PROBABILISTIC REASONING

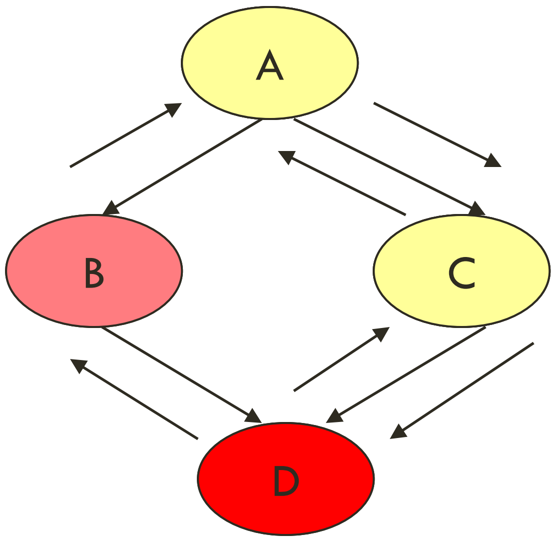
Unit # 09

INFERENCE IN MULTIPLY CONNECTED NETWORKS

We get evidence on node D.

As a result we need to update the marginal probabilities of the rest of the variables

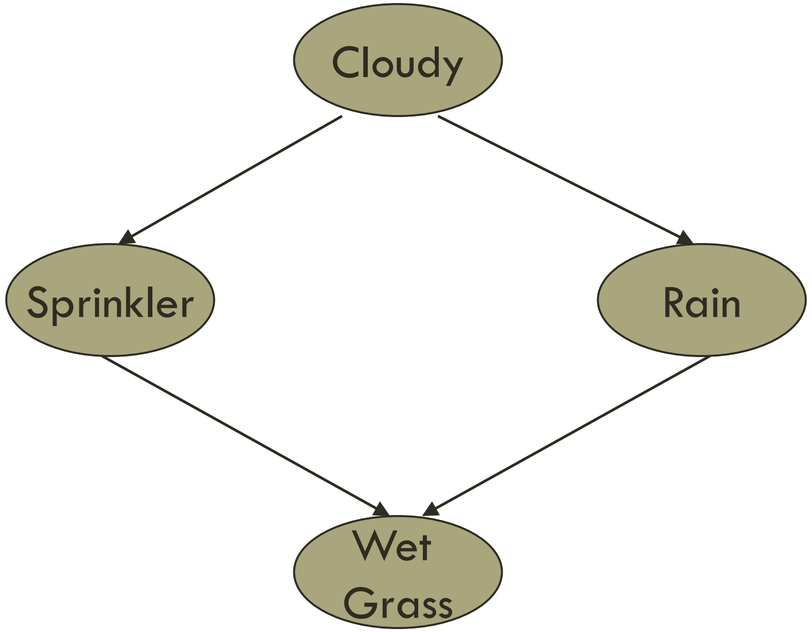
How to do it?

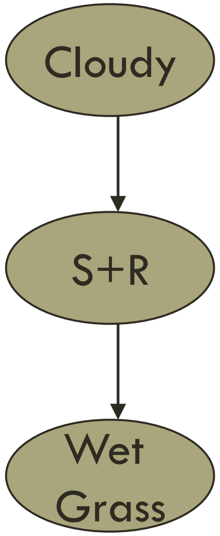


CLUSTERING

In clustering, we transform a multiply-connected network into a probabilistically equivalent poly-tree by merging nodes, removing the multiple paths between two nodes.

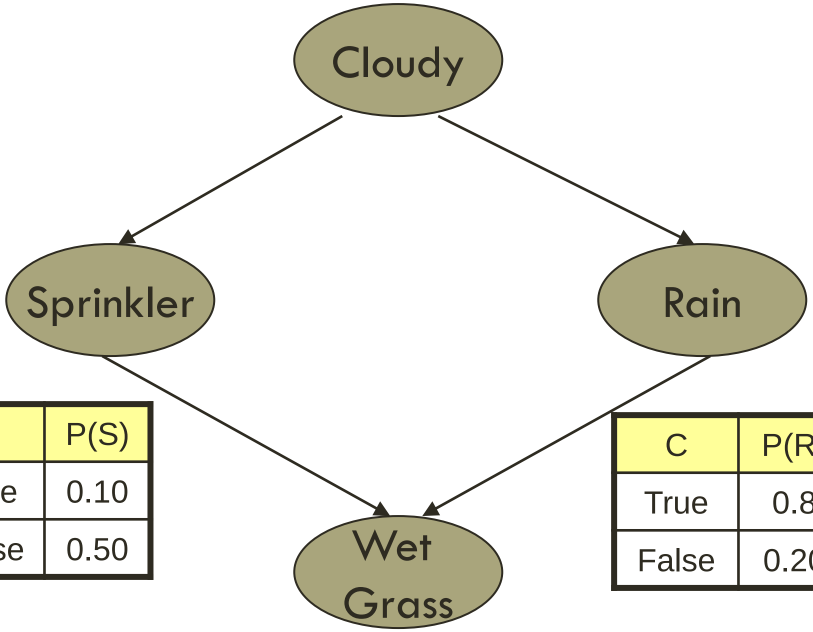
CLUSTERING





CLUSTERING

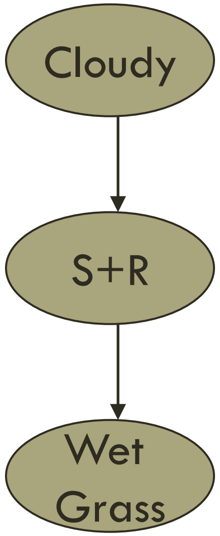
P(C)=0.5



C	P(S)
True	0.10
False	0.50

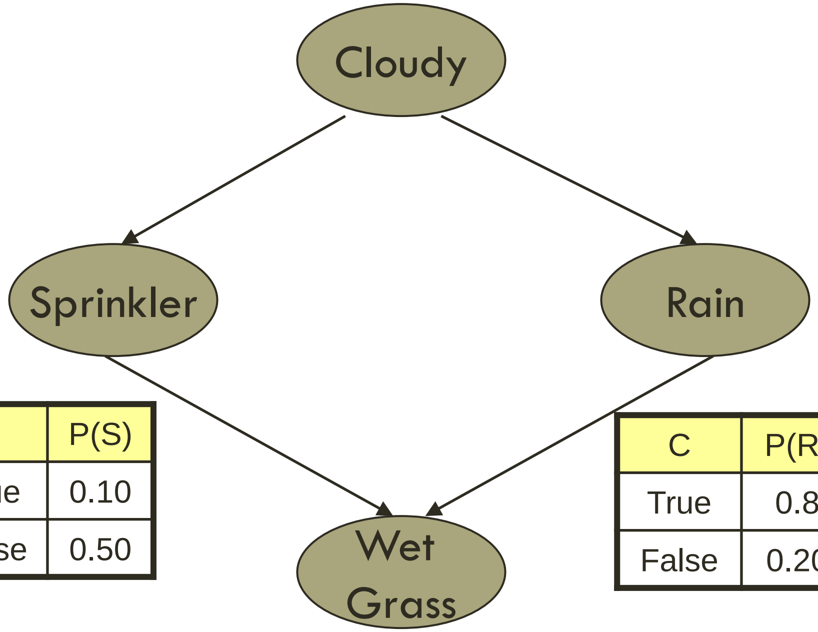
C	P(R)
True	0.8
False	0.20

S	R	P(W)
True	True	0.9
True	False	0.8
False	True	0.4
False	False	0.1



CLUSTERING

P(C)=0.5

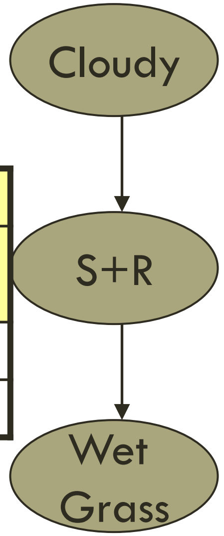


C	P(S)
True	0.10
False	0.50

C	P(R)
True	0.8
False	0.20

S	R	P(W)
True	True	0.9
True	False	0.8
False	True	0.4
False	False	0.1

P(C)=0.5



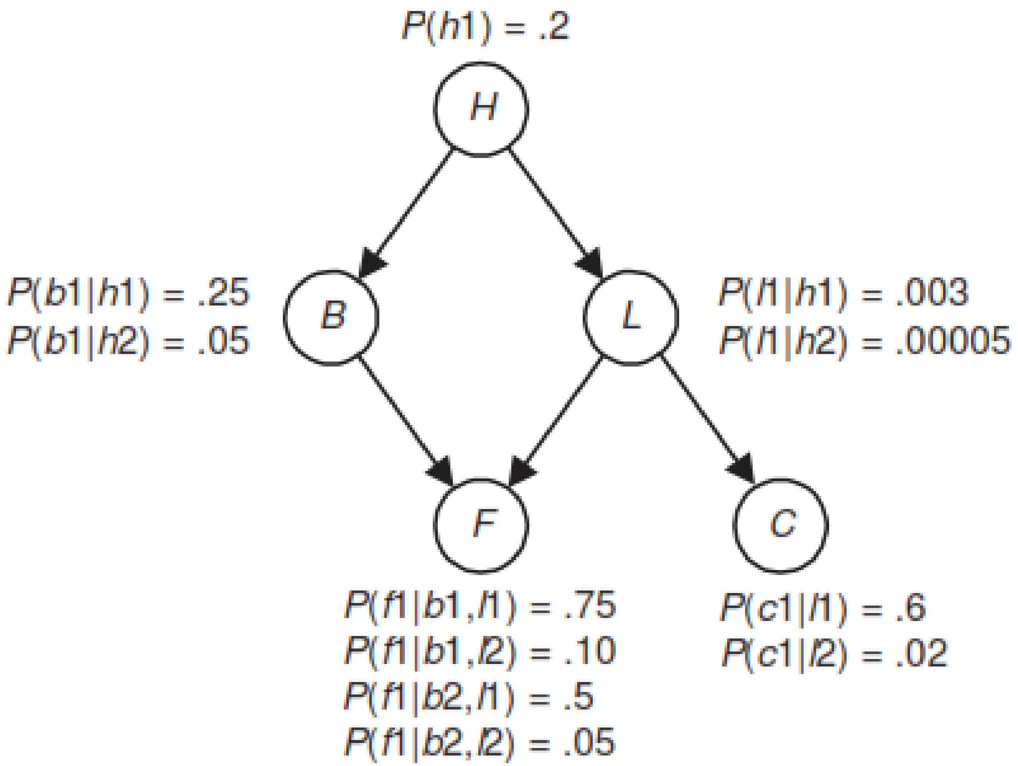
	S+R			
C	True True	True False	False True	False False
True	0.08	0.02	0.72	0.18
False	0.10	0.40	0.10	0.40

