

CS 583: PROBABILISTIC GRAPHICAL MODELS

TOPIC: JUNCTION TREE ALGORITHM



Mustafa Bilgic



<http://www.cs.iit.edu/~mbilgic>

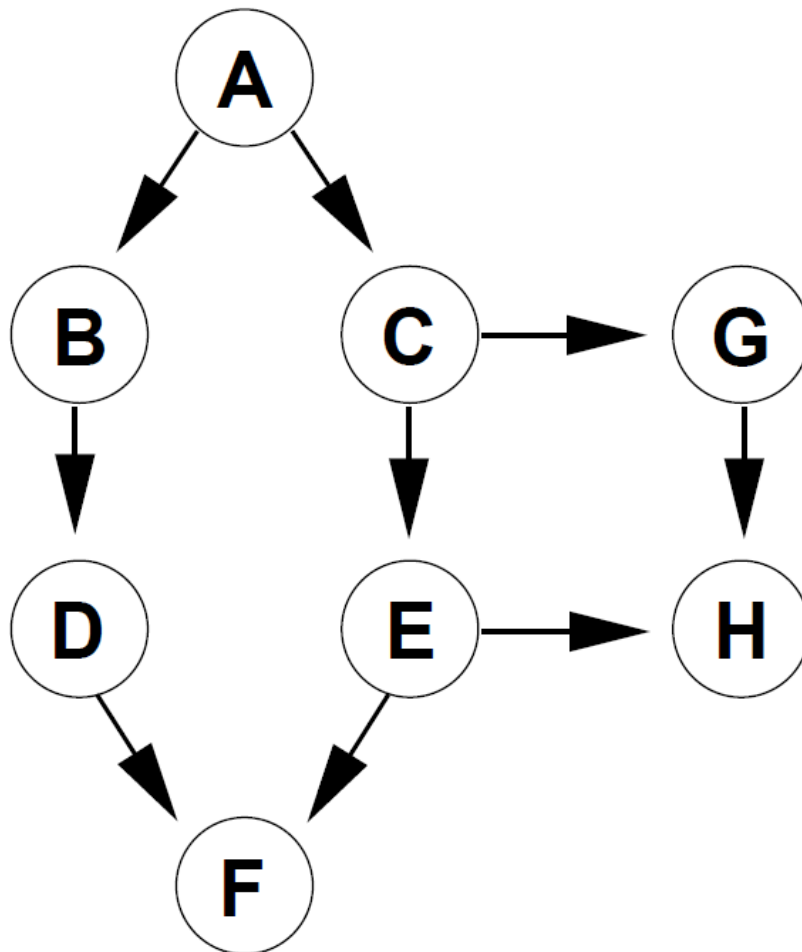


<https://twitter.com/bilgicm>

MOTIVATION

- We are interested in multiple marginal/conditional probabilities
- In variable elimination, we define our target upfront and then eliminate the others
- If we need probabilities for other variables, there is no apparent way of reusing shared computations
- In the student example, assume that I'm interested in $P(G)$ and $P(L)$. What are some of the shared computations?

