Exact Inference on Graphical Model

Reference:

Probabilistic Graphical Model Ch.9 , Ch. 10 (Koller & Friedman)
CMU, 10-708, Fall 2009 Probabilistic Graphical Models Lectures 8,9,10 (Eric Xing)

Probabilistic Inference

- A Graphical Model specifies a joint distribution $P_M(X)$ over a collection of variables X.

- How can we answer queries/questions about P(X) ?

  That is, how can we inference using P(X) ?

- Type of queries:

  - 1. Likelihood of evidence/assignments on variables
  - 2. Conditional Probability of some variables (given others).
  - 3. Most Probable Assignment for some variables (given others).

Query 1: Likelihood

- Given Evidence E = $\{X_1=x_1,\ldots, X_D=x_D\}$ specifying some variables' value and let $Z=\{Z_1, \ldots, Z_k\}$ be variables unspecified , the likelihood of a model M yielding this evidence can be computed by:

$$likelihood \ \ of \ \ E$$
$$= P_M(X) = \sum_{Z_1} \ldots \sum_{Z_K} P_M(Z_1,\ldots,Z_K, x_1,\ldots,x_D)$$

Naïve algorithm yield O($|Z|^K$) complexity…

Query 1: Likelihood

$$likelihood\ \ of\ \ E = \sum_{Z_1} \ldots \sum_{Z_K} P_M(Z_1,\ldots,Z_K, x_1,\ldots,x_D)$$

- This measure is often used as criteria for Model Selection.

Ex. In speech recognition,
Z: words (unspecified) , X: wave sample (specified evidence E)

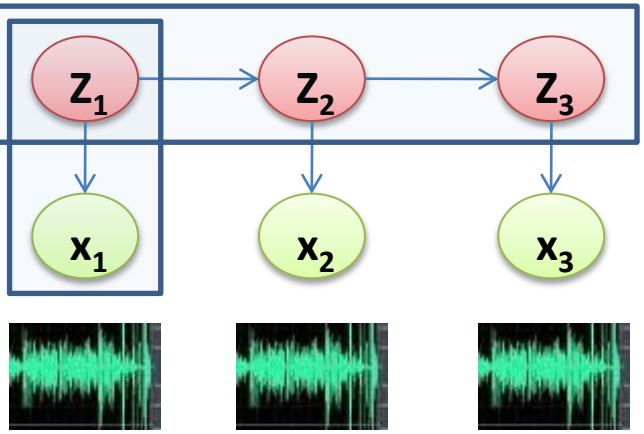
How likely a person M=(Language, Pronunciation) produce this wave sample can be Computed by:

$$likelihood\ \ of\ \ M\ \text{produce}\ \text{E}$$
$$= P_M(X)$$
$$= \sum_{Z_1}\sum_{Z_2}\sum_{Z_3} P_M(Z_1, Z_2, Z_3, x_1, x_2, x_3)$$

Summation over all possible words Z producing X

A person's "Language Model" $P(Z_{t+1}|Z_t)$

A person's "Pronunciation Model" $P(X_t|Z_t)$



Query 1: Likelihood

$$likelihood \;\; of \;\; E$$

$$= P_M(X) = \sum_{Z_1} \ldots \sum_{Z_K} P_M(Z_1, \ldots, Z_K, x_1, \ldots, x_D)$$

Taking special case E = empty , it can also be used to compute **Normalizing Const. = Z** in MRF as following :

$$\left(\; let \; \tilde{P}(Z_1 \ldots Z_K) = \prod_{clique C \, in \, M} \phi(C) \;\; be \, unnormalized \; dist. \; , \; \mathrm{P}(Z_1 \ldots Z_K) = \frac{1}{\mathrm{Z}} \tilde{P}(Z_1 \ldots Z_K) \; \right)$$

$$\sum_{Z_1} \ldots \sum_{Z_K} P(Z_1, \ldots, Z_K) \;\; = \;\; \frac{1}{Z} \sum_{Z_1} \ldots \sum_{Z_K} \tilde{P}(Z_1, \ldots, Z_K) \; = 1$$

$$==> \quad Z = \sum_{Z_1} \ldots \sum_{Z_K} \tilde{P}(Z_1, \ldots, Z_K)$$

Query 2: Conditional (marginal) Probability

- Given Evidence E = $\{X_1=x_1,\ldots, X_D=x_D\}$ and some other variables $Z=\{Z_1, \ldots, Z_k\}$ unspecified , Conditional Probability of Z is given by:

$$P(Z \mid X) = \frac{P(Z, X)}{P(X)}, \ \textit{where } P(X) \text{ is } \textit{given} \text{ by Query 1}$$

- Sometimes we are interested in only some variables Y in Z, where Z = { Y ,W }, then conditional (marginal) prob. of Y is

$$P(Y \mid X) = \sum_{W} P(Z \mid X) = \sum_{W_1} \ldots \sum_{W_K} P(Y, W_1 \ldots W_K \mid X)$$

Naïve summation over uninterested variables W yield O($|W|^K$) complexity…

Query 2: Conditional (marginal) Probability

Ex. In speech recognition,

Z: words (unspecified) , X: wave sample (specified evidence E)

$$P(Z \mid X) = \frac{P(Z, X)}{P(X)}, \ \textit{where } P(X) \text{ is } \textit{given} \text{ by Query 1}$$

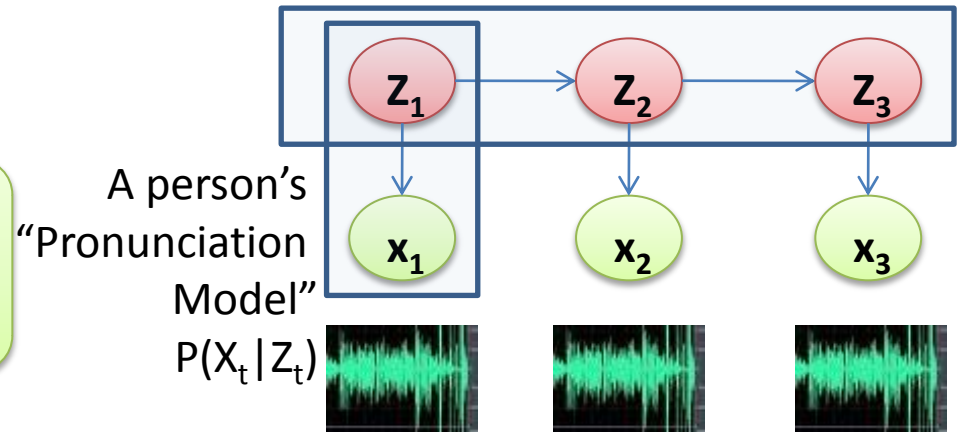
A word sequence $Z_1 \ldots Z_K$'s prob. given the wave sample $X_1 \ldots X_K$

If we only care the 1st word, then:

$$P(Z_1 \mid X) = \sum_{Z_2} \sum_{Z_3} P(Z_1, Z_2, Z_3 \mid X)$$

The 1st word Z_1's marginal distribution given the wave sample $X_1 \ldots X_K$
(naïve method is intractable for large K)

A person's "Language Model" $P(Z_{t+1} \mid Z_t)$



Query 3: Most Probable Assignment

- Given Evidence E = $\{X_1=x_1,\ldots, X_D=x_D\}$ and some other variables $Z=\{Z_1, \ldots, Z_k\}$ unspecified , Most Probable Assignment of Z is given by:

$$MPA(Z \mid X) = \arg\max_Z \; P(Z \mid X)$$

$$= \arg\max_Z \; \frac{P(X \mid Z)P(Z)}{\mathrm{P(X)}} = \arg\max_Z \; P(X \mid Z)P(Z)$$

- MPA is also called "maximum a posteriori configuration" or "MAP inference".

Note:

1. Even if we have computed **Query 2 = P(Z|X)** , it's intractable to enumerate all possible Z to get **$\mathbf{argmax_Z}$ P(Z|X).**

2. MPA cares "**Joint Maximum**", not "**Marginal Maximum**".

$$\operatorname*{argmax}_Z \; \mathrm{P(Z \mid X)} \neq \begin{cases} \arg\max_{Z1} \; P(Z_1 \mid X) \\ \ldots\ldots \\ \arg\max_{ZK} \; P(Z_K \mid X) \end{cases}$$

Query 3: Most Probable Assignment

We often just want to **"decode words "** from the wave sample,
That is, we care **$Z^* = \text{argmax}_Z P(Z|X)$** but not **$P(Z|X)$** itself.

Marginal Maximum:

$$\begin{cases} \arg\max_{Z1} P(Z_1 \mid X) \\ \arg\max_{Z2} P(Z_2 \mid X) \\ \arg\max_{Z3} P(Z_3 \mid X) \end{cases} \Rightarrow \text{may give } Z_1 = 'I', \quad Z_2 = 'comes', \quad Z_3 = 'front'$$

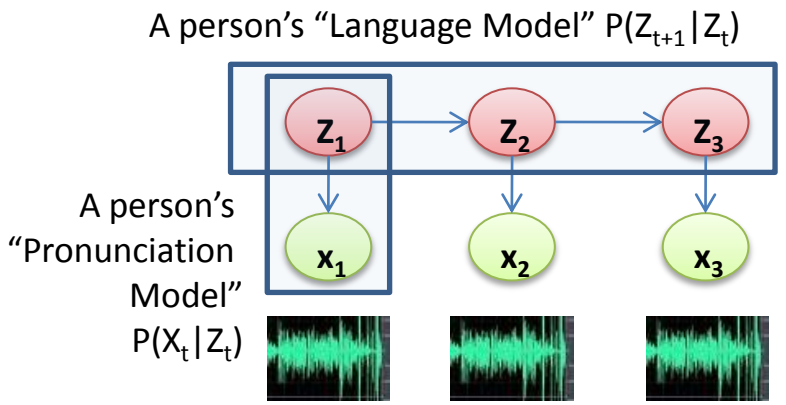
(inconsistent decoding)

Joint Maximum (MPA) :

$$\arg\max_{Z_1,Z_2,Z_3} P(Z_1, Z_2, Z_3 \mid X)$$

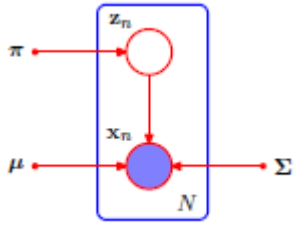
$$\Rightarrow \text{may give } 'I' \ 'come' \ 'from'$$

(consistent decoding)



In terms of difficulty, there are 3 types of inference problem.

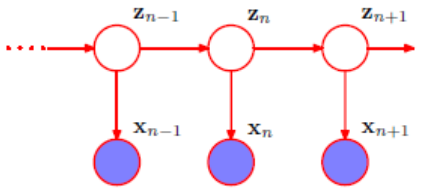
- Inference which is easily solved with Bayes rule.



Today's focus

- Inference which is tractable using some dynamic programming technique.

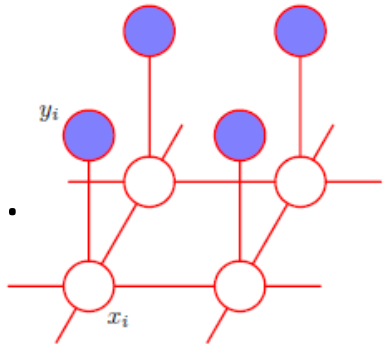
  (e.g. Variable Elimination or J-tree algorithm)



- Inference which is proved intractable

  & should be solved using some Approximate Method.

  (e.g. Approximation with Optimization or Sampling technique.)



Agenda

- Introduce the concept of "Variable Elimination" in special case of Tree-structured Factor Graph.
- Extend the idea of "VE" to general Factor graph with concept of "Clique Tree".
- See how to extend "VE" to "Most Probable Assignment" (MAP configuration) Problem.

Variable Elimination: Inference on a Chain

How to get P(E=e) ?

$$P(E=e)=\sum_{A}\sum_{B}\sum_{C}\sum_{D}P(A,B,C,D,E=e)$$

By structure of the BN:

$$P(A,B,C,D,E)=P(E\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

$$P(E)=\sum_{D}\sum_{C}\sum_{B}\sum_{A}P(E\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

We can put summation as right as possible…

Variable Elimination: Inference on a Chain

How to get P(E=e) ?

$$P(E=e)=\sum_{A}\sum_{B}\sum_{C}\sum_{D}P(A,B,C,D,E=e)$$

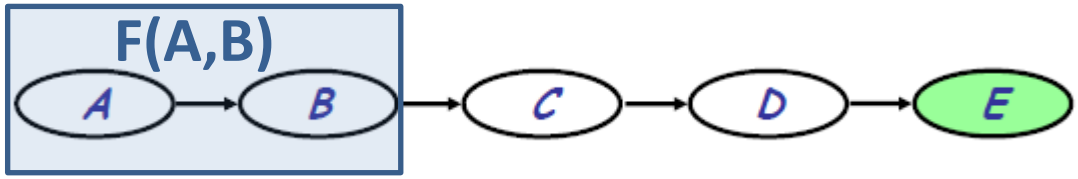
By structure of the BN:

$$P(A,B,C,D,E)=P(E\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

$$P(E)=\sum_{D}\sum_{C}\sum_{B}\sum_{A}P(E\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

$$=\sum_{D}P(E\mid D)\sum_{C}P(D\mid C)\sum_{B}P(C\mid B)\sum_{A}P(B\mid A)P(A)$$

Variable Elimination: Inference on a Chain



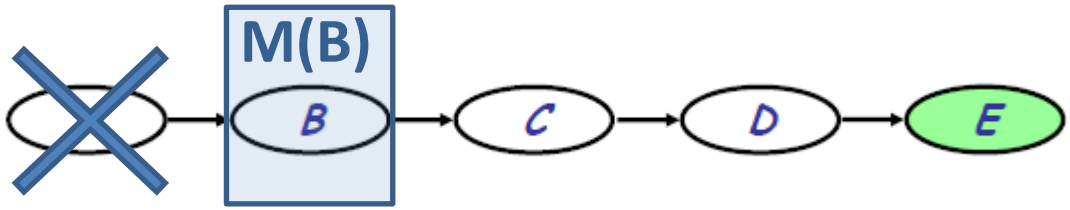
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_D P(E|D)\sum_C P(D|C)\sum_B P(C|B)\sum_A \underline{P(B|A)P(A)}$$

F(A,B)

A Table size=|A||B|

Variable Elimination: Inference on a Chain



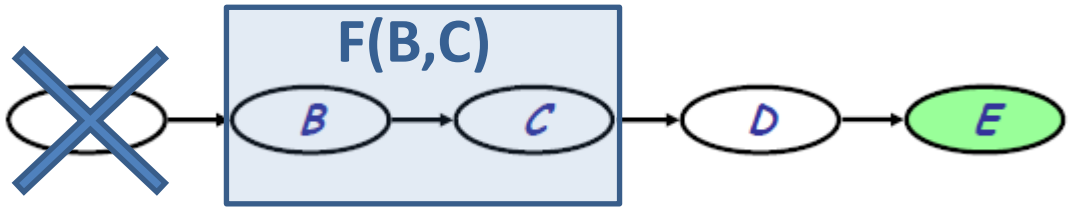
$$P(E=e)=\sum_{A}\sum_{B}\sum_{C}\sum_{D}P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_{D}P(E|D)\sum_{C}P(D|C)\sum_{B}P(C|B)\sum_{A}P(B|A)P(A)$$

$\sum_A F(A,B)=M(B)$

Eliminate “A”. A Table size=|B|.

Variable Elimination: Inference on a Chain



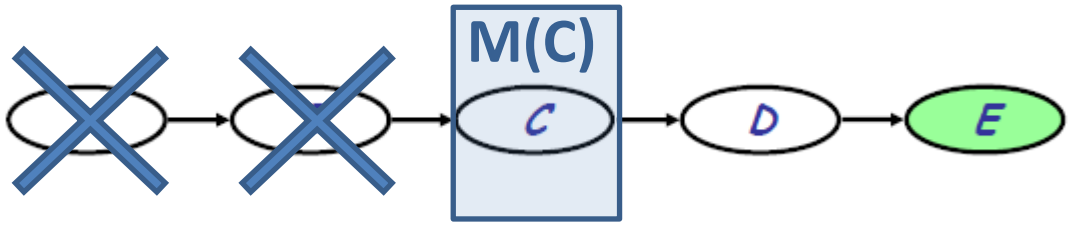
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_D P(E|D)\sum_C P(D|C)\sum_B P(C|B)\underbrace{\sum_A P(B|A)P(A)}$$

P(C|B)*M(B) = F(B,C)

A Table size=|B||C|.

Variable Elimination: Inference on a Chain



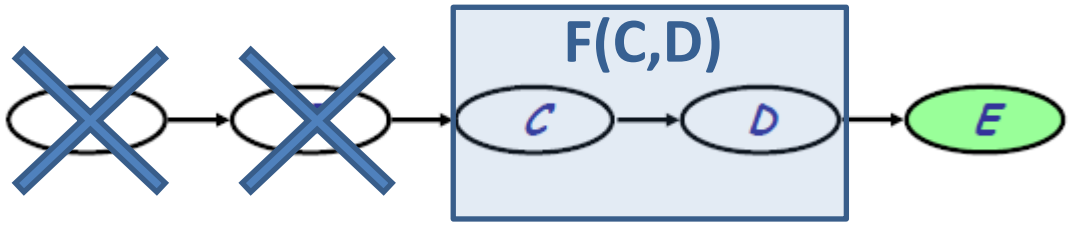
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_D P(E|D)\sum_C P(D|C)\sum_B P(C|B)\sum_A P(B|A)P(A)$$

$\sum_B F(B,C)=M(C)$

Eliminate “B”. A Table size=|C|.

Variable Elimination: Inference on a Chain



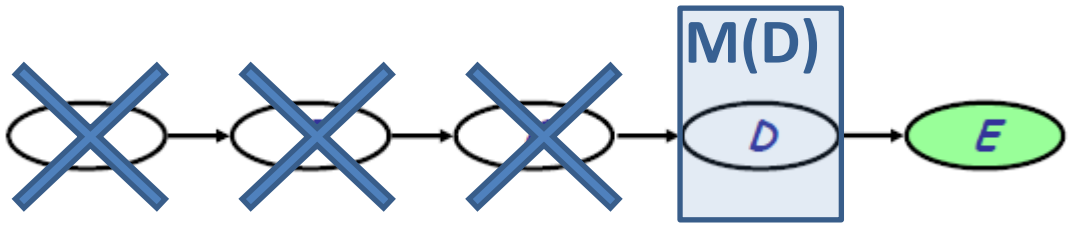
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_D P(E|D)\sum_C P(D|C)\sum_B P(C|B)\sum_A P(B|A)P(A)$$

P(D|C)M(C) = F(C,D)

A Table size=|C||D|.

Variable Elimination: Inference on a Chain



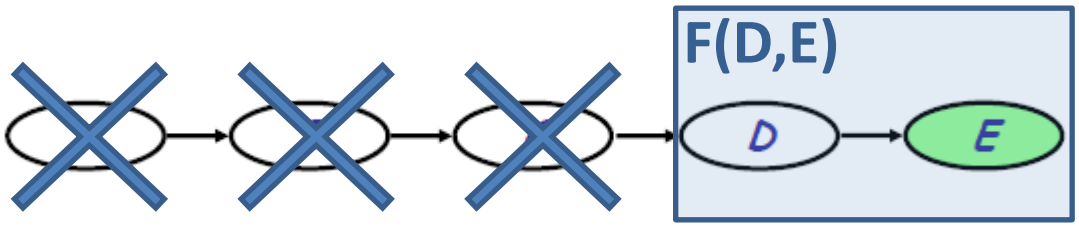
$$P(E=e)=\sum_A\sum_B\sum_C\sum_D P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_D P(E|D)\sum_C P(D|C)\sum_B P(C|B)\sum_A P(B|A)P(A)$$

$\sum_C F(C,D)= M(D)$

Eliminate “C”. A Table size=|D|.

Variable Elimination: Inference on a Chain



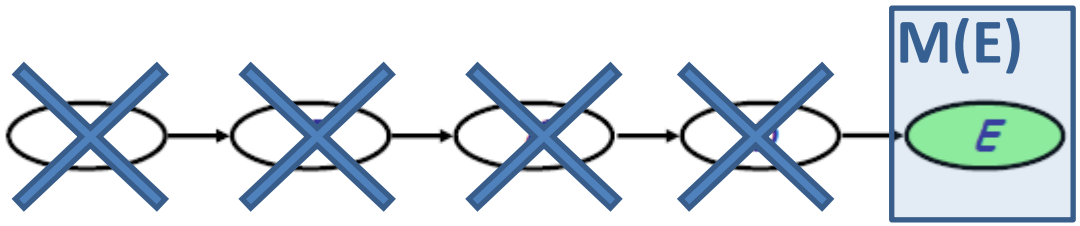
$$P(E=e)=\sum_{A}\sum_{B}\sum_{C}\sum_{D}P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_{D}P(E|D)\sum_{C}P(D|C)\sum_{B}P(C|B)\sum_{A}P(B|A)P(A)$$

P(E|D)M(D)=F(D,E)

A Table size=|D||E|.

Variable Elimination: Inference on a Chain



$$P(E=e)=\sum_{A}\sum_{B}\sum_{C}\sum_{D}P(A)P(B|A)P(C|B)P(D|C)P(E=e|D)$$

$$=\sum_{D}P(E|D)\sum_{C}P(D|C)\sum_{B}P(C|B)\sum_{A}P(B|A)P(A)$$

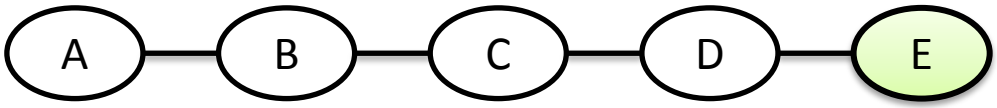
$\sum_D F(D,E)=M(E)=P(E)$

Eliminate D. Get the answer.

