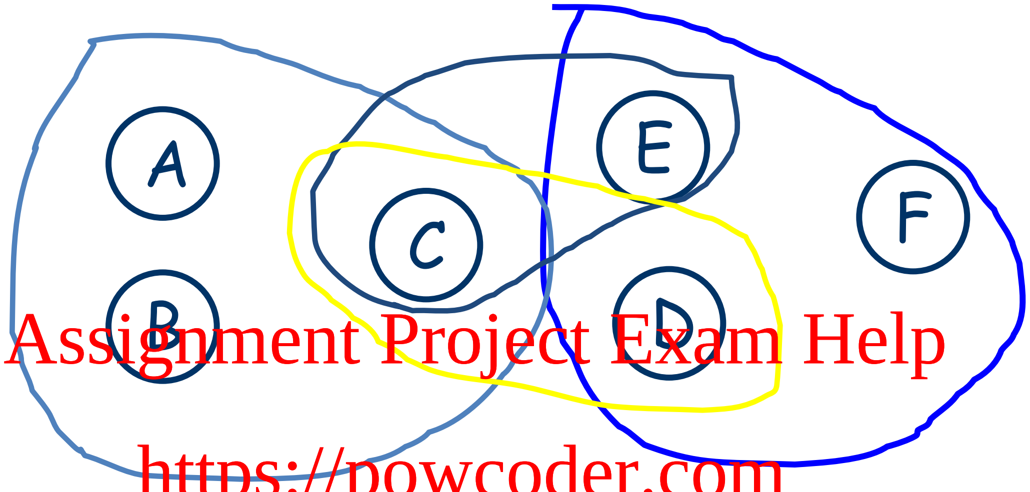
Complexity of Variable Elimination



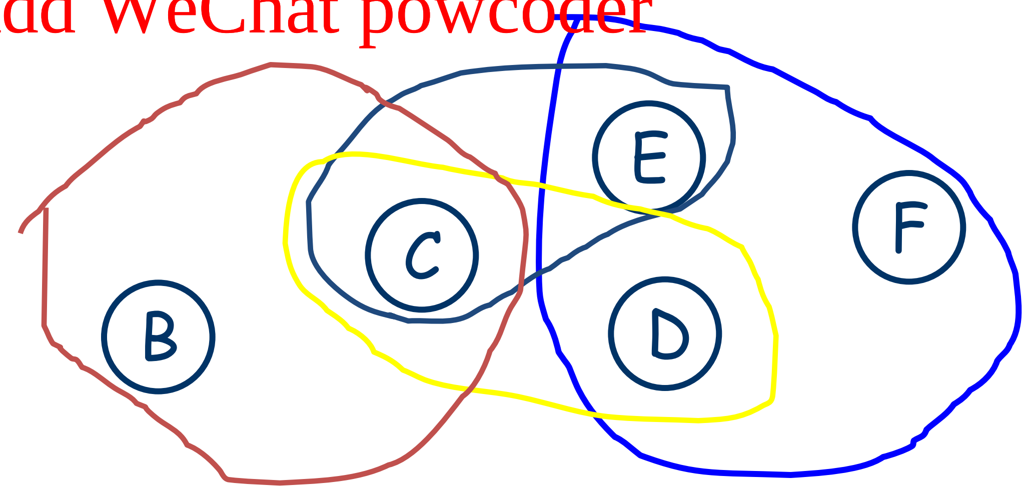
- Eliminate D



Complexity of Variable Elimination



- Eliminate A



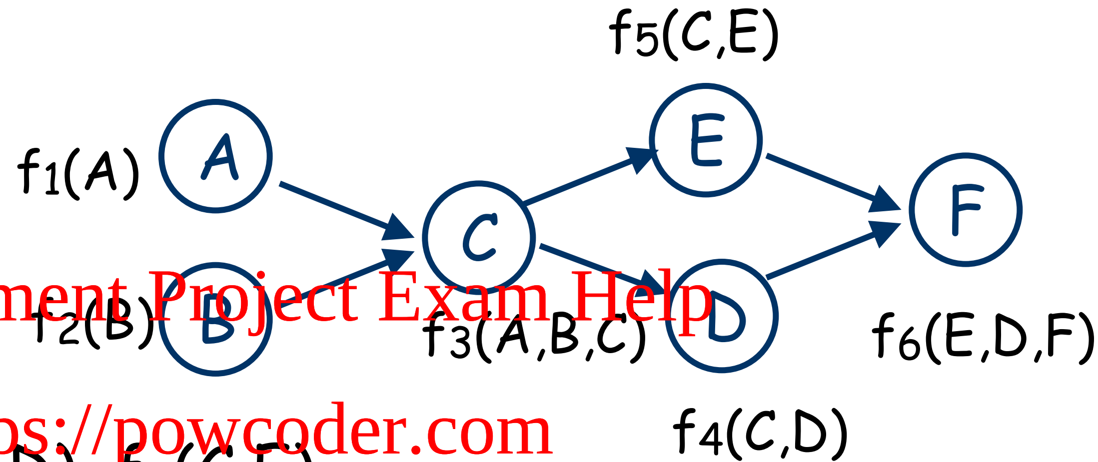
Variable Elimination

- Notice that when we start VE we have a set of factors consisting of the reduced CPTs. The unassigned variables for the vertices and the set of variables each factor depends on forms the hyperedges of a hypergraph H_1.

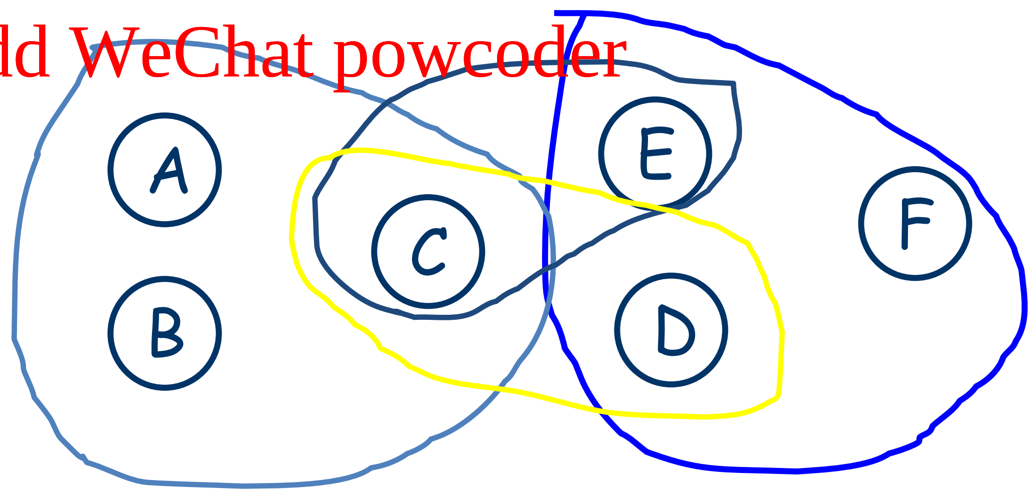
- If the first variable we eliminate is X, then we remove all factors containing X (all hyperedges) and add a new factor that has as variables the union of the variables in the factors containing X (we add a hyperdege that is the union of the removed hyperedges minus X).

VE: Place Original Factors in first applicable bucket.