S+R	P(W)
True True	0.9
True False	0.8
False True	0.4
False False	0.1

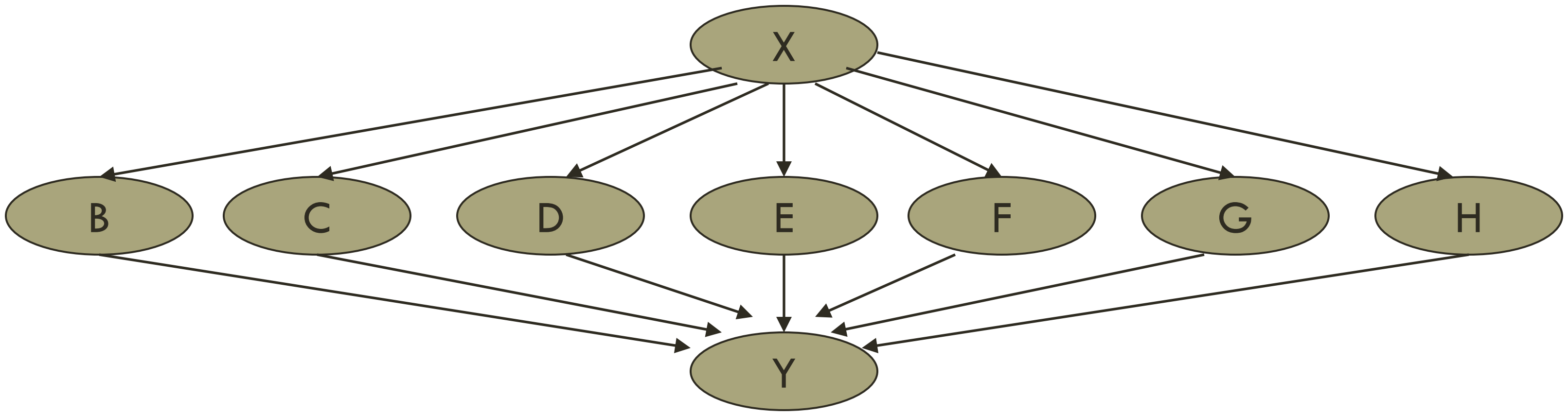
DEMO OF CLUSTERING IN GENIE

EXERCISE

Convert this graph into a singly-connected network via clustering.



PROBLEM WITH THE CLUSTERING APPROACH



We are back to the same number of computations as was in the enumeration approach

JUNCTION/CLIQUE TREE ALGORITHM

The junction tree method is based on a transformation of the BN to a junction tree, where each node in this tree is a group or cluster of variables from the original network.

The graph of clusters is singly connected and has the *junction tree property* (this property ensures that evidence propagates correctly)

The junction tree becomes a permanent part of the knowledge representation, and changes only if the graph changes

Beliefs are propagated in the junction tree using a local message-passing algorithm

HOW TO CONSTRUCT A JT

Moralize the graph by adding links between parents of common children.

Convert the given BN into an undirected graph by removing the arrowheads.

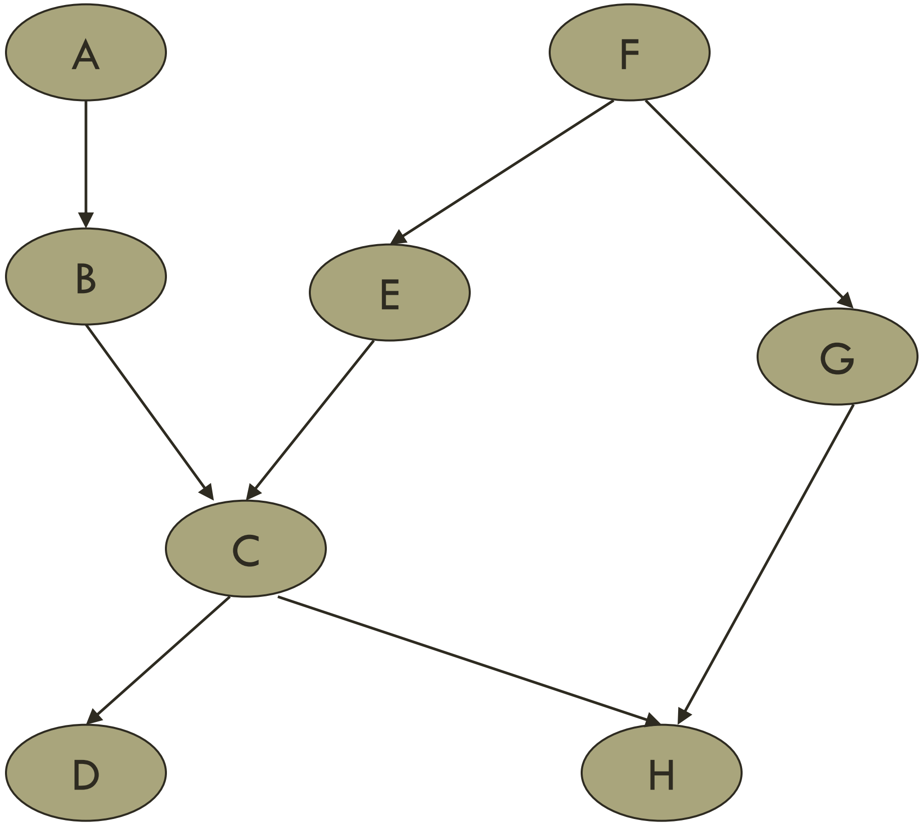
Triangulate the graph.

Order the nodes by maximum cardinality search.

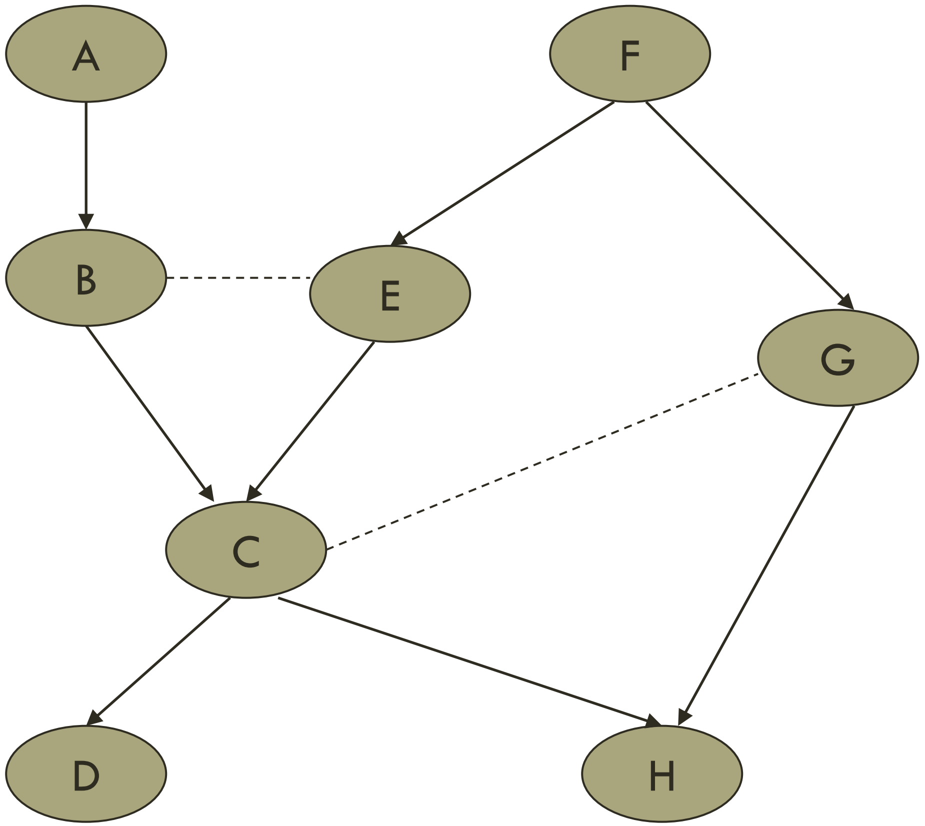
Find the cliques of the triangulated graph.

Arrange as a junction tree.