Both Time & Space Complexity are **O(|A||B|+|B||C|+|C||D|+|D||E|) → O(|Range|^2)**

Naïve method complexity = **O(|A||B||C||D||E|) → O(|Range|^N)**

Variable Elimination: Inference on a Chain



How about inference on Undirected Model (MRF) ?

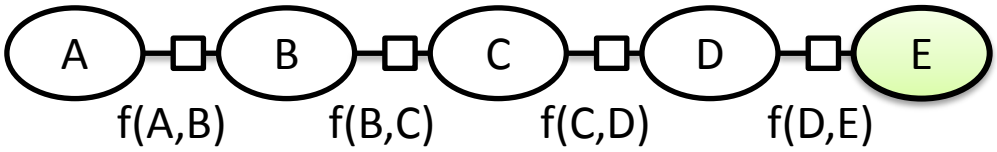
$$P(A,B,C,D,E) = \frac{1}{Z}\phi(E,D)\phi(D,C)\phi(C,B)\phi(B,A)$$

$$P(E) = \frac{1}{Z}\sum_{D}\sum_{C}\sum_{B}\sum_{A}\phi(E,D)\phi(D,C)\phi(C,B)\phi(B,A)$$

$$= \frac{1}{Z}\sum_{D}\phi(E,D)\sum_{C}\phi(D,C)\sum_{B}\phi(C,B)\sum_{A}\phi(B,A)$$

The same idea applies !!

Variable Elimination: Inference on a Chain



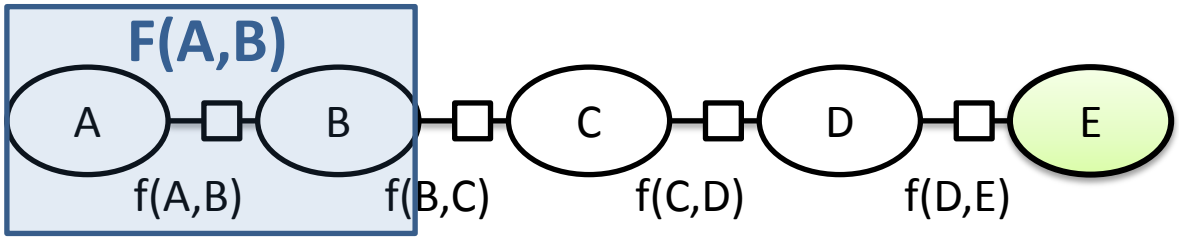
From now on, we won't distinguish between BN & MRF.
The same algorithm applies to them in a "Factor View".

$$P(A,B,C,D,E) = \frac{1}{Z}\phi(E,D)\phi(D,C)\phi(C,B)\phi(B,A)$$

$$P(A,B,C,D,E) = \mathbf{1}*P(E\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

All viewed as: $\frac{1}{Z} f(E,D) f(D,C) f(C,B) f(B,A)$

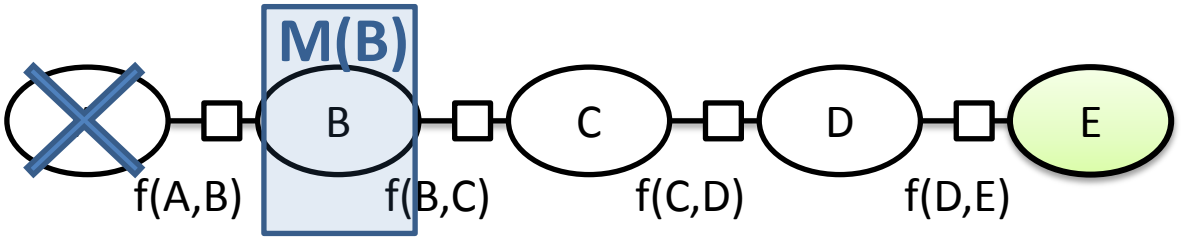
Variable Elimination: Inference on a Chain



f(A,B)=F(A,B)

The elimination process is sometimes called “Sum-Product algorithm”.

Variable Elimination: Inference on a Chain

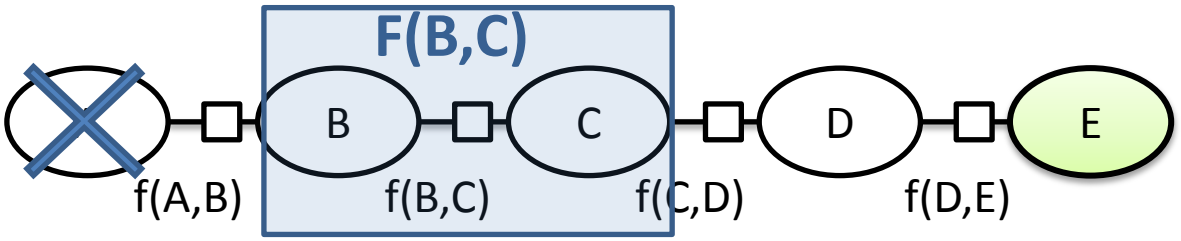


$\sum_A F(A,B) = M(B)$

Sum !

The elimination process is sometimes called "Sum-Product algorithm".

Variable Elimination: Inference on a Chain

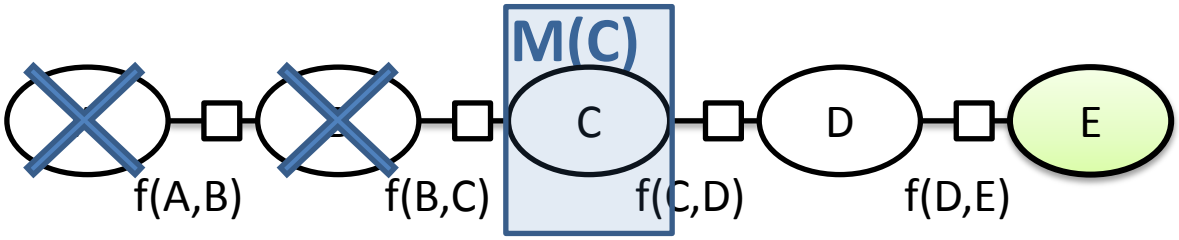


M(B)*f(B,C)= F(B,C)

Product !

The elimination process is sometimes called “Sum-Product algorithm”.

Variable Elimination: Inference on a Chain

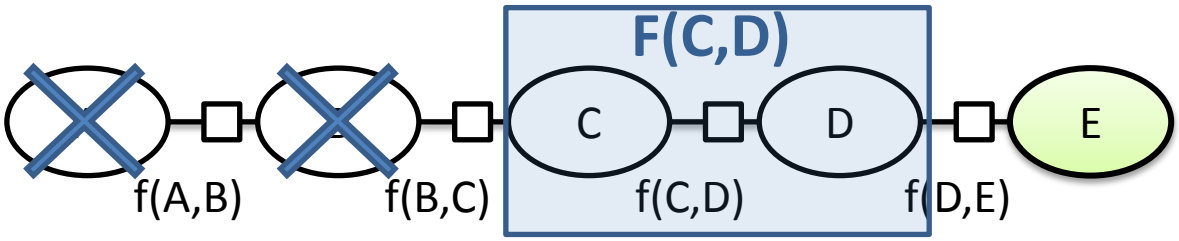


$\sum_B F(B,C)=M(C)$

Sum !

The elimination process is sometimes called “Sum-Product algorithm”.

Variable Elimination: Inference on a Chain

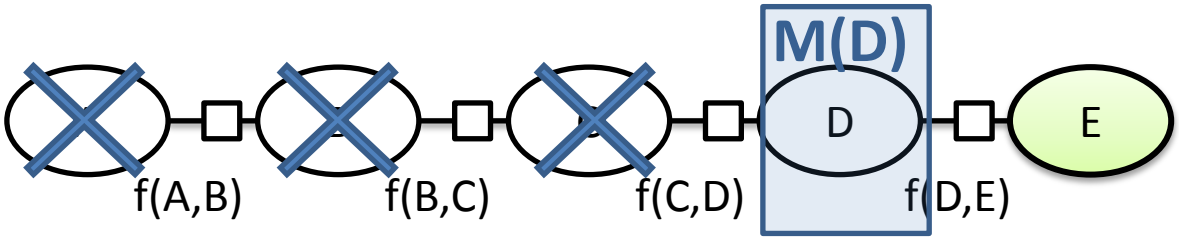


M(C)*f(C,D) = F(C,D)

Product !

The elimination process is sometimes called “Sum-Product algorithm”.

Variable Elimination: Inference on a Chain

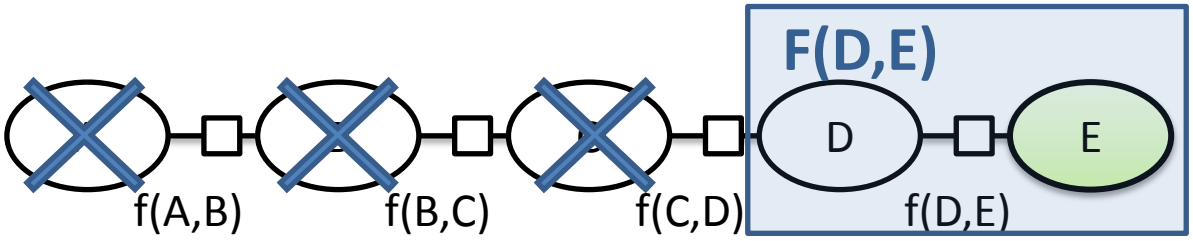


$\sum_C F(C,D) = M(D)$

Sum !

The elimination process is sometimes called "Sum-Product algorithm".

Variable Elimination: Inference on a Chain

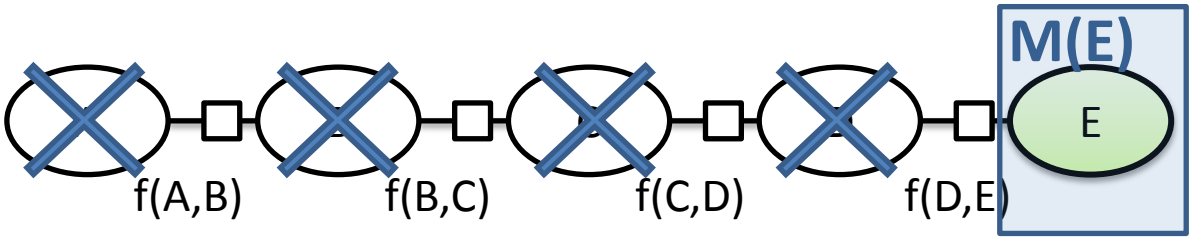


M(D)*f(D,E) = F(D,E)

Product !

The elimination process is sometimes called “Sum-Product algorithm”.

Variable Elimination: Inference on a Chain



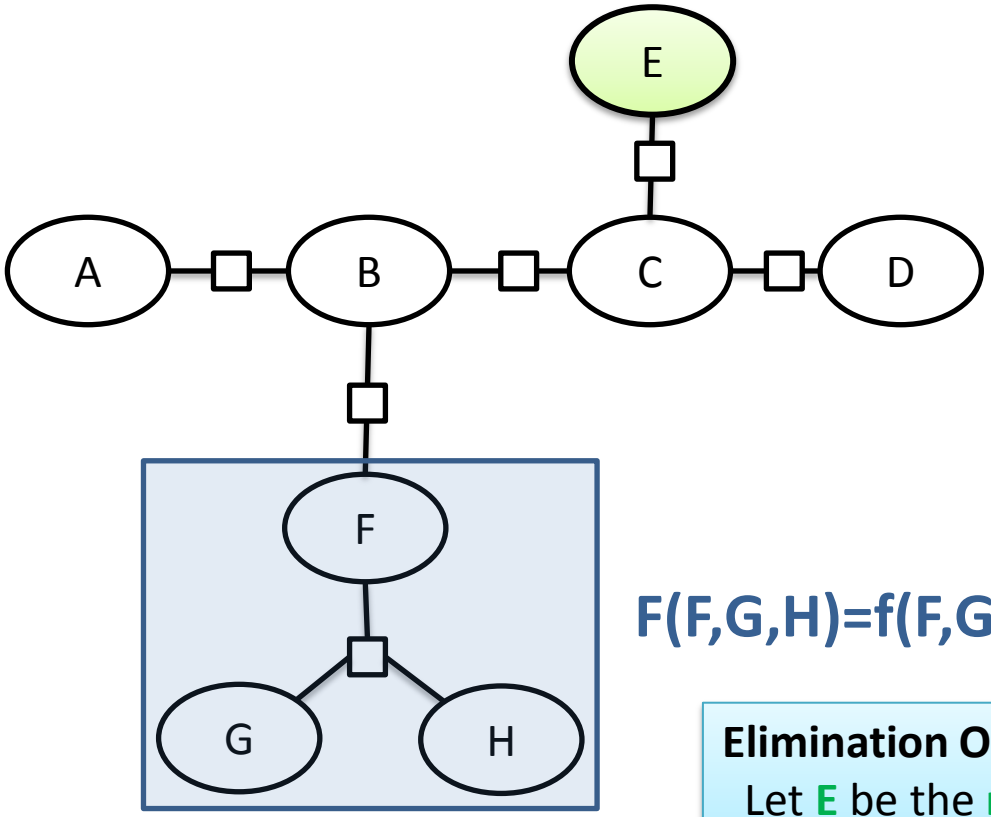
$$\sum_D F(D,E) = M(E) = P(E)$$

Sum !

The elimination process is sometimes called “Sum-Product algorithm”.

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle,
then VE can be applied in a similar way.

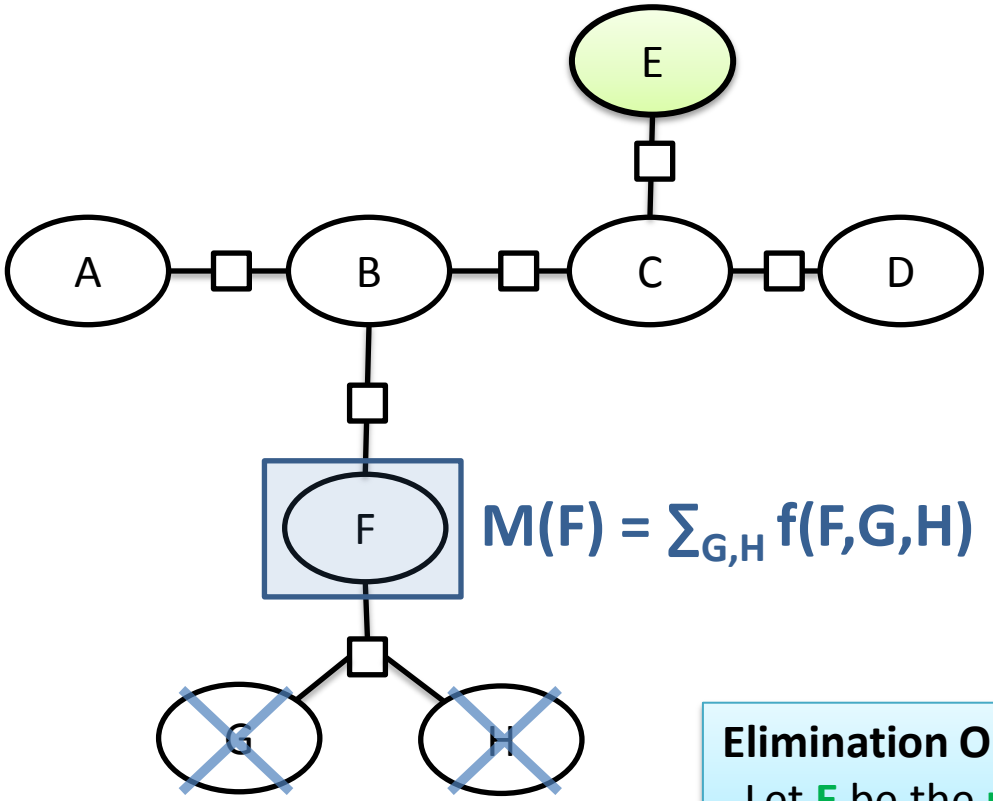


F(F,G,H)=f(F,G,H)

Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle,
then VE can be applied in a similar way.

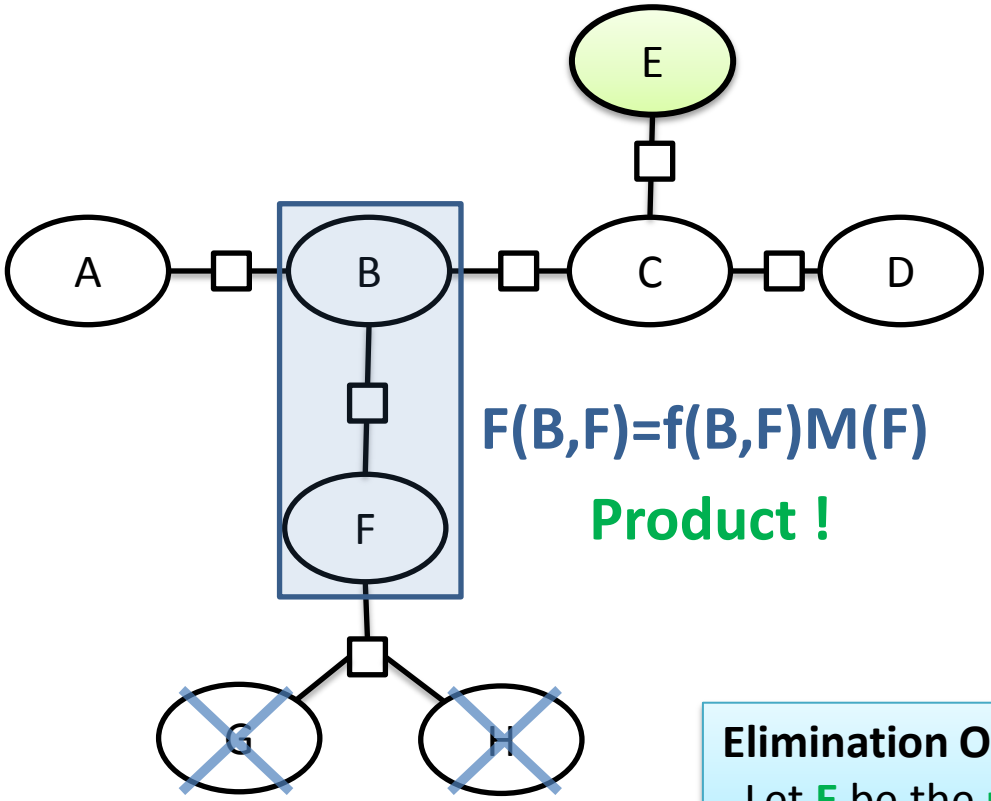


$M(F) = \sum_{G,H} f(F,G,H)$ **Sum !**

Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

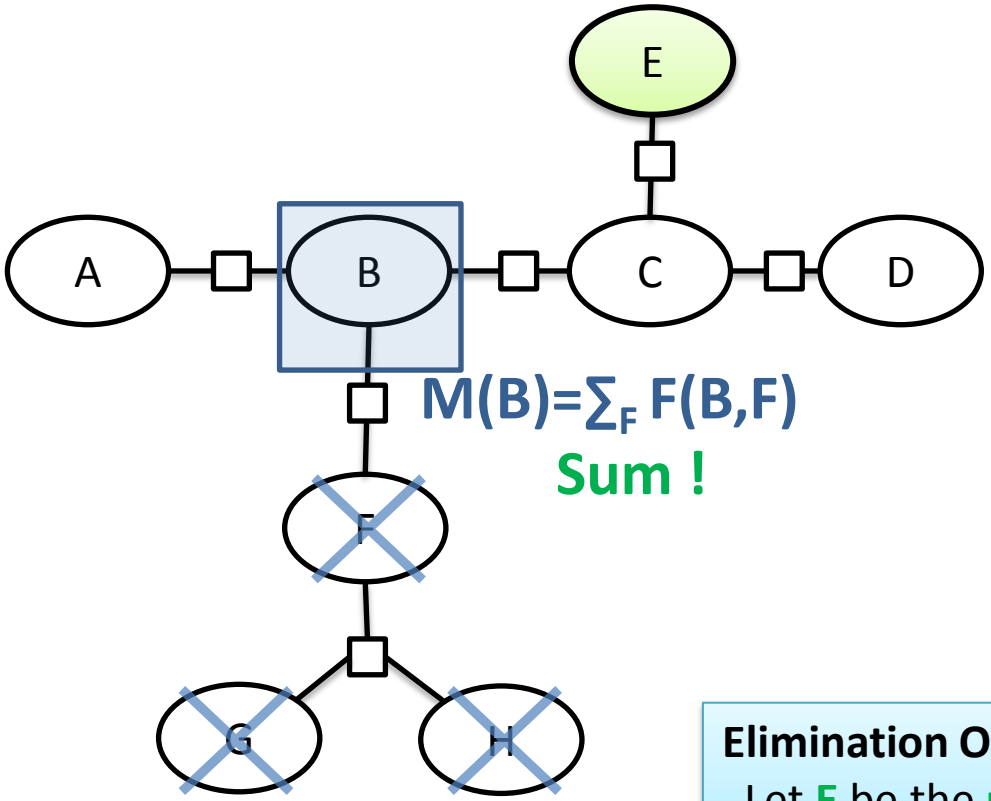
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

