Junction tree propagation - BNDG 4-4.6

Finn V. Jensen and Thomas D. Nielsen

Exact Inference

Message Passing in Join Trees

More sophisticated inference technique; used in most implemented Bayesian Network systems (e.g. Hugin).

Overview:

Given: Bayesian network for $P(\mathbf{V})$

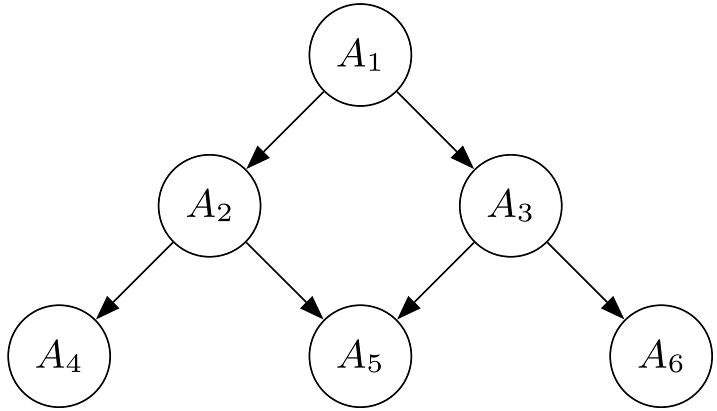
$\Downarrow$ Preprocessing

Construct a new internal representation for $P(\mathbf{V})$ called a *junction tree*

$\Downarrow$ Inference/Updating

Retrieve $P(A \mid \mathbf{E} = \mathbf{e})$ for single $A \in \mathbf{V}$

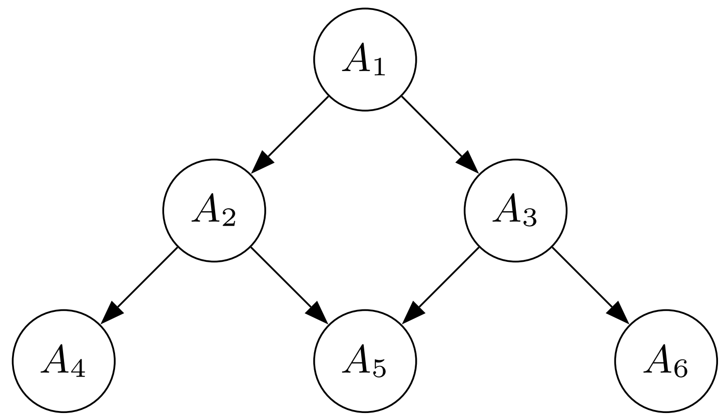
Calculate a marginal



$P(A_4)$

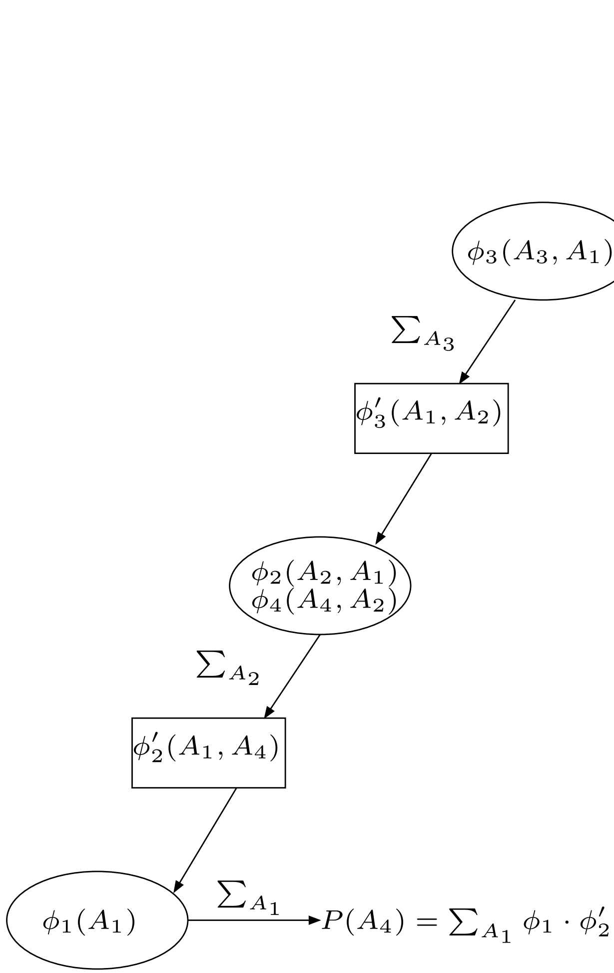
$$= \sum_{A_1} \sum_{A_2} \sum_{A_3} \sum_{A_5} \sum_{A_6} \phi_1(A_1)\phi_2(A_2, A_1)\phi_3(A_3, A_1)\phi_4(A_4, A_2)\phi_5(A_5, A_2, A_3)\phi_6(A_6, A_3)$$

Calculate a marginal



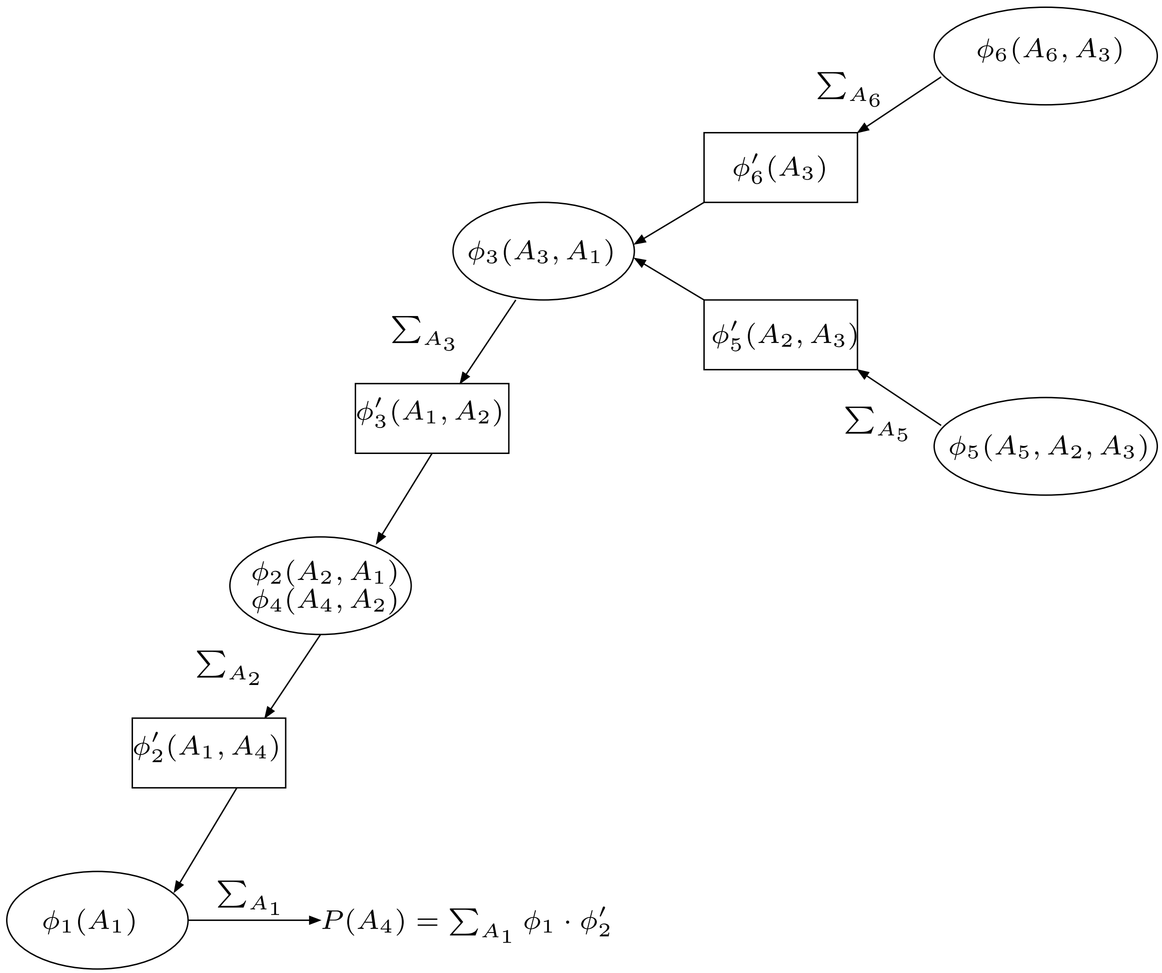
$$
\begin{aligned}
&P(A_4)\\
&\quad = \sum_{A_1}\sum_{A_2}\sum_{A_3}\sum_{A_5}\sum_{A_6}\phi_1(A_1)\phi_2(A_2,A_1)\phi_3(A_3,A_1)\phi_4(A_4,A_2)\phi_5(A_5,A_2,A_3)\phi_6(A_6,A_3)\\
&\quad = \sum_{A_1}\phi_1(A_1)\sum_{A_2}\phi_2(A_2,A_1)\phi_4(A_4,A_2)\sum_{A_3}\phi_3(A_3,A_1)\sum_{A_5}\phi_5(A_5,A_2,A_3)\sum_{A_6}\phi_6(A_6,A_3)
\end{aligned}
$$

Calculate a marginal



$$P(A_4)=\sum_{A_1} \phi_1(A_1) \sum_{A_2} \phi_2(A_2,A_1)\phi_4(A_4,A_2) \sum_{A_3} \phi_3(A_3,A_1) \sum_{A_5} \phi_5(A_5,A_2,A_3) \sum_{A_6} \phi_6(A_6,A_3)$$

Calculate a marginal



Some of the calculations can be simplified. E.g.

$$\sum_{A_6} P(A_6 \mid A_3) = \mathbf{1}$$

$$P(A_4)=\sum_{A_1} \phi_1(A_1) \sum_{A_2} \phi_2(A_2,A_1)\phi_4(A_4,A_2) \sum_{A_3} \phi_3(A_3,A_1) \sum_{A_5} \phi_5(A_5,A_2,A_3) \sum_{A_6} \phi_6(A_6,A_3)$$

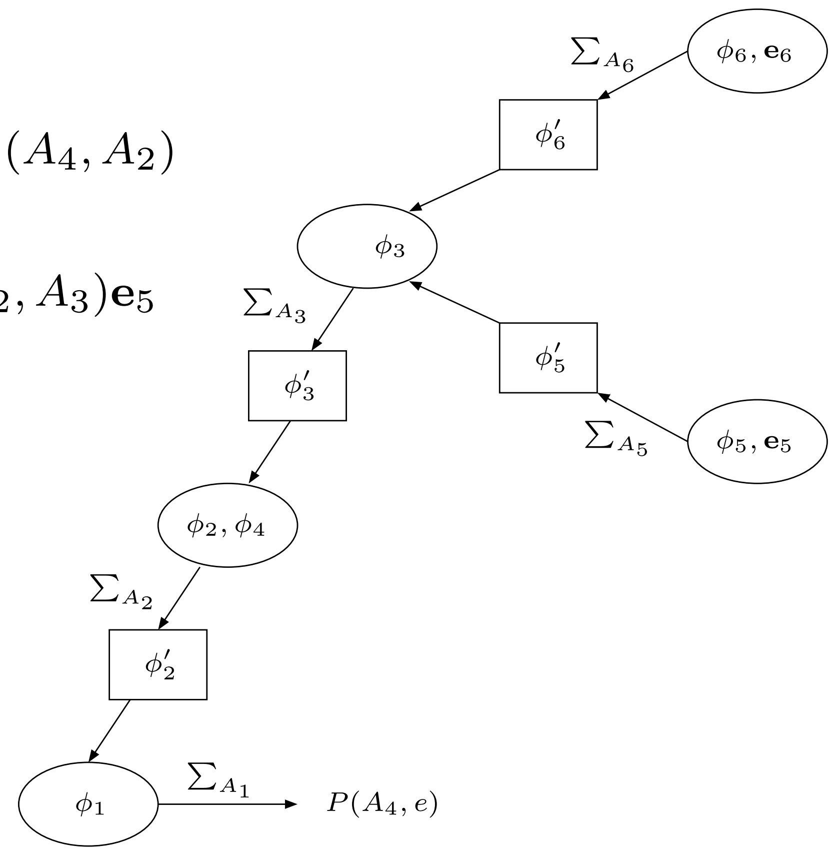
Calculate a marginal

Assume evidence $\mathbf{e} = (\mathbf{e}_5, \mathbf{e}_6)$; represented as 0/1 potentials. Then:

$$P(A_4, \mathbf{e}) = \sum_{A_1} \phi_1(A_1) \sum_{A_2} \phi_2(A_2, A_1)\phi_4(A_4, A_2)$$

$$\sum_{A_3} \phi_3(A_3, A_1) \sum_{A_5} \phi_5(A_5, A_2, A_3)\mathbf{e}_5$$

$$\sum_{A_6} \phi_6(A_6, A_3)\mathbf{e}_6$$



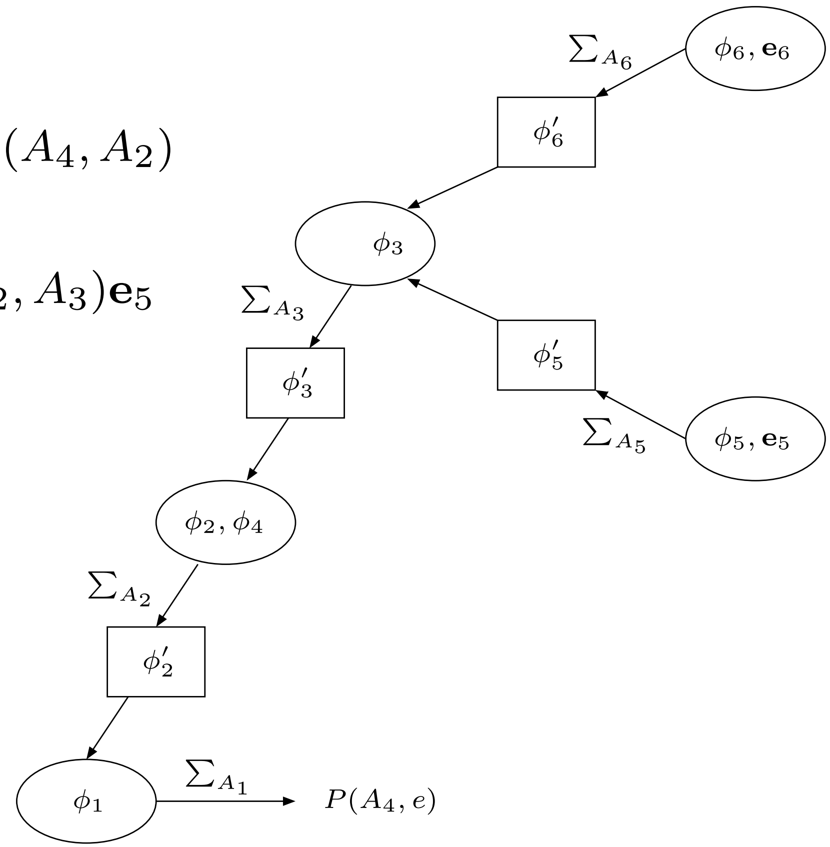
and

$$P(A_4 \mid \mathbf{e}) = \frac{P(A_4, \mathbf{e})}{\sum_{A_4} P(A_4, \mathbf{e})}$$