EXAMPLE



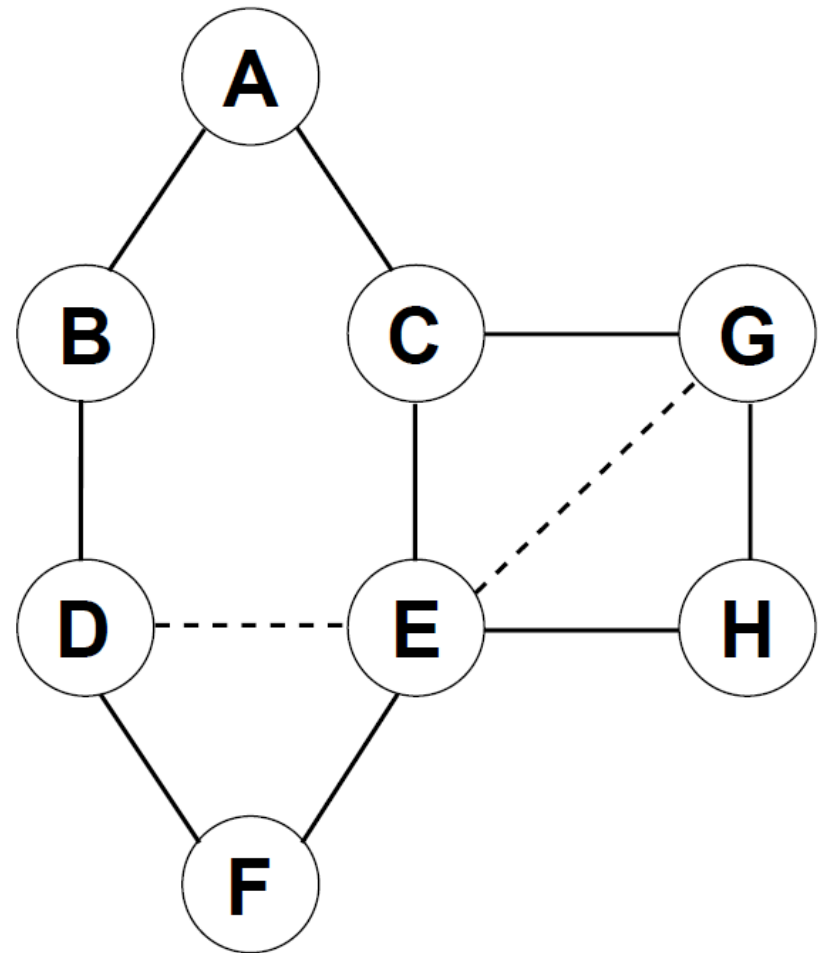
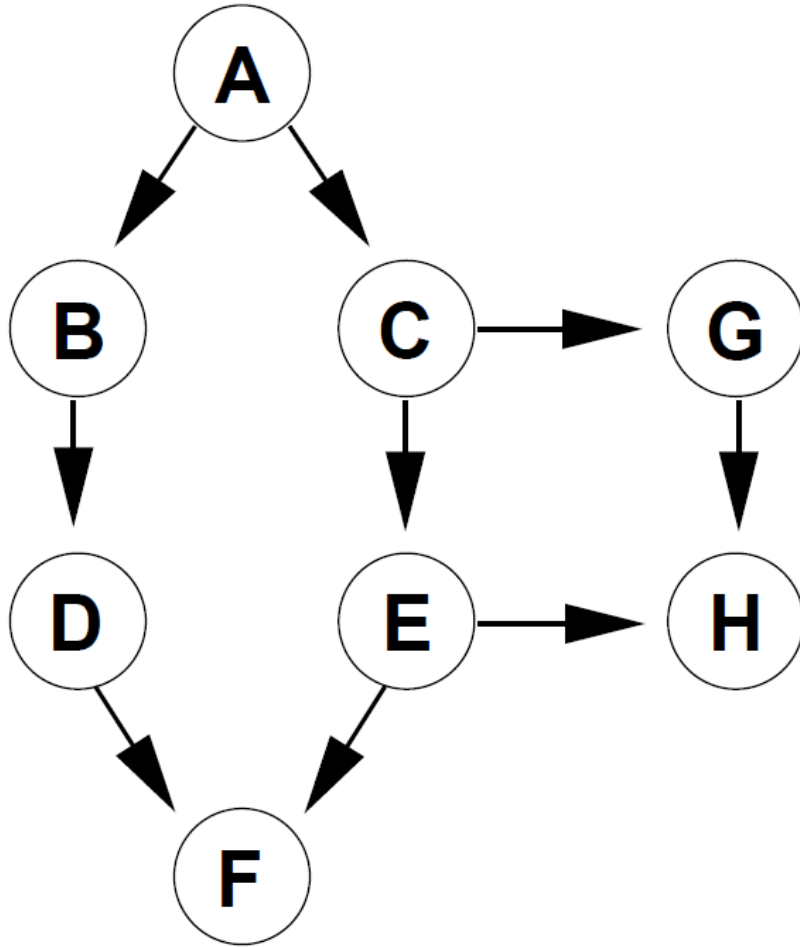
Calculate $P(H)$ using variable elimination