EXAMPLE I (LASKEY)

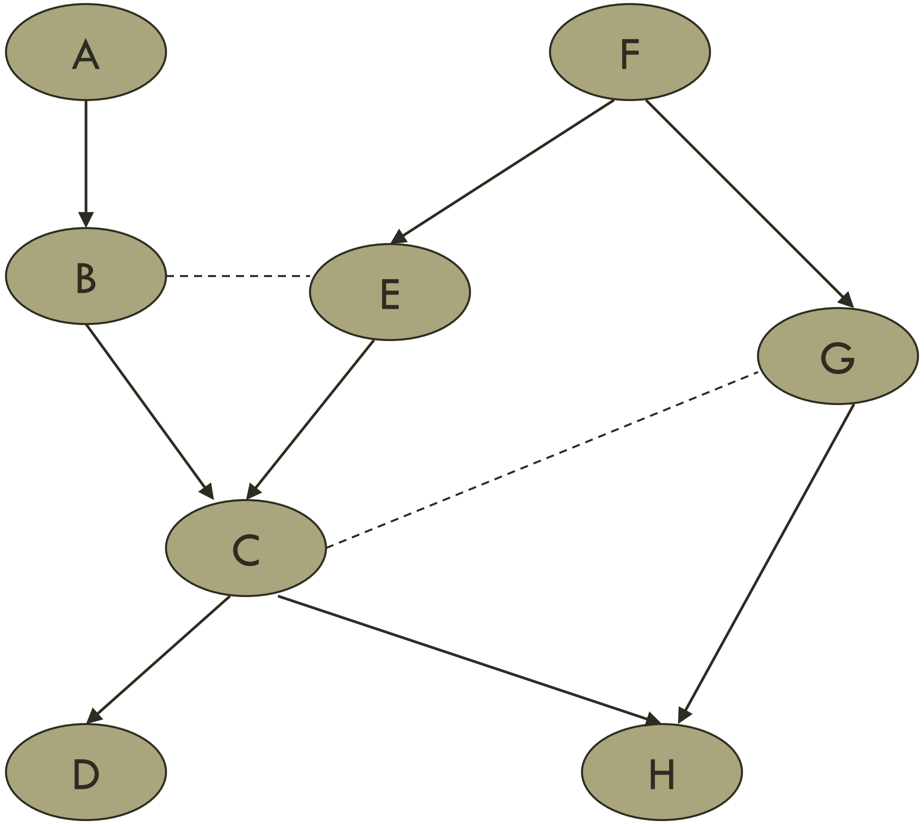


Example Bayesian Network

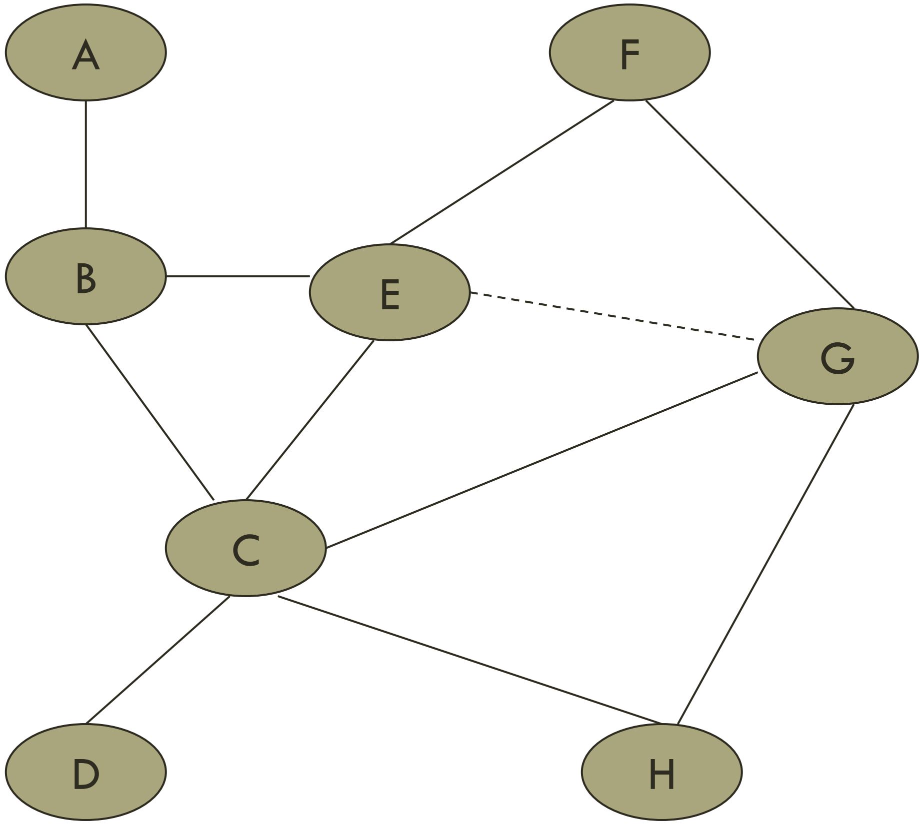


Moralize the graph

EXAMPLE I (CONT'D)

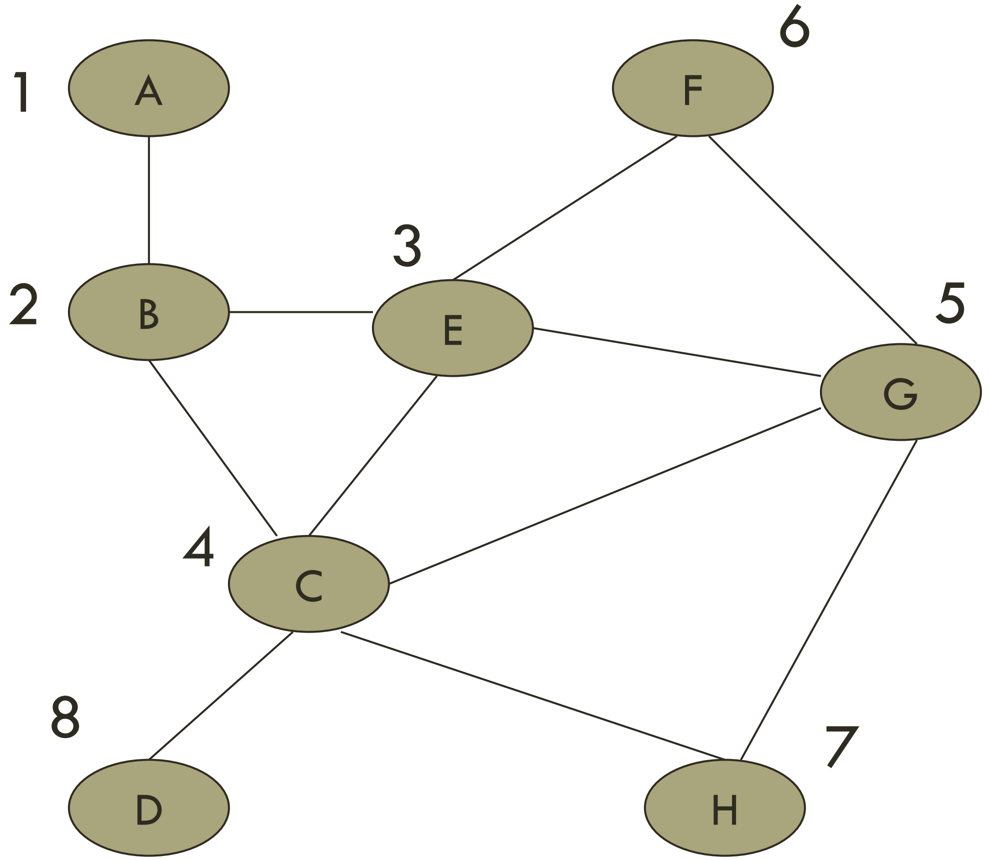


Moralize the graph



Remove Directions and
Triangulate the graph

EXAMPLE I (CONT'D)



Order Nodes in the Graph

Clique	Max. Vertex
AB	2
BEC	4
ECG	5
EGF	6
CGH	7
CD	8

Find Cliques of the Triangulated Graph

MAXIMUM CARDINALITY SEARCH

- The maximum cardinality search algorithm works as follows: Initialize $W \leftarrow V$ where $V = V(G)$, and set weight$(v) = 0$ for all $v \in V$. For each $i = 1, \ldots n$, let u be a node with maximal weight in W, set $v_i \leftarrow u$, and increment the weight of all neighbors of u in W by one. Then remove u from W and repeat. The resulting order is the sequence $v_1, \ldots, v_n$.

[03-chordal-mcs.pdf (cmu.edu)]

CLIQUE

A clique in an undirected graph is a complete subgraph of the given graph. A complete sub-graph is one in which all of its vertices are linked to all of its other vertices.

Many real-world issues make use of the Max clique. Consider a social networking program in which the vertices in a graph reflect people's profiles and the edges represent mutual acquaintance. A clique in this graph indicates a group of people who all know each other.