Calculate a marginal

Assume evidence $\mathbf{e} = (\mathbf{e}_5, \mathbf{e}_6)$; represented as 0/1 potentials. Then:

$$P(A_4, \mathbf{e}) = \sum_{A_1} \phi_1(A_1) \sum_{A_2} \phi_2(A_2, A_1)\phi_4(A_4, A_2)$$
$$\sum_{A_3} \phi_3(A_3, A_1) \sum_{A_5} \phi_5(A_5, A_2, A_3)\mathbf{e}_5$$
$$\sum_{A_6} \phi_6(A_6, A_3)\mathbf{e}_6$$



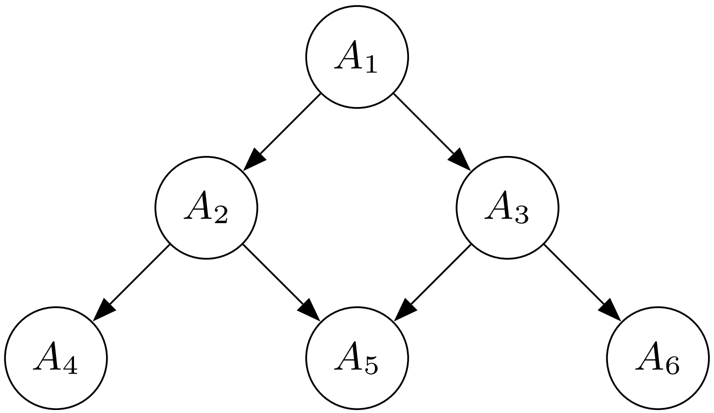
and

$$P(A_4 \,|\, \mathbf{e}) = \frac{P(A_4, \mathbf{e})}{\sum_{A_4} P(A_4, \mathbf{e})}$$

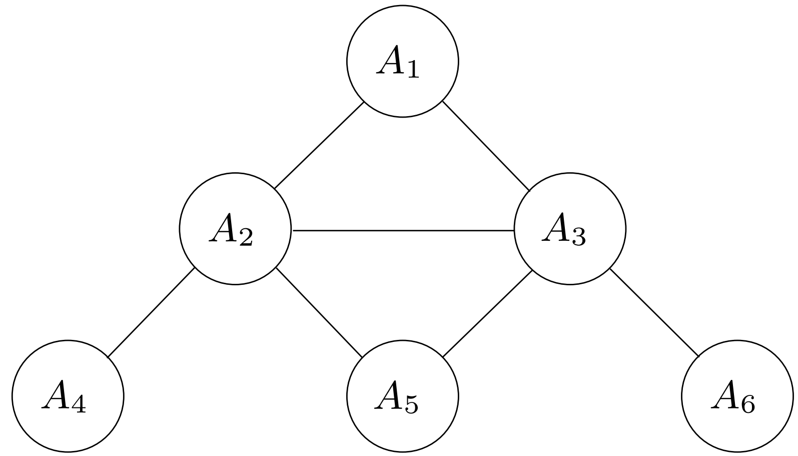
The process is sufficiently general to handle all evidence scenarios!

$\Rightarrow$ We look for a general structure in which these calculations can be performed for all variables.

Domain graphs



The Bayesian network



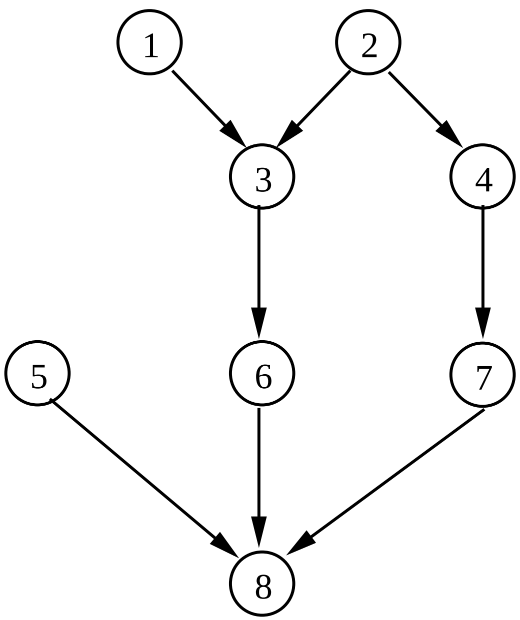
The domain graph

The link (A_2, A_3) is called a moral link.

Domain graphs

Moralization in general

- For all nodes X: Connect pairwise all parents of X with undirected links.
- Replace all original directed links by undirected ones.

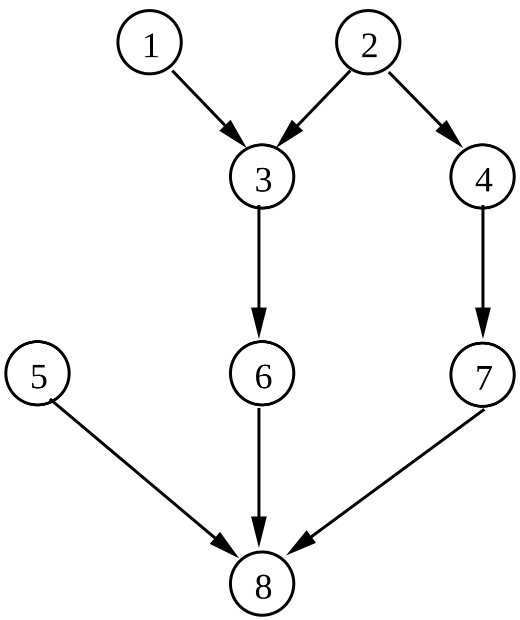


The Bayesian network

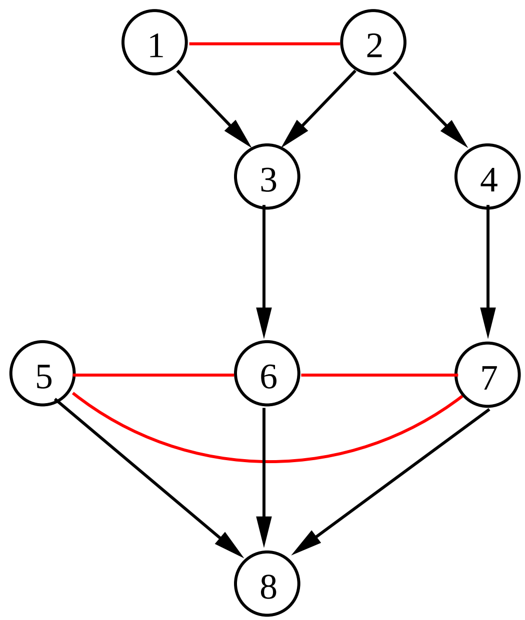
Domain graphs

Moralization in general

- For all nodes X: Connect pairwise all parents of X with undirected links.
- Replace all original directed links by undirected ones.



The Bayesian network

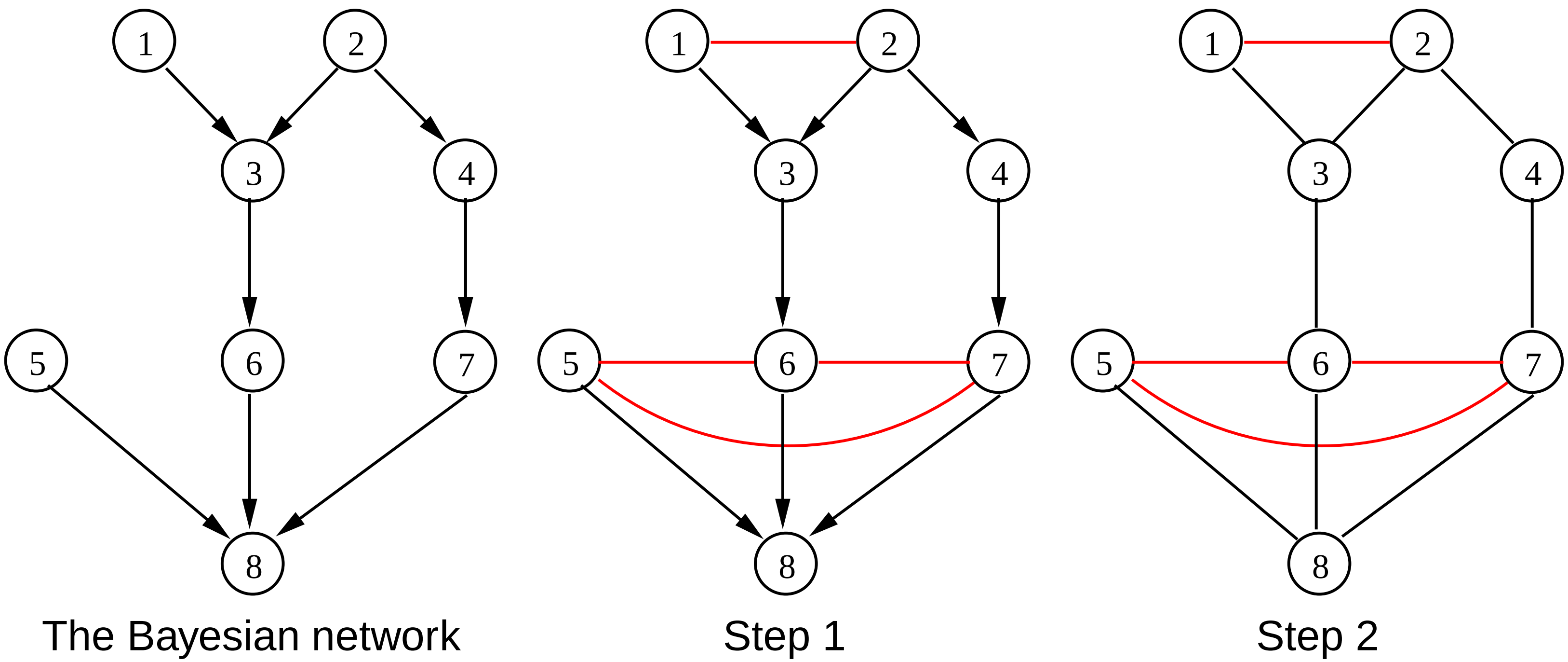


Step 1

Domain graphs

Moralization in general

- For all nodes X: Connect pairwise all parents of X with undirected links.
- Replace all original directed links by undirected ones.

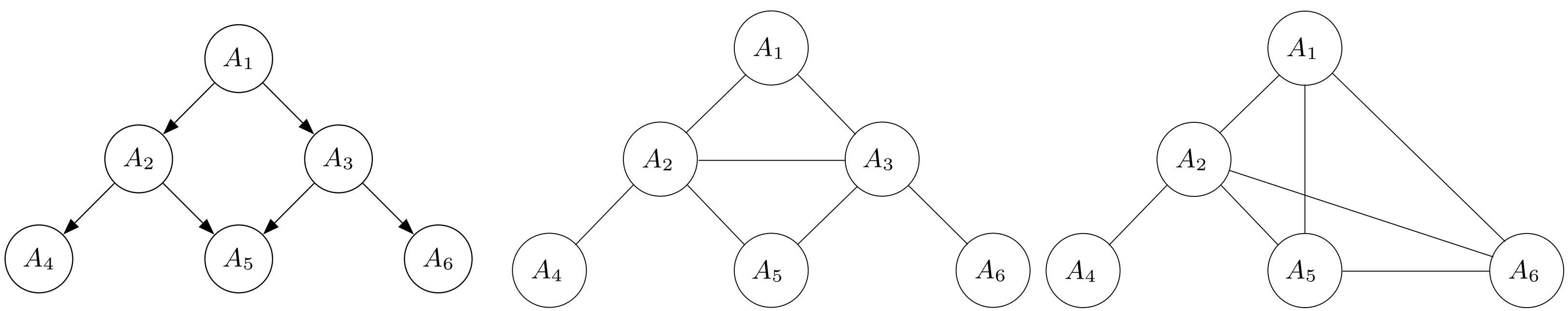


Eliminating a variable

Suppose that when calculating $P(A_4)$ we start off by eliminating A_3:

$$\phi'(A_1, A_2, A_5, A_6) = \sum_{A_3} \phi_3(A_3, A_1)\phi_4(A_4, A_2)\phi_5(A_5, A_2, A_3).$$

Graphically:



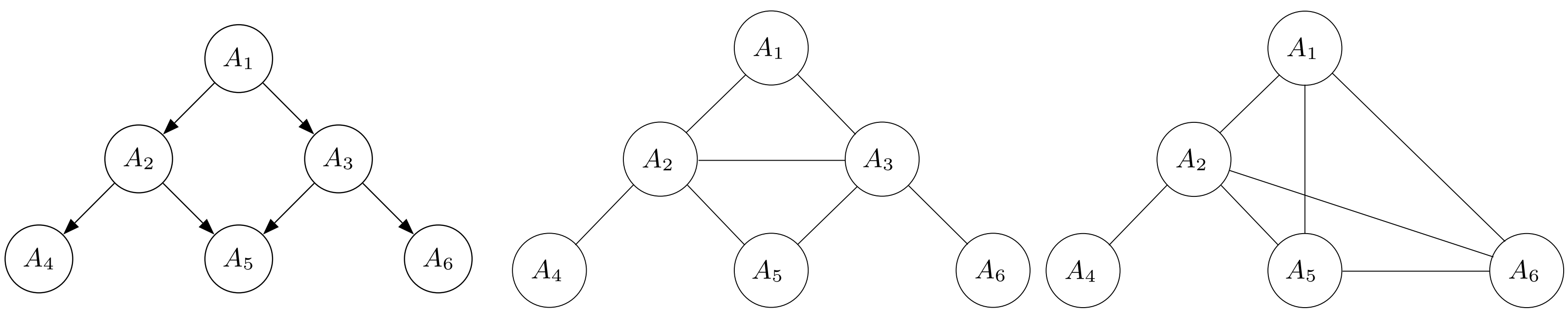
The Bayesian network   The domain graph   The new domain graph

Eliminating a variable

Suppose that when calculating $P(A_4)$ we start off by eliminating A_3:

$$\phi'(A_1, A_2, A_5, A_6) = \sum_{A_3} \phi_3(A_3, A_1)\phi_4(A_4, A_2)\phi_5(A_5, A_2, A_3).$$

Graphically:



The Bayesian network   The domain graph   The new domain graph

Perfect elimination sequences

If all variables can be eliminated without introducing fill-in edges, then the elimination sequence is called perfect. An example could be $A_5, A_6, A_3, A_1, A_2, A_4$.