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

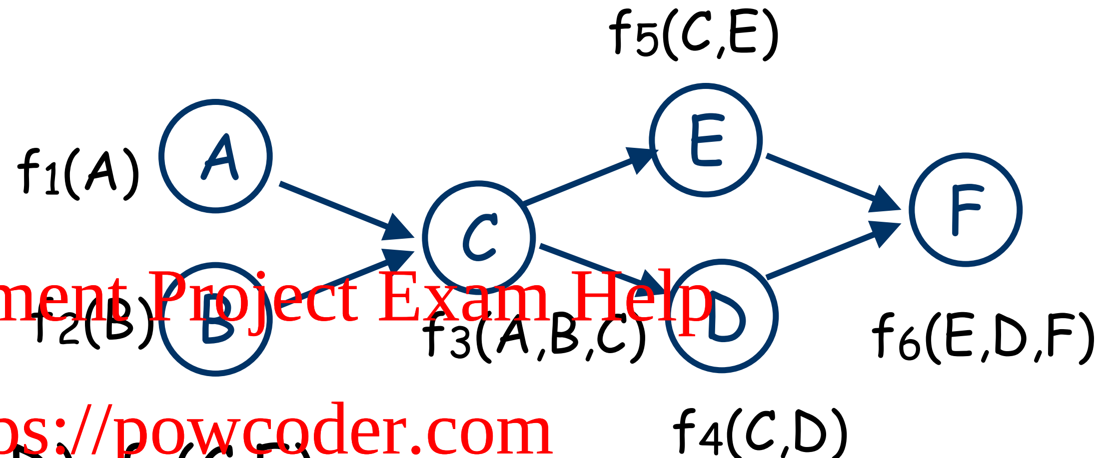


1. C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$
4. B: $f_2(B)$
5. E:
6. D:

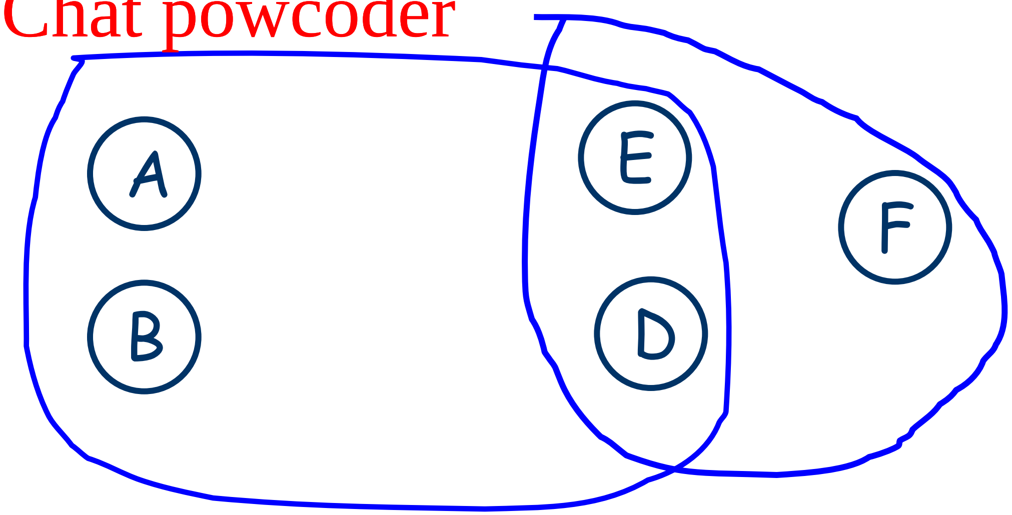


VE: Eliminate C, placing new factor f_7 in first applicable bucket.

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

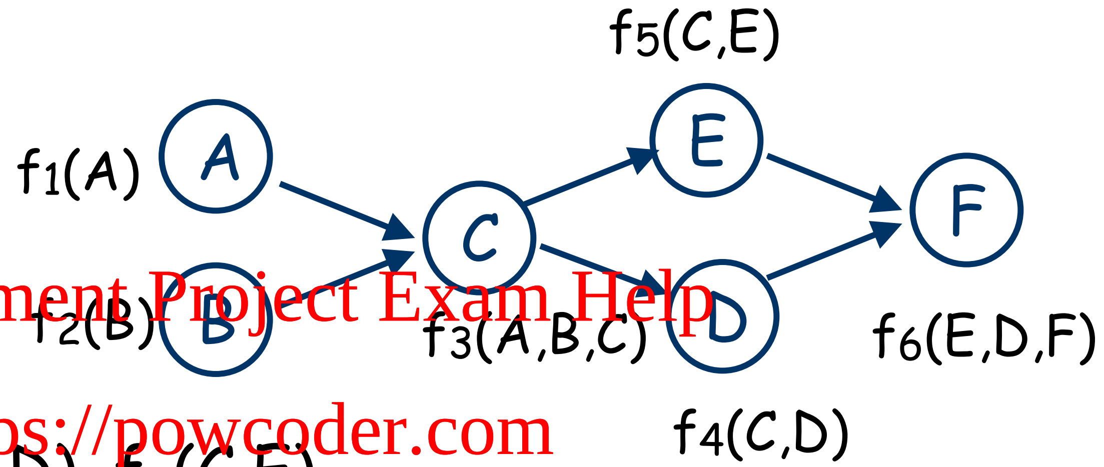


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. F: $f_6(E,D,F)$
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E:
6. D:

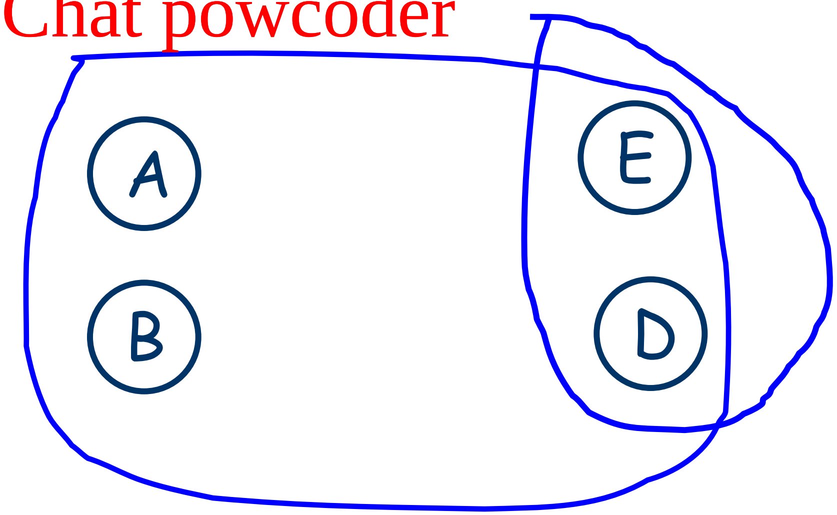


VE: Eliminate F, placing new factor f_8 in first applicable bucket.