Cliques in Graph - GeeksforGeeks

GRAPH FILLING

The filling of a graph consists of adding arcs to an original graph G, to obtain a new graph, G_t, such that G_t is triangulated. Given an undirected graph $G = (V, E)$, with n nodes, the following algorithm makes the graph triangulated:

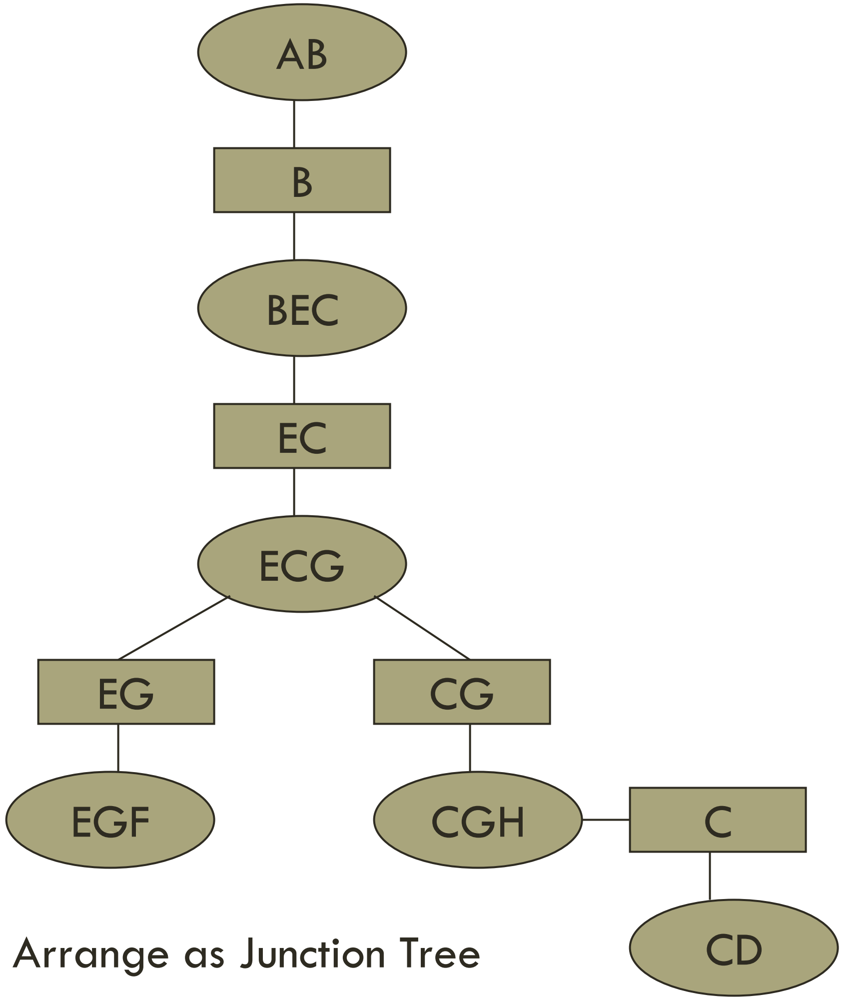
Algorithm 3.2 Graph Filling Algorithm

Order the vertices V with maximum cardinality search: $V_1, V_2, \ldots, V_n$.
for $i = n$ **to** $i = 1$ **do**
  For node V_i, select all its adjacent nodes V_j such that $j > i$. Call this set of nodes A_i.
  Add an arc from V_i to V_k if $k > i$ and $V_k \notin A_i$.
end for

EXAMPLE I(CONT'D)

Clique	Max. Vertex
AB	2
BEC	4
ECG	5
EGF	6
CGH	7
CD	8

Find Cliques of the Triangulated Graph



Arrange as Junction Tree

PROPERTIES OF CLIQUE TREE

A clique tree is an undirected tree, where each node represents a set of variables, which is called a clique.

A clique separator is the intersection between its two neighboring cliques.

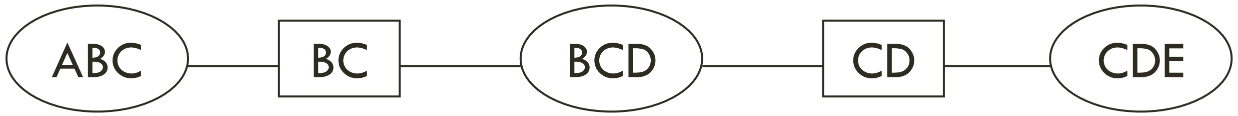
Note that not every clique tree is a junction tree.

A clique tree is a junction tree if and only if it has the *running intersection property or the junction tree property.*

PROPERTIES OF A JUNCTION TREE (ZHANG)

The junction tree has the following characteristics:

- it is an undirected tree, its nodes are clusters of variables
- given two clusters, C_1 and C_2, every node on the path between them contains their intersection $C_1 \cap C_2$
- a Separator, S, is associated with each edge and contains the variables in the intersection between neighbouring nodes



PROPERTIES OF A JUNCTION TREE (WILLIAMS)

A clique tree is a junction tree if it has the following junction tree property:

- if a node appears in two cliques, it appears everywhere on the path between the cliques.

For every triangulated graph there exists a clique tree which obeys the junction tree property

HOW TO CONSTRUCT A JT (EXTENDED VERSION)

Moralize the graph by adding links between parents of common children.

Convert the given BN into an undirected graph by removing the arrowheads.

Triangulate the graph.

- If there is a cycle of length > 3 with no chord, add a chord between two non-adjacent vertices in the cycle.

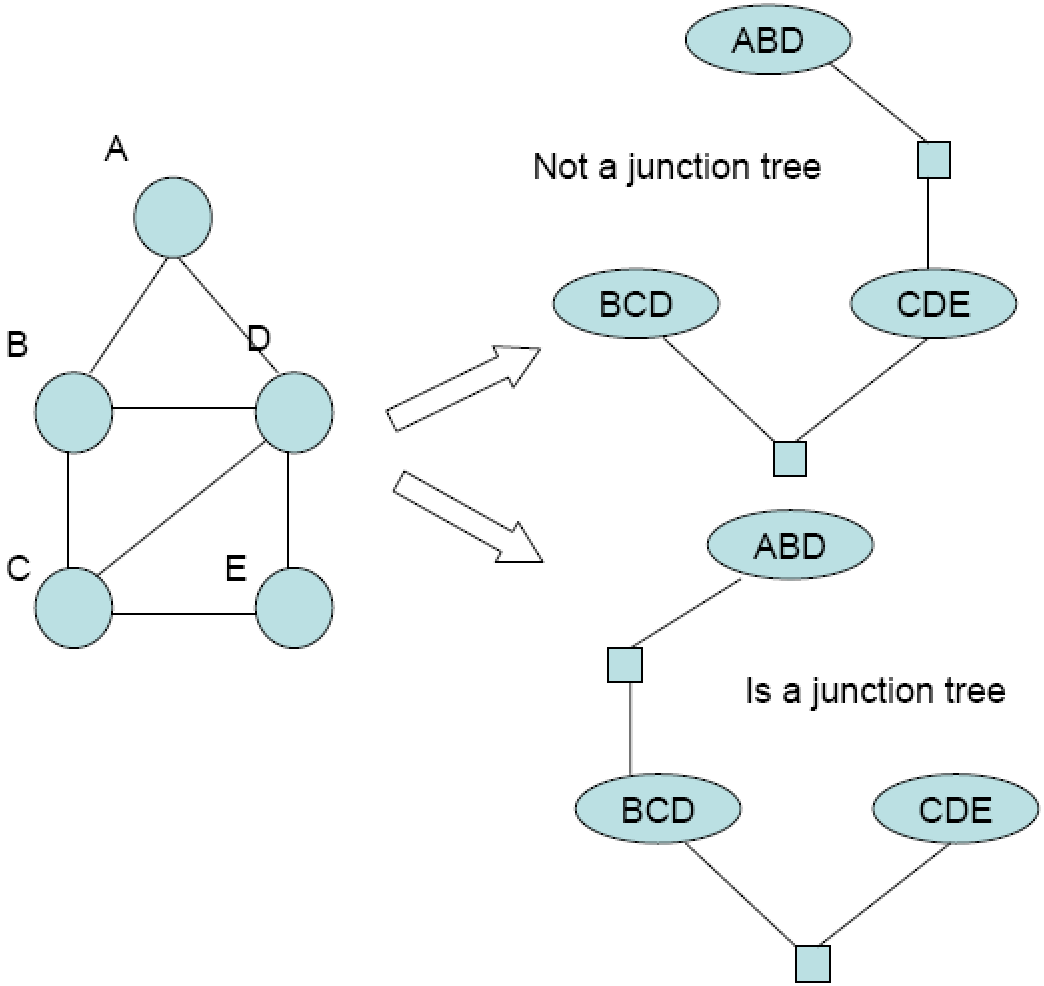
Order the nodes by maximum cardinality search.

- Choose any node in the graph and label it 1.
- For i=2 to n
  - Choose the node with the most labeled neighbors and label it i.
  - If any two labeled neighbors of i are not adjacent to each other
    - Add a link between those neighbors
    - Restart the algorithm.

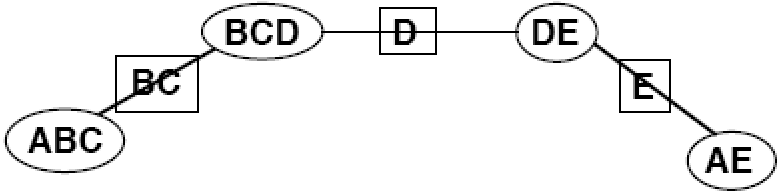
Find the cliques of the triangulated graph.

Arrange as a junction tree.