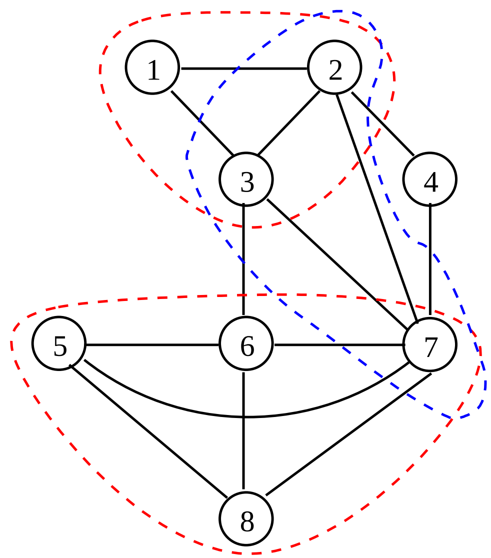
Perfect elimination sequences

Perfect elimination sequences

All perfect elimination sequences produce the same domain set, namely the set of cliques of the domain graph.

Clique

A set of nodes is complete if all nodes are pairwise linked. A complete set is a clique if it is not a subset of another complete set.



Property

Any perfect elimination sequence ending with A is optimal w.r.t. calculating $P(A)$.

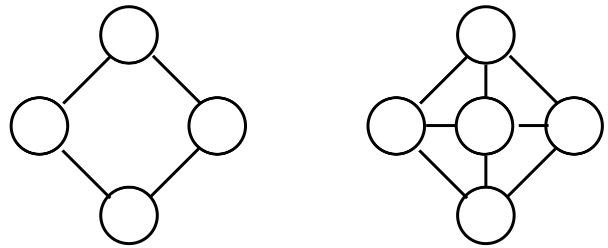
Triangulated graphs

Definition

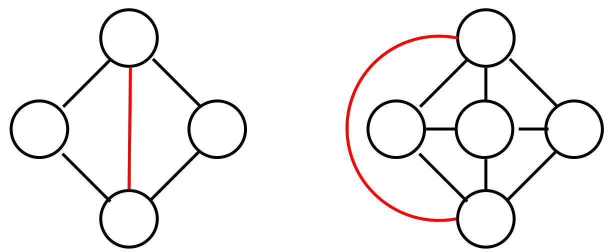
An undirected graph with a perfect elimination sequence is called a triangulated graph.

Example

not triangulated



triangulated



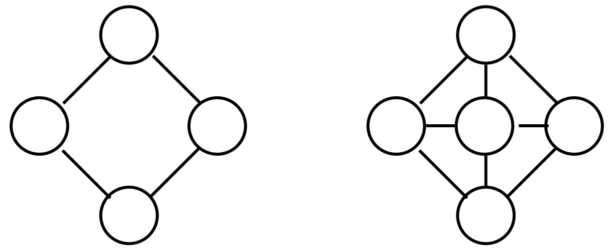
Triangulated graphs

Definition

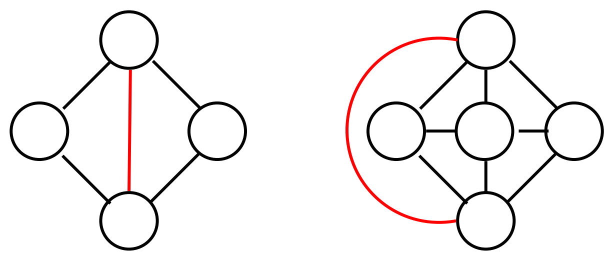
An undirected graph with a perfect elimination sequence is called a triangulated graph.

Example

not triangulated



triangulated



Corollary

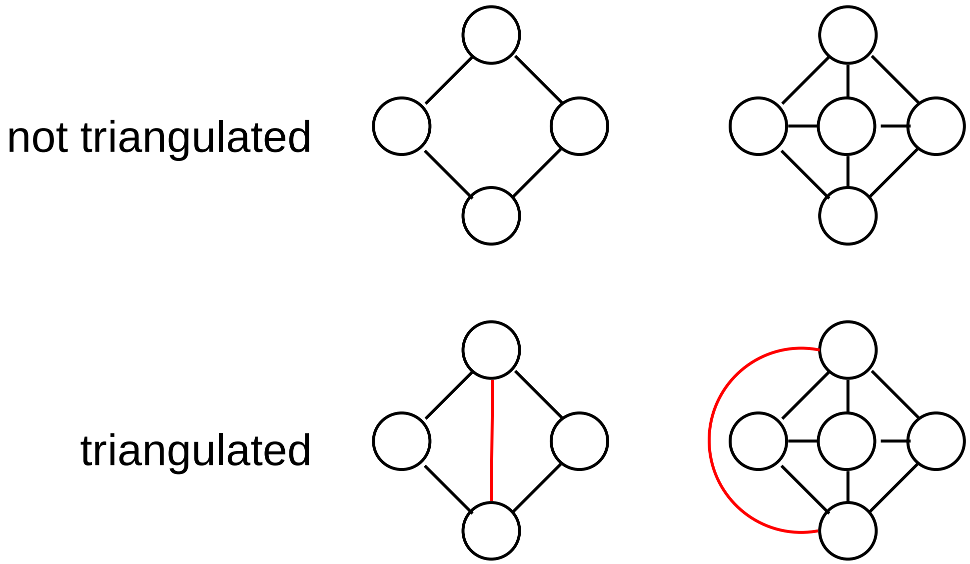
A graph is triangulated iff all nodes can successively be eliminated without introducing fill-in edges.

Triangulated graphs

Definition

An undirected graph with a perfect elimination sequence is called a triangulated graph.

Example



Corollary

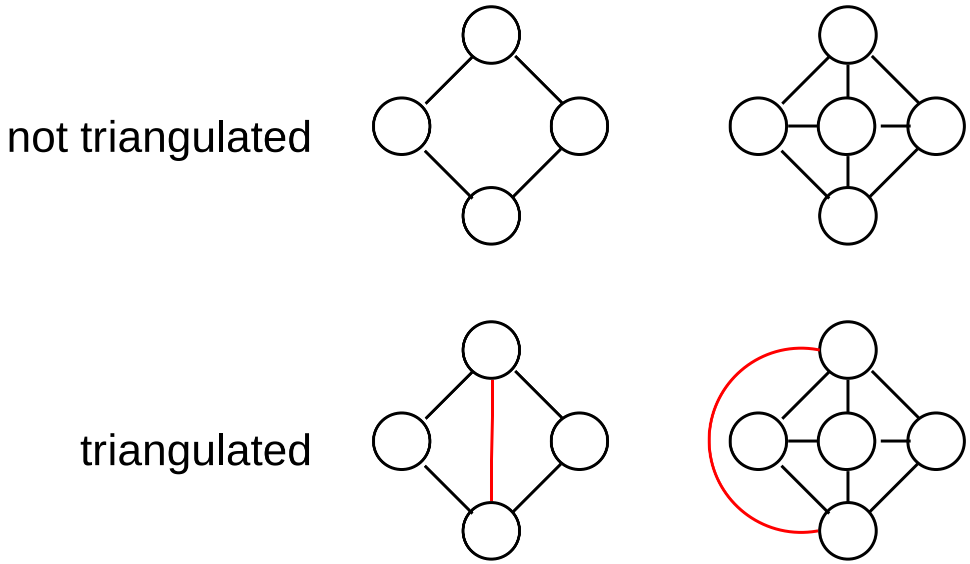
A graph is triangulated iff there does not exist a cycle of length ≥ 4 that is not cut by a cord.

Triangulated graphs

Definition

An undirected graph with a perfect elimination sequence is called a triangulated graph.

Example



Corollary

In a triangulated graph, each variable A has a perfect elimination sequence ending with A.

Triangulated graphs

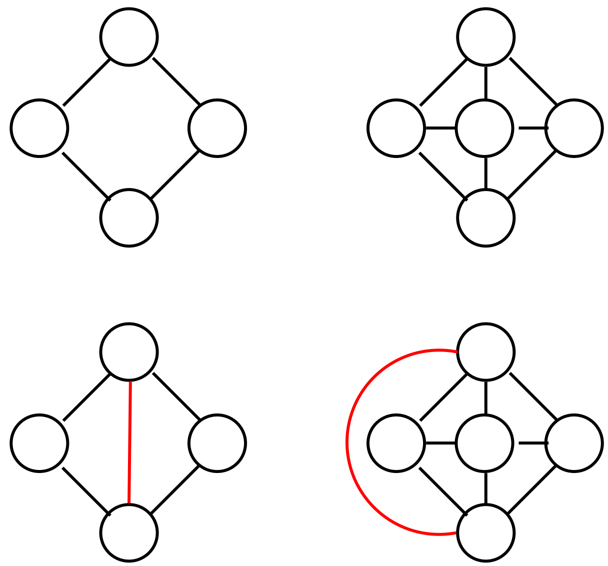
Definition

An undirected graph with a perfect elimination sequence is called a triangulated graph.

Example

not triangulated

triangulated



Assumption

For now we shall assume that the domain graph is triangulated!

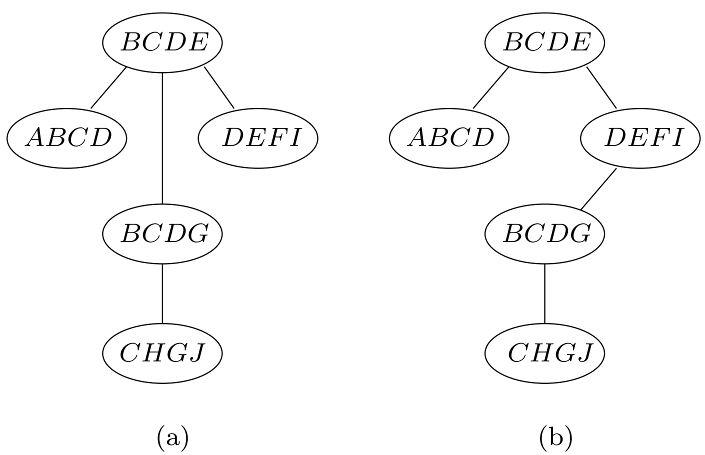
Exact Inference

Join Tree (Junction Tree)

Let $\mathcal{G}$ be the set of cliques from an undirected graph, and let the cliques of $\mathcal{G}$ be organized in a tree T. Then T is a join tree if

- a variable V that is contained in two nodes $\mathbf{C}, \mathbf{C}'$ also is contained in every node on the (unique) path connecting $\mathbf{C}$ and $\mathbf{C}'$.

Example:

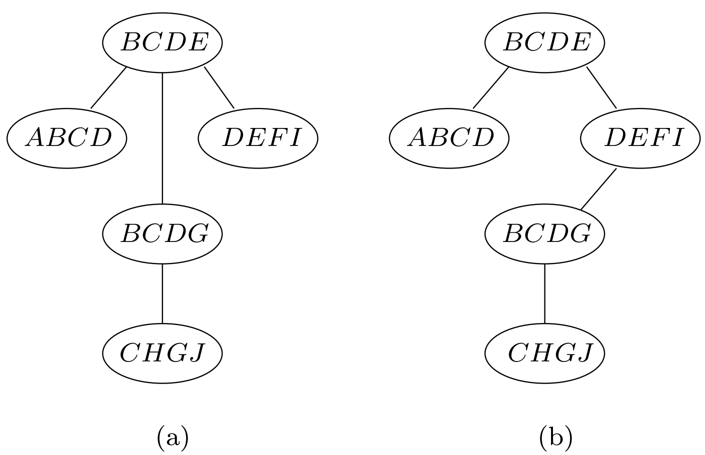


Join Tree (Junction Tree)

Let $\mathcal{G}$ be the set of cliques from an undirected graph, and let the cliques of $\mathcal{G}$ be organized in a tree T. Then T is a join tree if

- a variable V that is contained in two nodes $\mathbf{C}, \mathbf{C}'$ also is contained in every node on the (unique) path connecting $\mathbf{C}$ and $\mathbf{C}'$.

Example:



(a) A join tree

(b) Not a join tree

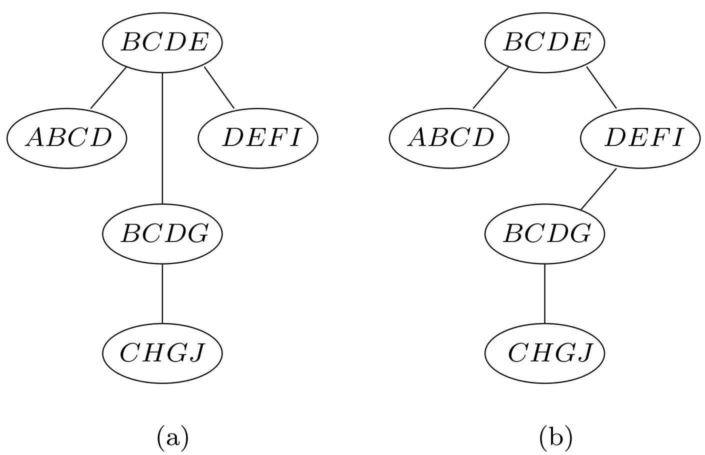
Exact Inference

Join Tree (Junction Tree)

Let $\mathcal{G}$ be the set of cliques from an undirected graph, and let the cliques of $\mathcal{G}$ be organized in a tree T. Then T is a join tree if

- a variable V that is contained in two nodes $\mathbf{C}, \mathbf{C}'$ also is contained in every node on the (unique) path connecting $\mathbf{C}$ and $\mathbf{C}'$.