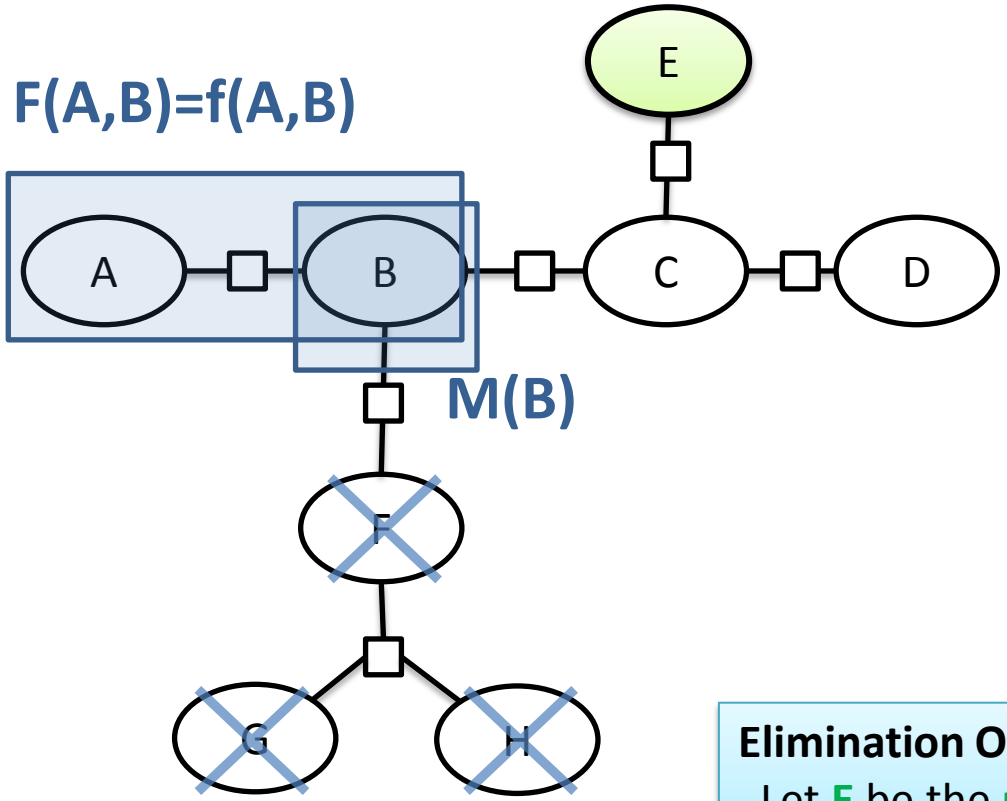
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

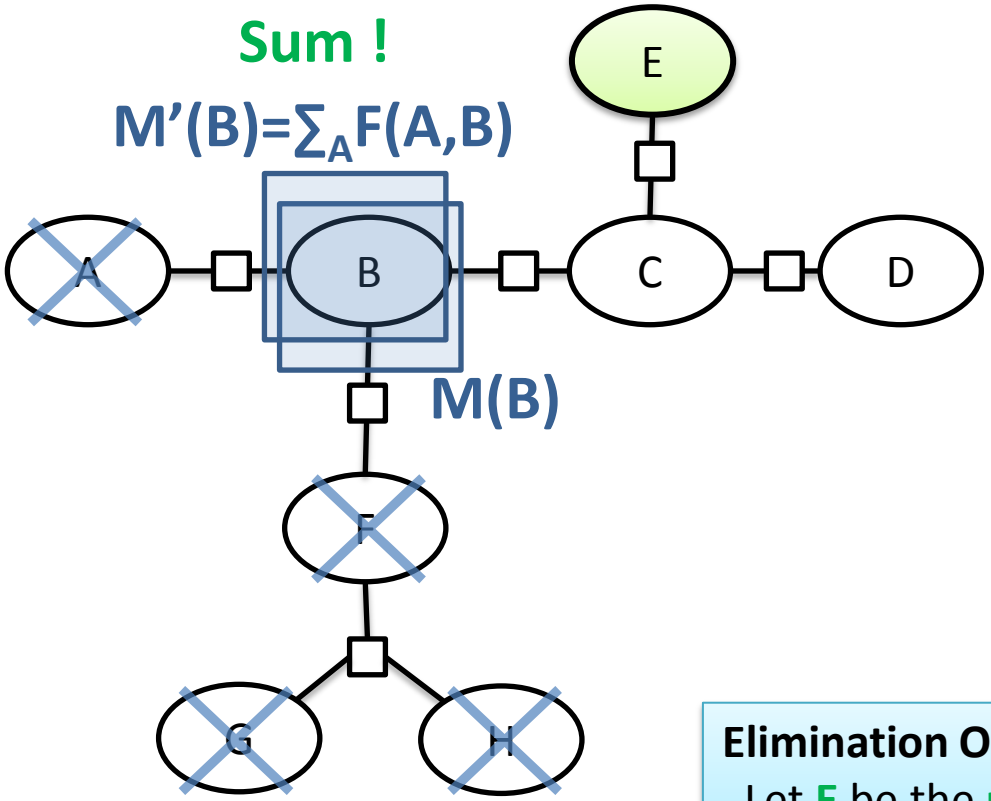
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

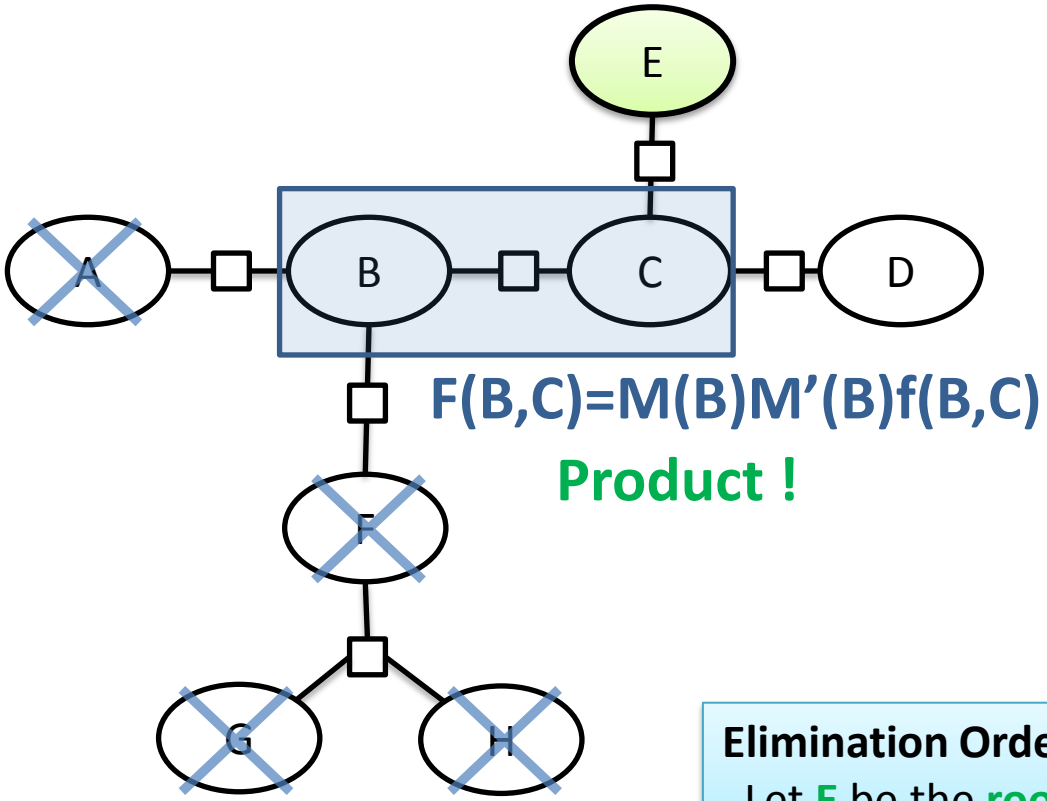
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

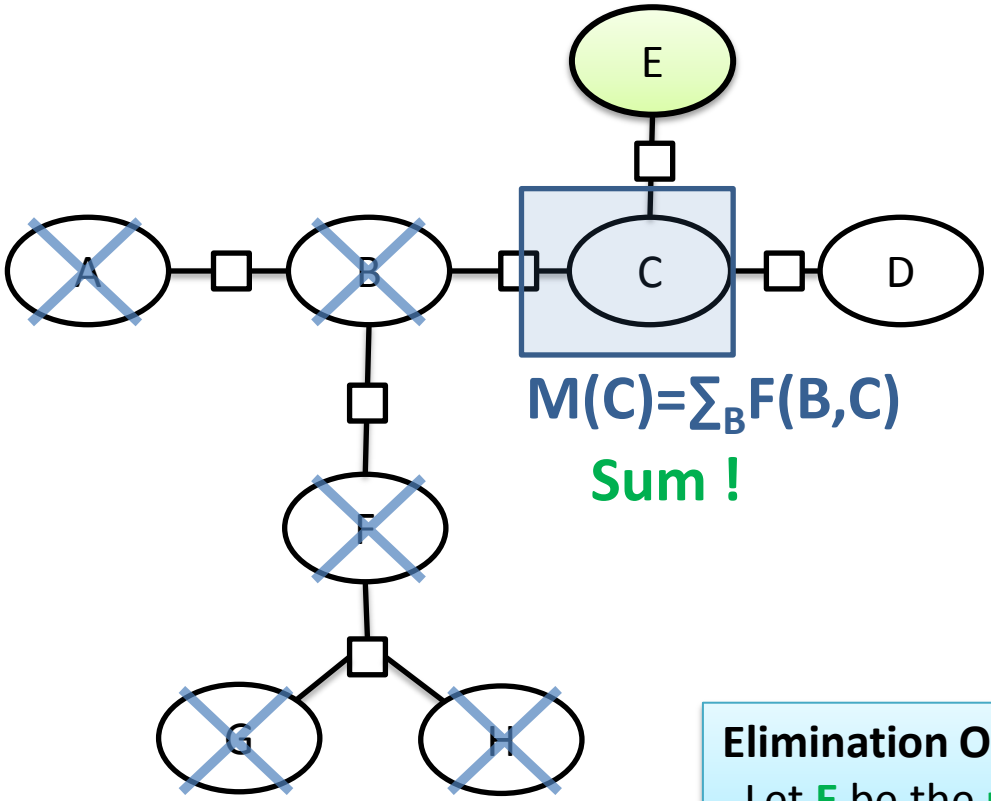
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

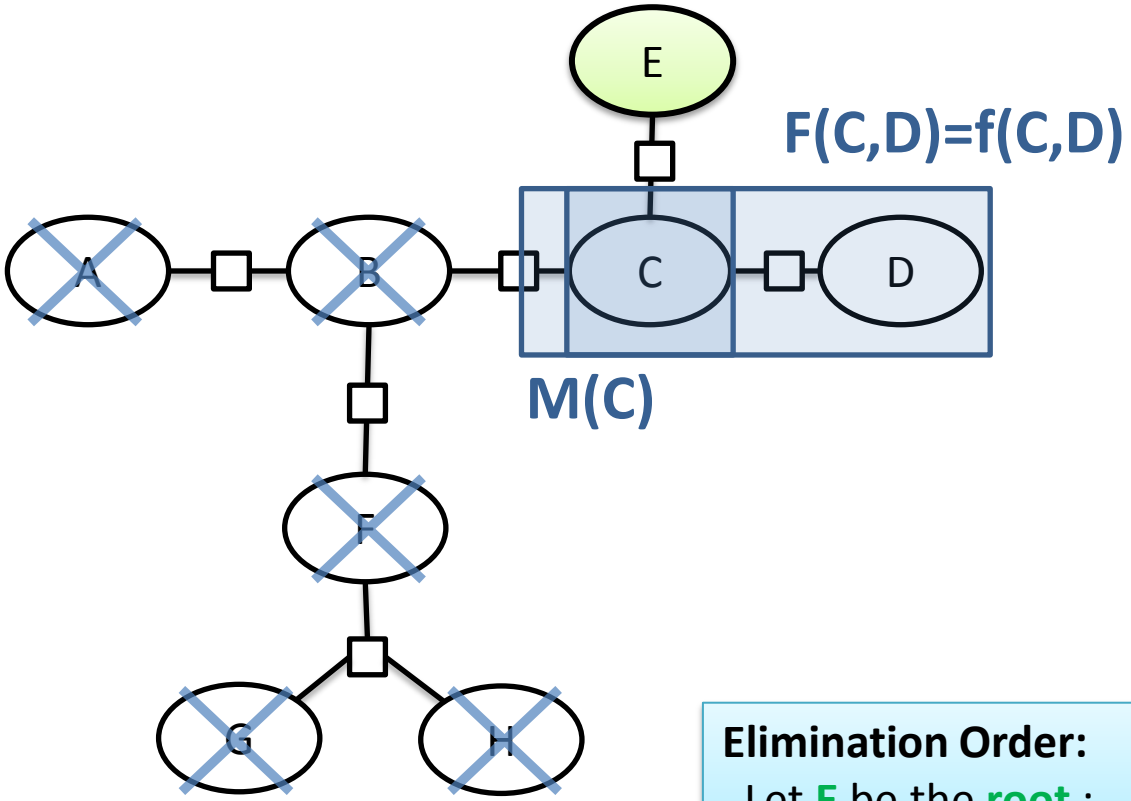
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

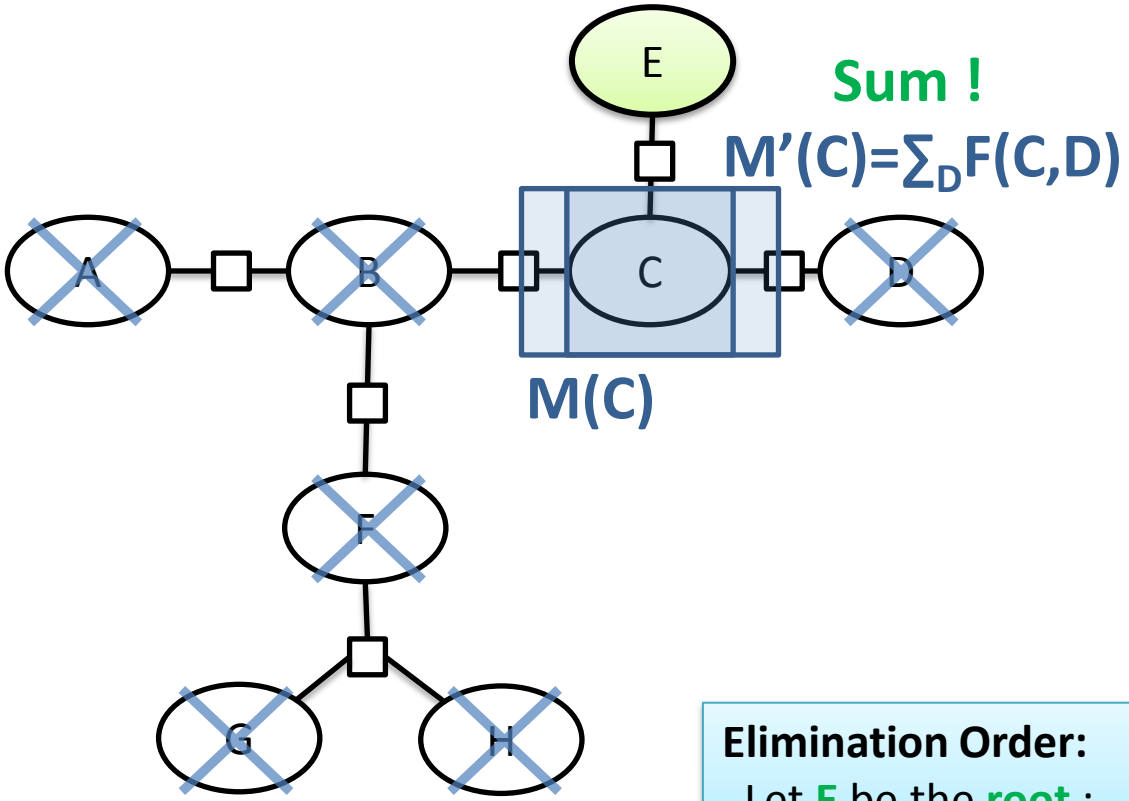
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

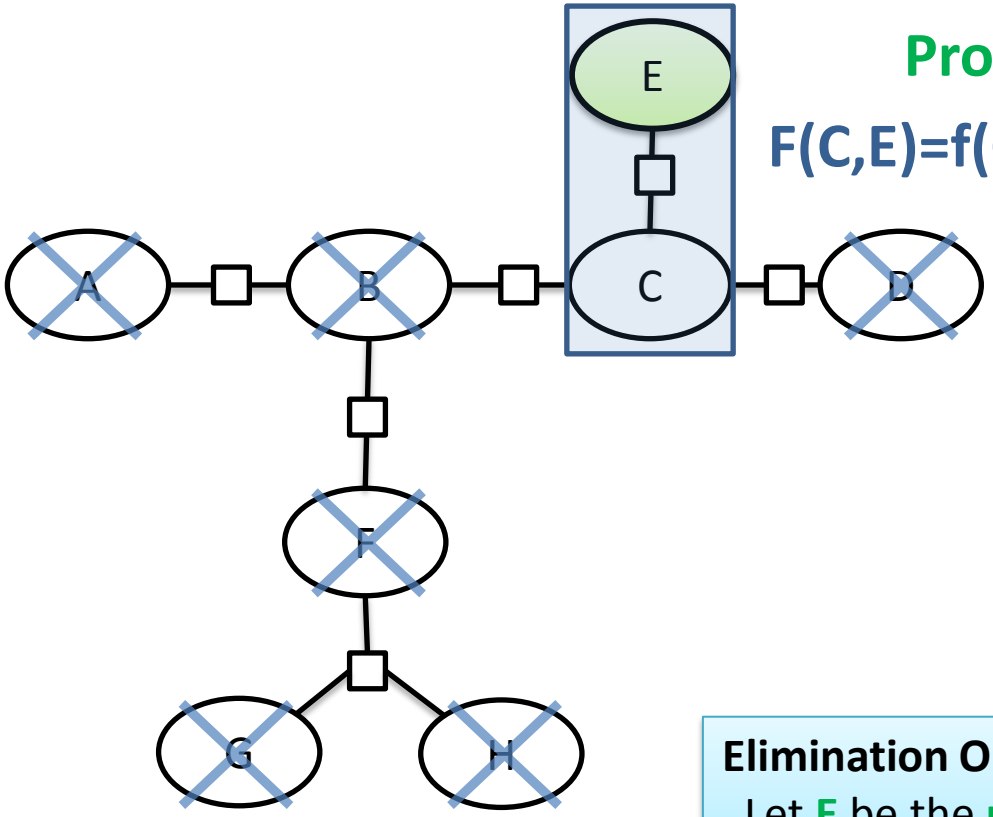
If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle,
then VE can be applied in a similar way.



Product !

F(C,E)=f(C,E)M(C)M'(C)

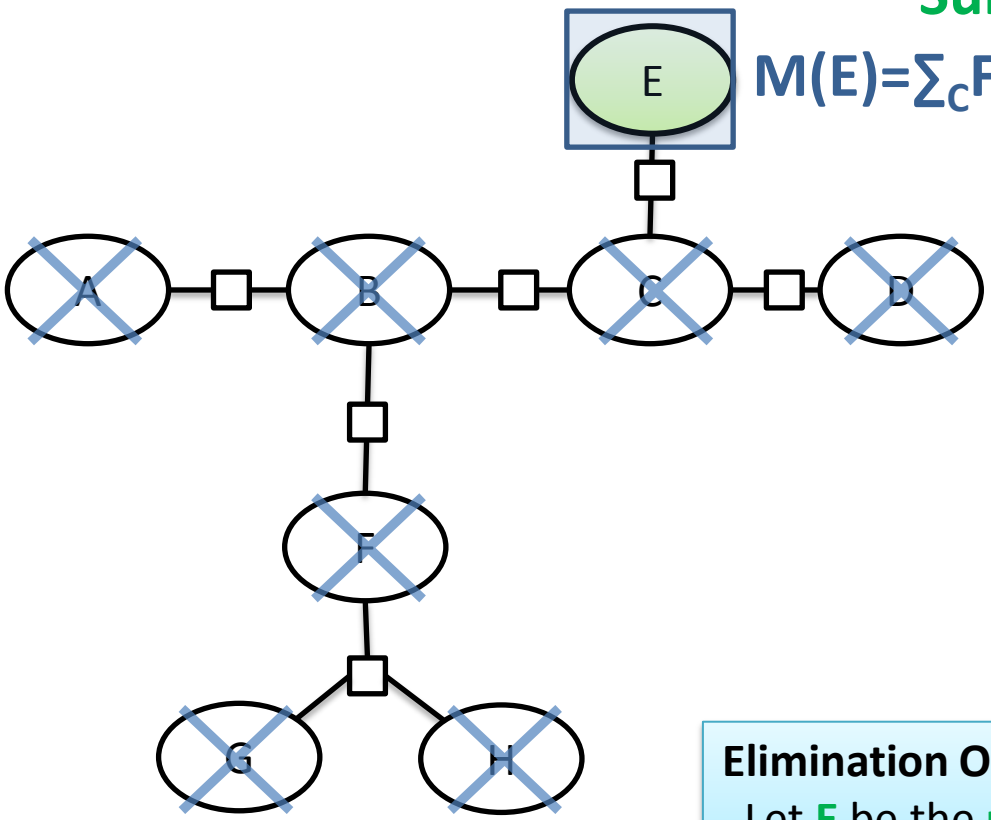
Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

If the factor graph is a tree without cycle,
then VE can be applied in a similar way.

Sum !