Now, calculate $P(G)$ using variable elimination

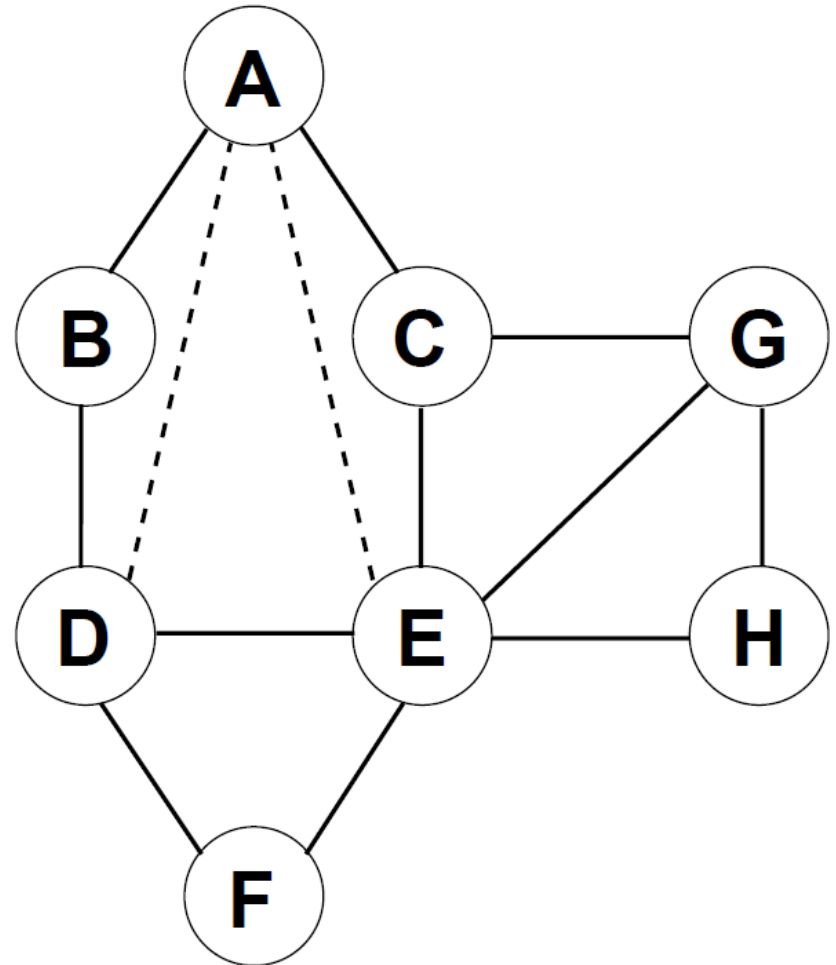
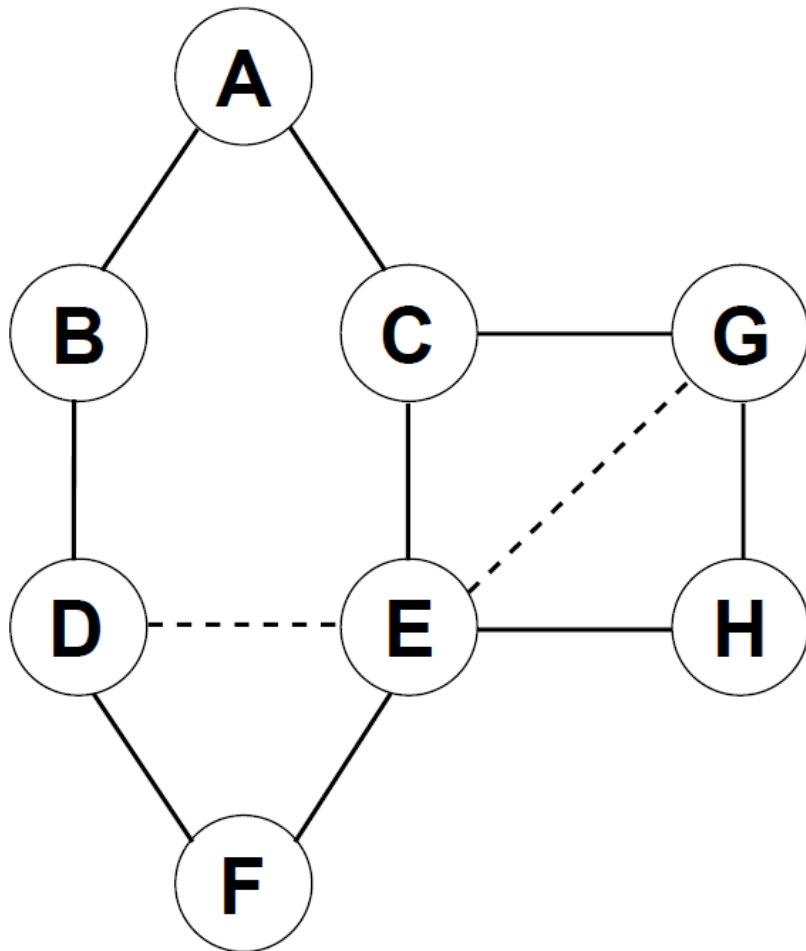
VARIABLE ELIMINATION AS GRAPH TRANSFORMATION