Example:



(a) A join tree

(b) Not a join tree

Theorem: An undirected graph G is triangulated if and only if the cliques of G can be organized in a join tree.

Join tree construction

From undirected graph to join graph

Define undirected graph $(\mathcal{C}, \mathcal{E}^*)$:

$$\begin{aligned} \mathcal{C}: &\quad \text{Set of cliques in triangulated moral graph} \\ \mathcal{E}^*: &\quad \{(\mathbf{C}_1, \mathbf{C}_2) \mid \mathbf{C}_1 \cap \mathbf{C}_2 \neq \emptyset\} \end{aligned}$$

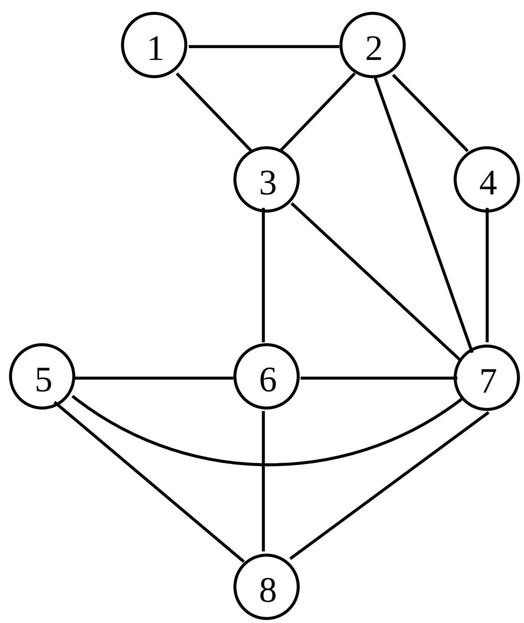
Join tree construction

From undirected graph to join graph

Define undirected graph $(\mathcal{C}, \mathcal{E}^*)$:

$$\mathcal{C} : \quad \text{Set of cliques in triangulated moral graph}$$

$$\mathcal{E}^* : \quad \{(\mathbf{C}_1, \mathbf{C}_2) \mid \mathbf{C}_1 \cap \mathbf{C}_2 \neq \emptyset\}$$



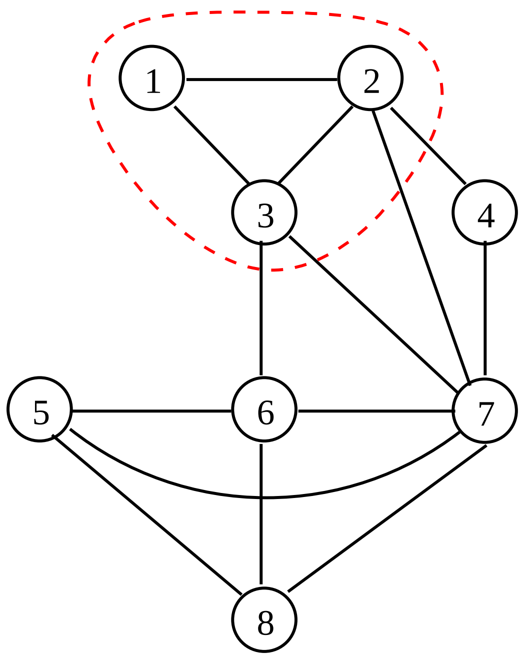
Join tree construction

From undirected graph to join graph

Define undirected graph $(\mathcal{C}, \mathcal{E}^*)$:

$\mathcal{C}$: Set of cliques in triangulated moral graph

$\mathcal{E}^*$: $\{(\mathbf{C}_1, \mathbf{C}_2) \mid \mathbf{C}_1 \cap \mathbf{C}_2 \neq \emptyset\}$

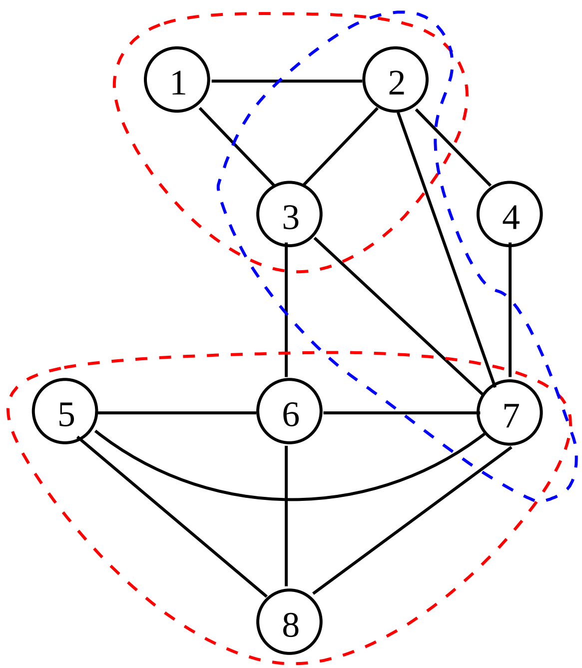


Join tree construction

From undirected graph to join graph

Define undirected graph $(\mathcal{C}, \mathcal{E}^*)$:

$$\begin{array}{ll} \mathcal{C}: & \text{Set of cliques in triangulated moral graph} \\ \mathcal{E}^*: & \{(\mathbf{C}_1, \mathbf{C}_2) \mid \mathbf{C}_1 \cap \mathbf{C}_2 \neq \emptyset\} \end{array}$$

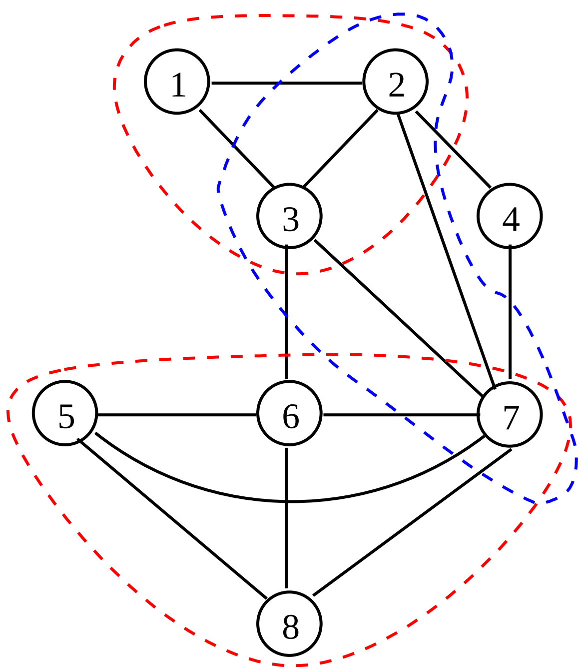


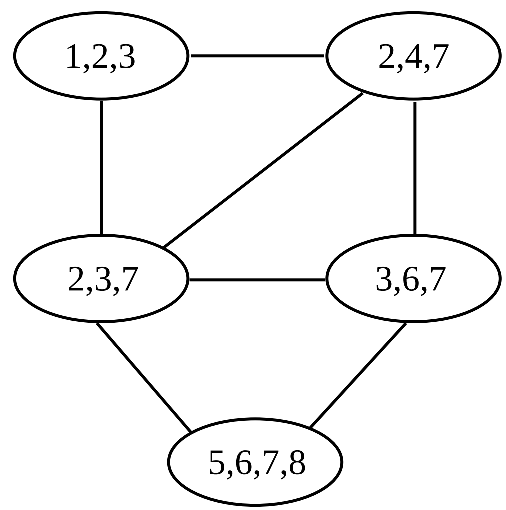
Join tree construction

From undirected graph to join graph

Define undirected graph $(\mathcal{C}, \mathcal{E}^*)$:

$$\mathcal{C}: \quad \text{Set of cliques in triangulated moral graph}$$
$$\mathcal{E}^*: \quad \{(\mathbf{C}_1, \mathbf{C}_2) \mid \mathbf{C}_1 \cap \mathbf{C}_2 \neq \emptyset\}$$





Join tree construction

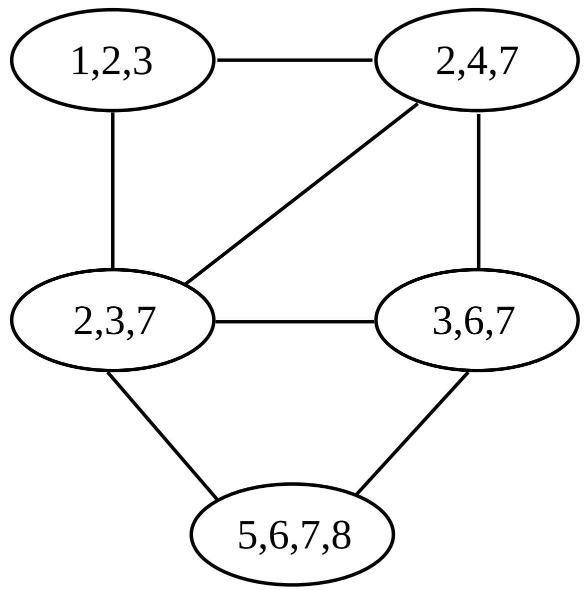
From join graph to join tree

Find maximal spanning tree: for edge $e = (\mathbf{C}_1, \mathbf{C}_2)$ define weight $w(e) := |\mathbf{C}_1 \cap \mathbf{C}_2|$. Let $(\mathcal{C}, \mathcal{E})$ be a maximal spanning tree of $(\mathcal{C}, \mathcal{E}^*)$ with respect to edge weights.

Join tree construction

From join graph to join tree

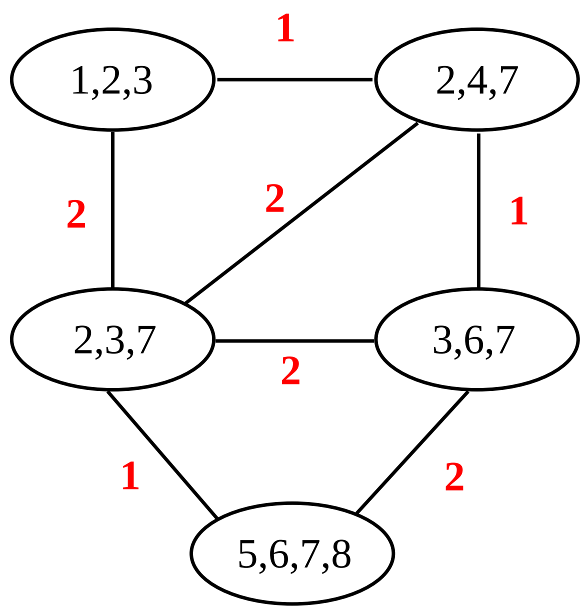
Find maximal spanning tree: for edge $e = (\mathbf{C}_1, \mathbf{C}_2)$ define weight $w(e) := |\mathbf{C}_1 \cap \mathbf{C}_2|$. Let $(\mathcal{C}, \mathcal{E})$ be a maximal spanning tree of $(\mathcal{C}, \mathcal{E}^*)$ with respect to edge weights.



Join tree construction

From join graph to join tree

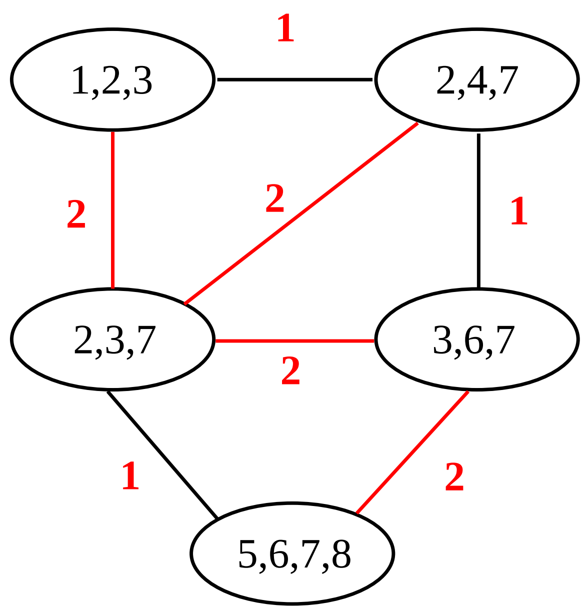
Find maximal spanning tree: for edge $e = (\mathbf{C}_1, \mathbf{C}_2)$ define weight $w(e) := |\mathbf{C}_1 \cap \mathbf{C}_2|$. Let $(\mathcal{C}, \mathcal{E})$ be a maximal spanning tree of $(\mathcal{C}, \mathcal{E}^*)$ with respect to edge weights.



Join tree construction

From join graph to join tree

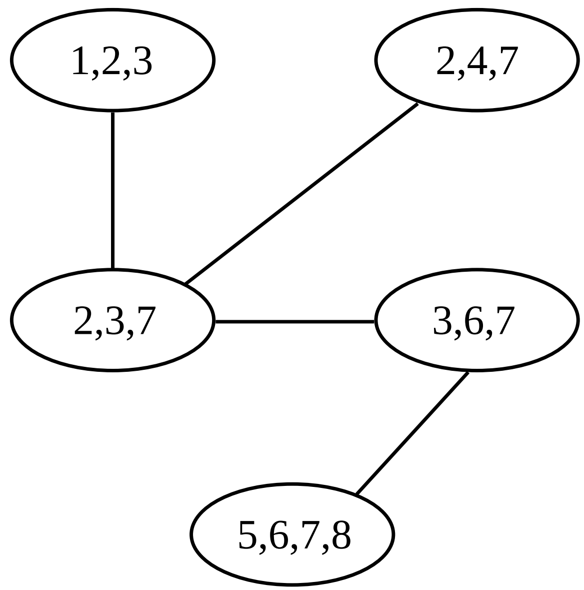
Find maximal spanning tree: for edge $e = (\mathbf{C}_1, \mathbf{C}_2)$ define weight $w(e) := |\mathbf{C}_1 \cap \mathbf{C}_2|$. Let $(\mathcal{C}, \mathcal{E})$ be a maximal spanning tree of $(\mathcal{C}, \mathcal{E}^*)$ with respect to edge weights.