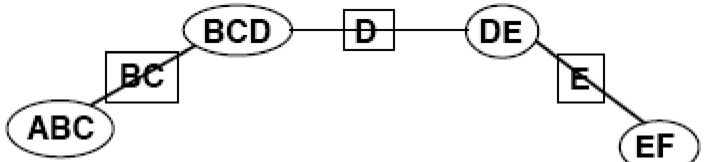
JUNCTION TREE EXAMPLES (JORDAN)



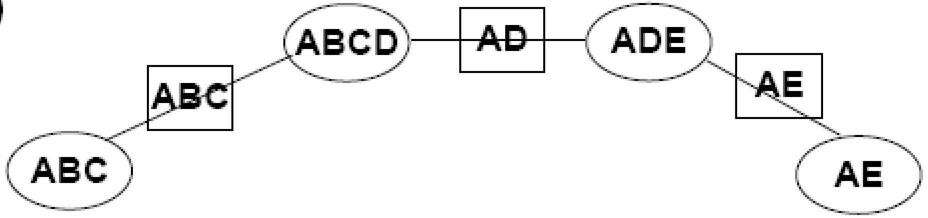
JUNCTION TREE EXAMPLES (LASKEY)



A cluster tree that is not a junction tree
(*Why not?*)

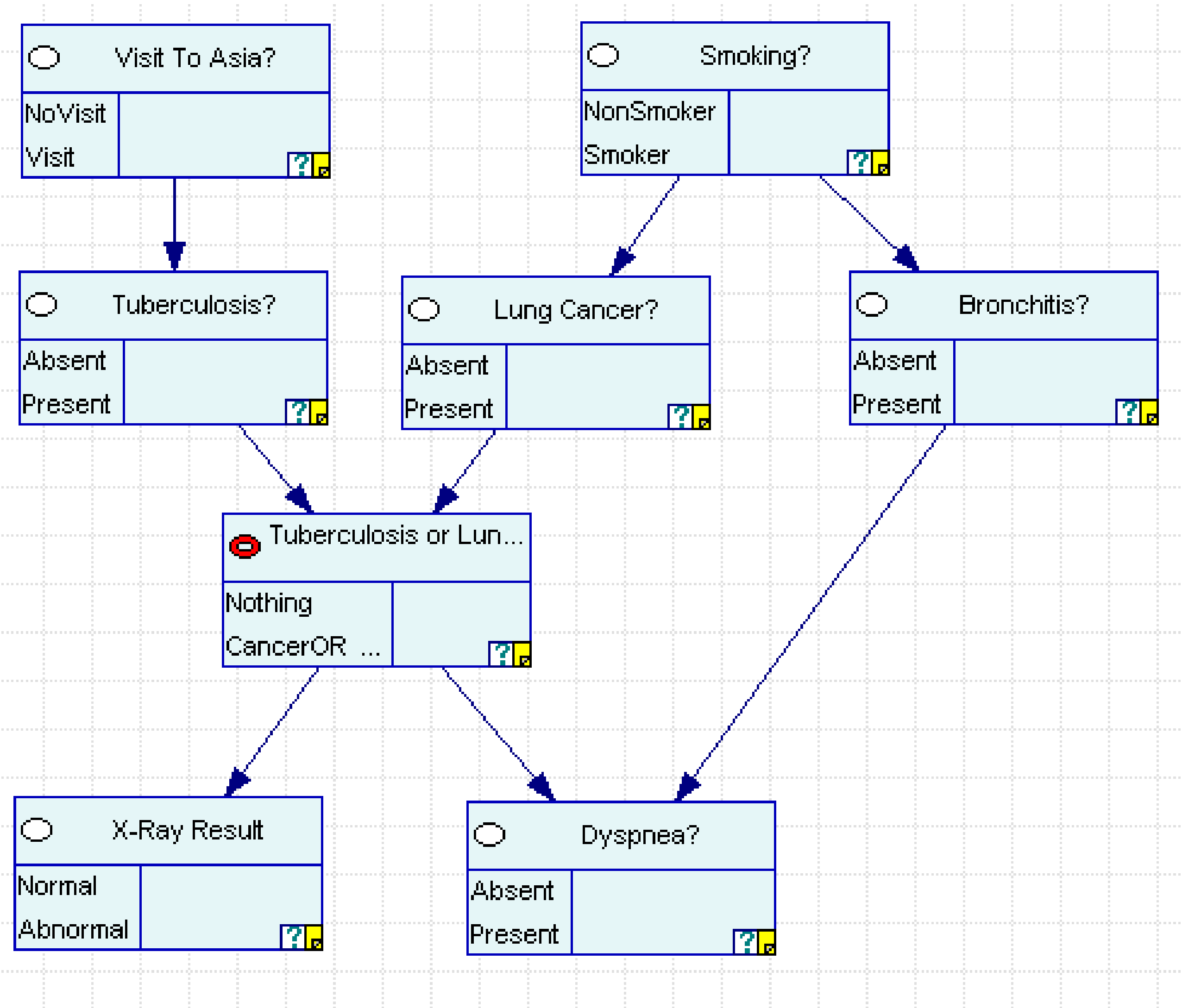


A junction tree (*Why?*)



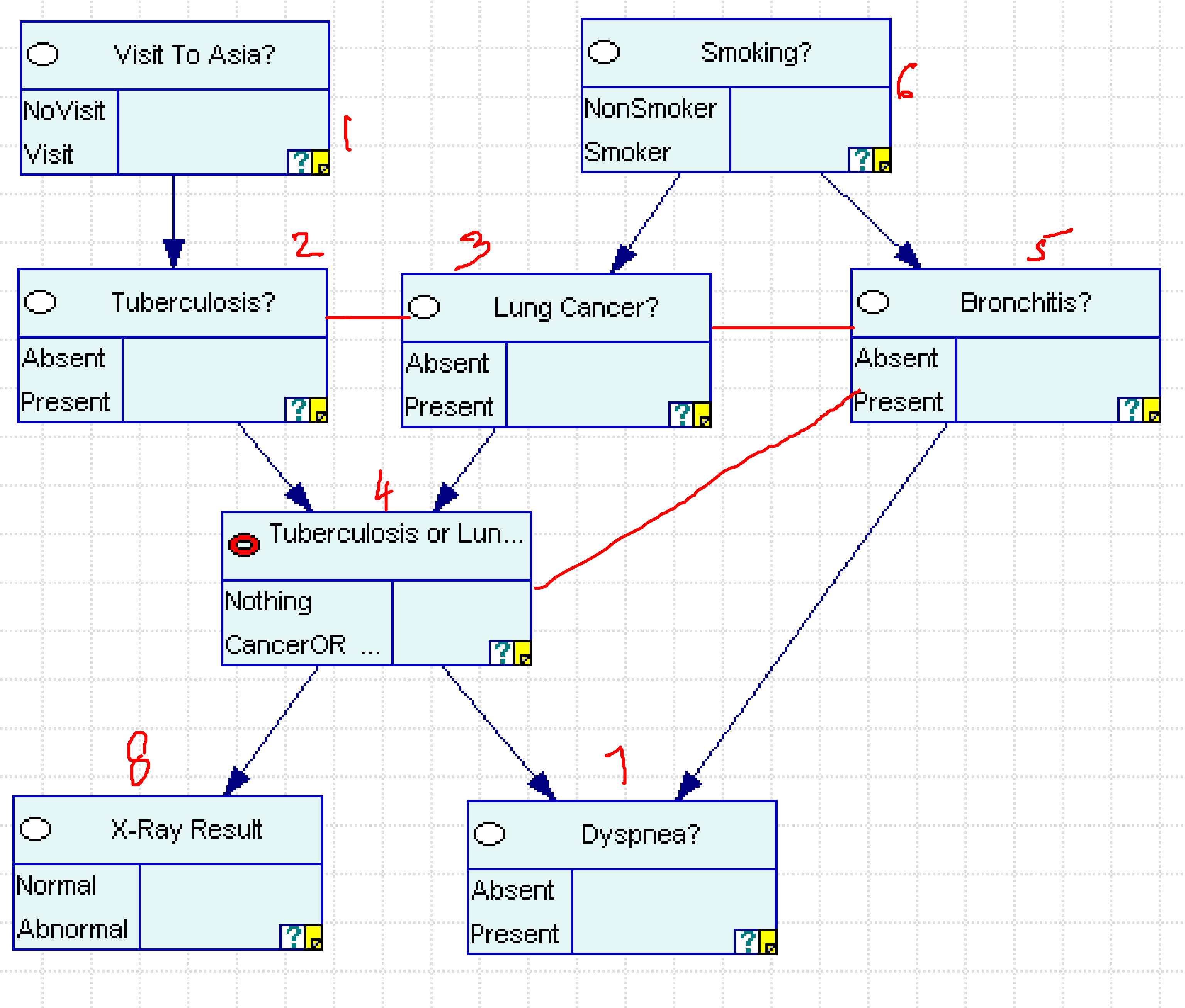
Also a junction tree (*Why?*)

EXAMPLE II



Visit To Asia?
NoVisit
Visit
Smoking?
NonSmoker
Smoker
Tuberculosis?
Absent
Present
Lung Cancer?
Absent
Present
Bronchitis?
Absent
Present
Tuberculosis or Lun...
Nothing
CancerOR ...
X-Ray Result
Normal
Abnormal
Dyspnea?
Absent
Present

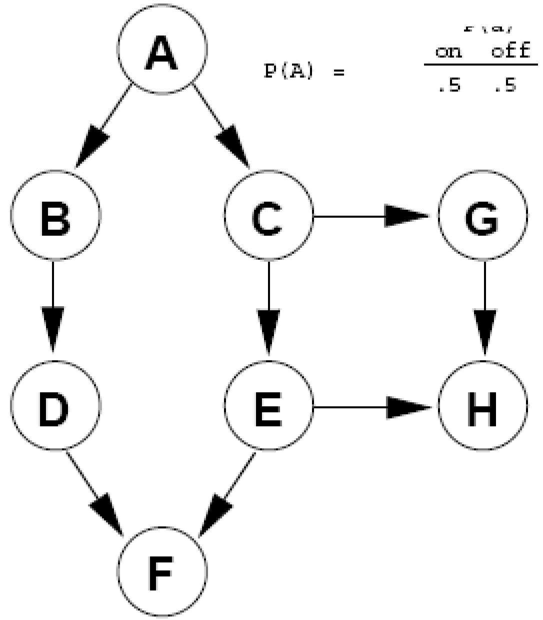
EXAM



Visit To Asia?
NoVisit
Visit
1
Smoking?
NonSmoker
Smoker
6
2
Tuberculosis?
Absent
Present
3
Lung Cancer?
Absent
Present
5
Bronchitis?
Absent
Present
4
Tuberculosis or Lun...
Nothing
CancerOR ...
8
X-Ray Result
Normal
Abnormal
7
Dyspnea?
Absent
Present