- First, construct the moral graph
- Then, eliminate variables so that each elimination introduces the fewest number of edges
- Take note of the factors

EXAMPLE - MORALIZE



ELIMINATION ORDER: H, G, F, C, B, D, E, A



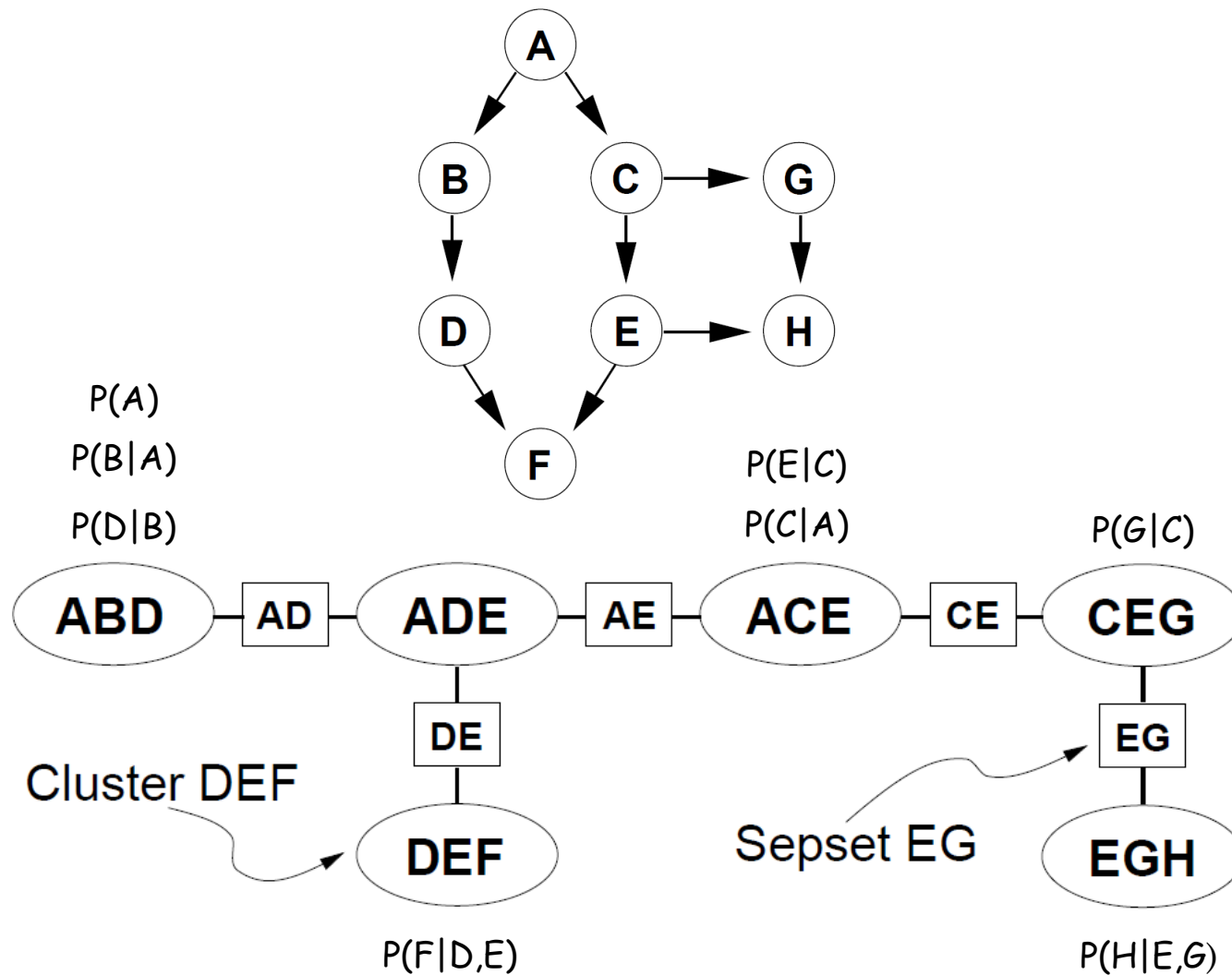
CLUSTER GRAPH

- A *cluster graph* \mathcal{U} for a set of factors Φ over \mathcal{X} is an undirected graph, each of whose nodes are associated with a cluster $C_i \subseteq \mathcal{X}$.
- A cluster graph must be *family preserving* – each factor $\phi \in \Phi$ must be associated with a cluster C_i , denoted as $\alpha(\phi)$, such that $\text{Scope}[\phi] \subseteq C_i$.
- Each edge between a pair of clusters C_i and C_j is associated with a *sepset* $S_{ij} \subseteq C_i \cap C_j$

RUNNING INTERSECTION PROPERTY

- Let \mathcal{T} be a cluster tree. \mathcal{T} has *running intersection property* if, whenever there is a variable X such that $X \in \mathbf{C}_i$ and $X \in \mathbf{C}_j$, then X is also in every cluster in the unique path in \mathcal{T} between \mathbf{C}_i and \mathbf{C}_j .
- A cluster tree that satisfies the running intersection property is also called the *join/clique/junction tree*.
- **Theorem:** A cluster tree obtained through a run of variable elimination satisfies the running intersection property; that is, it is a clique tree.