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

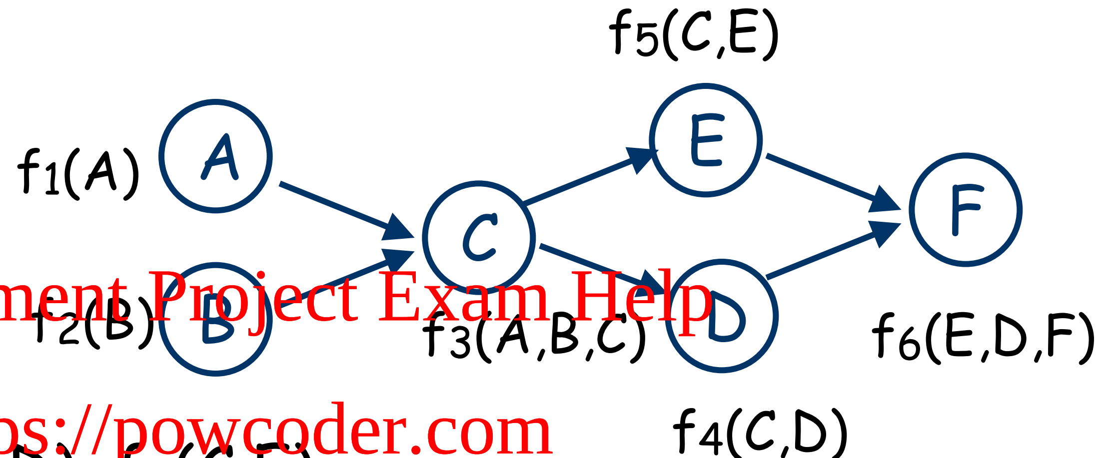


1. ~~C: $f_3(A,B,C)$, $f_4(C,D)$, $f_5(C,E)$~~
2. ~~F: $f_6(E,D,F)$~~
3. A: $f_1(A)$, $f_7(A,B,D,E)$
4. B: $f_2(B)$
5. E: $f_8(E,D)$
6. D:

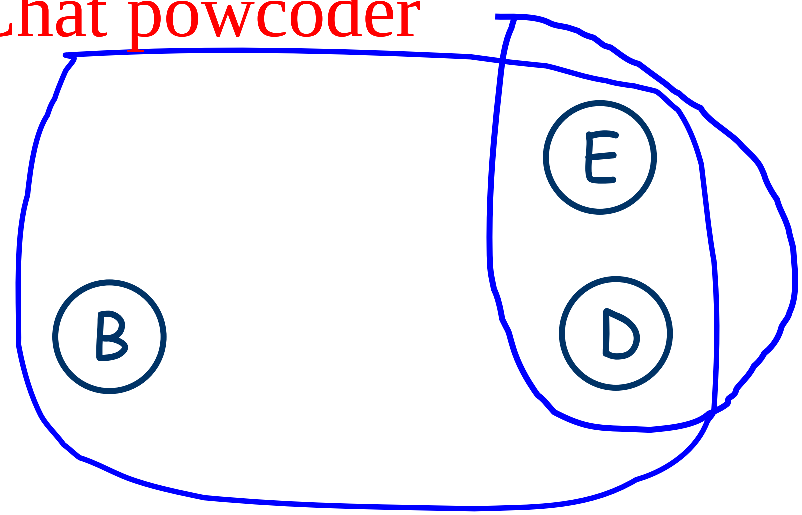


VE: Eliminate A, placing new factor f_9 in first applicable bucket.

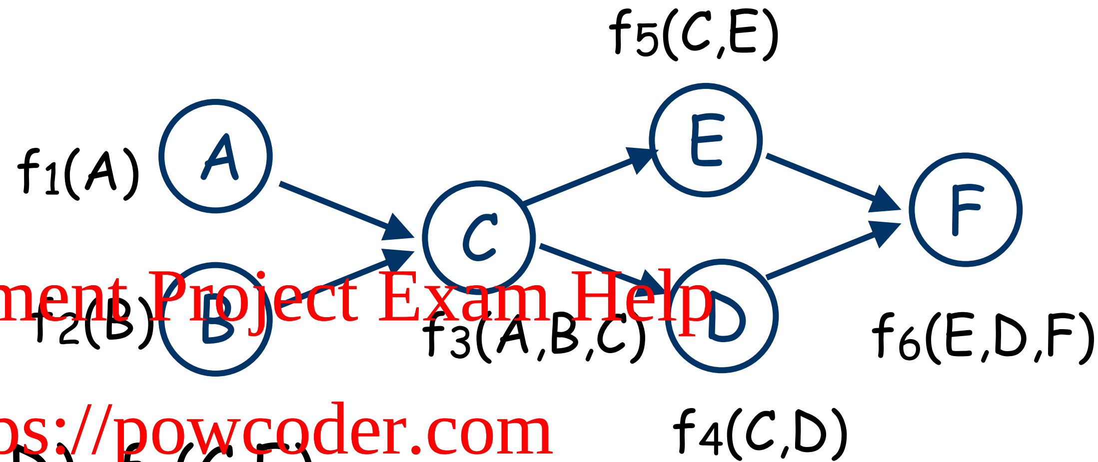
Ordering:
C,F,A,B,E,D



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:

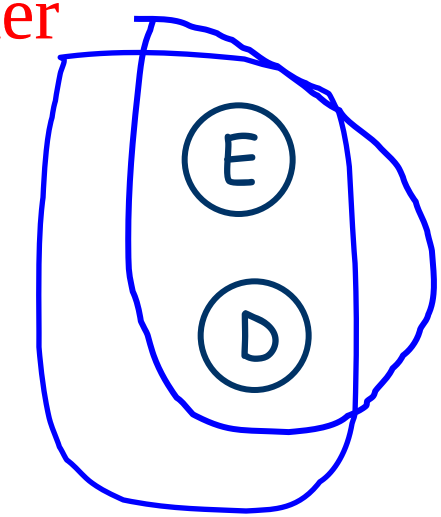


VE: Eliminate B, placing new factor f_{10} in first applicable bucket.



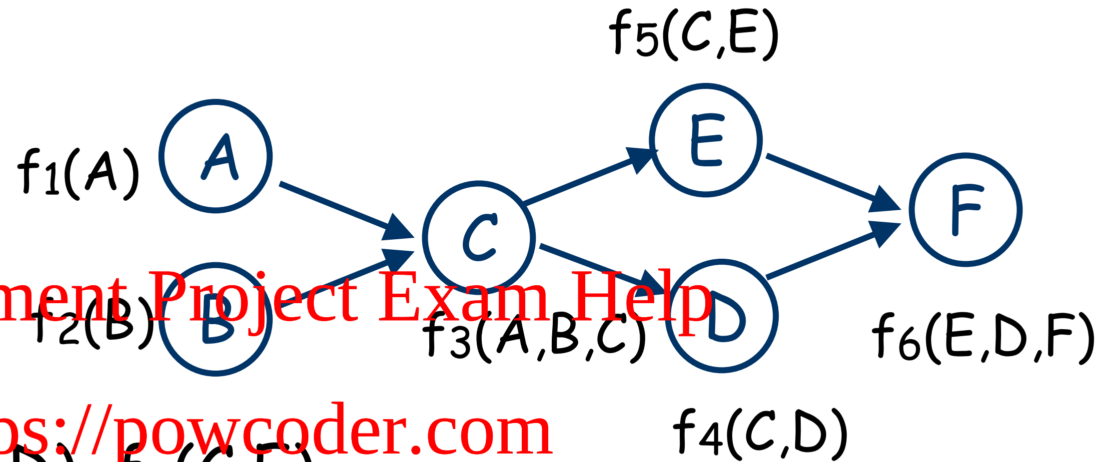
Ordering:
C,F,A,B,E,D

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

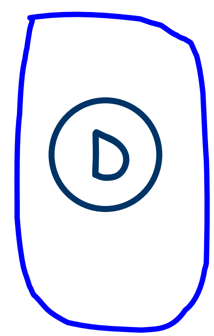


VE: Eliminate E, placing new factor f_{11} in first applicable bucket.