$M(E)=\sum_C F(C,E)=P(E)$

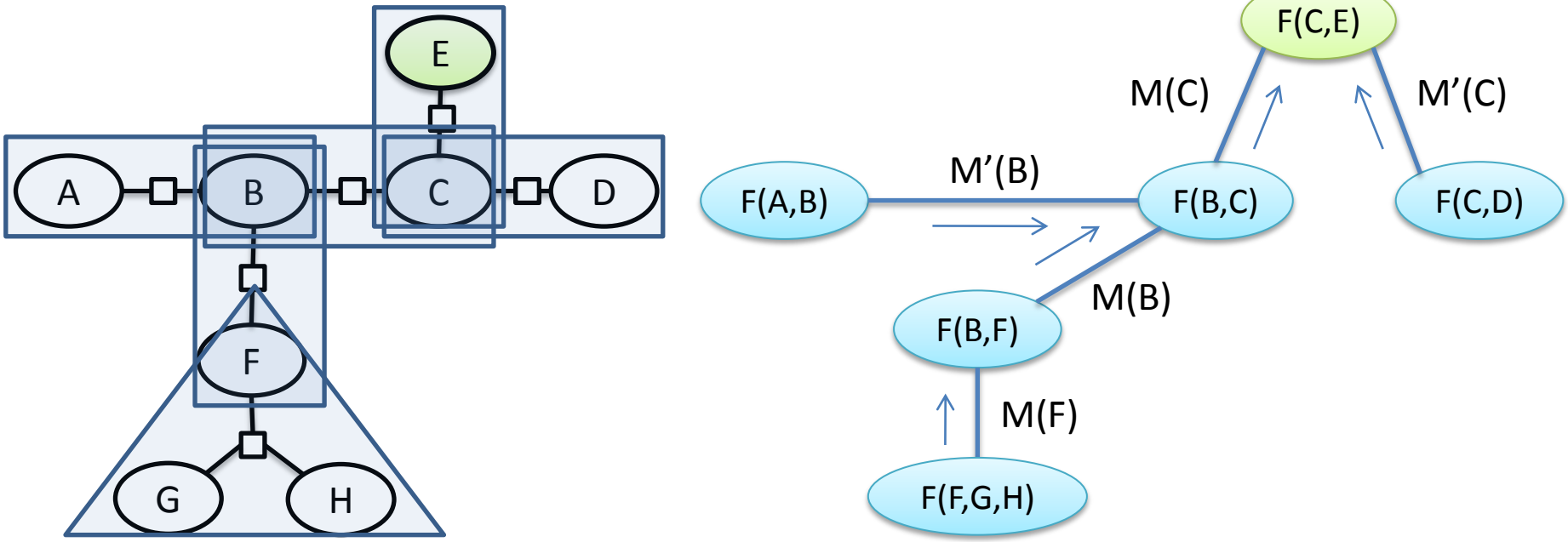


Elimination Order:
Let **E** be the **root** ;
Eliminate **from leaves to root**.

Variable Elimination: Inference on a Tree

Follow the Elimination Process, we can build a "Clique Tree". In which:

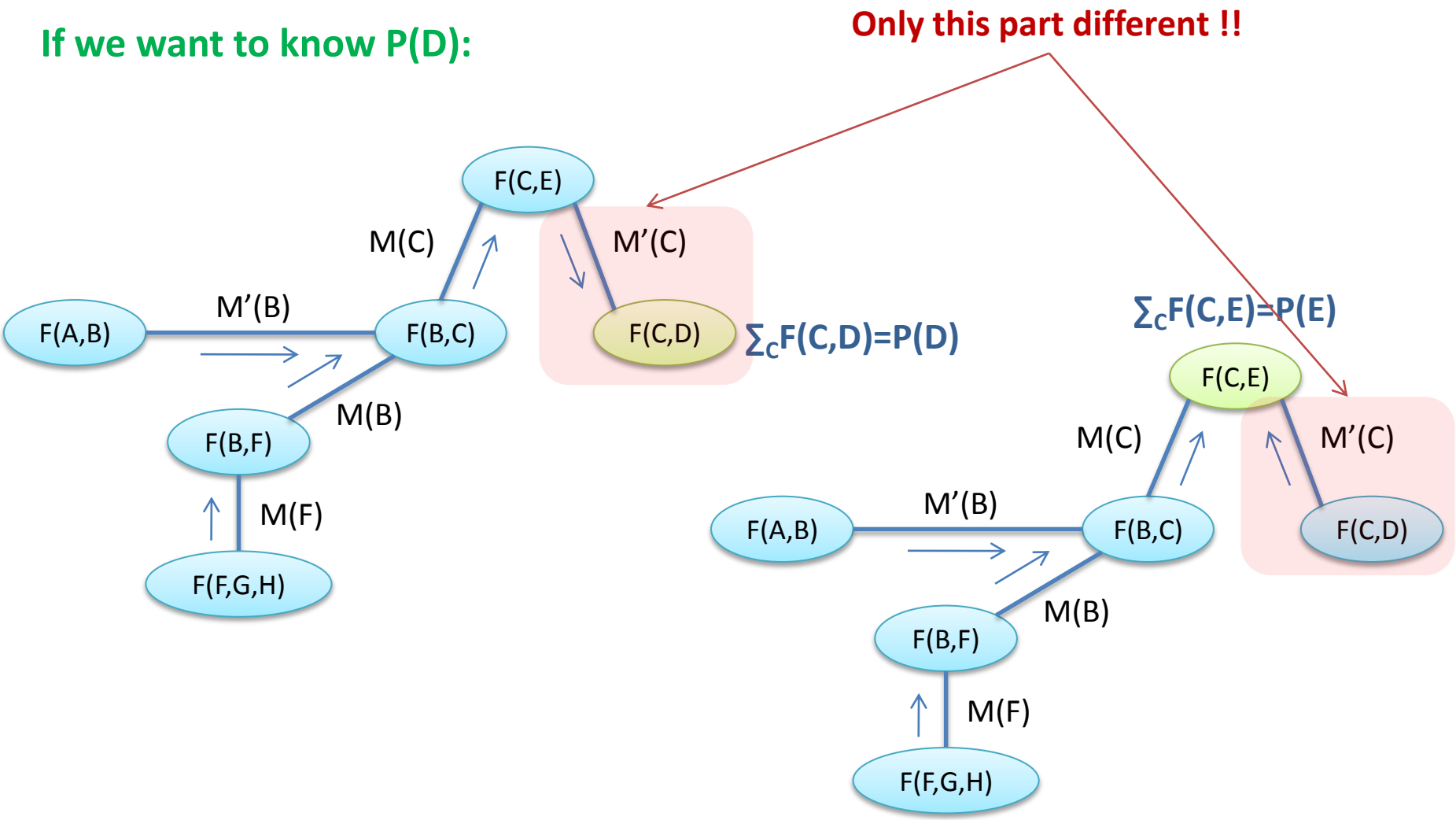
1. Every node is a F(.) before elimination.
2. Every edge is a "message" M(.) passed from F(.) to F(.).



Variable Elimination: Inference on a Tree

Why is Clique Tree useful ?

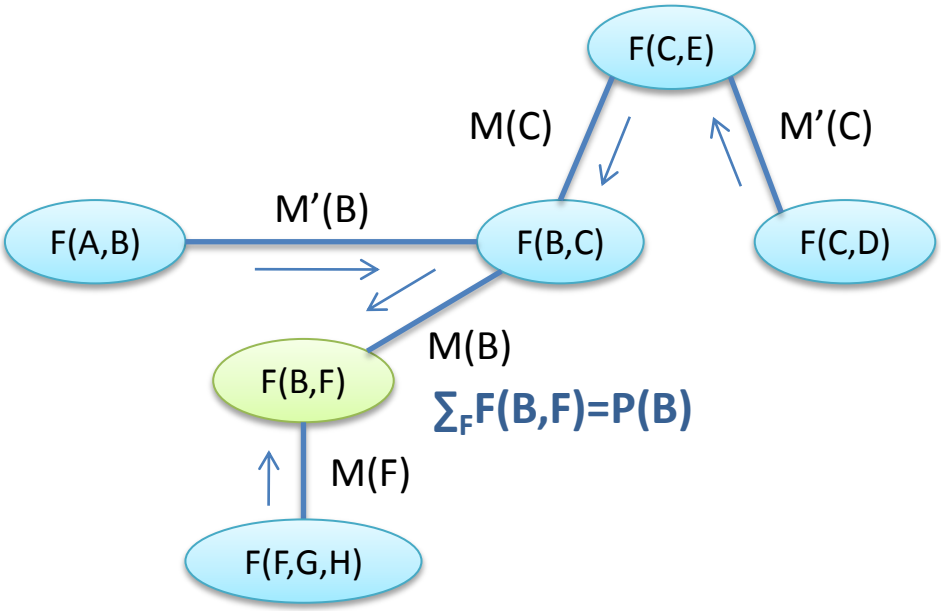
If we want to know P(D):

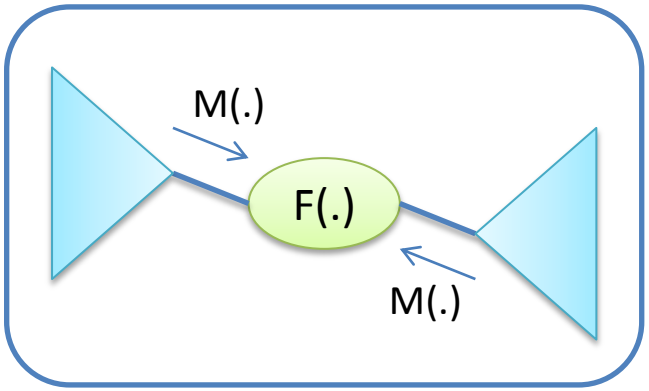


Variable Elimination: Inference on a Tree

Why is Clique Tree useful ?

If we want to know P(B):



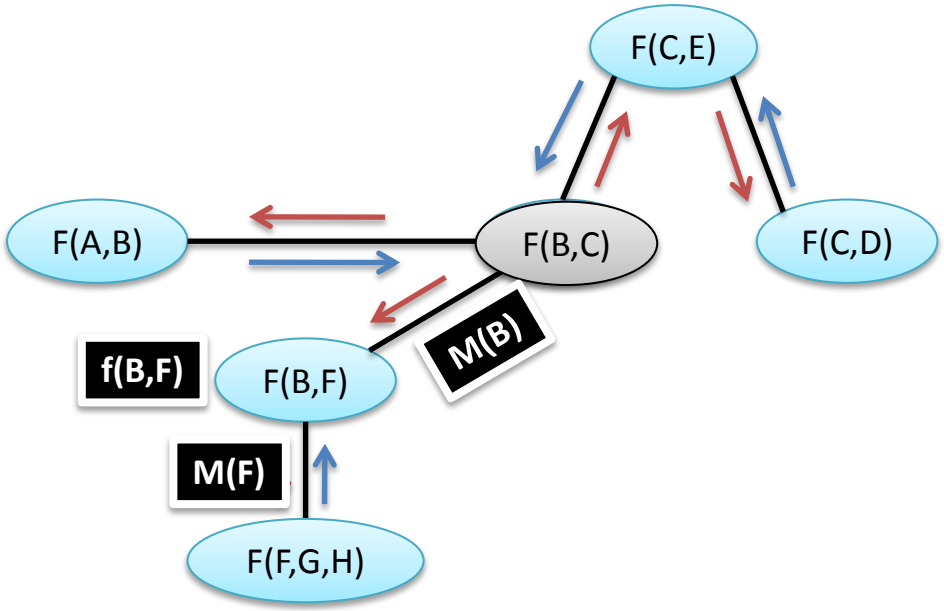


The queried node should get messages from all nodes on the tree to get the marginal distribution.

Variable Elimination: Inference on a Tree

Why is Clique Tree useful ?

To get marginal distribution of N nodes, we don't need run VE "N times", "2 times" are enough to get all possible messages.



$$P(B,F) = M(B)\, f(B,F) M(F)$$

1st pass:
Take a node as root.
Run Sum-Product from leaves to root.

2nd pass:
Run Sum-Product from root to leaves.

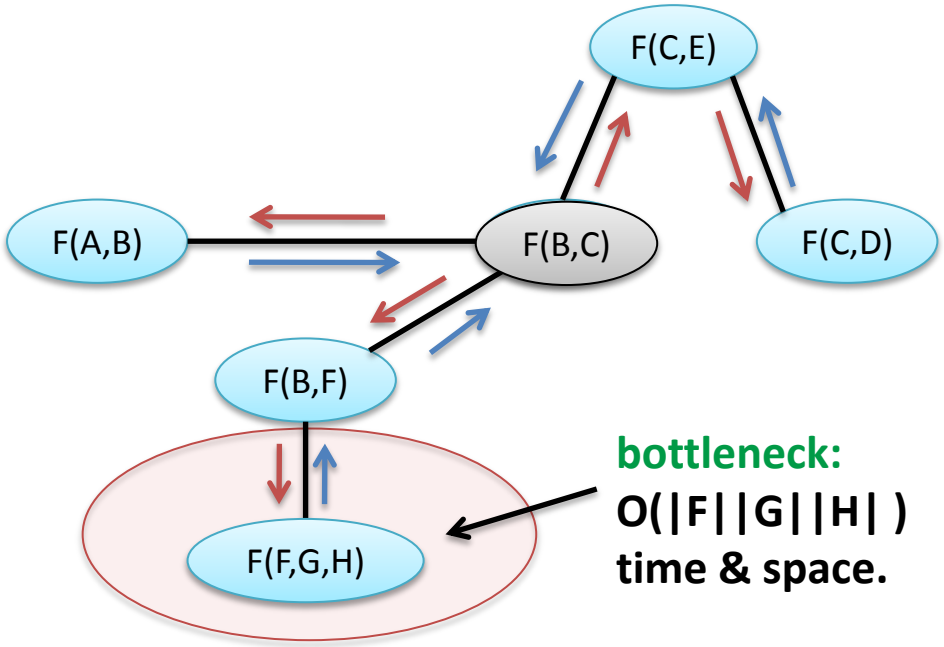
All marginal dist. can be derived from

1. Multiply all M(.) from neighbors by f(.) on this node.
2. Eliminate unwanted variables.

Variable Elimination: Inference on a Tree

Why is Clique Tree useful ?

To get marginal distribution of N nodes, we don't need run VE "N times", "2 times" are enough to get all possible messages.



Complexity :

Elimination on a node F(A,B,C) takes O(|A||B||C|) space & time.

So the algorithm's bottleneck is on elimination for the "**Largest Node**" on clique tree.

Agenda

- Introduce the concept of “Variable Elimination” in special case of Tree-structured Factor Graph.

- **Extend the idea of “VE” to General Factor Graph with concept of “Clique Tree”.**

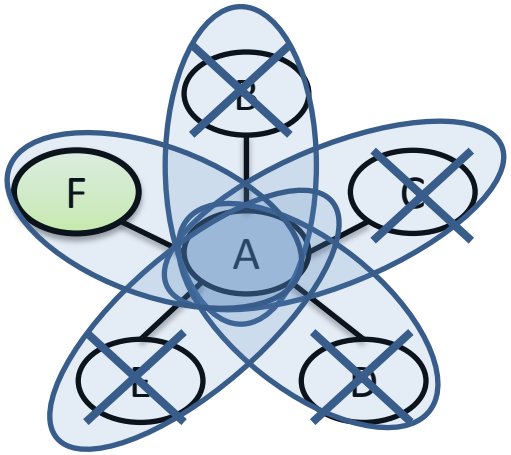
- See how to extend “VE” to “Most Probable Assignment” (MAP configuration) Problem.

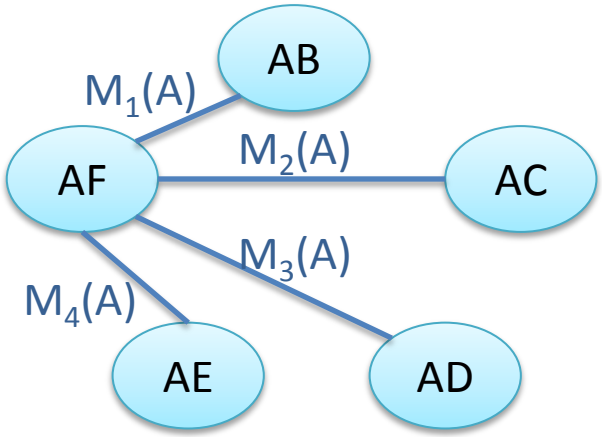
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 1: B C D E A





$$P(F)$$

$$= \frac{1}{Z} \sum_A \sum_E \sum_D \sum_C \sum_B f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)$$

$$= \frac{1}{Z} \sum_A f(A,F) \sum_E f(A,E) \sum_D f(A,D) \sum_C f(A,C) \sum_B f(A,B)$$

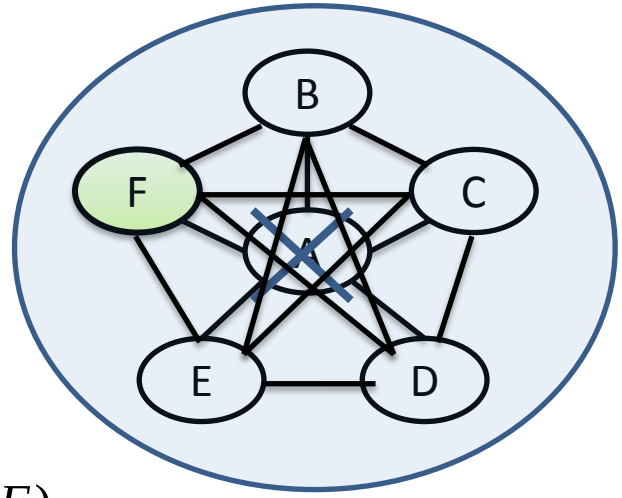
Maximum Node Size=2

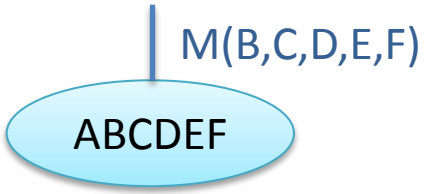
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





$$P(F)$$

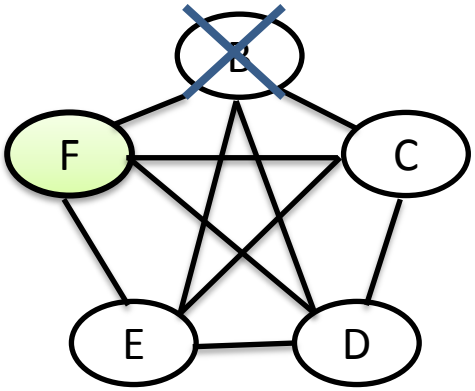
$$= \frac{1}{Z} \sum_{E} \sum_{D} \sum_{C} \sum_{B} \underbrace{\sum_{A} f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)}_{\text{M(B,C,D,E,F)}}$$

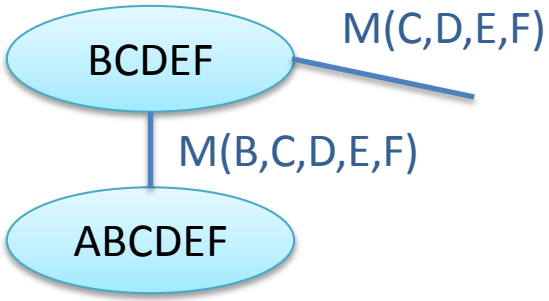
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





$$P(F)$$

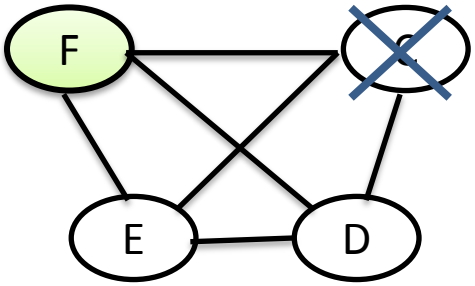
$$= \frac{1}{Z} \sum_{E} \sum_{D} \sum_{C} \sum_{B} \sum_{A} f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)$$

M(C,D,E,F)

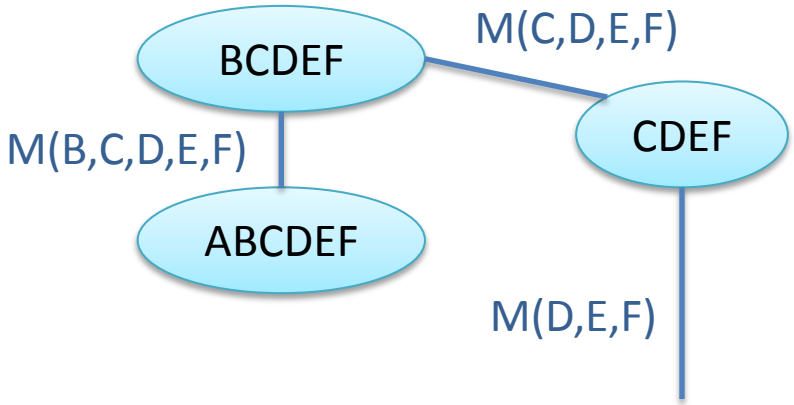
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.