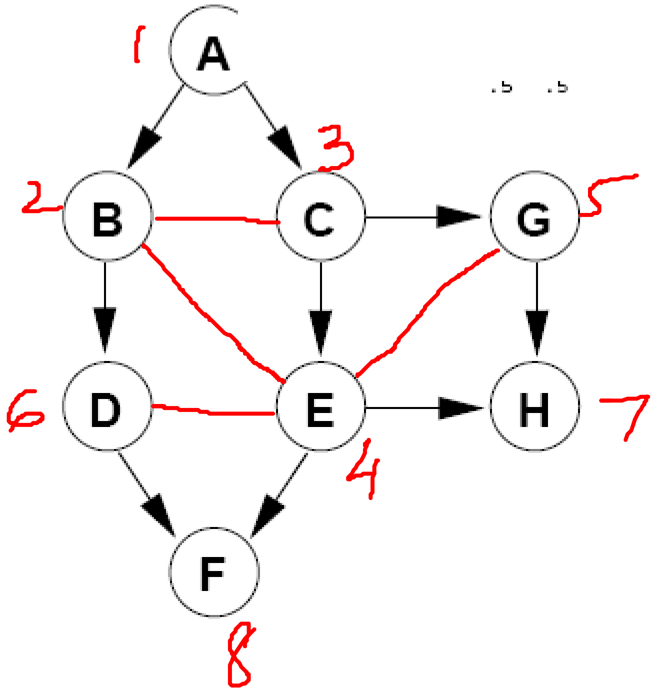
EXAMPLE III (HUANG & DARWICHE)

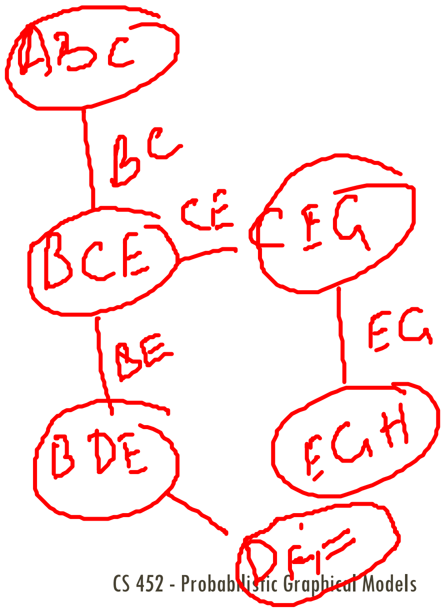
Consider this BN and transform it into a junction tree by performing moralization, triangulation and maximum cardinality search.



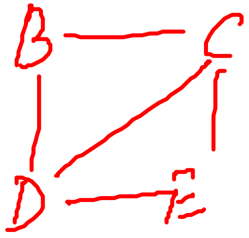
EXAMPLE III (HUANG & DARWICHE)

Consider this BN and transform it into a junction tree by performing moralization, triangulation and maximum cardinality search.