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: $f_{11}(D)$



Elimination Width

- Given an ordering π 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, …, H_n

- Where H_n contains only one vertex (the query variable).

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

Elimination Width

- If the elimination width of an ordering π is k, then the complexity of VE using that ordering is $2^{O(k)}$
- Elimination width k means that at some stage in the elimination process a factor involving k variables was generated.
- That factor will require $2^{O(k)}$ space to store
  - VE will require $2^{O(k)}$ space using this ordering
- And it will require $2^{O(k)}$ operations to process (either to compute in the first place, or when it is being processed to eliminate one of its variables).
  - VE will require $2^{O(k)}$ time using this ordering.
- NOTE, that k is the elimination width of this particular ordering.

Elimination Width

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

- In the worst case the elimination width can be equal to the number of variables—exponential complexity.

- Note that there are many measures similar to elimination width—tree width is another common measure.

Complexity of Variable Elimination

- Under the best ordering VE will generate factors of size $2^{O(\omega)}$ where ω is the **elimination width** of the initial Bayes Net, and it will require this much space and time

Complexity of Variable Elimination

- Note that VE input can already be larger than the number of variables.

- VE's inputs are the conditional probability tables $P(X|Par(X))$. If the largest CPT has k variables then VE's input will be $2^{O(k)}$
  - The table will have size equal to the product of the domain sizes of X and its parents.

- ω is always bigger k (the input factors are part of the first hypergraph.

- In some cases, however the elimination width is equal to k. In these cases VE operates in time linear in the size of its input

Elimination Width

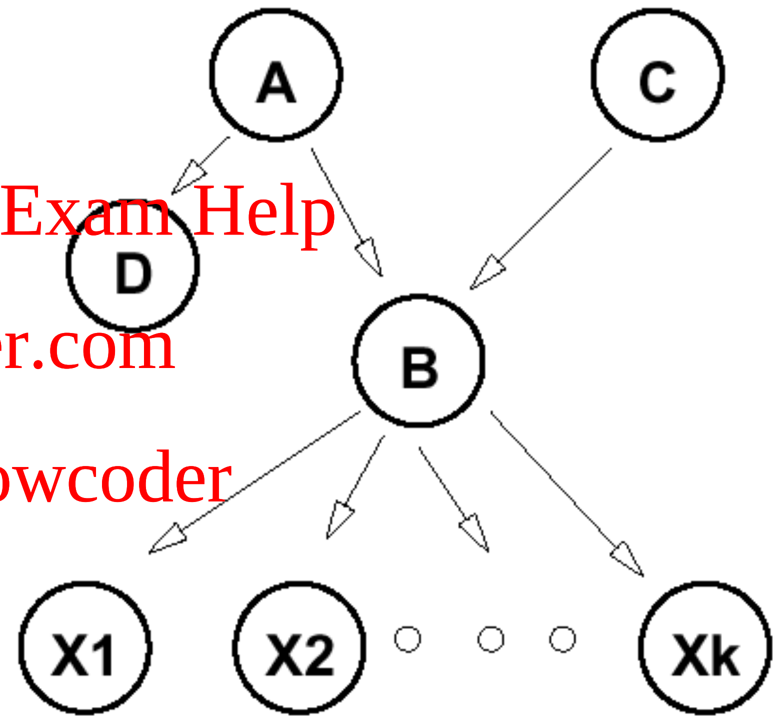
- Exponential in the tree width is the best that VE can do.
  - Finding an ordering that has minimum elimination width is NP-Hard.
    - so in practice there is no point in trying to speed up VE by finding the best possible elimination ordering.
  - Heuristics are used to find orderings with good (low) 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 elimination width is equal to the number of variables.

Finding Good Orderings

- A *polytrees* is a singly connected Bayes Net: in particular there is only one path between any two nodes.
- A node can have multiple parents, but we have no cycles.
- 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 width = size of largest input CPT

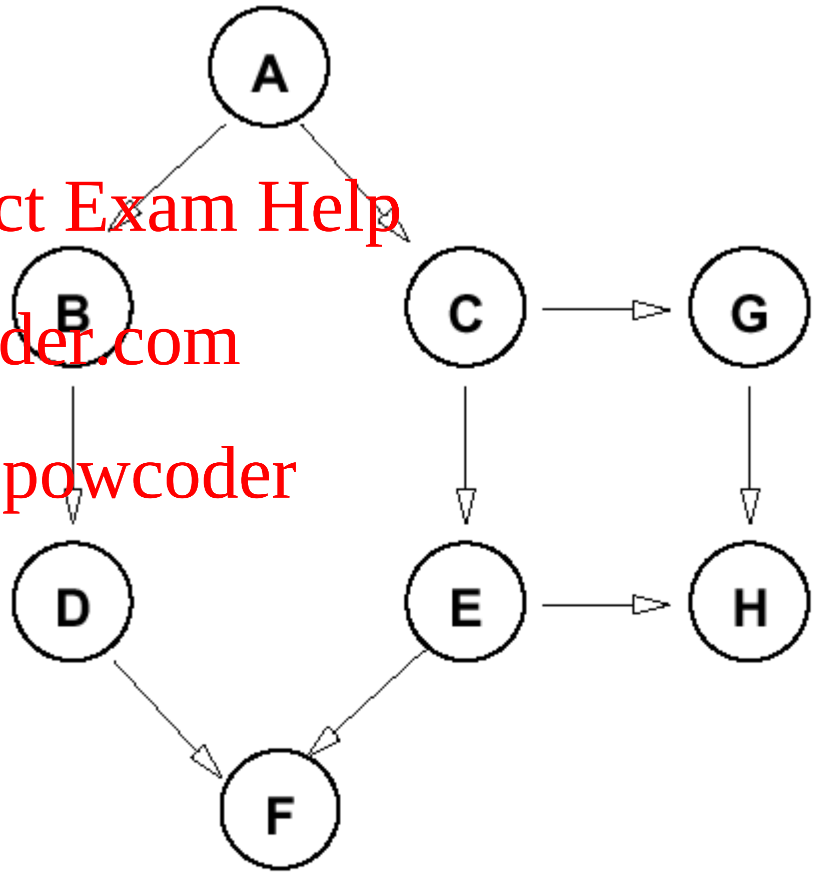
Elimination Ordering: Polytrees

- Eliminating singly connected nodes allows VE to run in time linear in size of network (not linear in the number of variables)
  - e.g., in this network, eliminate D, A, C, X1,…; or eliminate X1,… Xk, D, A C; or mix up…
  - result: no factor ever larger than original CPTs
  - eliminating B before these gives factors that include all of A,C, X1,… Xk !!!