Join tree construction

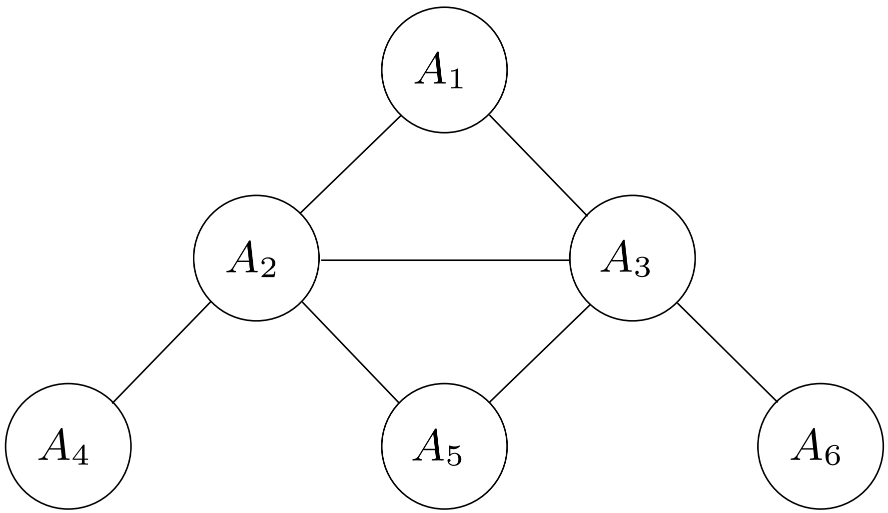
From join graph to join tree

Find maximal spanning tree: for edge $e = (\mathbf{C}_1, \mathbf{C}_2)$ define weight $w(e) := |\mathbf{C}_1 \cap \mathbf{C}_2|$. Let $(\mathcal{C}, \mathcal{E})$ be a maximal spanning tree of $(\mathcal{C}, \mathcal{E}^*)$ with respect to edge weights.



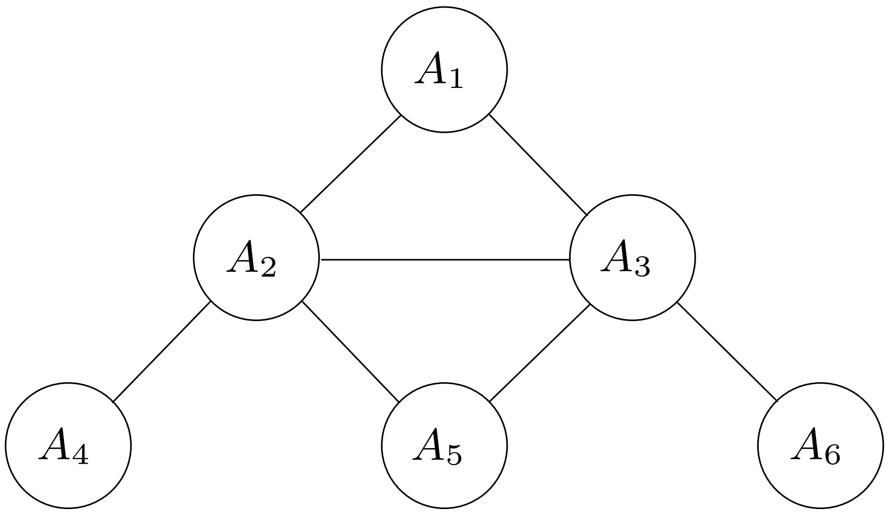
Join tree construction

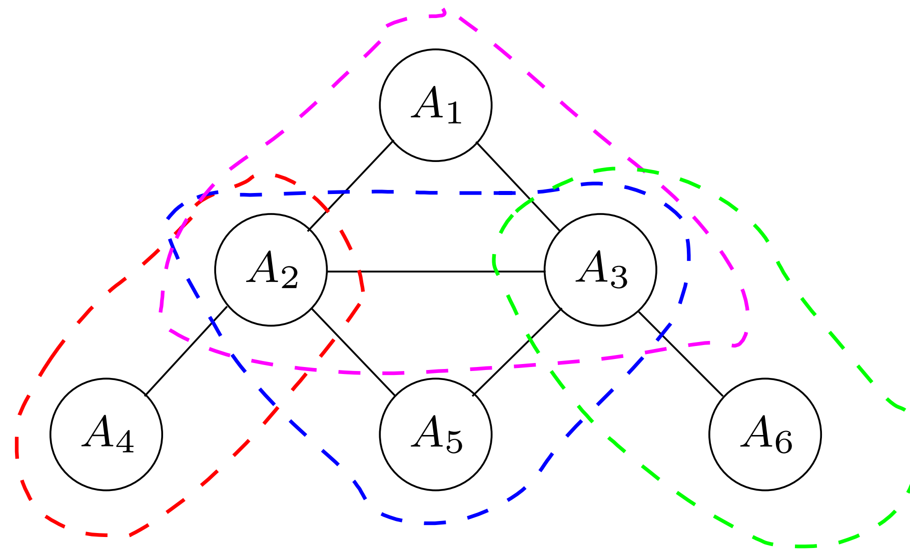
Another example



Join tree construction

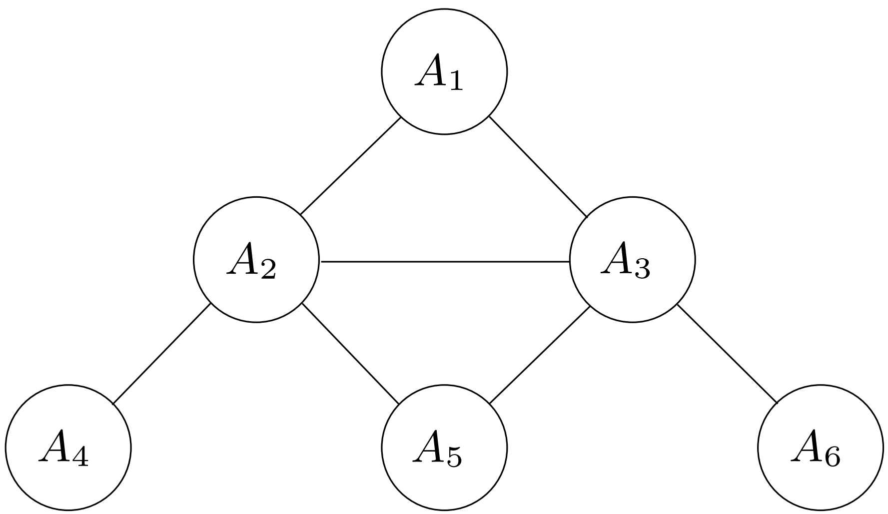
Another example

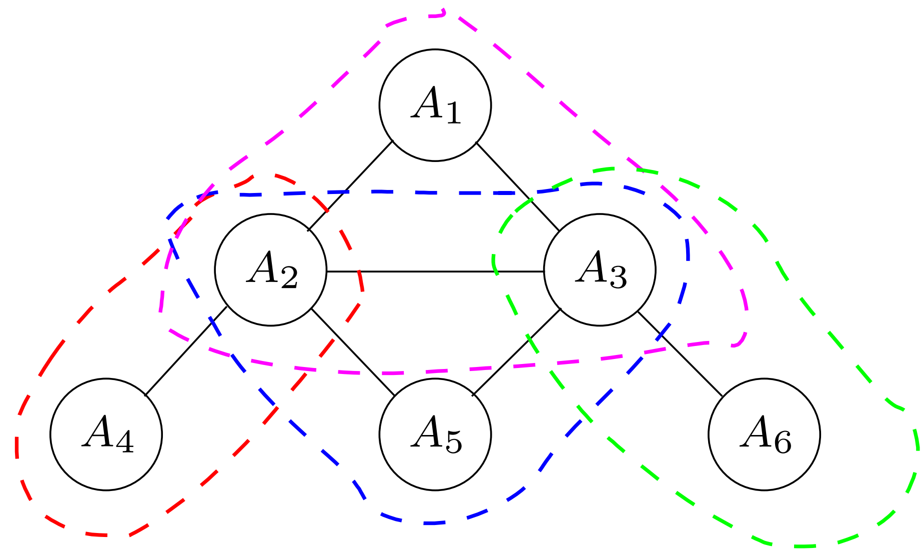


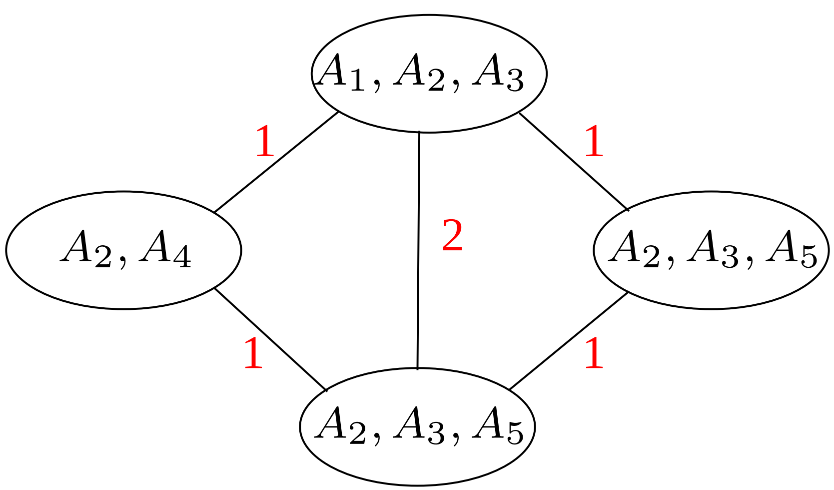


Join tree construction

Another example

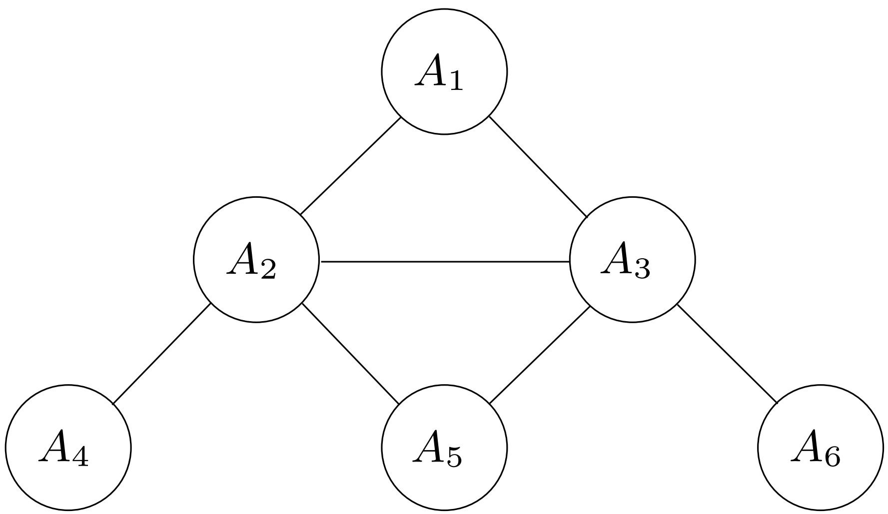


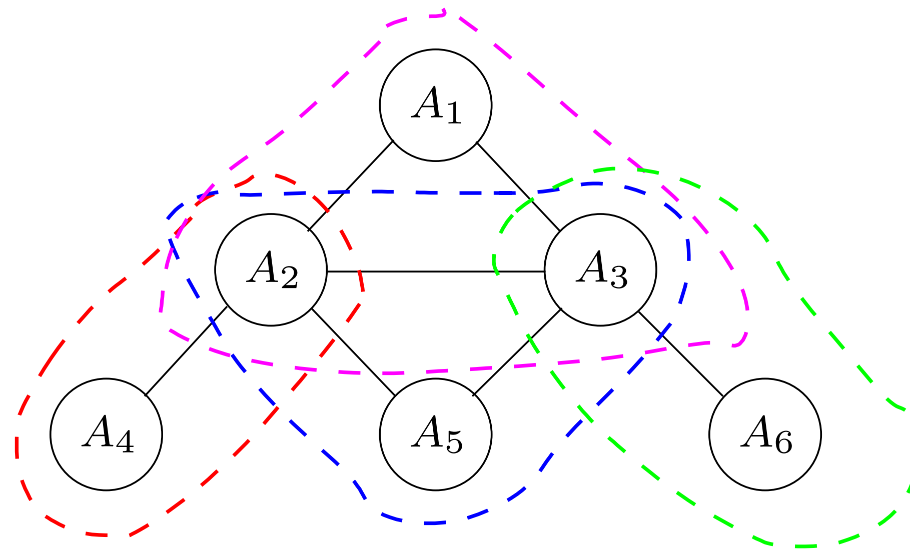


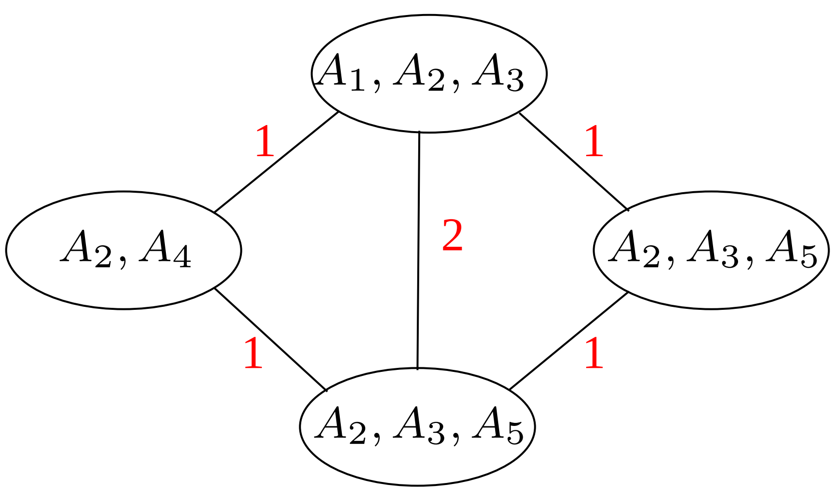


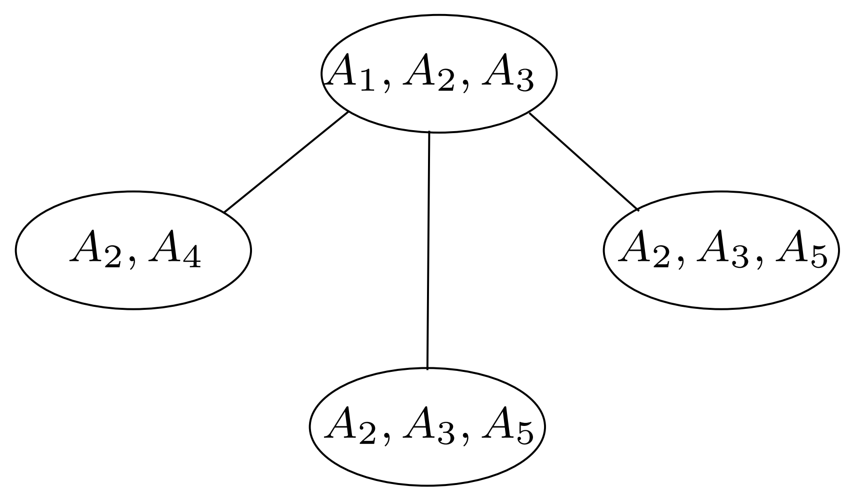
Join tree construction

Another example









Junction tree construction

Correctness

Let $(\mathbf{V}, \rightarrow)$ be a triangulated graph, $(\mathcal{C}, \mathcal{E}^*)$. Every maximal spanning tree $(\mathcal{C}, \mathcal{E})$ of $(\mathcal{C}, \mathcal{E}^*)$ is a join tree for $\mathbf{V}$ (Jensen,Jensen 1994).