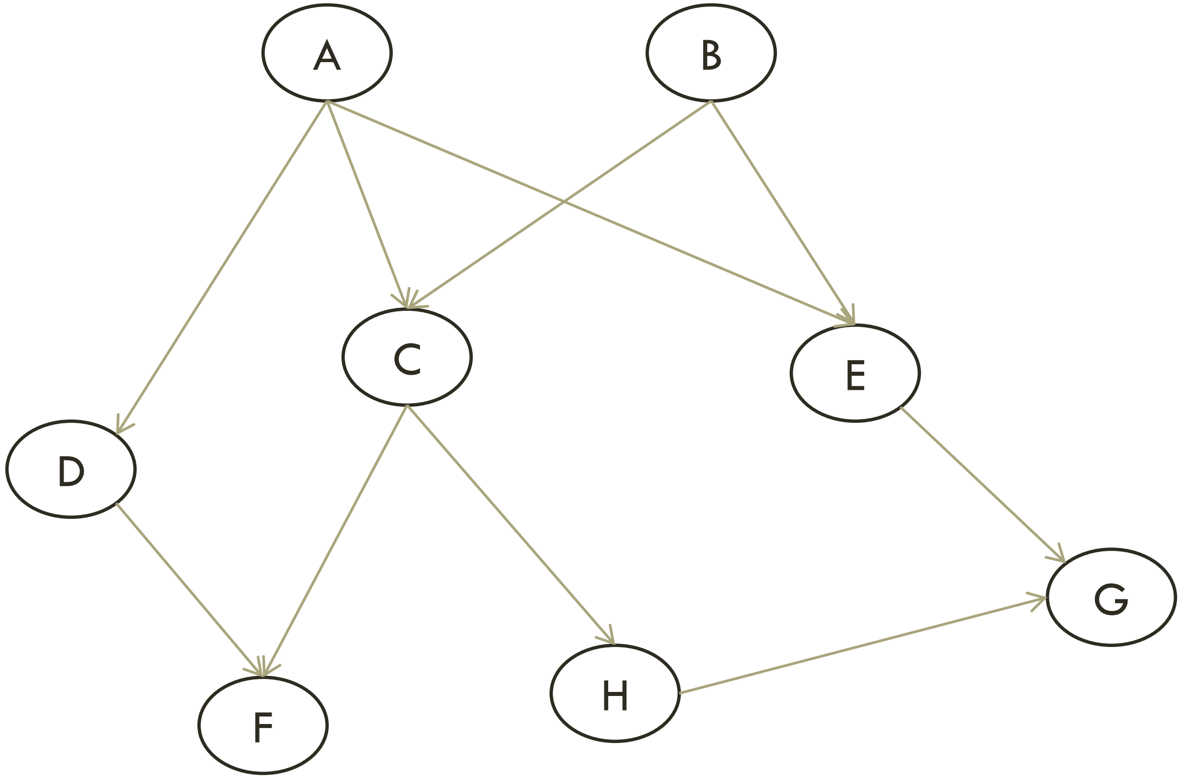




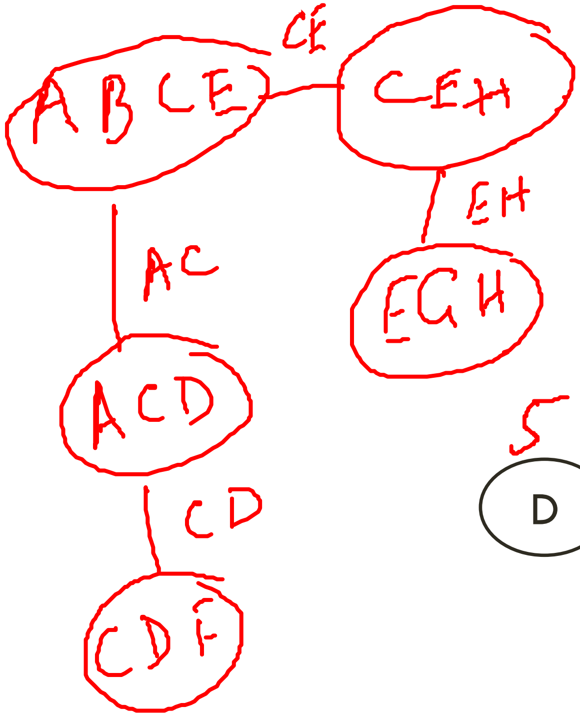
ABC 3
BCE 4
BDE 6
CEG 5
EGH 7
DEF 8

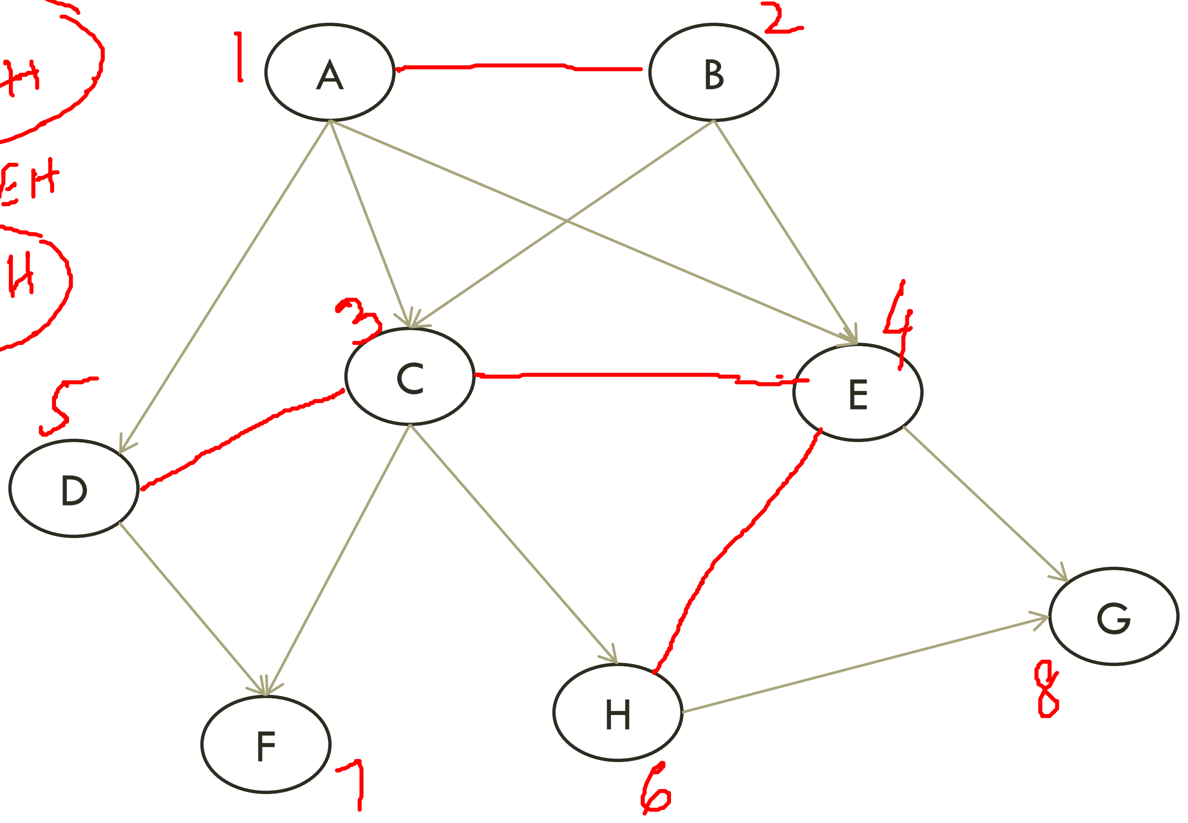


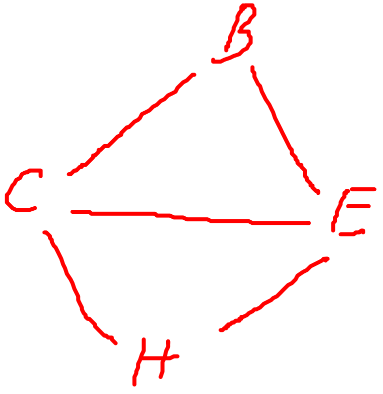
EXAMPLE IV (SOURCE: LASKEY)



EXAMPLE IV (SOURCE: LASKEY)

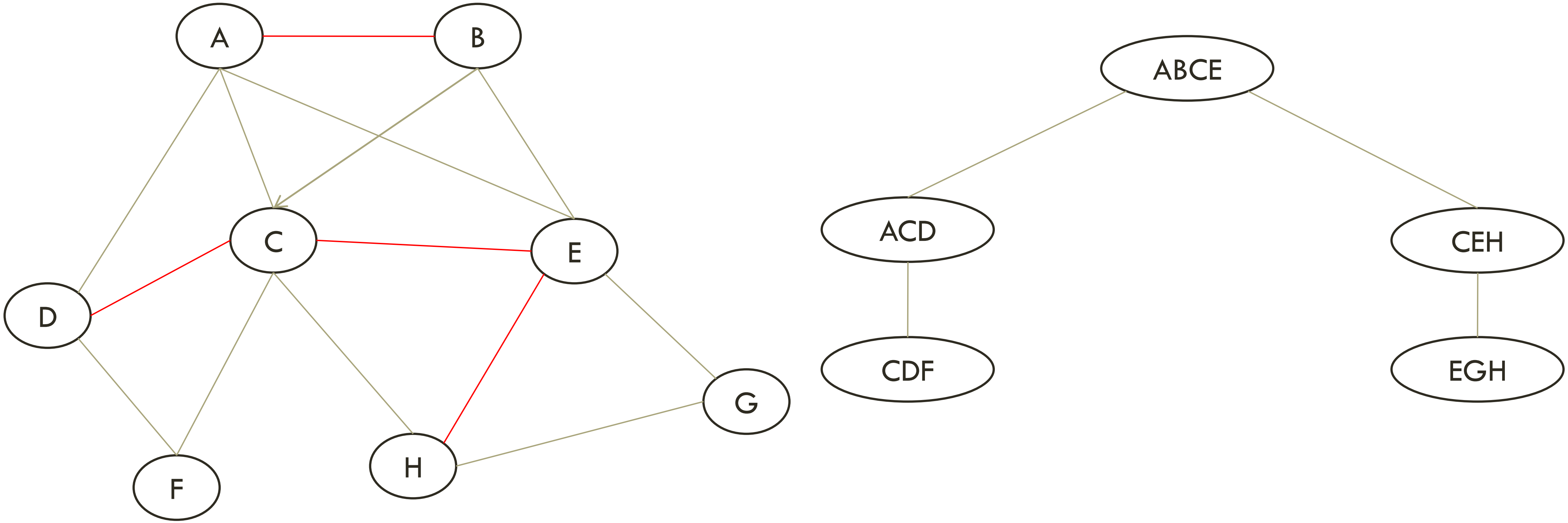




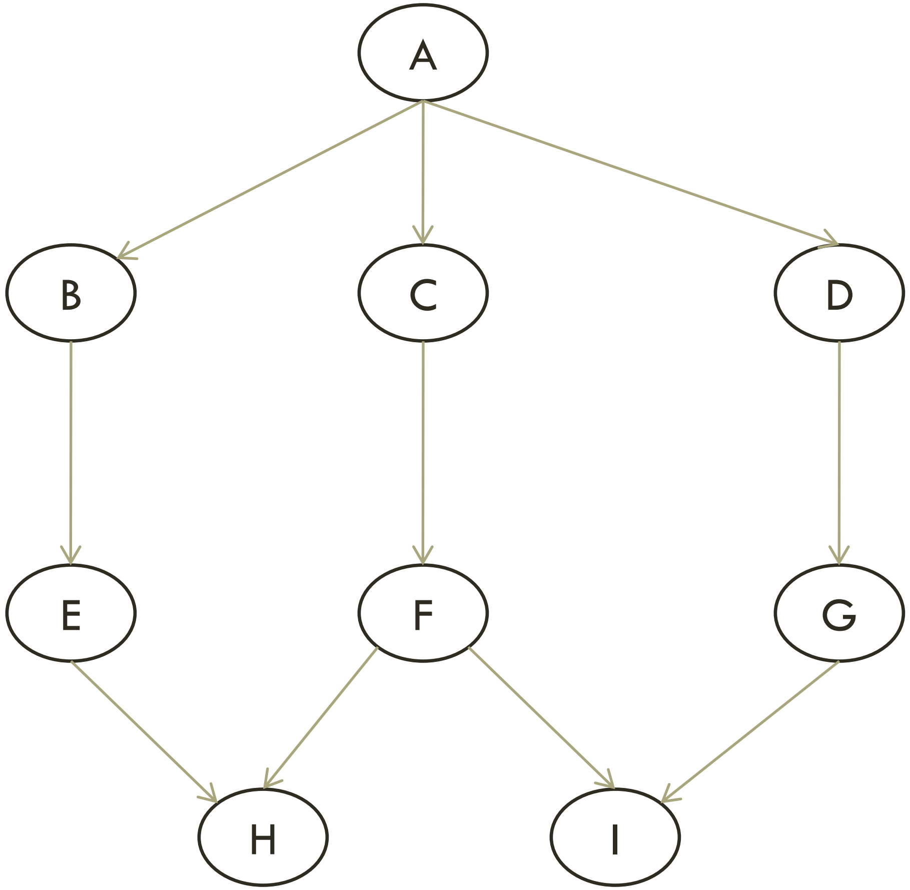


ABCE 4
ACD 5
CDF 7
CEH 6
EGH 8

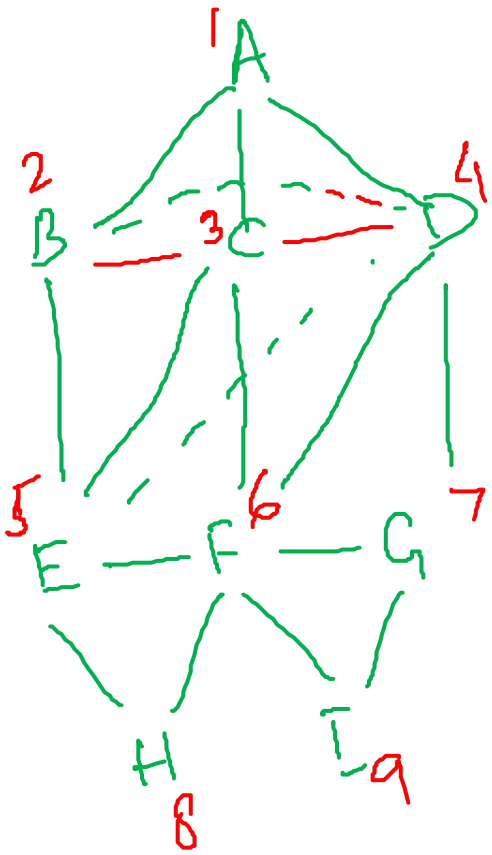
EXAMPLE IV (CONT'D)