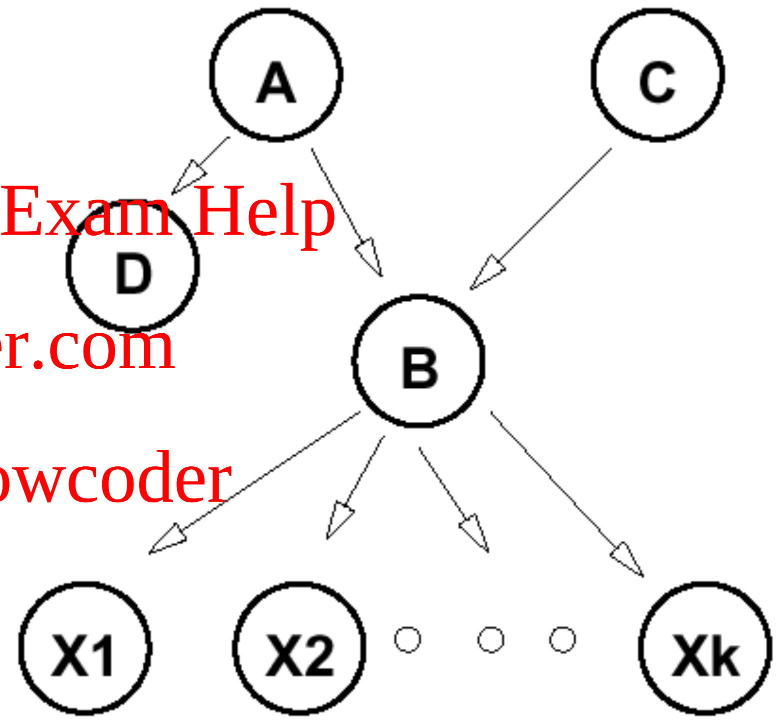
Effect of Different Orderings

- Suppose query variable is D. Consider different orderings for this network (not a polytree!)
  - A,F,H,G,B,C,E:
    - good
  - E,C,A,B,G,H,F:
    - bad



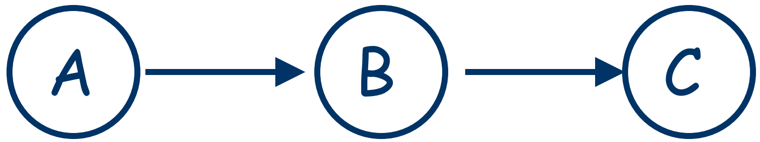
Min Fill Heuristic

- A fairly effective heuristic is 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



- The heuristic always solves polytrees in linear time.

Relevance



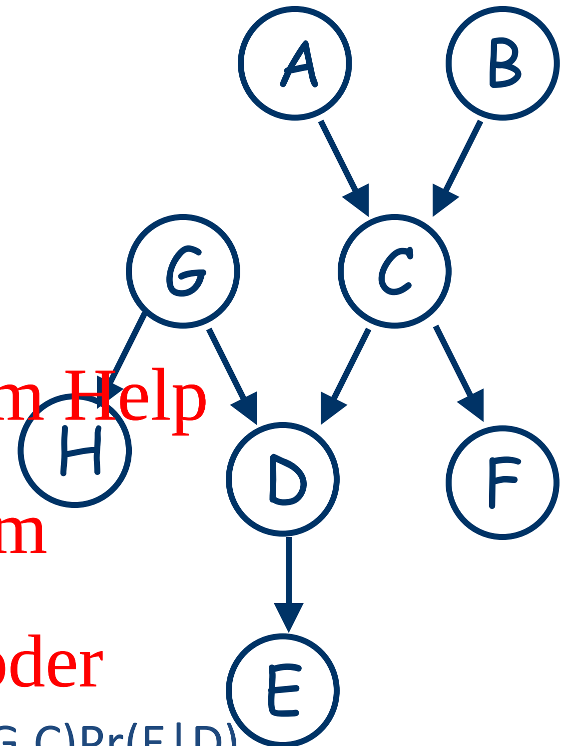
- Certain variables have no impact on the query. In network ABC, computing Pr(A) with no evidence requires elimination of B and C.
  - But when you sum out these vars, you compute a trivial factor (whose value are all ones); for example:
  - eliminating C: Σ_C Pr(C|B)
  - 1 for any value of B (e.g., Pr(c|b) + Pr(~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 *subnetwork comprising only relevant variables* when evaluating a query Q

Relevance: Examples

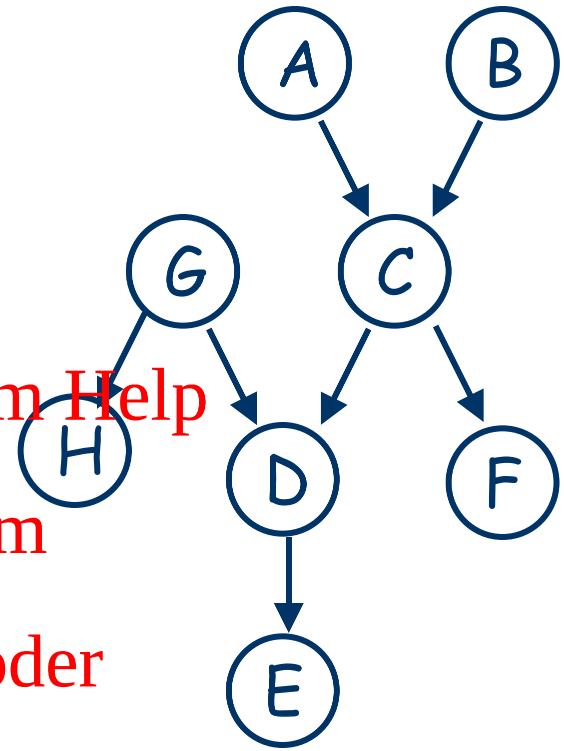
- Query: P(F)
  - relevant: F, C, B, A
- Query: P(F|E)
  - relevant: F, C, B, A
  - **also: E, hence D, G**
  - intuitively, we need to compute P(C|E) to compute P(F|E)
- Query: P(F|H)
  - relevant F,C,A,B.



Pr(A)Pr(B)Pr(C|A,B)Pr(F|C) Pr(G)Pr(h|G)Pr(D|G,C)Pr(E|D)
= … Pr(G)Pr(h|G)Pr(D|G,C) $\sum_E$Pr(E|D) = a table of 1's
= … Pr(G)Pr(h|G) $\sum_D$ Pr(D|G,C) = a table of 1's
= [Pr(A)Pr(B)Pr(C|A,B)Pr(F|C)] [Pr(G)Pr(h|G)]

[Pr(G)Pr(h|G)] $\neq$ 1 but irrelevant once we normalize, as it multiplies each value of F by the same number.

Relevance: Examples



- Query: P(F|E,C)
  - algorithm says all vars except H are relevant; but really none except C, F (since C cuts of all influence of others)
  - algorithm is overestimating relevant set

Independence in a Bayes Net

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

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

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

More generally

- Many conditional independencies hold in a given BN.
- These independencies are useful in computation, explanation, etc.
- Some of these independencies can be detected using a graphical condition called **D-Separation**.

More generally…

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

Simple graphical property: 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 cond. independent given E.

Blocking in D-Separation

- Let *P* be an **undirected path** from X to Y in a BN. Let **E** *(evidence)* be a set of variables.