If $(\mathbf{V}, \rightarrow)$ is the triangulated moral graph of a Bayesian network, then there exists for every $V \in \mathbf{V}$ a clique $\mathbf{C} \in \mathcal{C}$ with $\{V\} \cup \mathrm{pa}(V) \subseteq \mathbf{C}$.

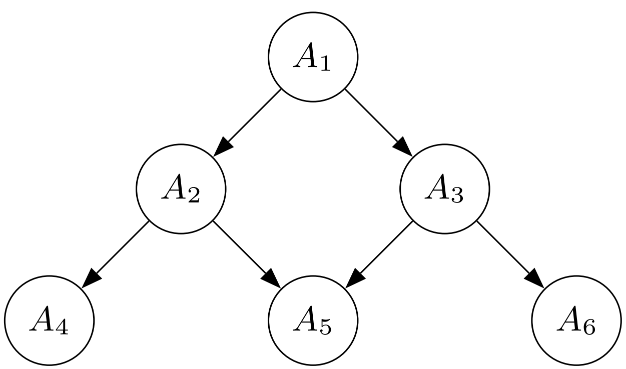
Propagation in junction trees

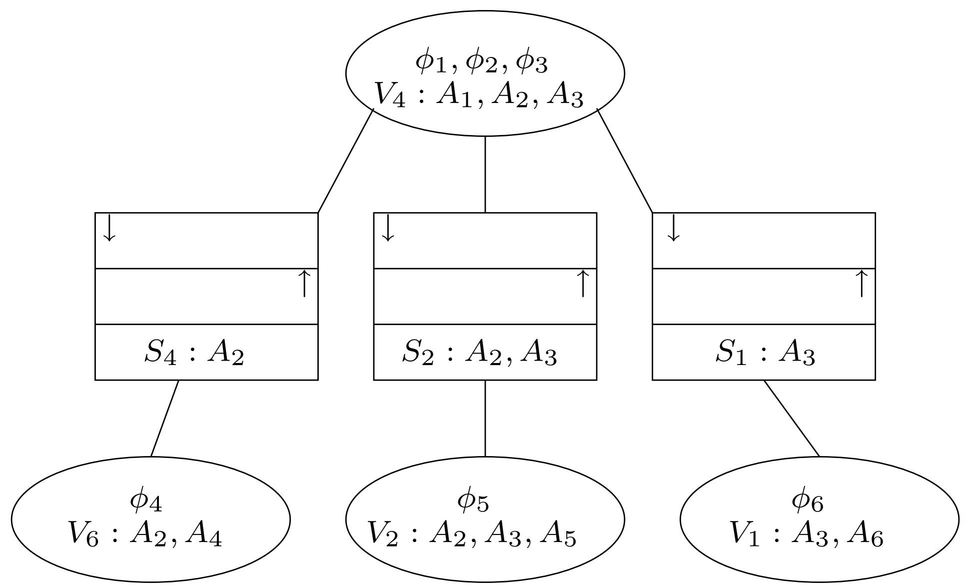
Initialization

Let T be a join tree for the domain graph G with potentials Φ. A junction tree for G consists of T with the additions:

- each potential ϕ from Φ is assigned to a clique containing $\mathrm{dom}\,(\phi)$.
- each link has a separator attached.
- each separator contains two mailboxes, one for each direction.

Example



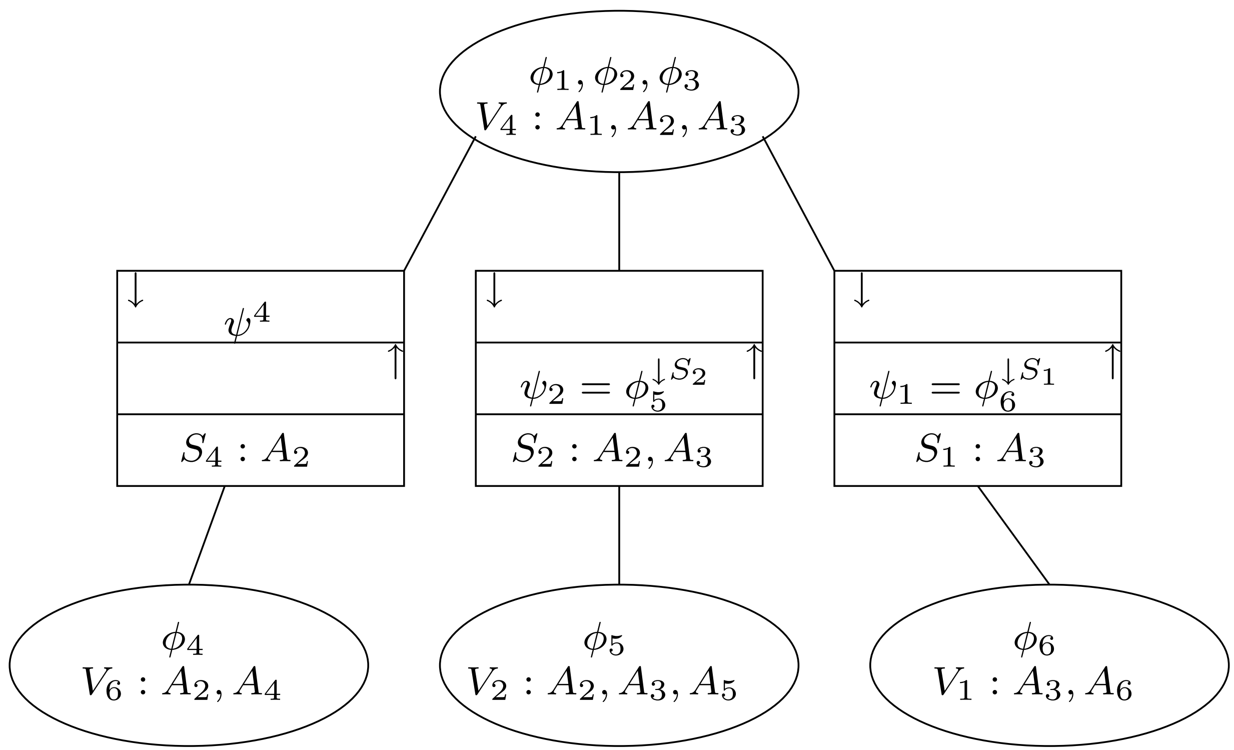


Propagation in junction trees

Propagation

To calculate $P(A_4)$ we find a clique (V_6) containing A_4 and send messages to that clique.

Example

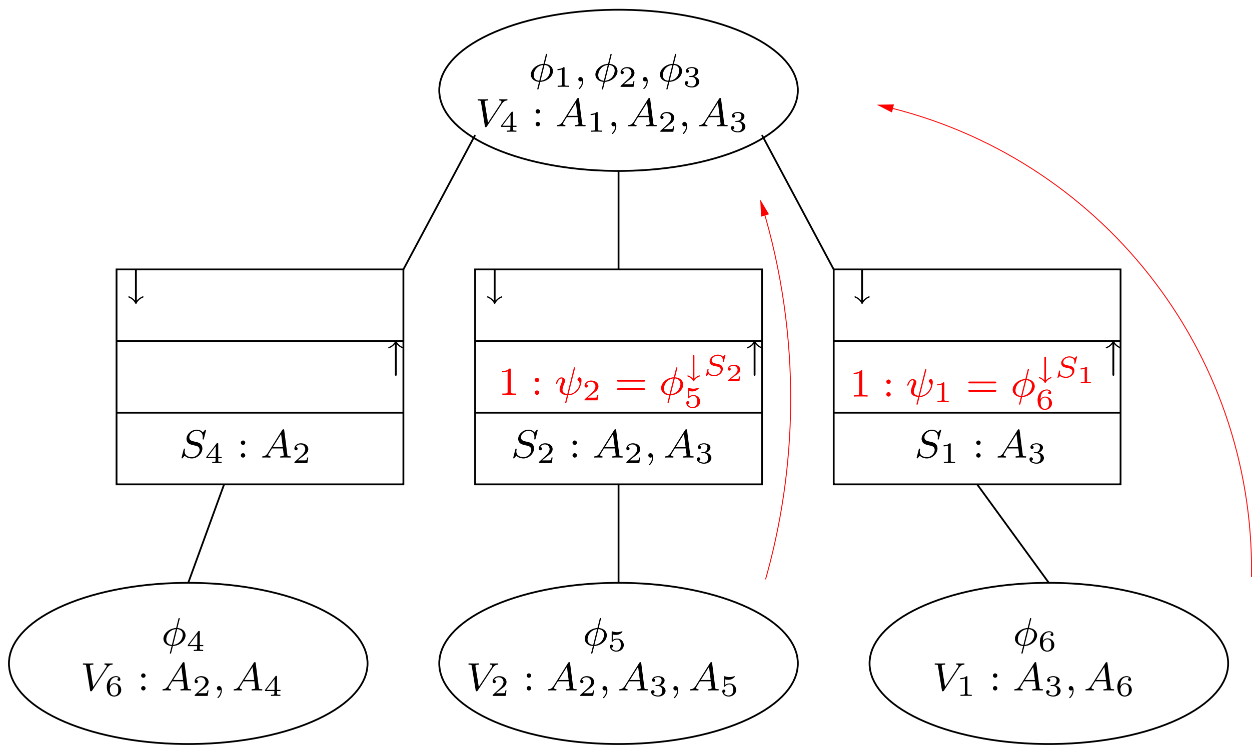


Propagation in junction trees

Propagation

To calculate $P(A_4)$ we find a clique (V_6) containing A_4 and send messages to that clique.

Example

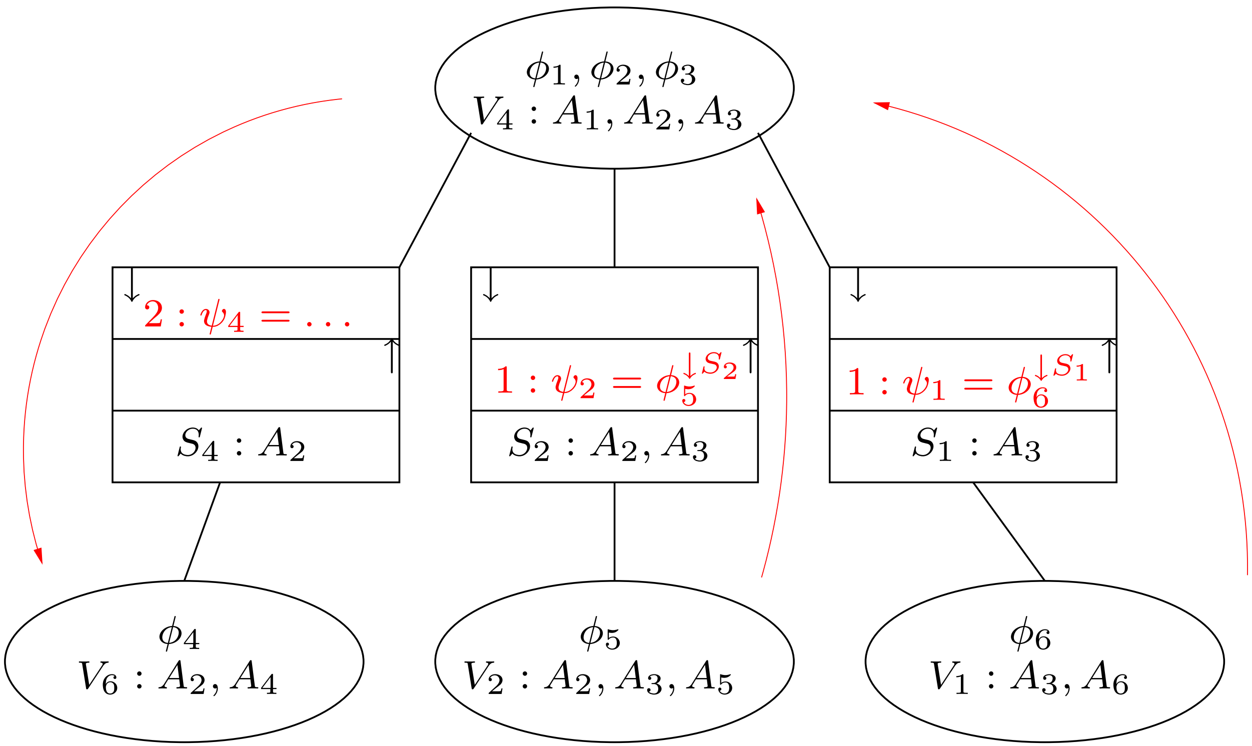


Propagation in junction trees

Propagation

To calculate $P(A_4)$ we find a clique (V_6) containing A_4 and send messages to that clique.