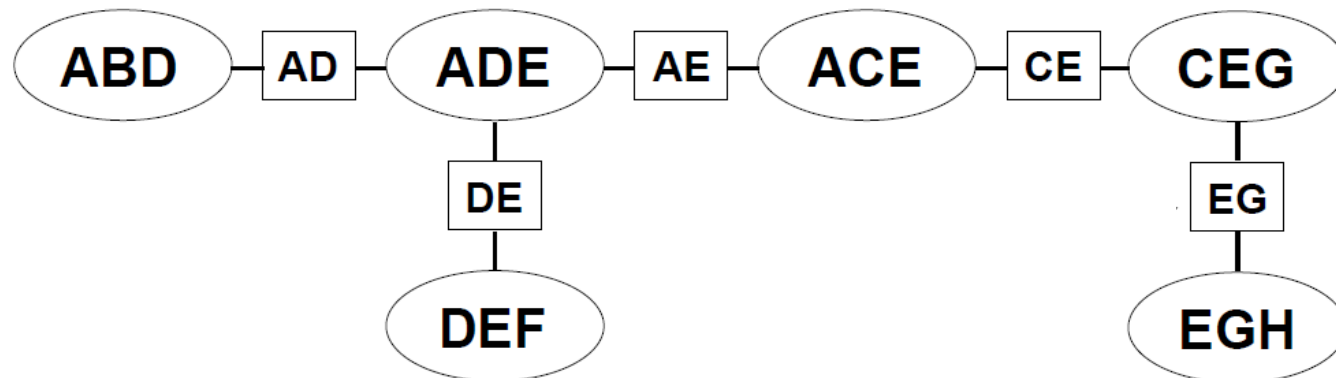
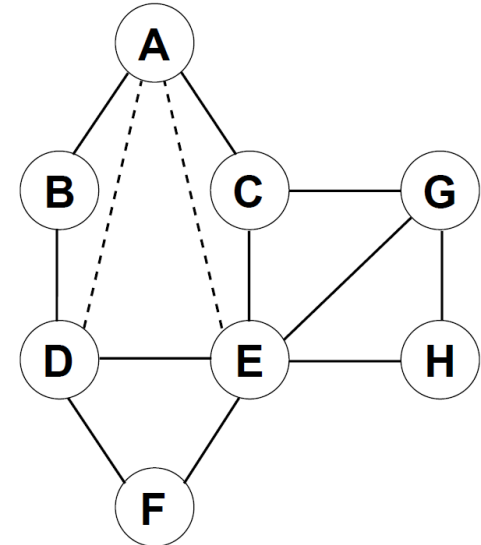
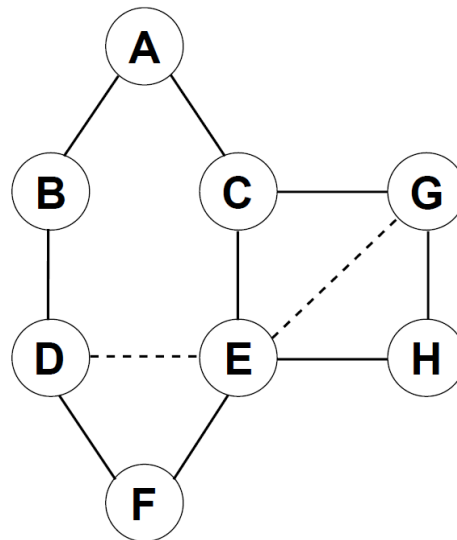
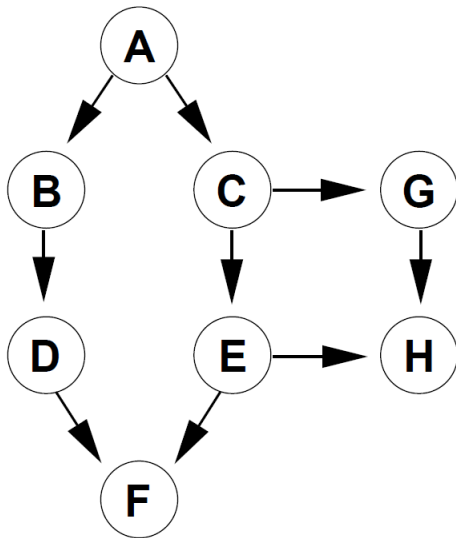
EXAMPLE CLIQUE TREE