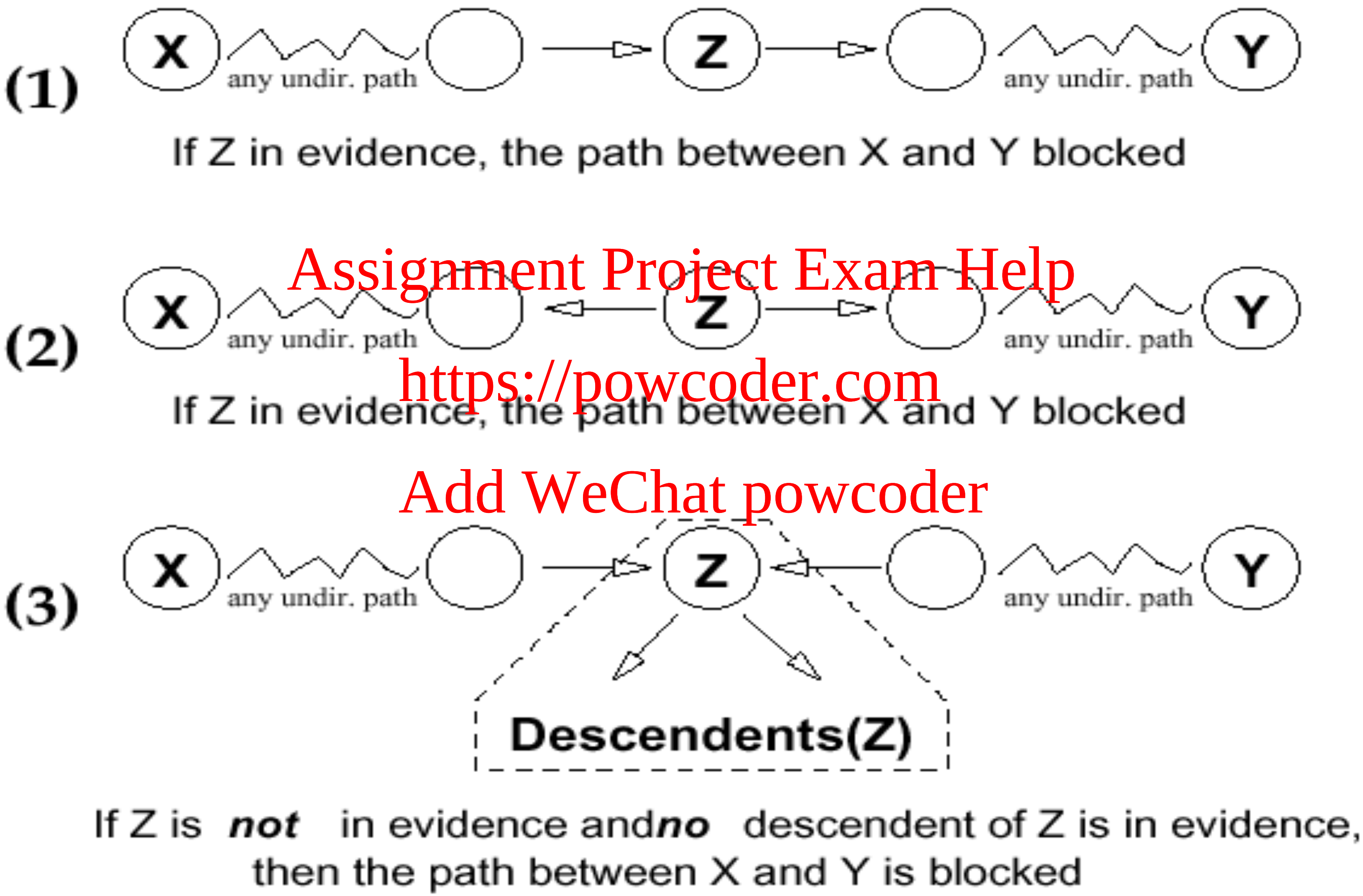
We say **E** *blocks path P*
iff **there is some** node Z on the path P such that:

- **Case 1:** Z∈**E** and one arc on P *enters (goes into)* Z and one *leaves (goes out of)* Z; or

- **Case 2:** Z∈**E** and both arcs on P leave Z; or

- **Case 3:** both arcs on P enter Z and *neither Z, nor any of its descendents*, are in **E**.

Blocking: Graphical View

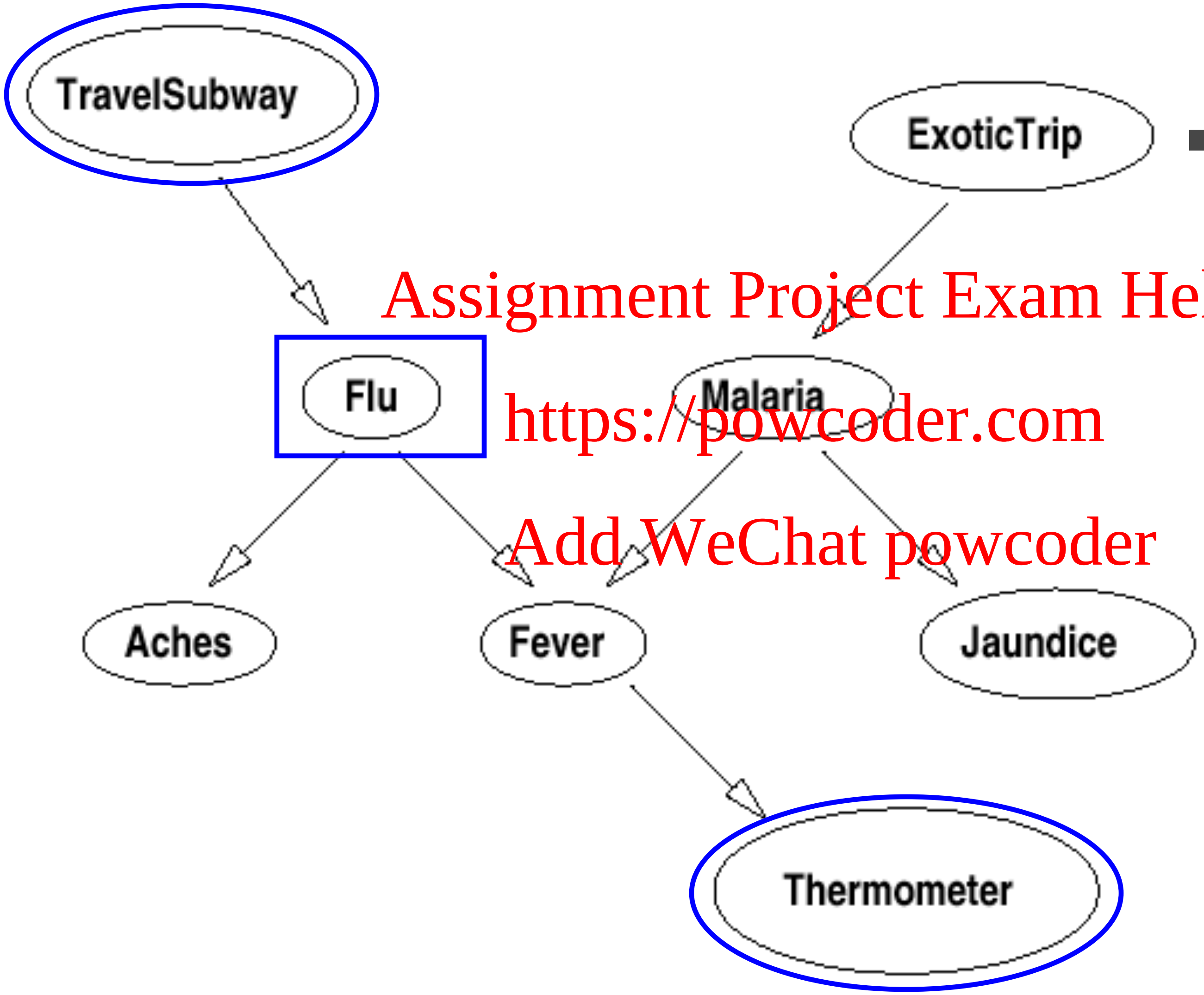


Recall: D-Separation

D-separation:

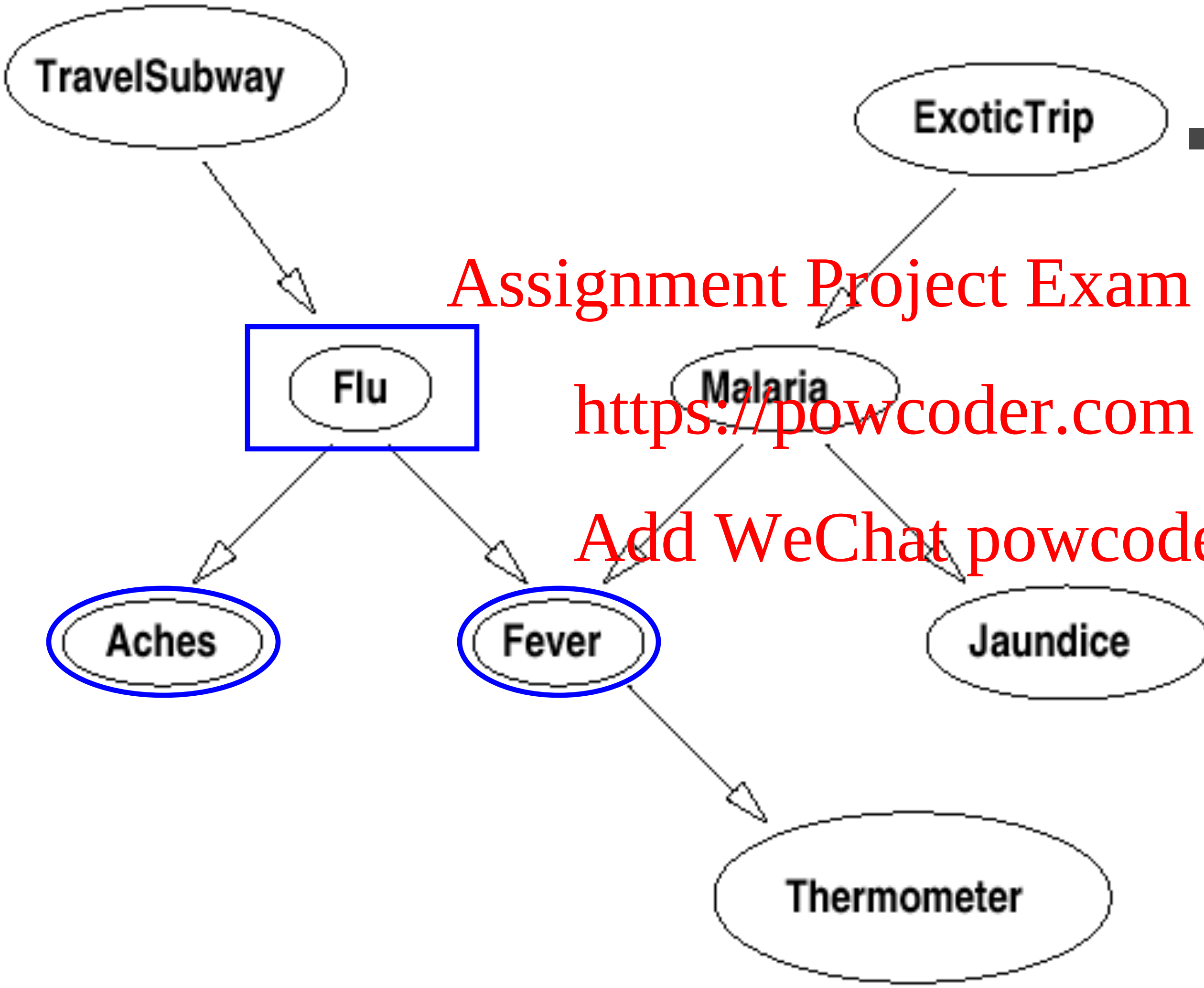
A set of variables **E** *d-separates* X and Y if it *blocks every undirected path* in the BN between X and Y.

D-Separation: Intuitions



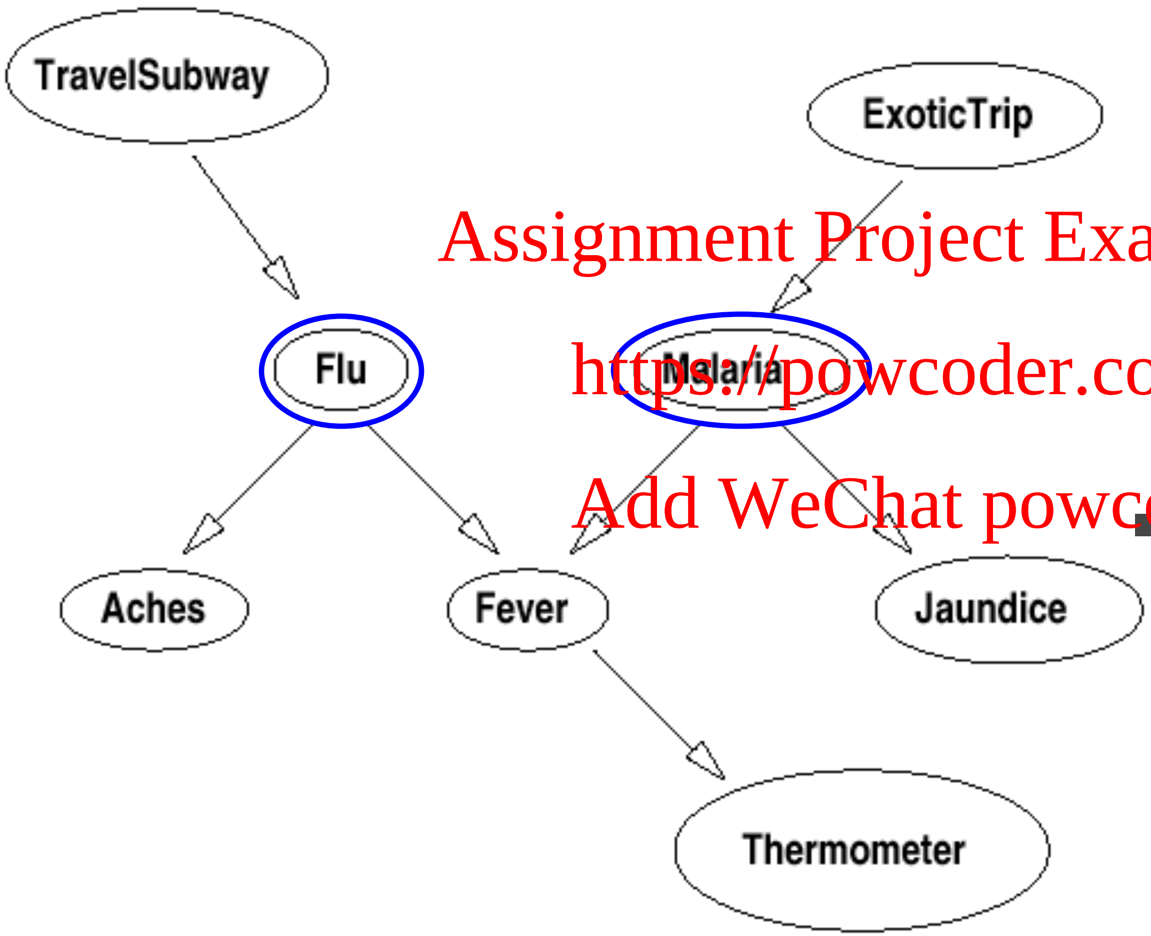
- Subway and Thermometer are **dependent**; but are **independent given Flu** (since Flu blocks the only path)