Example



$\psi_4 = \sum_{A_1} \phi_1 \phi_2 \sum_{A_3} \phi_3 \psi_1 \psi_2$

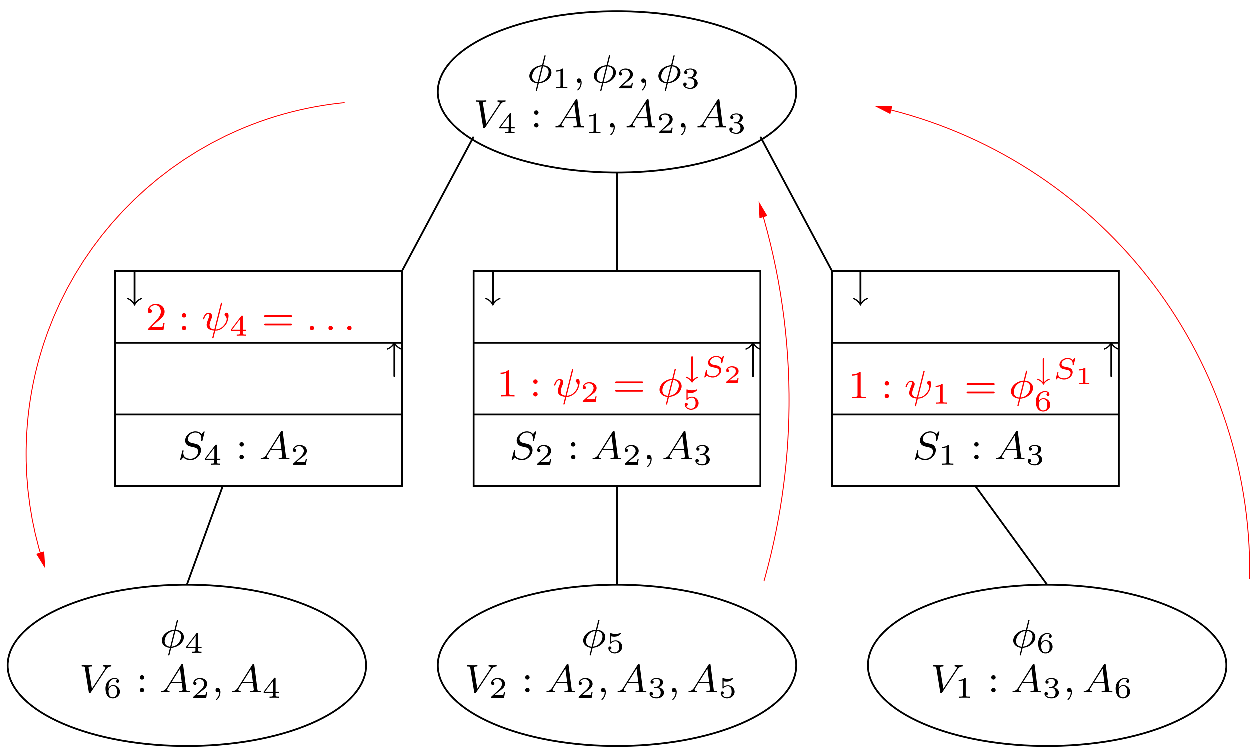
Now evidence has been collected to V_4 and we get $P(A_4) = \sum_{A_2} \phi_4 \psi_4$.

Propagation in junction trees

Propagation

To calculate $P(A_i)$ for any A_i we send messages away from V_6.

Example continued



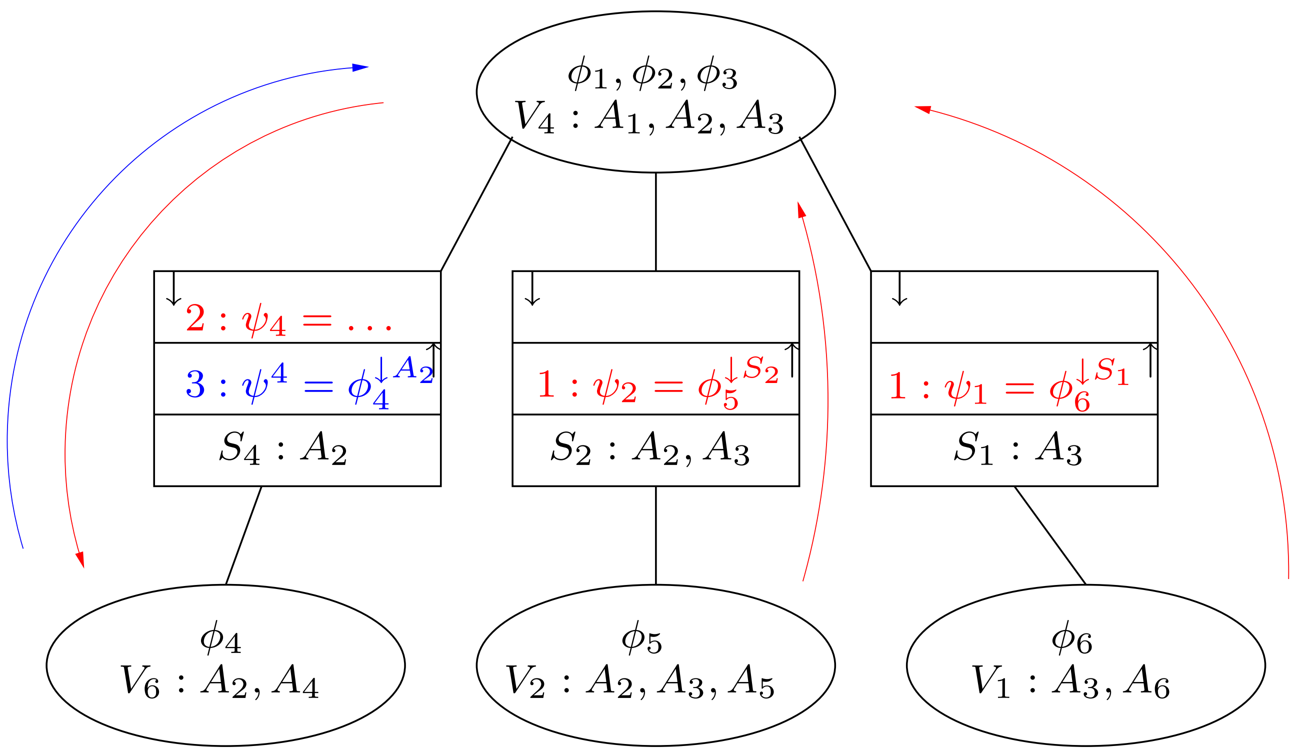
$$\psi_4 = \sum_{A_1} \phi_1 \phi_2 \sum_{A_3} \phi_3 \psi_1 \psi_2$$

Propagation in junction trees

Propagation

To calculate $P(A_i)$ for any A_i we send messages away from V_6.

Example continued

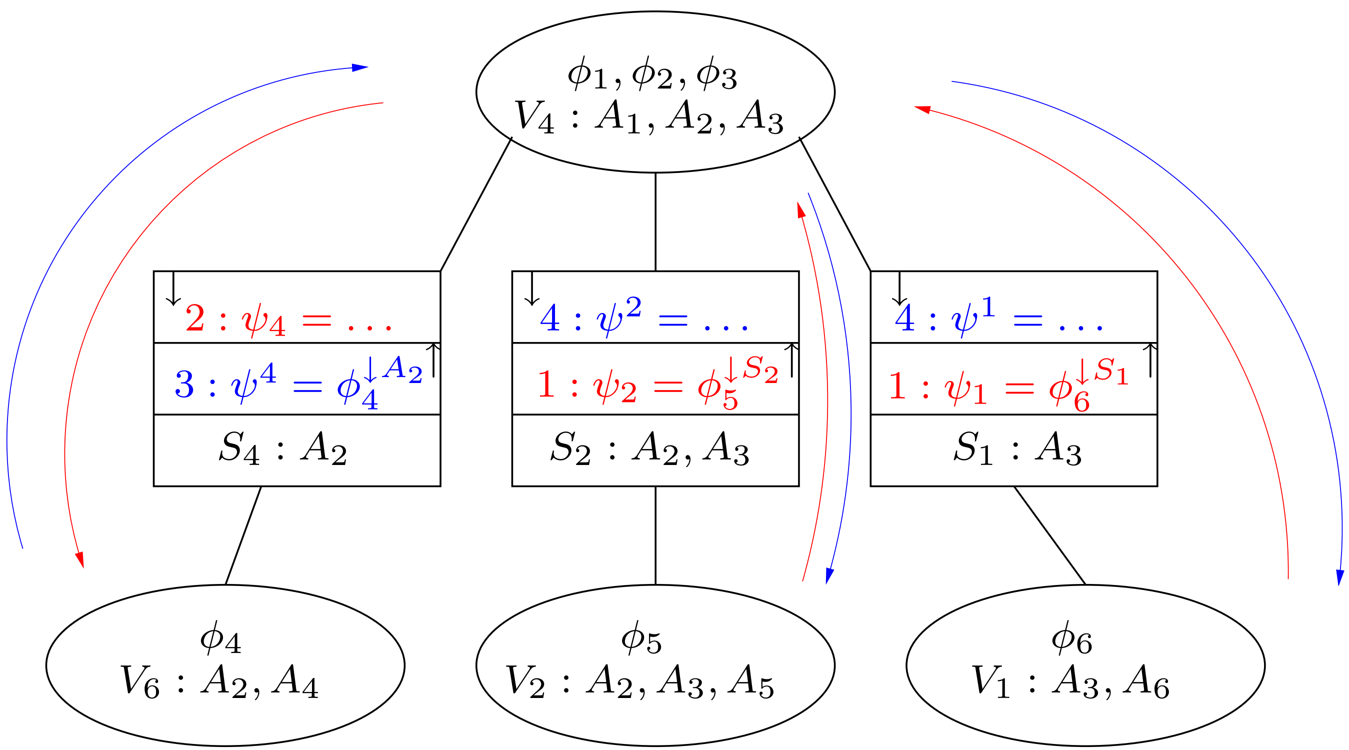


Propagation in junction trees

Propagation

To calculate $P(A_i)$ for any A_i we send messages away from V_6.

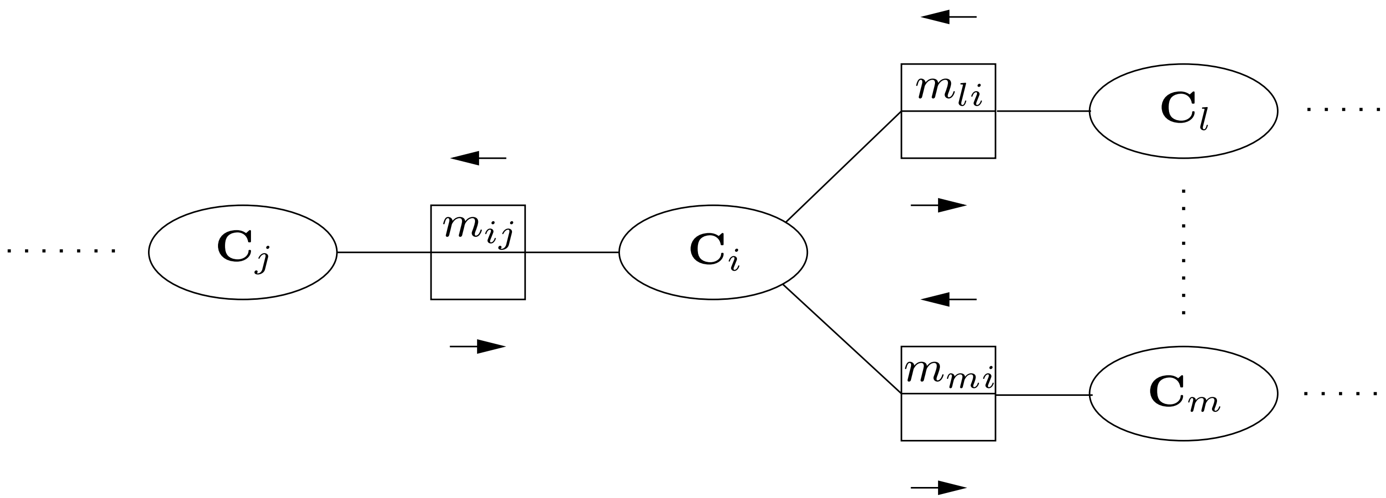
Example continued



$$\psi^2 = \sum_{A_1} \psi_1 \phi_1 \phi_2 \phi_3 \psi^4 \sim \{\psi^4, \sum_{A_1} \phi_1 \phi_2 \phi_3, \psi_1\}$$
$$\psi^1 = \sum_{A_2} \psi_2 \psi_4 \sum_{A_1} \phi_1 \phi_2 \phi_3$$

Propagation in junction trees

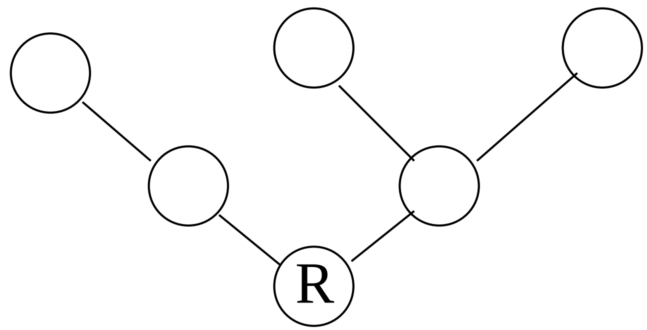
In general: Sending a message from $\mathbf{C}_i$ to $\mathbf{C}_j$



$$m_{ij} := \sum_{\mathbf{C}_j \setminus \mathbf{C}_i} \prod_{\phi \in \Phi_{\mathbf{C}_i}} \phi \prod_{\mathbf{C}_k \in \mathrm{nd}(\mathbf{C}_i)} m_{ki}$$