Elimination Order 2: A B C D E



$$P(F)$$

$$= \frac{1}{Z} \sum_{E} \sum_{D} \sum_{C} \sum_{B} \sum_{A} f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)$$

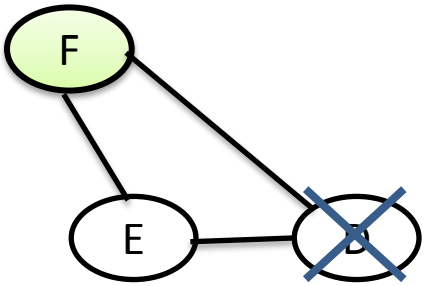
M(D,E,F)

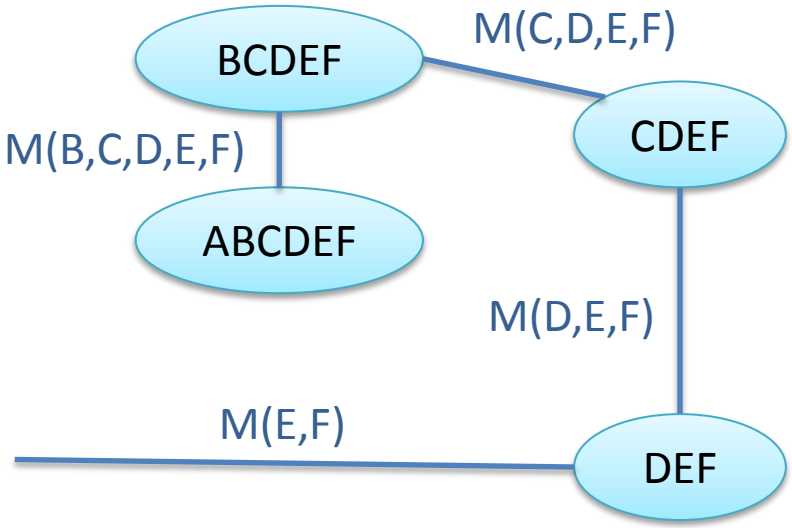
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





$$P(F)$$

$$= \frac{1}{Z} \sum_{E} \sum_{D} \sum_{C} \sum_{B} \sum_{A} f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)$$

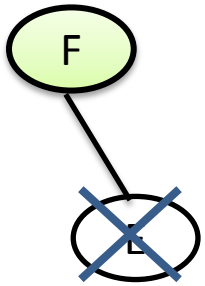
$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{M(E,F)}$$

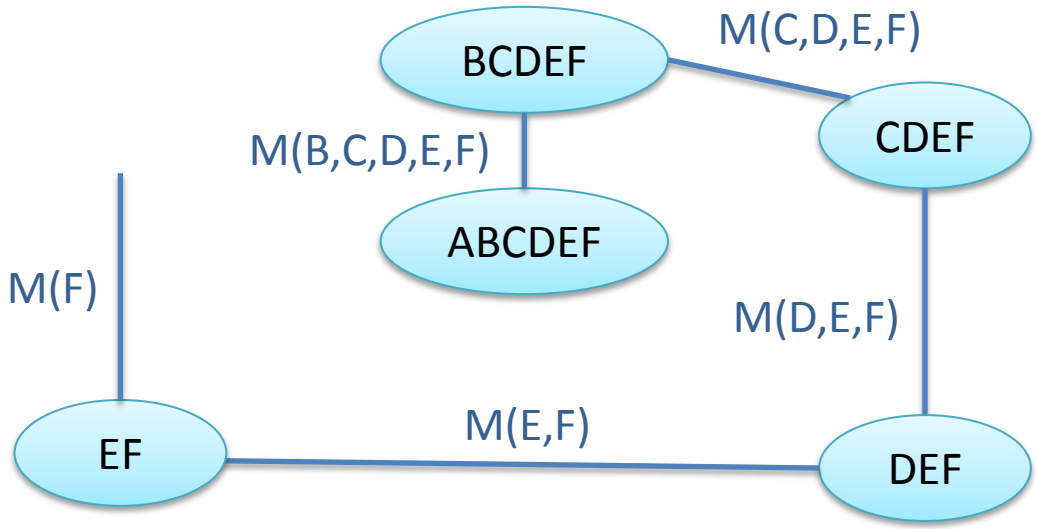
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E





$$P(F) = \frac{1}{Z} \sum_{E} \sum_{D} \sum_{C} \sum_{B} \sum_{A} f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)$$

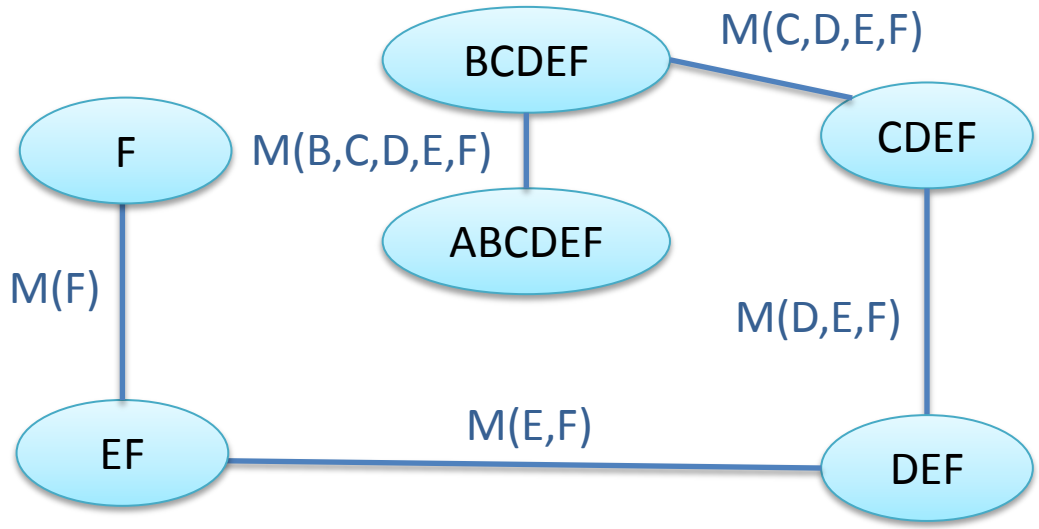
M(F)

Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 2: A B C D E



$$P(F)$$

$$= \frac{1}{Z} \sum_{E} \sum_{D} \sum_{C} \sum_{B} \sum_{A} f(A,F) f(A,E) f(A,D) f(A,C) f(A,B)$$

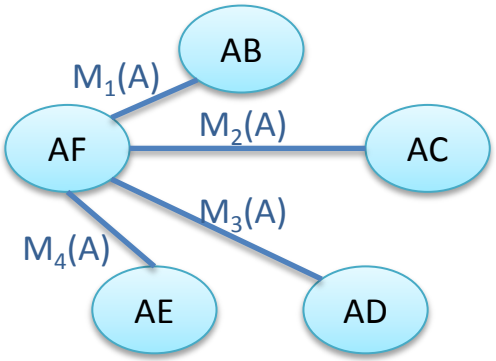
Maximum Node Size=6

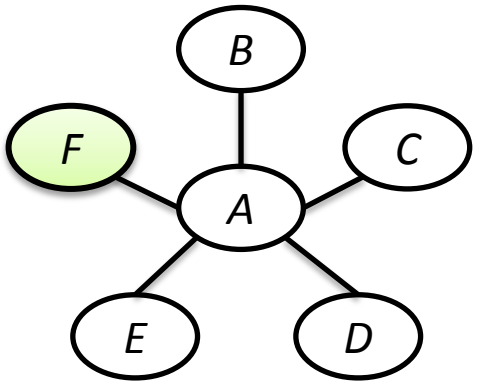
Variable Elimination: Inference on General Graph

Some problem ignored earlier:

Different "Elimination Orders" have different effect.

Elimination Order 1: B C D E A





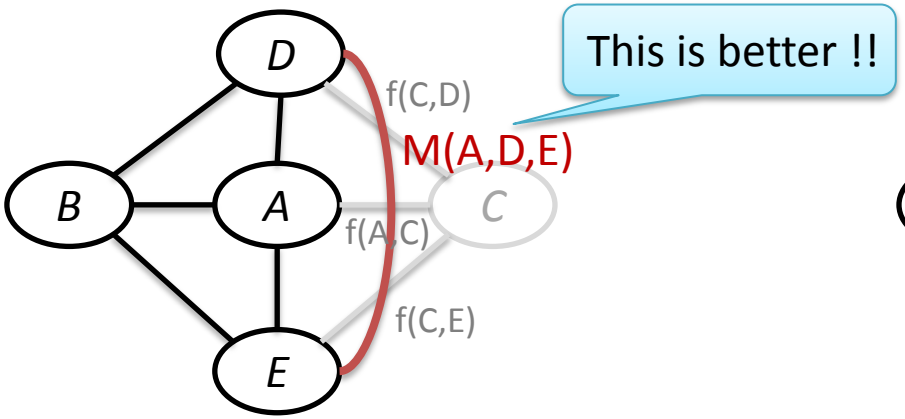
In "Tree" structure factor graph, the optimal "Elimination Order" is just "Elimination from leaves".

If factor graph is not Tree, what's the best elimination order ???

Variable Elimination: Inference on General Graph

When factor graph is not Tree, we want a Elimination Order "introducing as fewer edges as possible" (then we will have factor size smaller).

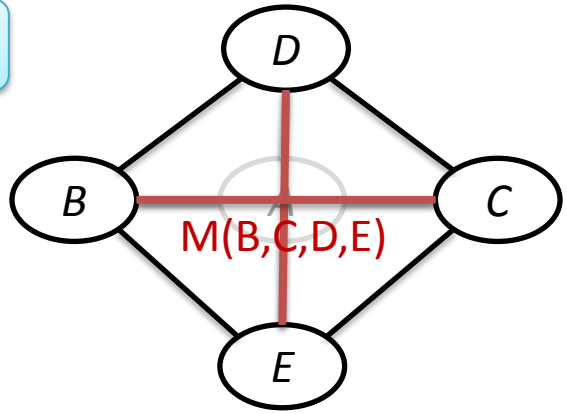
Eliminate "C" ➔ fill 1 edge



$$M(A,D,E) = \sum_C f(A,C) f(C,D) f(C,E)$$

Produce a factor of size 3

Eliminate "A" ➔ fill 2 edges



$$M(B,C,D,E) = \sum_A f(A,B) f(A,C) f(A,D) f(A,E)$$

Produce a factor of size 4

Variable Elimination: Inference on General Graph

Unfortunately, Finding Elimination order with "smallest maximum factor" is NP-hard.

It's fortunate that greedy algorithm works quite well in practical, in which, we just search for the "least-cost" variable to eliminate:

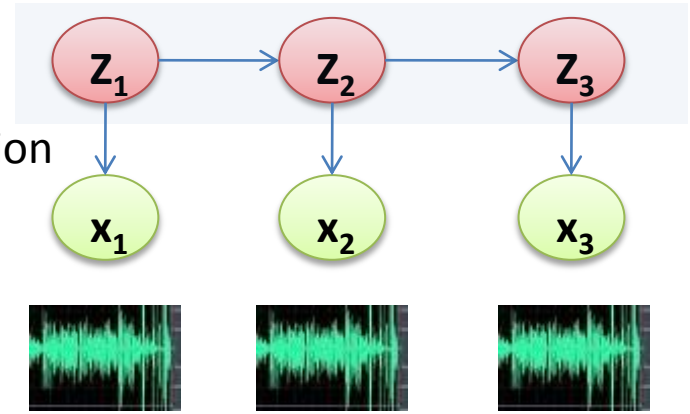
1. If variables have same cardinality
   ➔ cost = (# of edges introduced by elimination).

2. If variables have different cardinality
   ➔ cost = (# of edges)*(weight by cardinality of node involved)

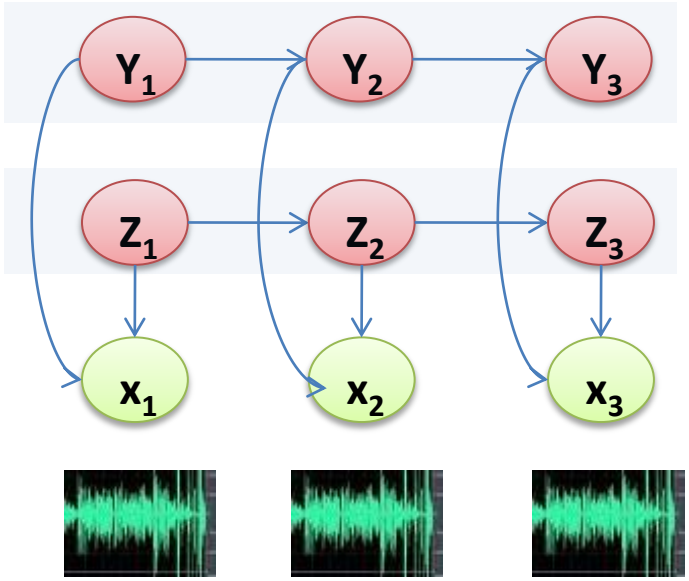
Example: Factorial HMM

Language Model of "words sequence"

Speech Recognition:



Decoding 2 person's speech from waves:



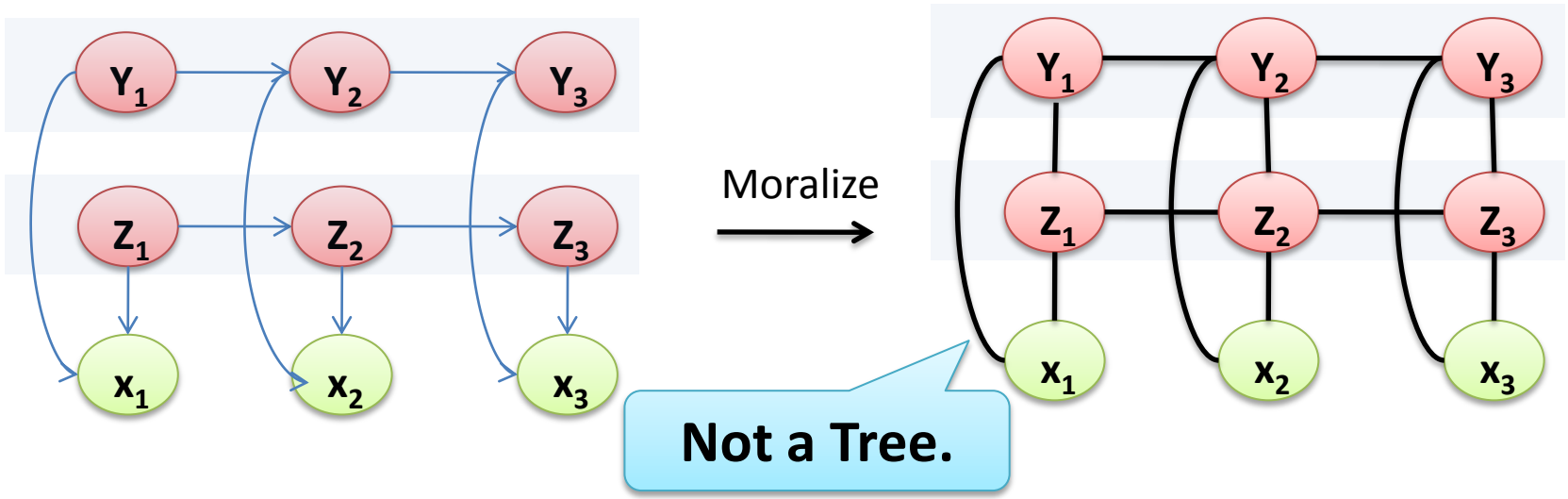
Language Model of 1st person

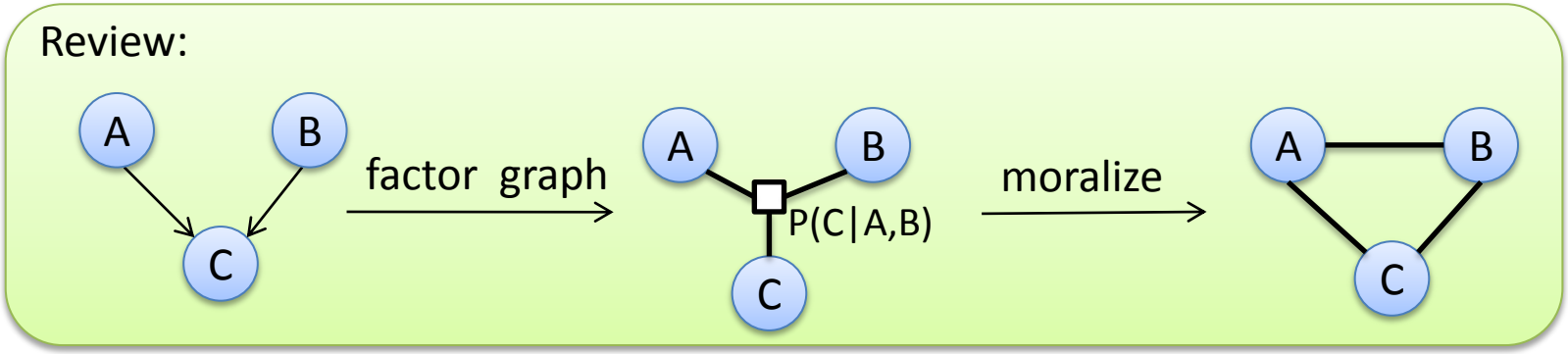
Language Model of 2nd person

Superposition of "2 waves"

Example: Factorial HMM

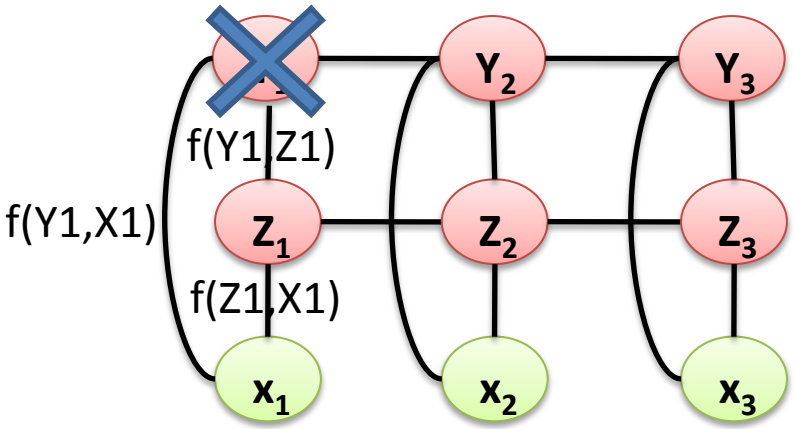
Because a factor is a "clique" in undirected representation,
we transform Factorial HMM into "undirected" before running VE.





Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)



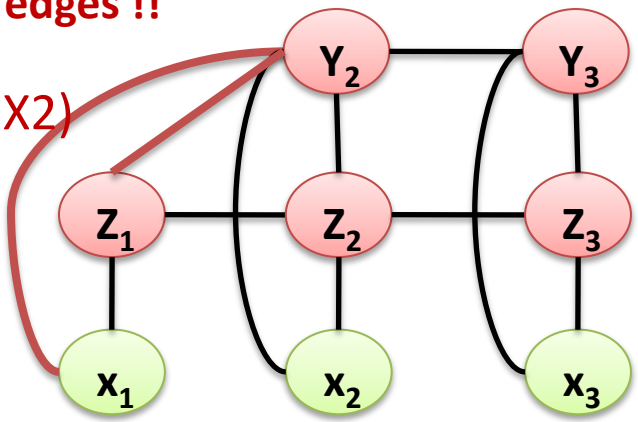
$$M(Z_1, Y_2, X_1) = \sum_{Y_1} f(Y_1, Z_1) f(Y_1, Y_2) f(Y_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)

Intro. 2 edges !!

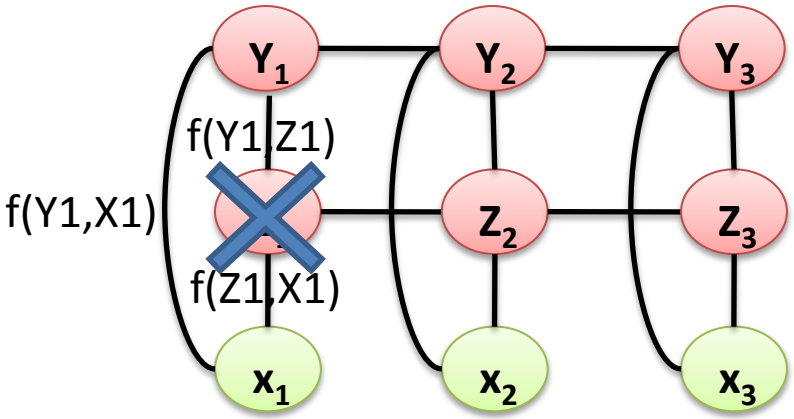
M(Z1,Y2,X2)



$$M(Z_1, Y_2, X_1) = \sum_{Y_1} f(Y_1, Z_1)\, f(Y_1, Y_2)\, f(Y_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)

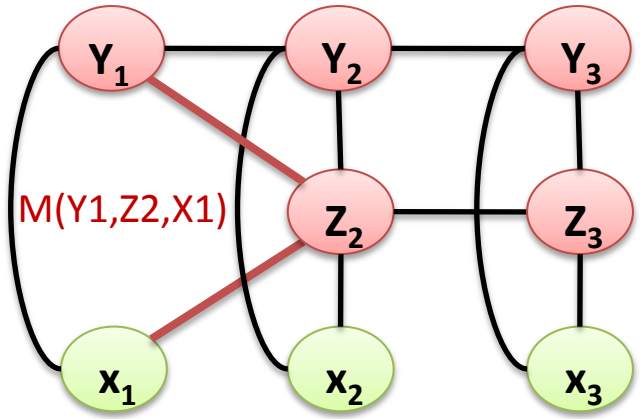


$$M(Y_1, Z_2, X_1) = \sum_{Z_1} f(Y_1, Z_1) f(Z_1, Z_2) f(Z_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)

Intro. 2 edges !!