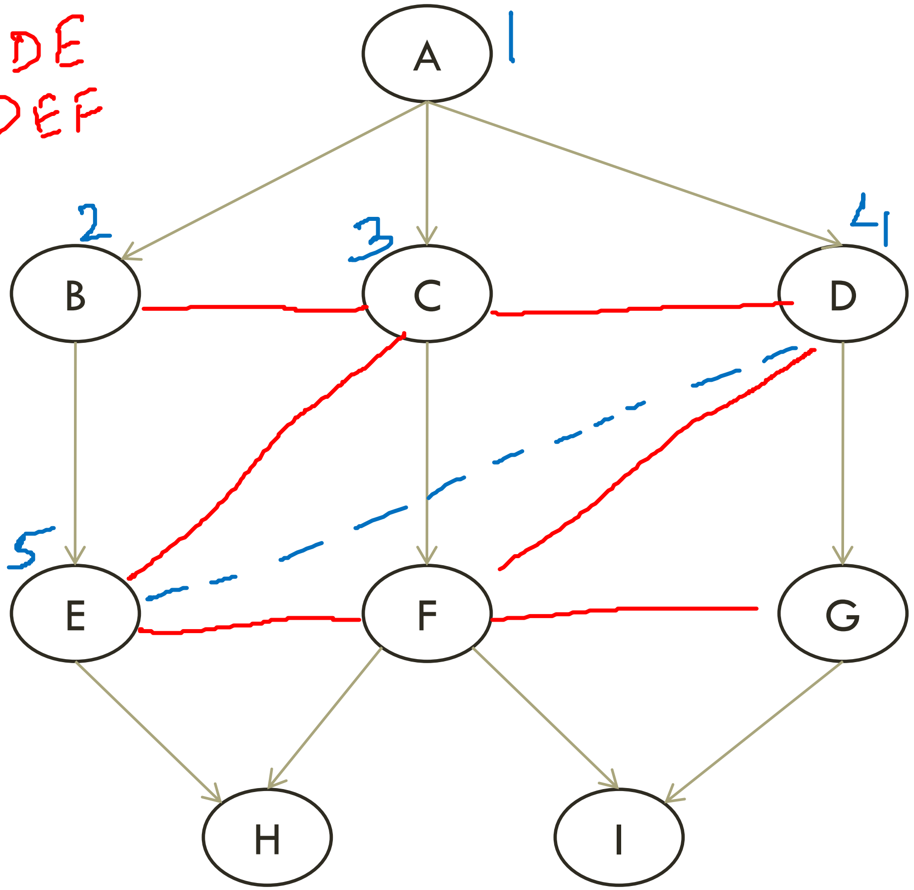
EXAMPLE V (SOURCE: KORB & NICHOLSON)

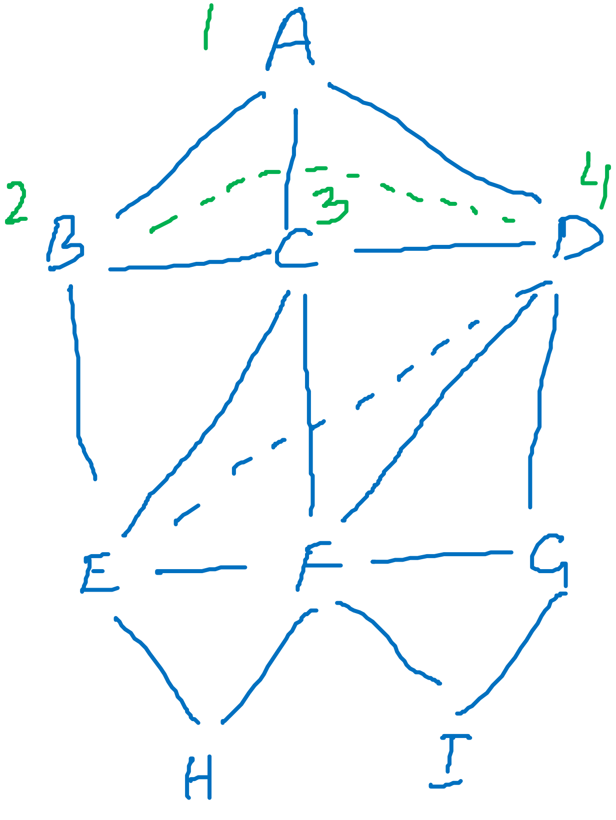


EXAMPLE V (SOURCE: KORB & NICHOLSON)

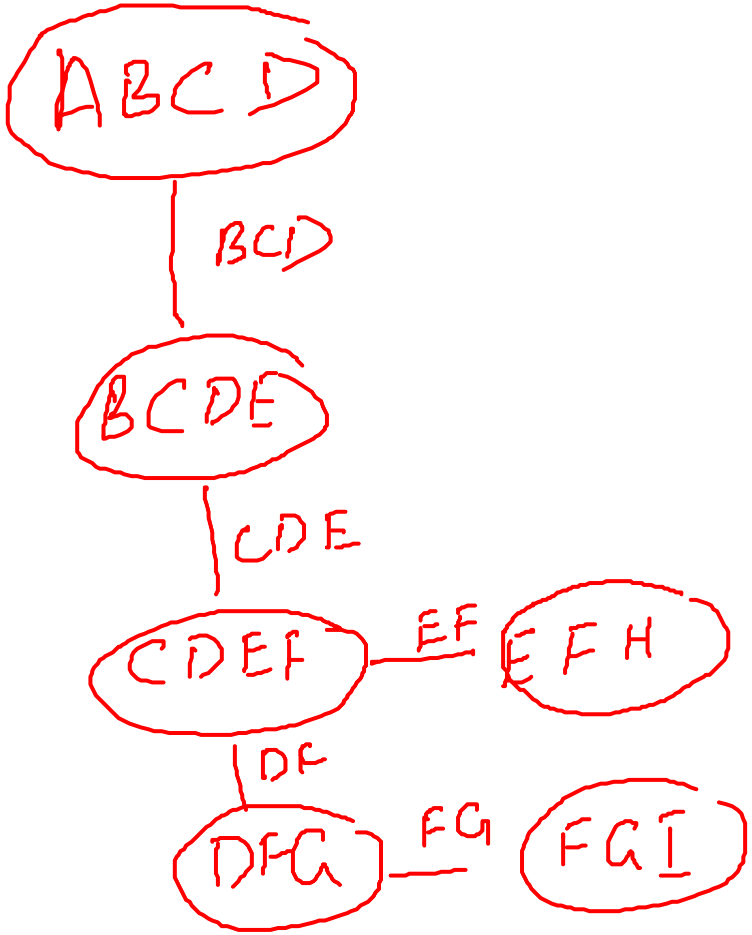


ABCD
BCDE
CDEF

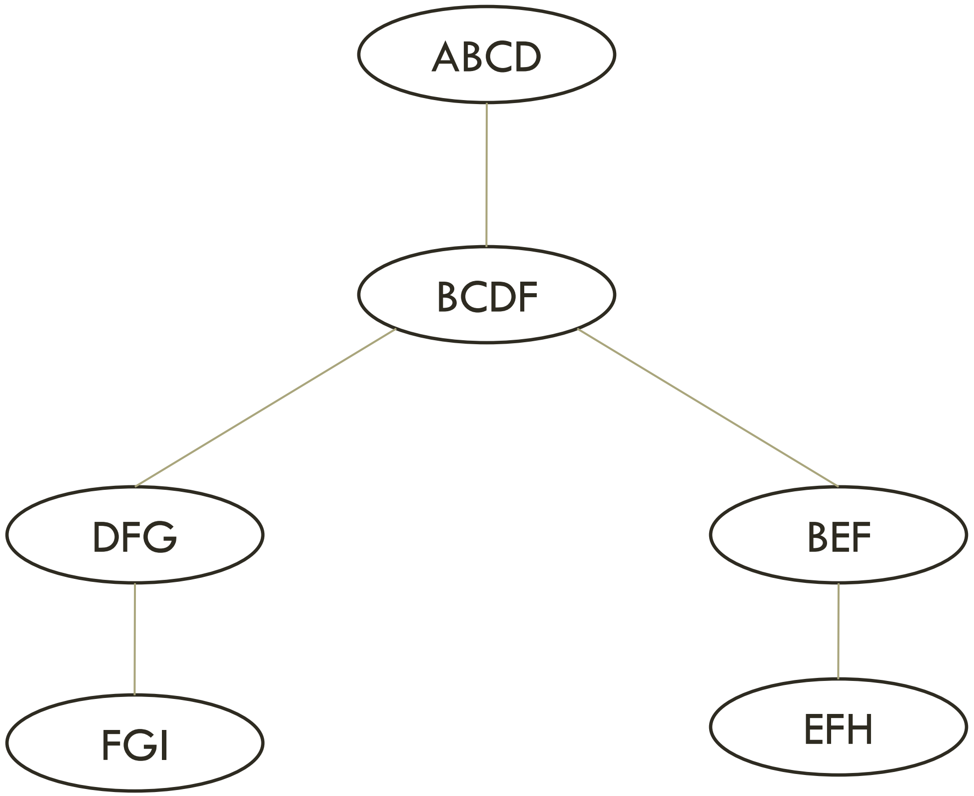




ABCD	4
BCDE	5
CDEF	6
DFG	7
~~FGH~~ EFH	8
FGI	9



EXAMPLE V (CONT'D)



PARAMETERS OF CLIQUES

The potentials of each clique are obtained following the next steps:

1. Determine the set of variables for each clique, C_i.
2. Determine the set of variables that are common with the previous (parent) clique, S_i.
3. Determine the variables that are in C_i but not in S_i: $R_i = C_i - S_i$.
4. Calculate the potential of each clique, clq_i, as the product of the corresponding CPTs: $\psi(clq_i) = \prod_j P(X_j \mid Pa(X_j))$; where X_j are the variables in clq_i.

REFERENCES

[machine-learning-uiuc/docs/Probabilistic Graphical Models - Principles and Techniques.pdf at master · Zhenye-Na/machine-learning-uiuc · GitHub]

[PPT - Junction tree Algorithm PowerPoint Presentation, free download - ID:819513]

[Junction Tree Algorithm (ermongroup.github.io)]