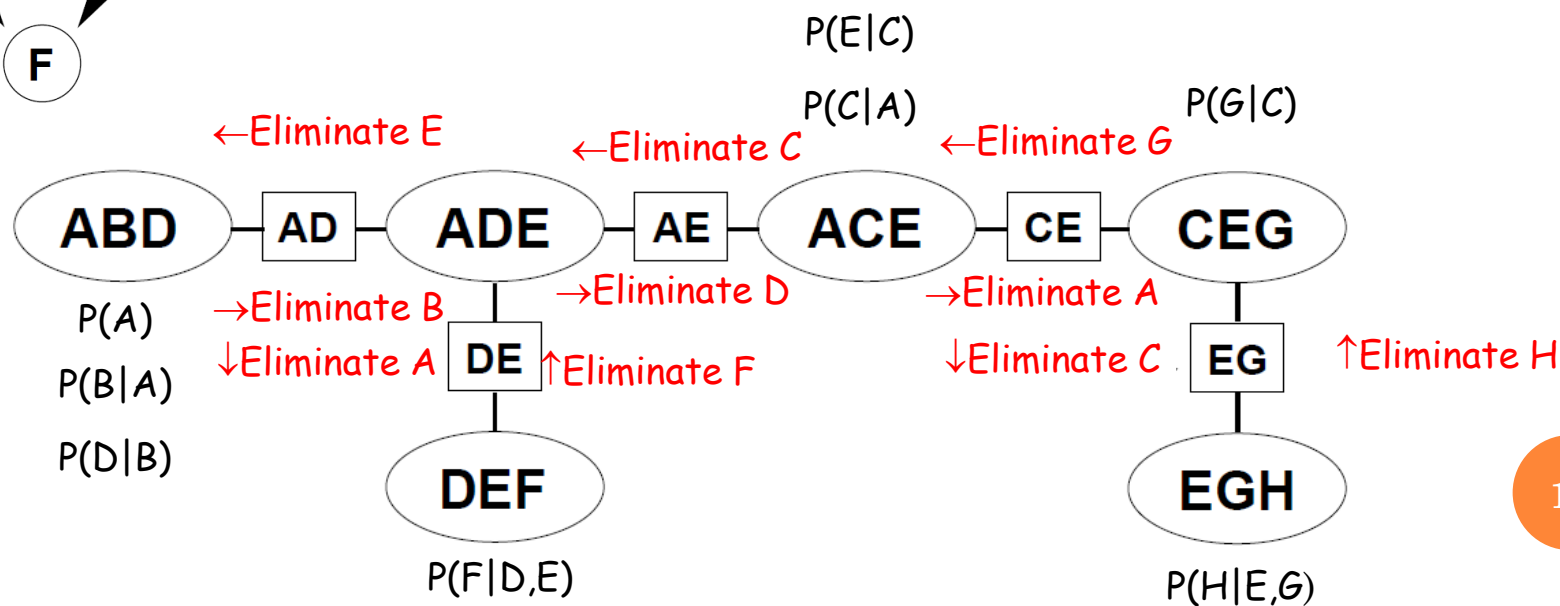
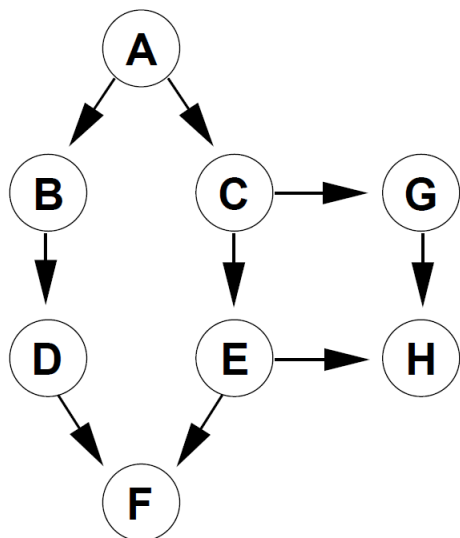
CONSTRUCT A CLIQUE TREE

1. Moralize the graph
2. Pick a variable elimination order
3. Eliminate the variables, noting the maximal cliques
4. The cliques are the nodes of the tree
5. Until a tree is formed (i.e., $n-1$ edges are added)
 - a) Connect two disconnected components by a maximal size sepset

ELIMINATION ORDER: H, G, F, C, B, D, E, A



VARIABLE ELIMINATION ON JUNCTION TREE



MESSAGE PASSING ON JUNCTION TREE

- Clusters receive from and send messages to its neighbors
- Each message pass consists of elimination of one or more variables
- A cluster C_i is ready to send a message to its neighbor C_j , when it receives messages from its *all other* neighbors
- A message from C_i to C_j is computed as follows
 - C_i multiplies all the factors assigned to it, and all the messages it received from its *other* neighbors
 - It sums out $C_i \setminus S_{ij}$

A MESSAGE

$$\delta_{i \rightarrow j} = \sum_{C_i \setminus S_{ij}} \left(\left(\prod_{\phi: \alpha(\phi)=i} \phi \right) \times \left(\prod_{k \in (Nb_i - \{j\})} \delta_{k \rightarrow i} \right) \right)$$

BELIEF

$$\beta_i \propto \left(\prod_{\phi: \alpha(\phi)=i} \phi \right) \times \left(\prod_{k \in Nb_i} \delta_{k \rightarrow i} \right)$$