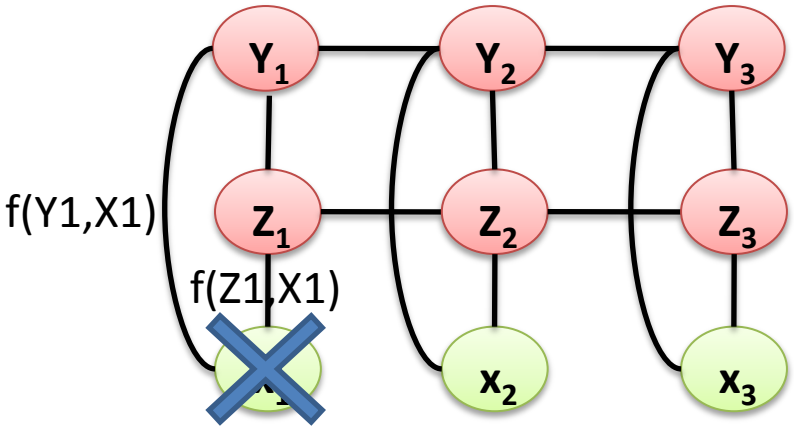


$$M(Y_1, Z_2, X_1) = \sum_{Z_1} f(Y_1, Z_1) f(Z_1, Z_2) f(Z_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)

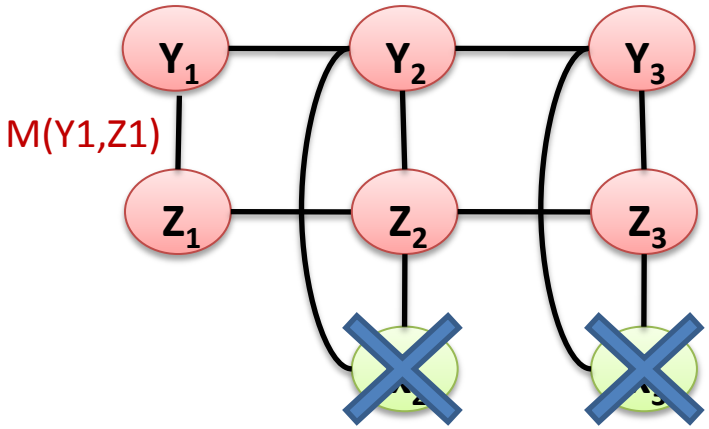


$$M(Y_1, Z_1) = \sum_{X_1} f(Y_1, X_1) f(Z_1, X_1)$$

Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)

Intro. no edges !! ➔ **Let's eliminate !!**

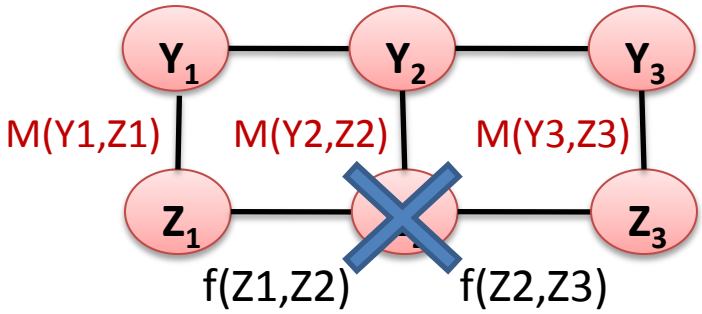


$$M(Y_1, Z_1) = \sum_{X_1} f(Y_1, X_1) f(Z_1, X_1)$$

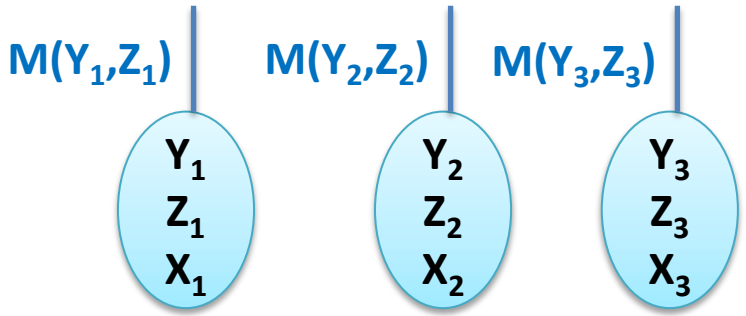
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



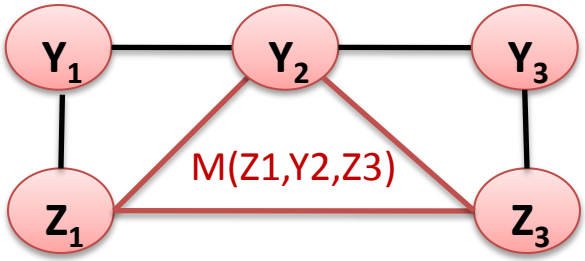
$$M(Z_1, Y_2, Z_3) = \sum_{Z_2} M(Y_2, Z_2)\, f(Z_1, Z_2)\, f(Z_2, Z_3)$$



Example: Factorial HMM

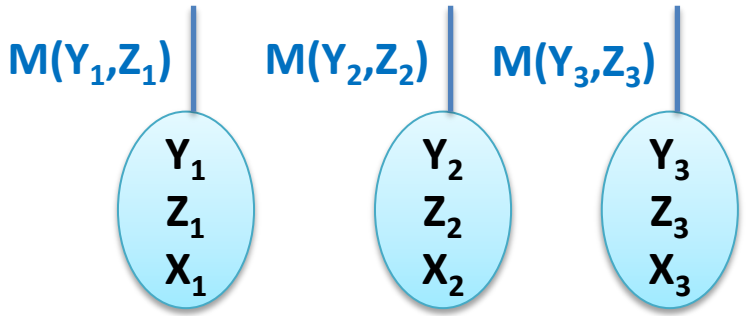
Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



Intro. 3 edges !!

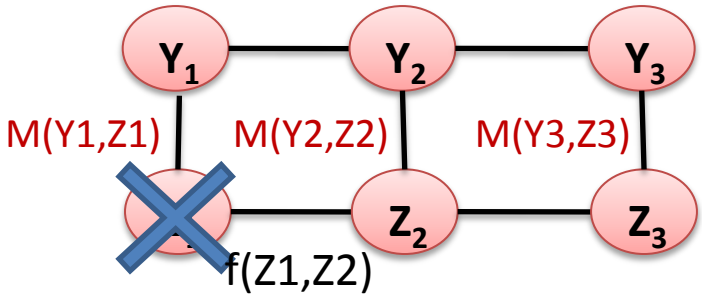
$$M(Z_1, Y_2, Z_3) = \sum_{Z_2} M(Y_2, Z_2)\, f(Z_1, Z_2)\, f(Z_2, Z_3)$$



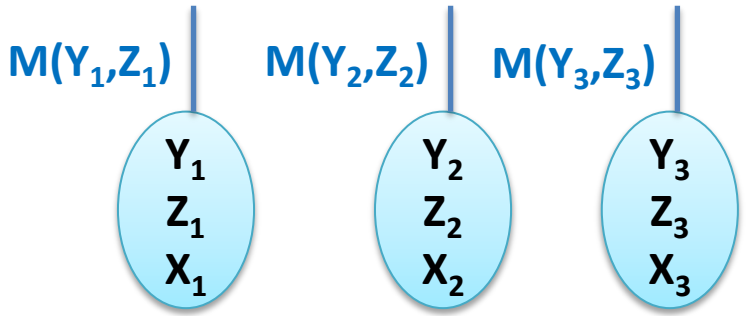
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



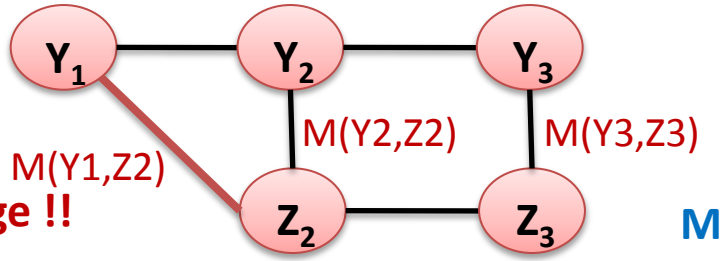
$$M(Y_1, Z_2) = \sum_{Z_1} M(Y_1, Z_1) f(Z_1, Z_2)$$



Example: Factorial HMM

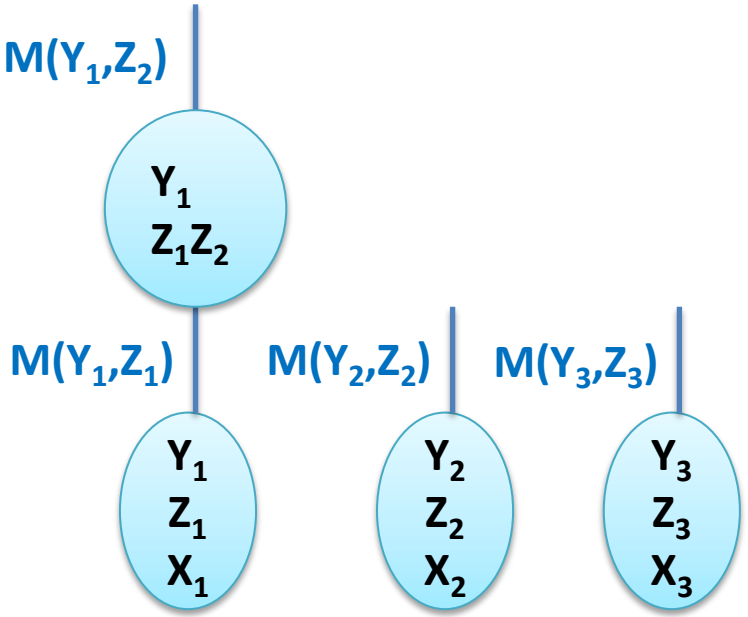
Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



Intro. 1 edge !!
(the best)

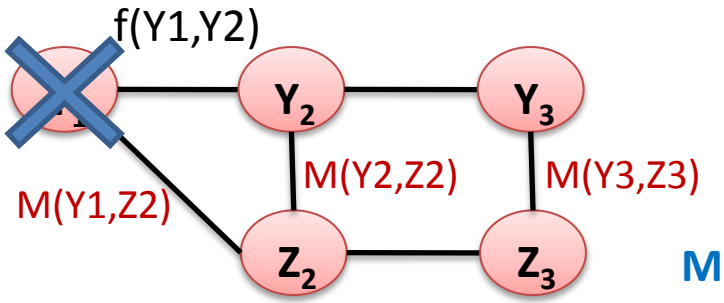
$$M(Y_1, Z_2) = \sum_{Z_1} M(Y_1, Z_1) f(Z_1, Z_2)$$



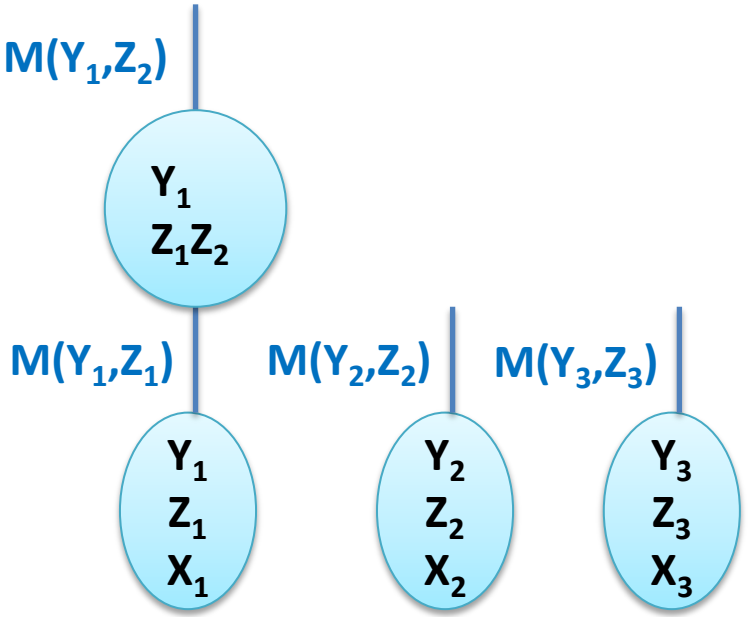
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



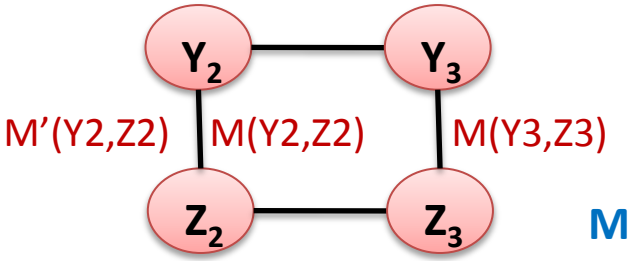
$$M'(Y_2, Z_2) = \sum_{Y_1} M(Y_1, Z_2)\, f(Y_1, Y_2)$$



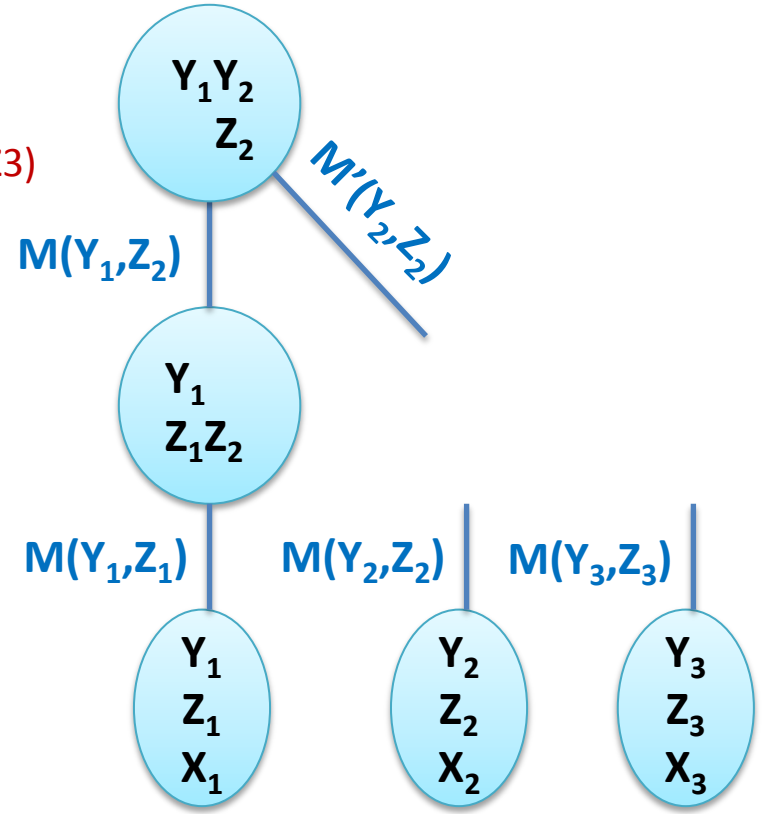
Example: Factorial HMM

Finding Elimination Order:
Find elimination adding as fewer edges as possible. (greedily)

Intro. 0 edge !!
(the best)



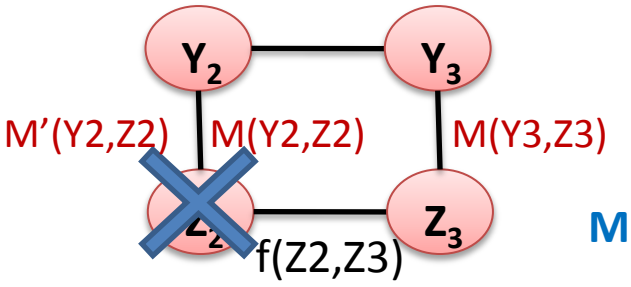
$$M'(Y_2, Z_2) = \sum_{Y_1} M(Y_1, Z_2)\, f(Y_1, Y_2)$$



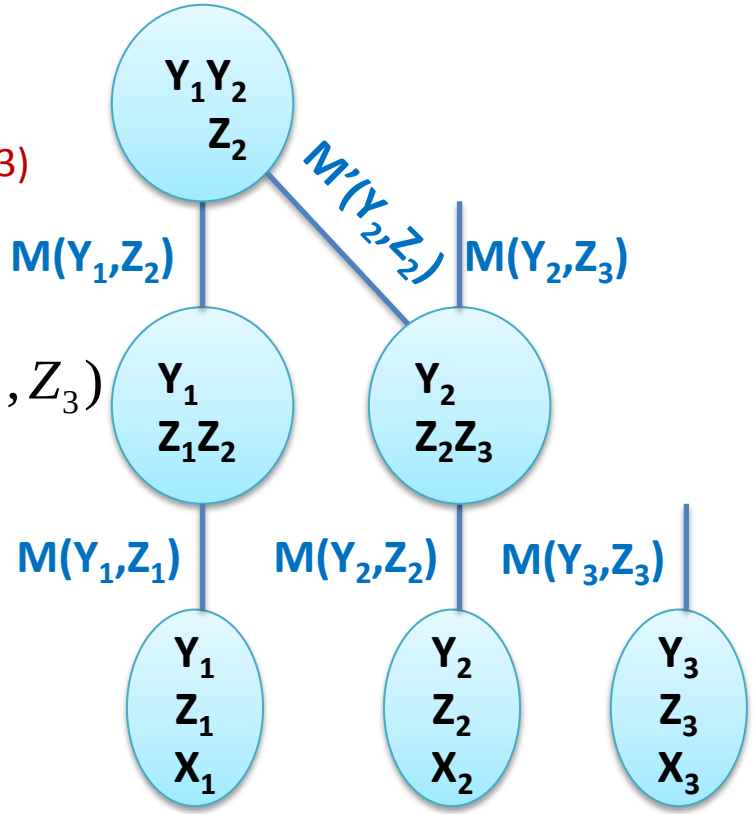
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



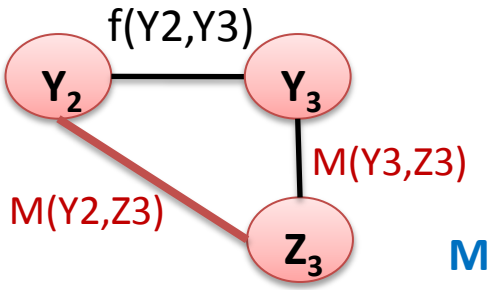
$$M(Y_2, Z_3) = \sum_{Z_2} M'(Y_2, Z_2) M(Y_2, Z_2) f(Z_2, Z_3)$$



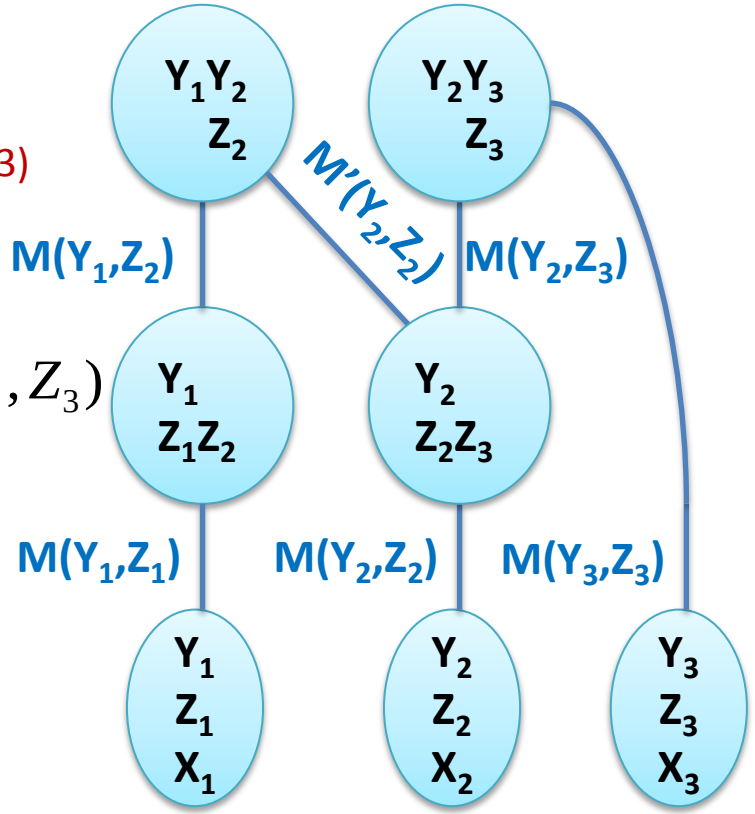
Example: Factorial HMM

Finding Elimination Order:

Find elimination adding as fewer edges as possible. (greedily)



$$M(Y_2, Z_3) = \sum_{Z_2} M'(Y_2, Z_2) M(Y_2, Z_2) f(Z_2, Z_3)$$



Example: Factorial HMM

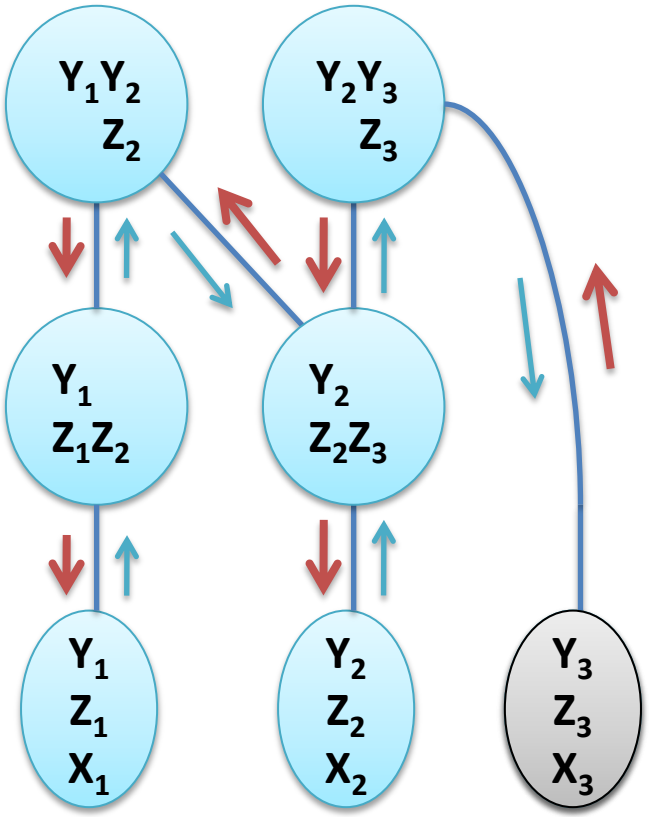
After building a clique tree, we can run "2 passes" on the tree to get all messages M(.) needed for computing marginal.

1st pass:
Take a node as root.
Run Sum-Product from leaves to root.

2nd pass:
Run Sum-Product from root to leaves.

All marginal dist. can be derived from

1. Multiply all M(.) from neighbors by f(.) on this node.
2. Eliminate unwanted variable.



Example: Factorial HMM

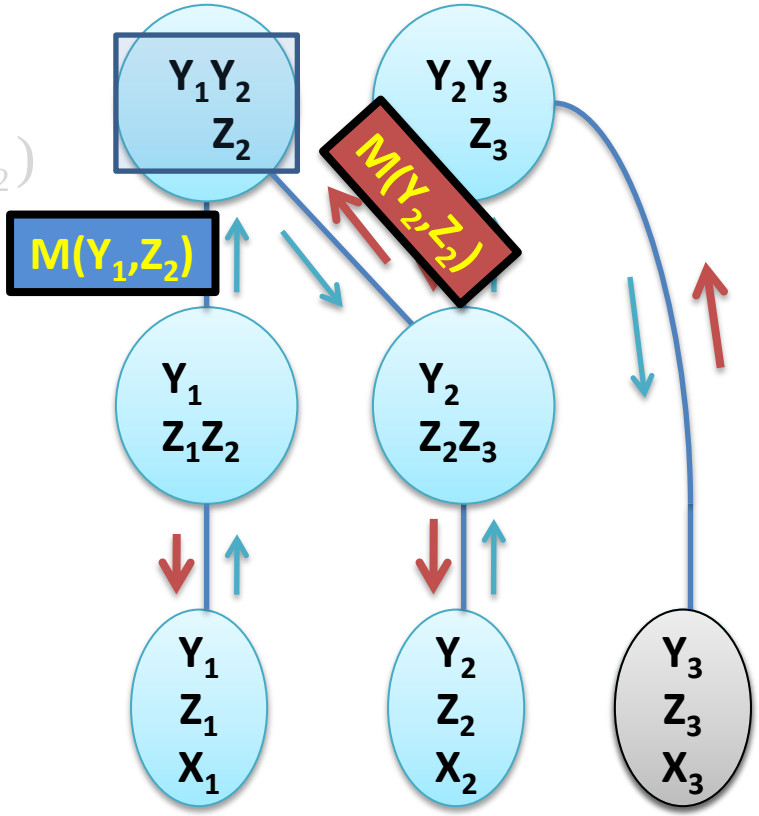
After building a clique tree, we can run "2 passes" on the tree to get all messages M(.) needed for computing marginal.