D-Separation: Intuitions



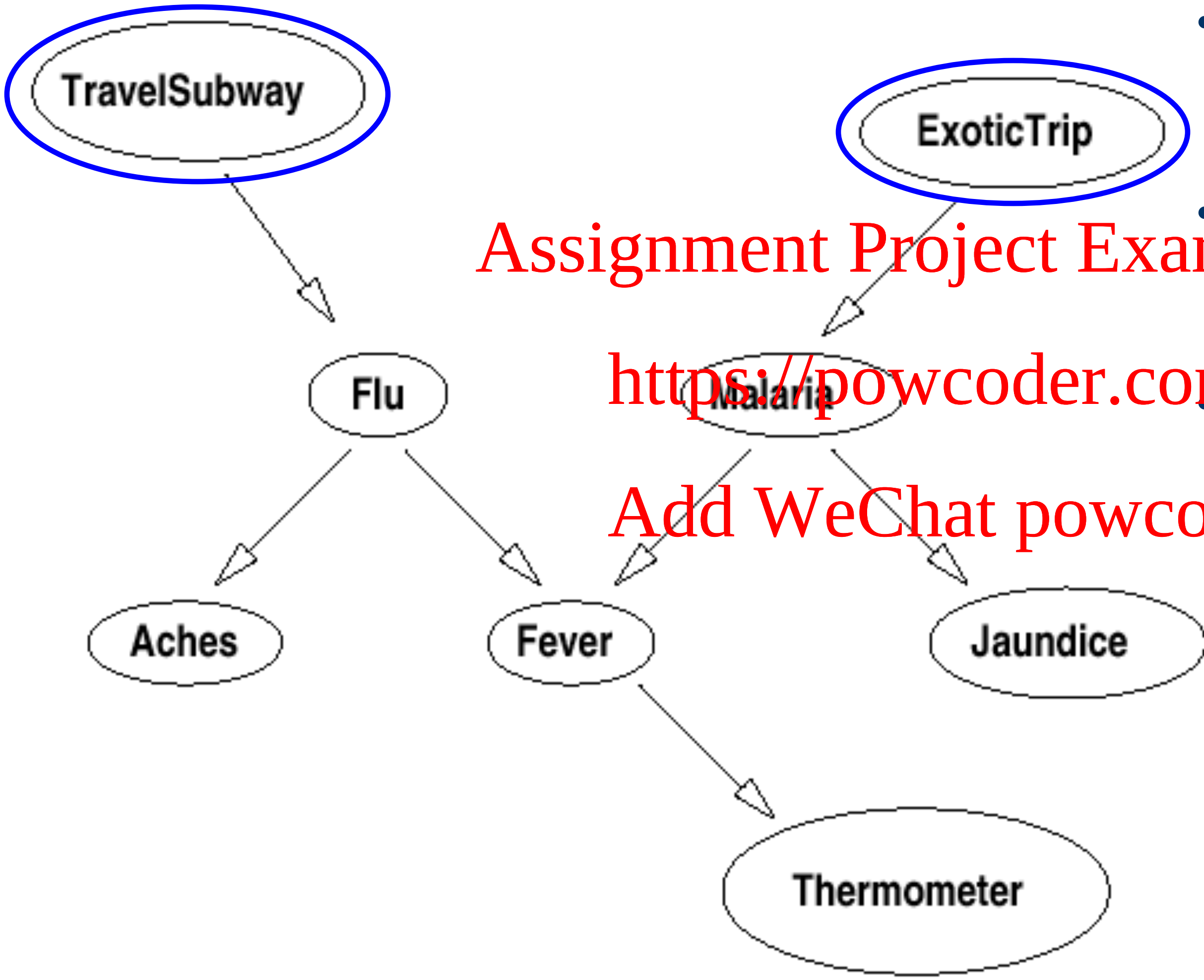
- Aches and Fever are **dependent**; but are **independent given Flu** (since Flu blocks the only path). Similarly for Aches and Therm (dependent, but indep. given Flu).

D-Separation: Intuitions



- Flu and Mal are **independent (given no evidence):** Fever blocks the path, since it is *not in evidence*, nor is its decsendant Therm.
- Flu and Mal are **dependent** given Fever (or given Therm): nothing blocks path now. **What's the intuition?**

D-Separation: Intuitions

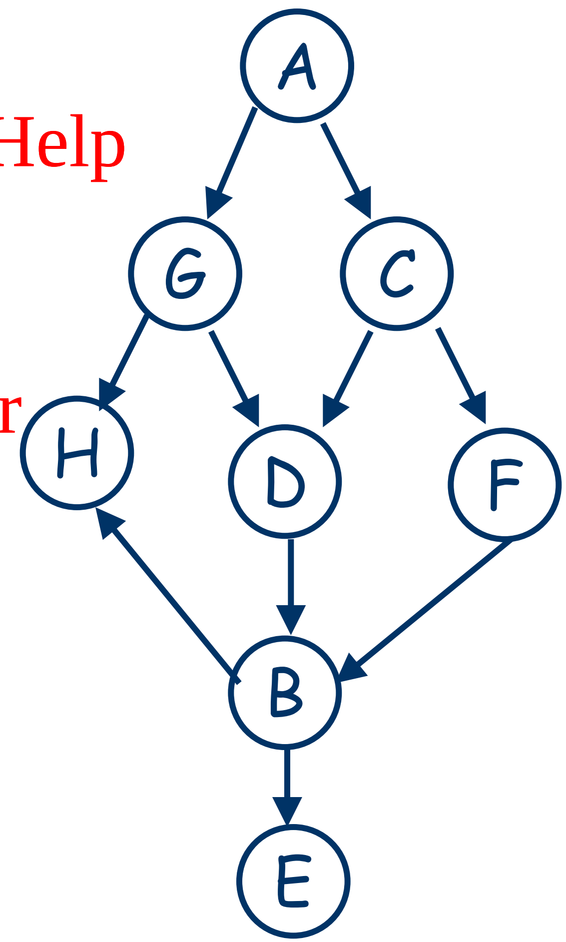


- Subway, ExoticTrip are **independent**;
- They are **dependent given Therm**;
- They are **independent given Therm and Malaria**. This for exactly the same reasons for Flu/Mal above.

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