Assume we want : $P(Y_1,Y_2)$

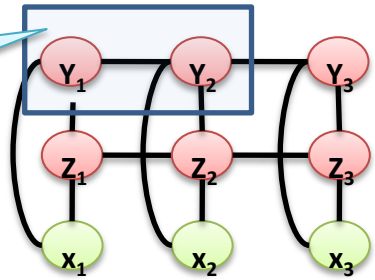
belief (or maginal dist.)

$$b(Y_1,Y_2,Z_2) = M(Y_1,Z_2)\, f(Y_1,Y_2,Z_2) M(Y_2,Z_2)$$

$$P(Y_1,Y_2) = \sum_{Z_2} b(Y_1,Y_2,Z_2)$$



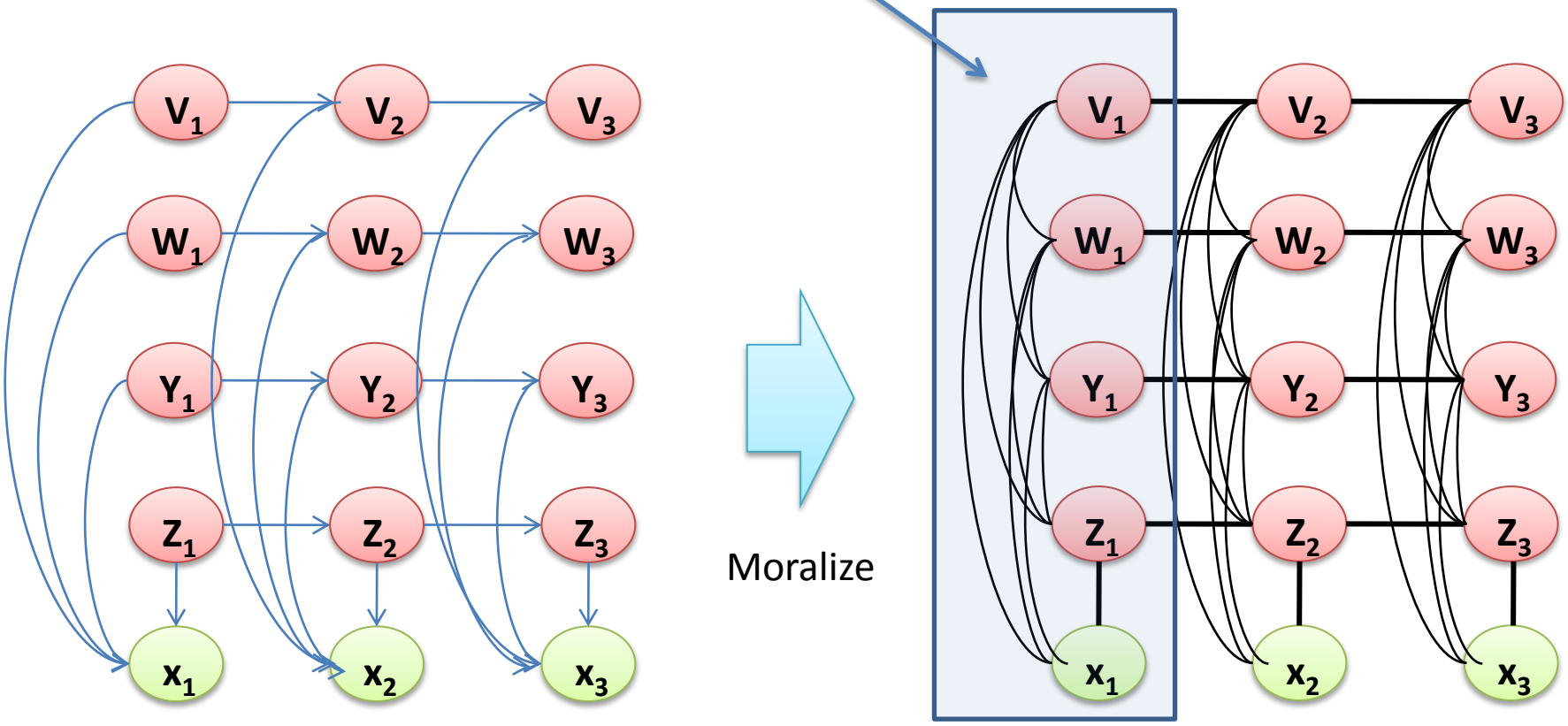
Y_1, Y_2 are **dependent.**



➔There **must be a node** in clique tree containing (Y_1,Y_2).

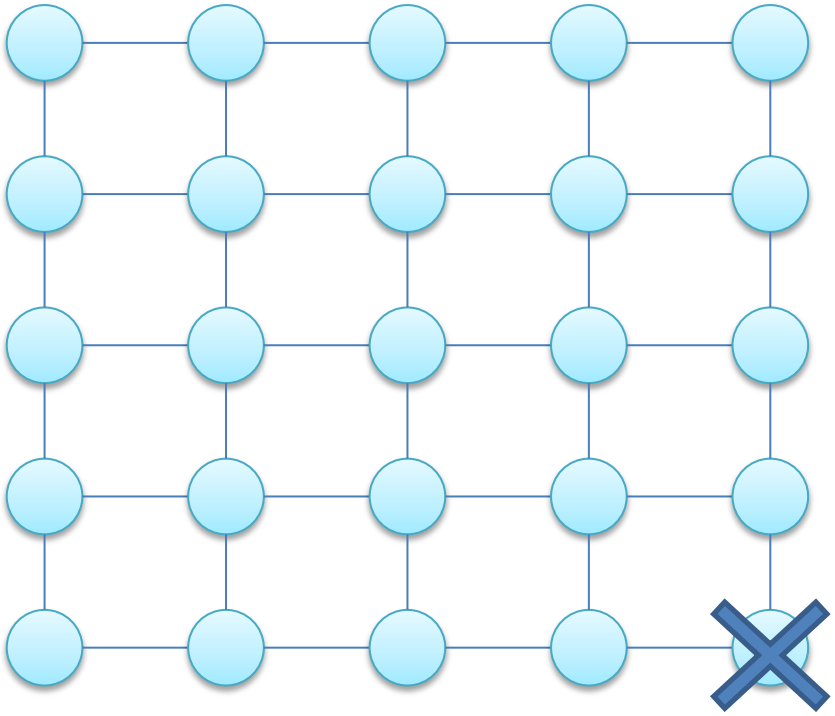
Example : General Factorial HMM

A clique size=5,
intractable most of times.
(No tractable elimination exist…)



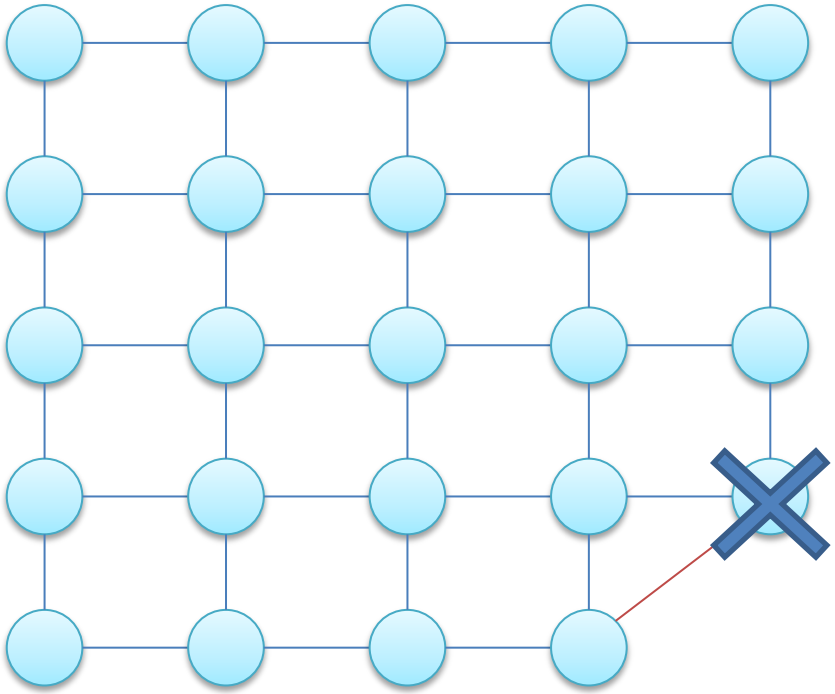
Some Model are Intractable for Exact Inference

Example: A Grid MRF



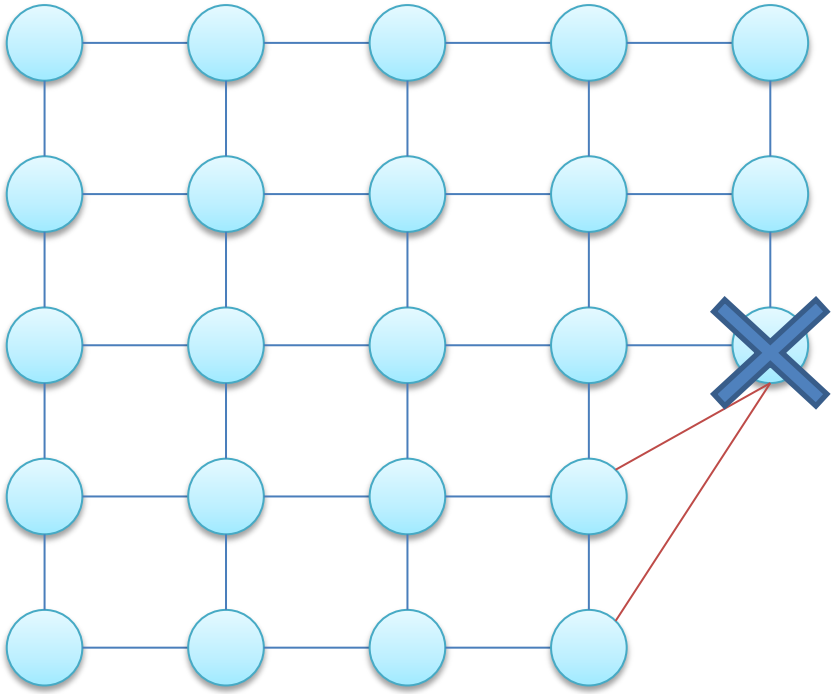
Some Model are Intractable for Exact Inference

Example: A Grid MRF



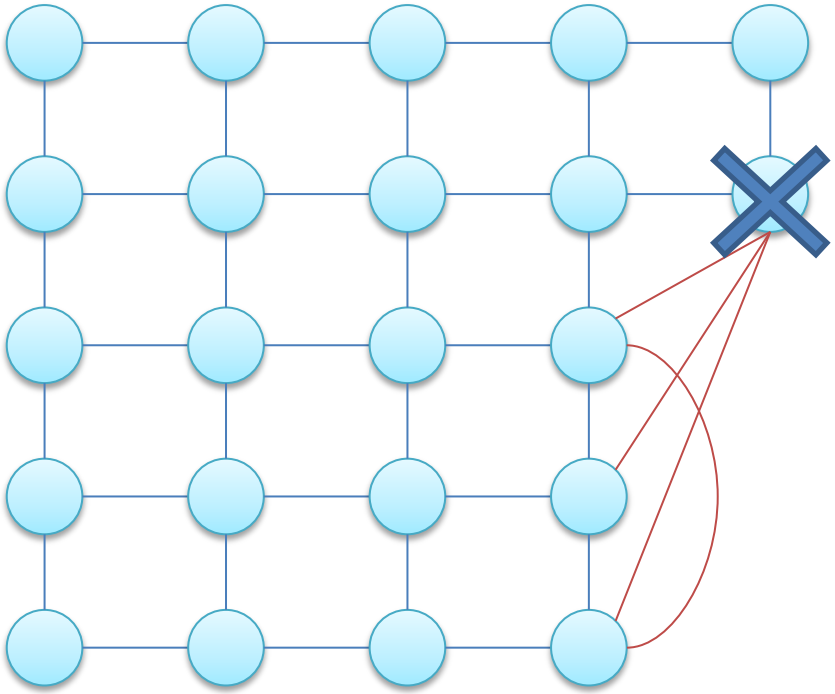
Some Model are Intractable for Exact Inference

Example: A Grid MRF



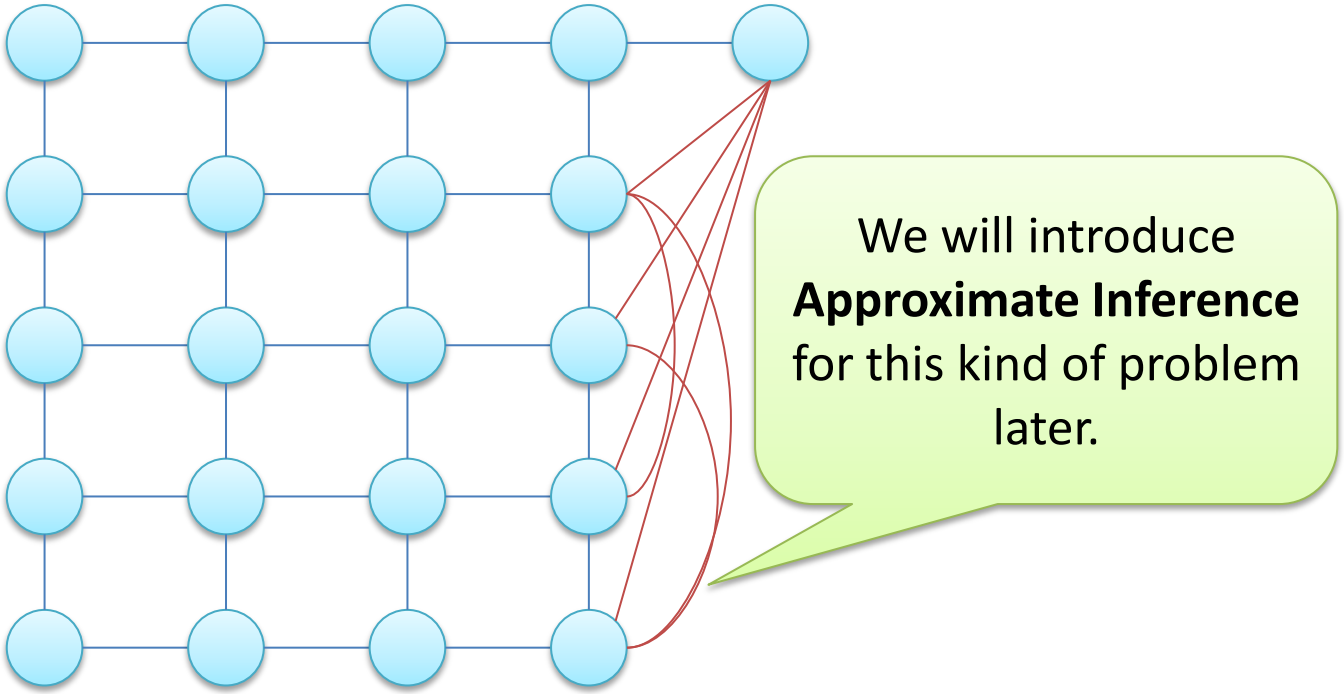
Some Model are Intractable for Exact Inference

Example: A Grid MRF



Some Model are Intractable for Exact Inference

Example: A Grid MRF

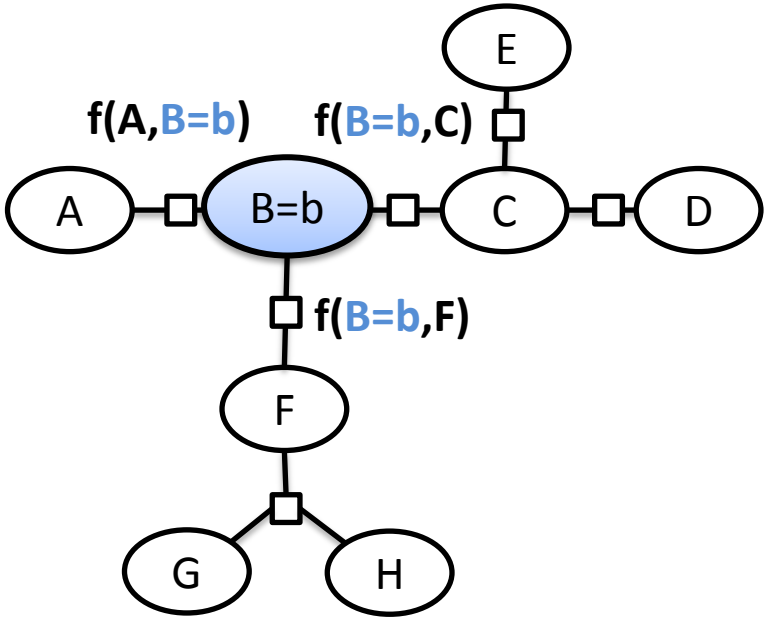


Generally, we will have clique of "size N"
for a N*N grid, which is indeed intractable.

Variable Elimination: Dealing with Evidence

What if some variables $X=\{X_1 \ldots X_D\}$ are given in Evidence :

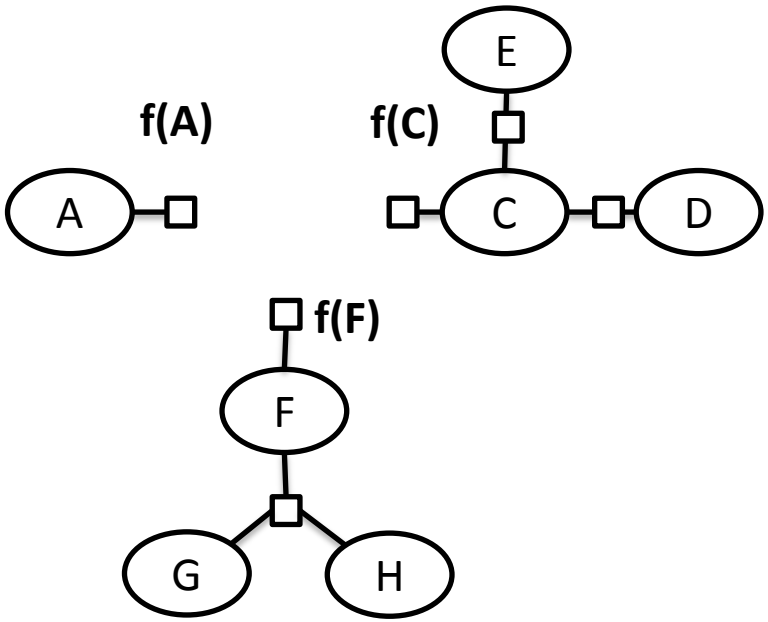
Given Evidence { B=b } :



Variable Elimination: Dealing with Evidence

What if some variables $X=\{X_1 \ldots X_D\}$ are given in Evidence :

Given Evidence { B=b } :



A model with evidence equivalent to another model without evidence.

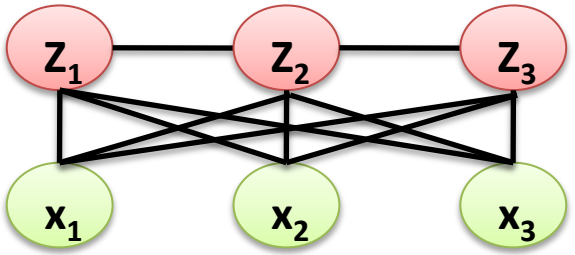
To infer $P_M(Z|X)$, we transform M to another model M' and infer $P_{M'}(Z)$.

Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(X,Z)=\frac{1}{Z}\, f(Z_1,X_1)\, f(Z_1,X_2)\, f(Z_1,X_3)$$
$$f(Z_2,X_1)\, f(Z_2,X_2)\, f(Z_2,X_3)$$
$$f(Z_3,X_1)\, f(Z_3,X_2)\, f(Z_3,X_3)$$
$$f(Z_1,Z_2)\, f(Z_2,Z_3)$$



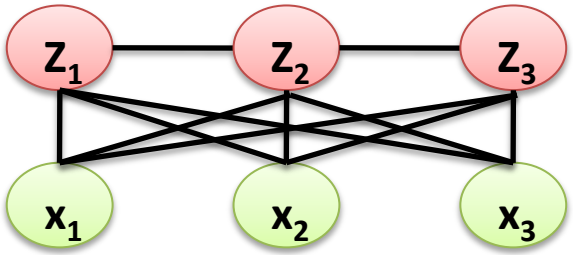
But given X1~X3, we actually run inference on another model M'.

Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(X = x, Z) = \frac{1}{Z} f_x(Z_1) f_x'(Z_1) f_x''(Z_1)$$
$$f_x(Z_2) f_x'(Z_2) f_x''(Z_2)$$
$$f_x(Z_3) f_x'(Z_3) f_x''(Z_3)$$
$$f(Z_1, Z_2) f(Z_2, Z_3)$$



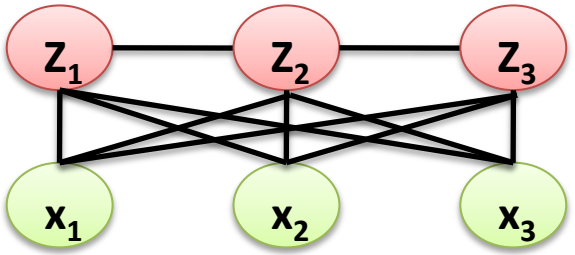
But given X1~X3, we actually run inference on another model M'.

Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(X = x, Z) = \frac{1}{Z} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)$$



But given X1~X3, we actually run inference on another model M'.

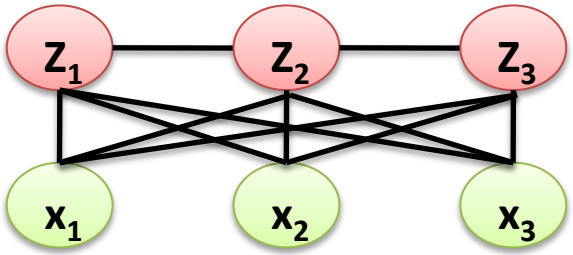
Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(Z \mid X = x) = \frac{P(X = x, Z)}{P(X = x)} = \frac{\frac{1}{Z} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}{\frac{1}{Z} \sum_{Z1} \sum_{Z2} \sum_{Z3} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}$$

$$= \frac{1}{Z'(x)} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)$$



But given X1~X3, we actually run inference on another model M'.

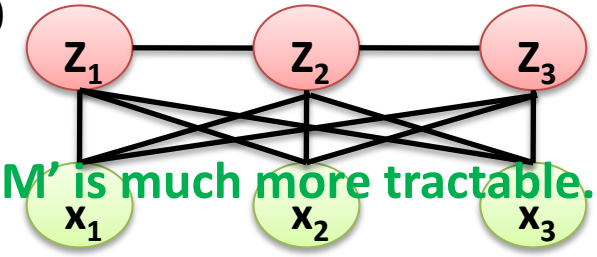
Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(Z \mid X = x) = \frac{P(X = x, Z)}{P(X = x)} = \frac{\frac{1}{Z} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}{\frac{1}{Z} \sum_{Z1} \sum_{Z2} \sum_{Z3} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}$$

$$= \frac{1}{Z'(x)} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)$$



But given X1~X3, we actually run inference on another model M'.

Variable Elimination: Dealing with Evidence

If we can know "which variables will be given", then a intractable model will become a tractable one.

Sometimes we want capture more dependency in a model, which induce intractable inference.

$$P(Z \mid X = x) = \frac{P(X = x, Z)}{P(X = x)} = \frac{\frac{1}{Z} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}{\frac{1}{Z} \sum_{Z1} \sum_{Z2} \sum_{Z3} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)}$$

$$= \frac{1}{Z'(x)} f_x'(Z_1, Z_2) f_x'(Z_2, Z_3)$$

M' is much more tractable.

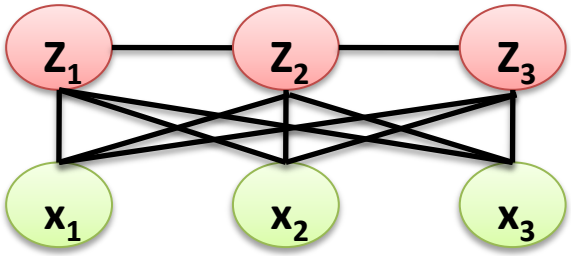
New normalize const. can be computed Using VE.

But given X1~X3, we actually run inference on another model M'.

Variable Elimination: Dealing with Evidence

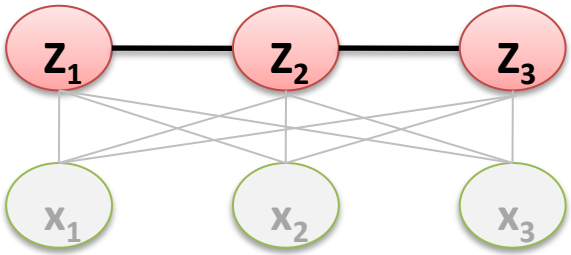
Even if P(Z|X) can be inferred efficiently, "learning P(X,Z)" is intractable.
One solution is model P(Z|X) directly , yielding "CRF" model.

$$P(X,Z) = \frac{1}{Z} f(Z_1,X_1) f(Z_1,X_2) f(Z_1,X_3)$$
$$f(Z_2,X_1) f(Z_2,X_2) f(Z_2,X_3)$$
$$f(Z_3,X_1) f(Z_3,X_2) f(Z_3,X_3)$$
$$f(Z_1,Z_2) f(Z_2,Z_3)$$



Intractable MRF model

$$P(Z \mid X) = \frac{1}{Z'(X)} f_X'(Z_1,Z_2) f_X'(Z_2,Z_3)$$



Tractable CRF Model

Agenda

- Introduce the concept of “Variable Elimination” in special case of Tree-structured Factor Graph.
- Extend the idea of “VE” to General Factor Graph with concept of “Clique Tree”.
- **See how to extend “VE” to “Most Probable Assignment” (MAP configuration) Problem.**

Query 3: Most Probable Assignment

- Given Evidence E = $\{X_1=x_1,\ldots, X_D=x_D\}$ and some other variables $Z=\{Z_1, \ldots, Z_k\}$ unspecified , Most Probable Assignment of Z is given by:

$$MPA(Z \mid X) = \arg\max_Z \; P(Z \mid X)$$

$$= \arg\max_Z \; \frac{P(X \mid Z)P(Z)}{\mathrm{P(X)}} = \arg\max_Z \; P(X \mid Z)P(Z)$$

$$\operatorname*{argmax}_Z \; \mathrm{P(Z \mid X)} \neq \begin{cases} \arg\max_{Z1} \; P(Z_1 \mid X) \\ \ldots\ldots \\ \arg\max_{ZK} \; P(Z_K \mid X) \end{cases}$$

What's the different ?

MPA Goal:

$$\max_{Z} P(Z \mid X) = \max_{Z_1} \ldots \max_{Z_K} P(Z_1 \ldots Z_K \mid X)$$

Likelihood Goal::

(Solved using VE) $$P(X) = \sum_{Z_1} \ldots \sum_{Z_K} P(Z_1 \ldots Z_K, X)$$

Exploring the similarity between "**max**" & "**∑**" is the key to solve MPA using VE.

What's the different ?

Review:

$$P(E=e)=\sum_{D}\sum_{C}\sum_{B}\sum_{A}P(E=e\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

$$=\sum_{D}P(E\mid D)\sum_{C}P(D\mid C)\sum_{B}P(C\mid B)\sum_{A}P(B\mid A)P(A)$$

M(B) : marginal of B

$$\max_{A,B,C,D}P(A,B,C,D,E=e)$$

$$=\max_{D}\max_{C}\max_{B}\max_{A}P(E=e\mid D)P(D\mid C)P(C\mid B)P(B\mid A)P(A)$$

$$=\max_{D}P(E=e\mid D)\max_{C}P(D\mid C)\max_{B}P(C\mid B)\max_{A}P(B\mid A)P(A)$$

M'(B) : ???

M(B) = $\max_A$ F(A,B) : maxMarginal of B

$$M(B) = \max_A F(A,B)$$

For every choice of B, we decide an A*(B)=argmax_A F(A,B) with M(B) = F(A*(B), B).

F(A,B)	b1	b2	b3
a1	1	3	9
a2	2	5	8
a3	4	7	6

M(B) = $\max_A$ F(A,B) : maxMarginal of B

$$M(B) = \max_A F(A,B)$$

For every choice of B, we decide an A*(B)=argmax_A F(A,B) with M(B) = F(A*(B), B).

F(A,B)	b1	b2	b3
a1	1	3	9
a2	2	5	8
a3	4	7	6

B	b1	b2	b3
A*(B)	a3		

B	b1	b2	b3
M(B)	4		

$M(B) = \max_A F(A,B)$: maxMarginal of B

$$M(B) = \max_A F(A,B)$$

For every choice of B, we decide an $A^*(B)=\text{argmax}_A F(A,B)$ with $M(B) = F(A^*(B), B)$.

F(A,B)	b1	b2	b3
a1	1	3	9
a2	2	5	8
a3	4	7	6

B	b1	b2	b3
A*(B)	a3	a3	

B	b1	b2	b3
M(B)	4	7	

$M(B) = \max_A F(A,B)$: maxMarginal of B

$$M(B) = \max_A F(A,B)$$

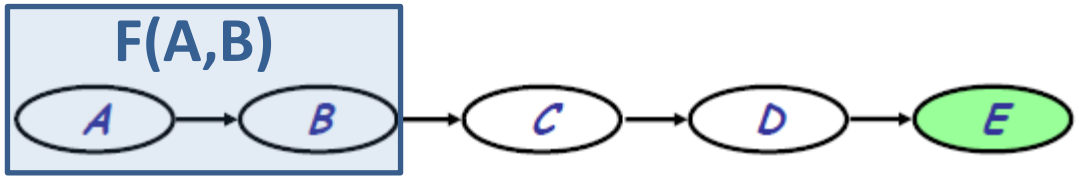
For every choice of B, we decide an $A^*(B)=\text{argmax}_A F(A,B)$ with $M(B) = F(A^*(B), B)$.

F(A,B)	b1	b2	b3
a1	1	3	9
a2	2	5	8
a3	4	7	6

B	b1	b2	b3
A*(B)	a3	a3	a1

B	b1	b2	b3
M(B)	4	7	9

Most Probable Assignment on a Chain



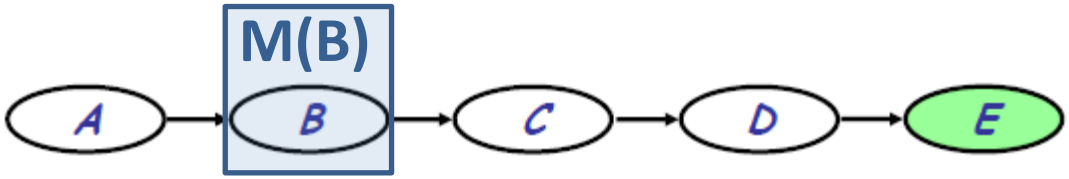
$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} \underbrace{P(B \mid A)P(A)}_{F(A,B)}$$

F(A,B)	a1	a2	a3
b1	…	…	…
b2	…	…	…
b3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

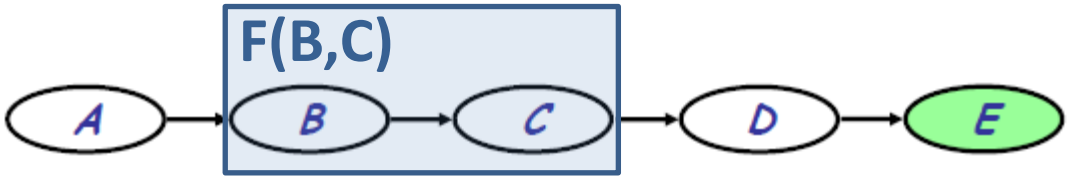
$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

$M(B) = \max_A F(A,B)$

B	A*(B)	M(B)
b1	a1	…
b2	a3	…
b3	a2	…

F(A,B)	a1	a2	a3
b1	…	…	…
b2	…	…	…
b3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

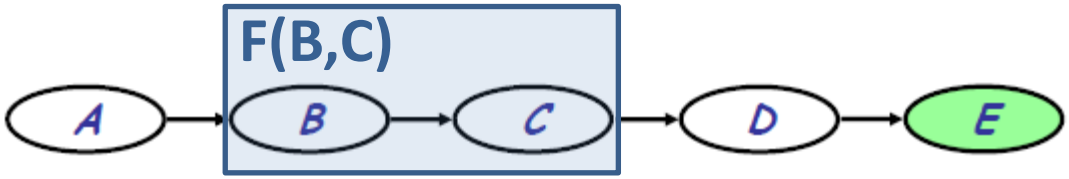
$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \underbrace{\max_{A} P(B \mid A)P(A)}_{F(B,C)=P(C|B)M(B)}$$

P(C|B)	b1	b2	b3
c1	…	…	…
c2	…	…	…
c3	…	…	…

B	A*(B)	M(B)
b1	a1	…
b2	a3	…
b3	a2	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

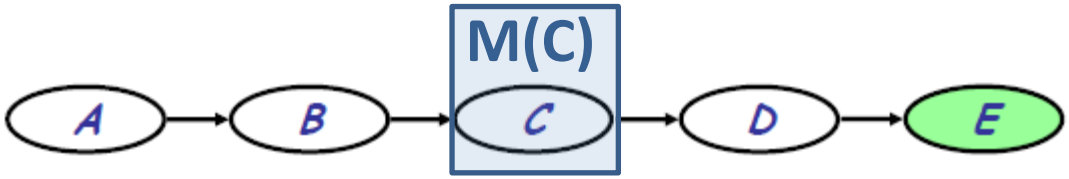
$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

$$\underbrace{\qquad\qquad\qquad\qquad}_{F(B,C)=P(C|B)M(B)}$$

F(B,C)	b1	b2	b3
c1	…	…	…
c2	…	…	…
c3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

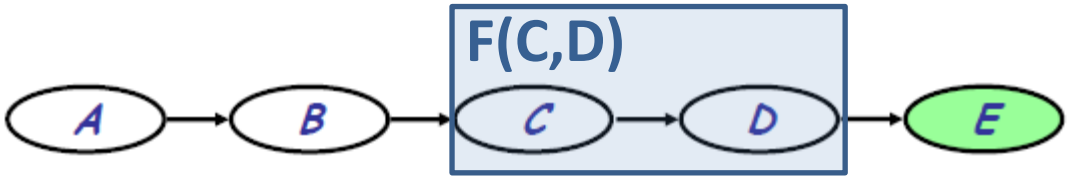
$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$$\mathbf{M(C) = \max_B F(B,C)}$$

C	B*(C)	M(C)
c1	b3	…
c2	b1	…
c3	b2	…

F(B,C)	b1	b2	b3
c1	…	…	…
c2	…	…	…
c3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

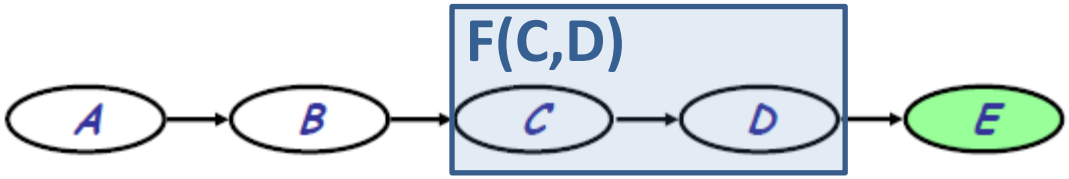
$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

F(C,D) = P(D|C)M(C)

P(D|C)	c1	c2	c3
d1	…	…	…
d2	…	…	…
d3	…	…	…

C	B*(C)	M(C)
c1	b3	…
c2	b1	…
c3	b2	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

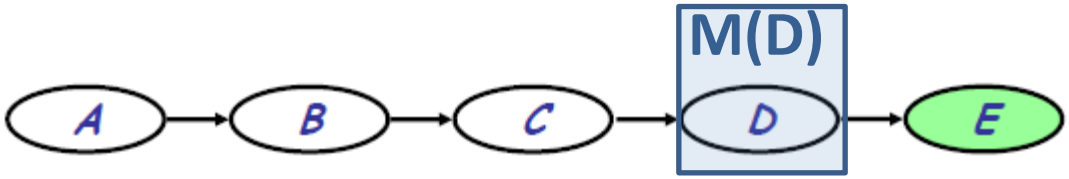
$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

F(C,D) = P(D|C)M(C)

F(C,D)	c1	c2	c3
d1	…	…	…
d2	…	…	…
d3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

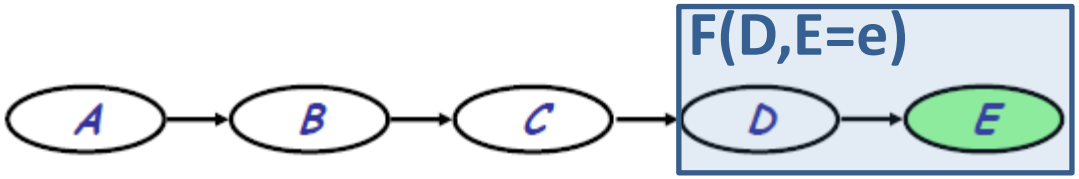
$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

$$M(D)=\max_{C} F(C,D)$$

D	C*(D)	M(D)
d1	c1	…
d2	c2	…
d3	c3	…

F(C,D)	c1	c2	c3
d1	…	…	…
d2	…	…	…
d3	…	…	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

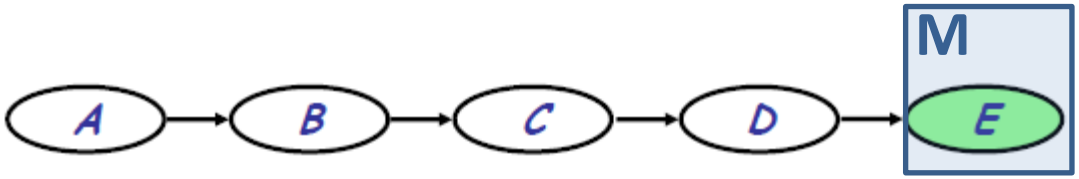
$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

F(D)=P(E=e|D)M(D)

P(E=e|D)	d1	d2	d3
e	…	…	…

D	C*(D)	M(D)
d1	c1	…
d2	c2	…
d3	c3	…

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

M= max$_D$ F(D)

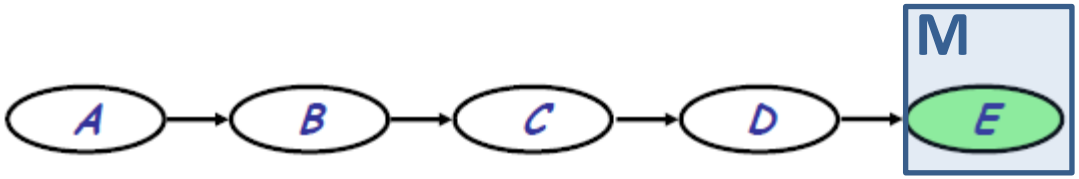
D*	M
d2	…

F(D,E=e)	d1	d2	d3
e	…	…	…

What we get ? ➔ **M = max$_{ABCD}$ P(A,B,C,D,E=e)**

What we want ? ➔ **(A*,B*,C*,D*) = argmax$_{ABCD}$ P(A,B,C,D,E=e)**

Most Probable Assignment on a Chain



$$\max_{A,B,C,D} P(A,B,C,D,E=e)$$

$$= \max_{D} \max_{C} \max_{B} \max_{A} P(E=e \mid D)P(D \mid C)P(C \mid B)P(B \mid A)P(A)$$

$$= \max_{D} P(E=e \mid D) \max_{C} P(D \mid C) \max_{B} P(C \mid B) \max_{A} P(B \mid A)P(A)$$

What we want ? ➔ (A*,B*,C*,D*) = argmax$_{ABCD}$ P(A,B,C,D,E=e)

(a1 , b1, c2, d2)

D*	M
d2	…

D	C*(D)	M(D)
d1	c1	…
d2	c2	…
d3	c3	…

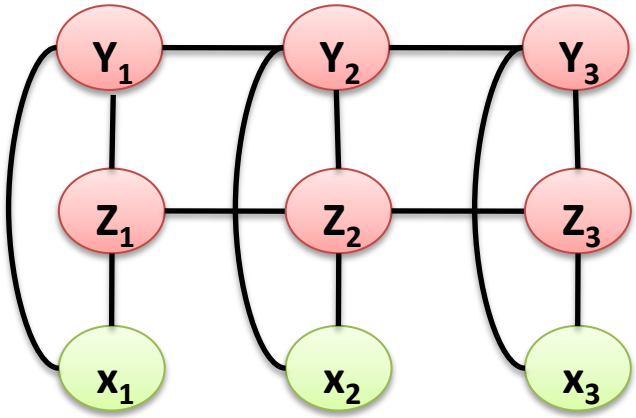
C	B*(C)	M(C)
c1	b3	…
c2	b1	…
c3	b2	…

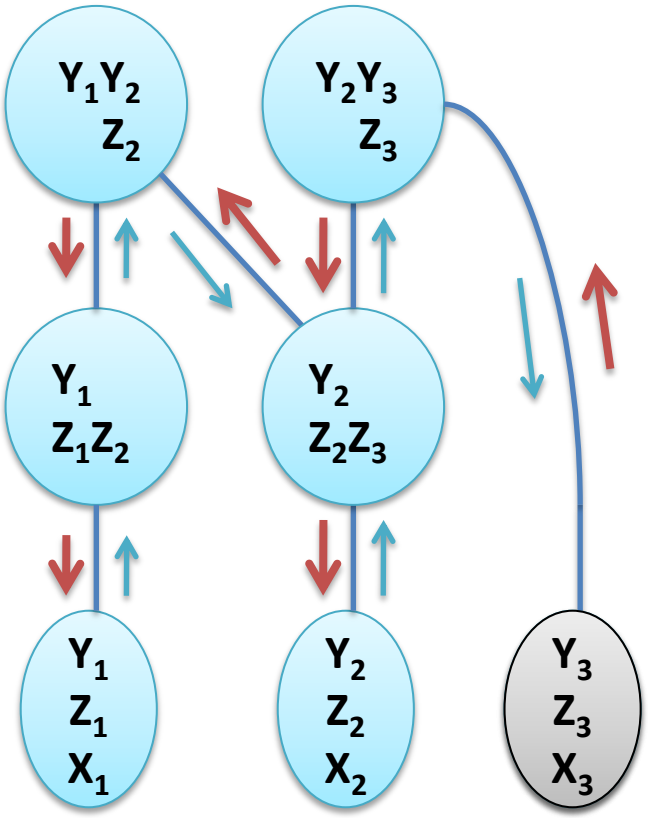
B	A*(B)	M(B)
b1	a1	…
b2	a3	…
b3	a2	…

Most Probable Assignment on general Graph

It's straight forward to generalize algorithm above to case of **general graph** with similarity of "**∑**" and "**max**".

(The difference is there must be a "traceback" procedure to find the "argmax" after we get "max".)





Summary

- To solve inference problems like "likelihood of X", "P(Z|X)", "Most Probable Assignment", we can use Variable Elimination (e.g. Sum-Product) algorithm

- In case of tree-structured factor graph, we just run "2 passes" of VE from leaves to a root & the reverse.

- In case of general-structured graph, we must find a "good" elimination order inducing smallest "maximum clique", which is often done with greedy method.

- When we know which variables will be given in advance, we can derive much easier model M' from original M with evidence, which is more tractable in Inference & Learning.