Exact Inference

Messages passing in general

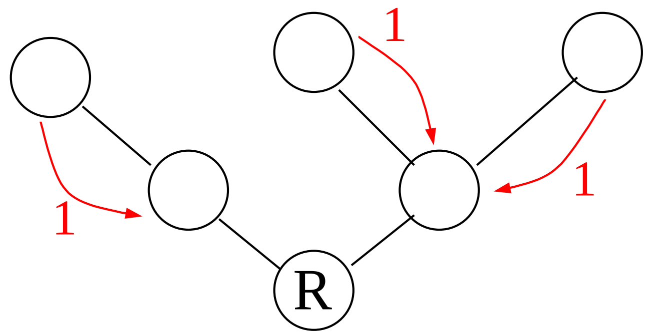


Collect and distribute messages

i) Collect messages to a preselected root R

Exact Inference

Messages passing in general

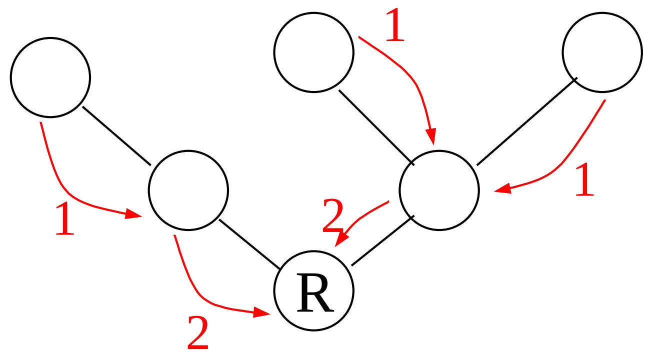


Collect and distribute messages

i) Collect messages to a preselected root R

Exact Inference

Messages passing in general

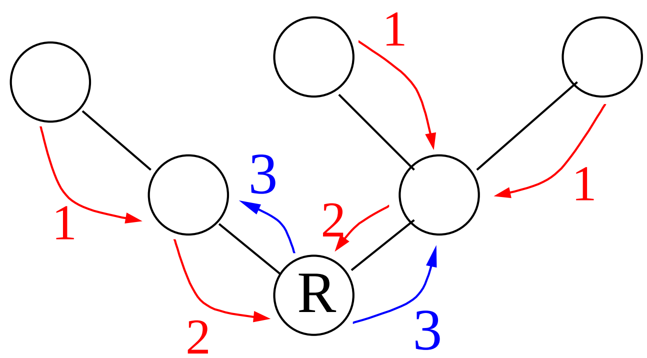


Collect and distribute messages

i) Collect messages to a preselected root R

Exact Inference

Messages passing in general



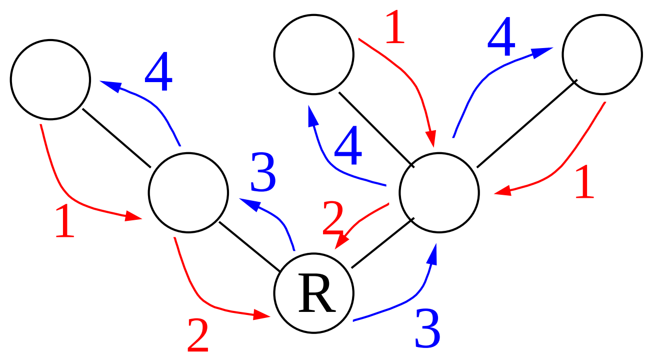
Collect and distribute messages

i) Collect messages to a preselected root R

ii) Distribute messages away from the root R

Exact Inference

Messages passing in general



Collect and distribute messages

i) Collect messages to a preselected root R

ii) Distribute messages away from the root R

Propagation in junction trees

Theorem

Let the junction tree T represent the Bayesian network BN over the universe U and with evidence e. Assume that T is full.

1. Let V be a clique with set of potentials Φ_V, and let $S_1, \ldots, S_k$ be V's neighboring separators and with V-directed messages $\Psi_1, \ldots, \Psi_k$. Then

$$P(V, e) = \prod \Phi_V \prod \Psi_1 \cdot \ldots \cdot \prod \Psi_k.$$

2. Let S be a separator with the sets Ψ_S and Ψ^S in the mailboxes. Then

$$P(S, e) = \prod \Psi_S \prod \Psi^S.$$

Propagation in junction trees

Theorem

Let the junction tree T represent the Bayesian network BN over the universe U and with evidence e. Assume that T is full.

1. Let V be a clique with set of potentials Φ_V, and let $S_1, \ldots, S_k$ be V's neighboring separators and with V-directed messages $\Psi_1, \ldots, \Psi_k$. Then

$$P(V, e) = \prod \Phi_V \prod \Psi_1 \cdot \ldots \cdot \prod \Psi_k.$$

2. Let S be a separator with the sets Ψ_S and Ψ^S in the mailboxes. Then

$$P(S, e) = \prod \Psi_S \prod \Psi^S.$$

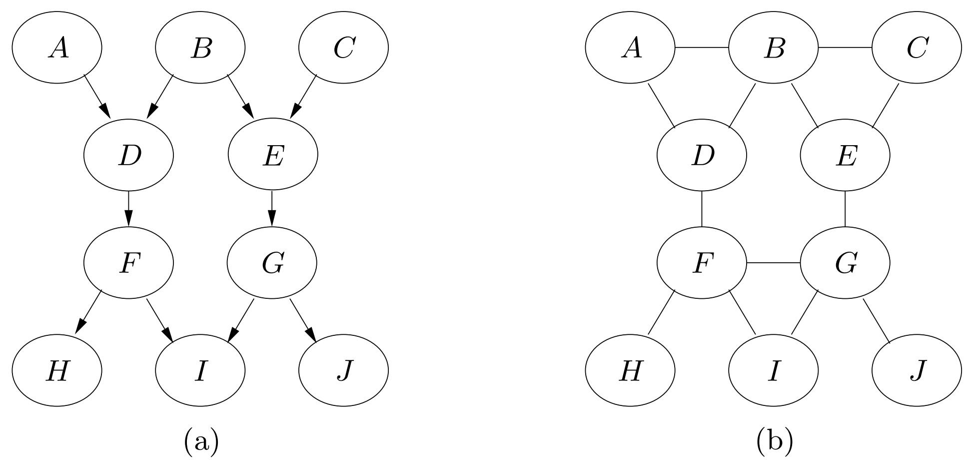
Evidence

Evidence is inserted by adding corresponding $0/1$-potentials to the appropriate cliques.

Triangulation of graphs

So far we have assumed that the domain graph is triangulated. If this is not the case, then we embed it in a triangulated graph.

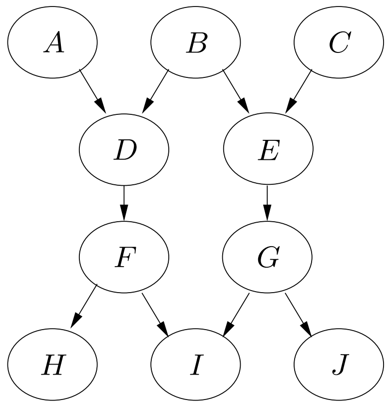
Example



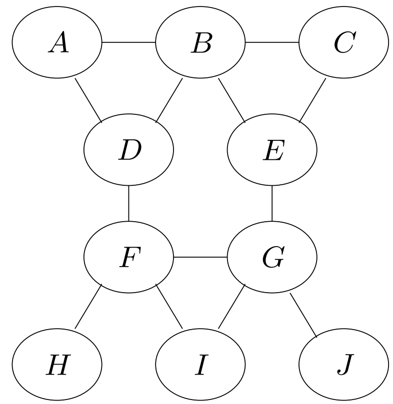
Triangulation of graphs

So far we have assumed that the domain graph is triangulated. If this is not the case, then we embed it in a triangulated graph.

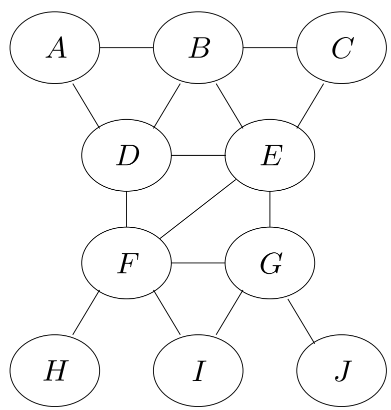
Example



(a)

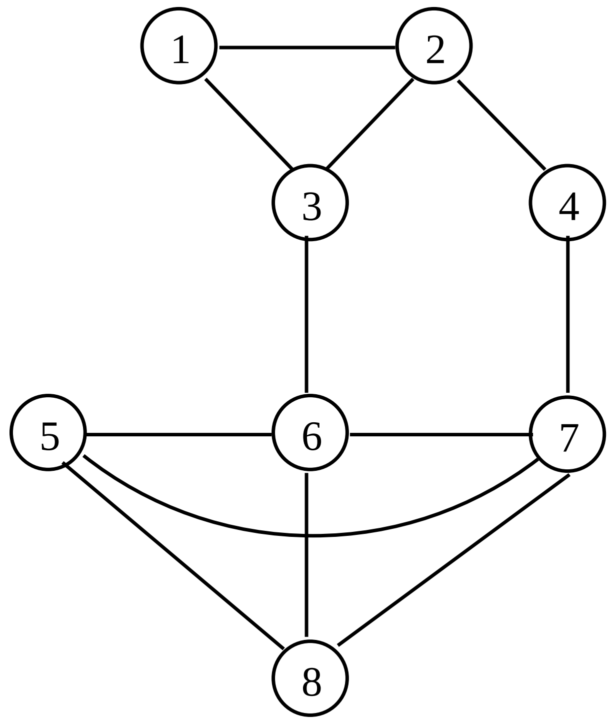


(b)

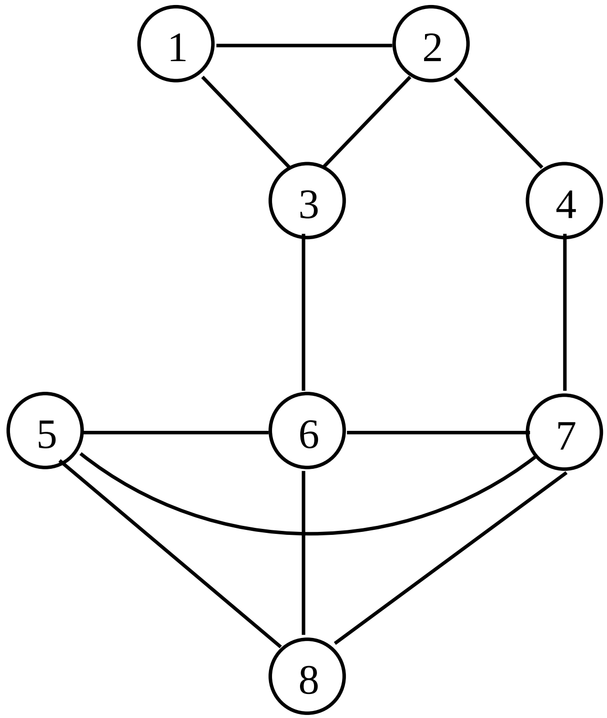


A triangulated graph extending (b)

Triangulation of graphs



Triangulation of graphs

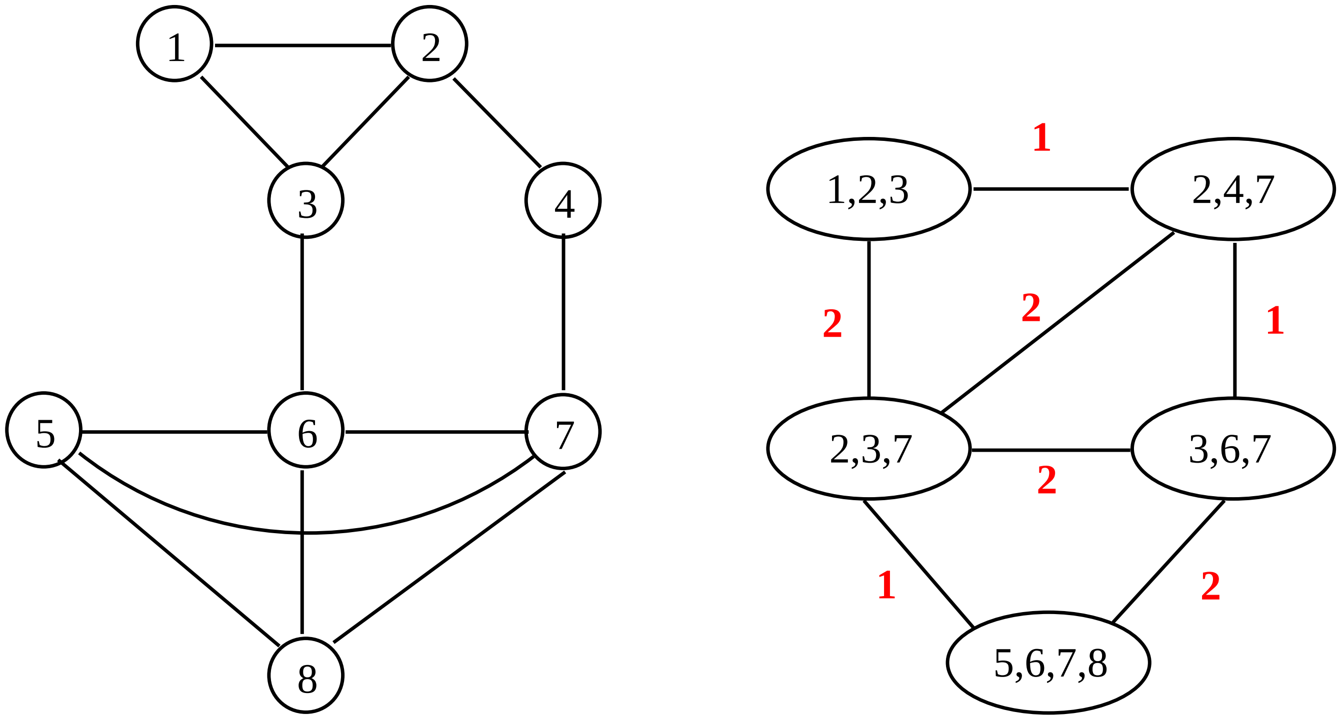


A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs

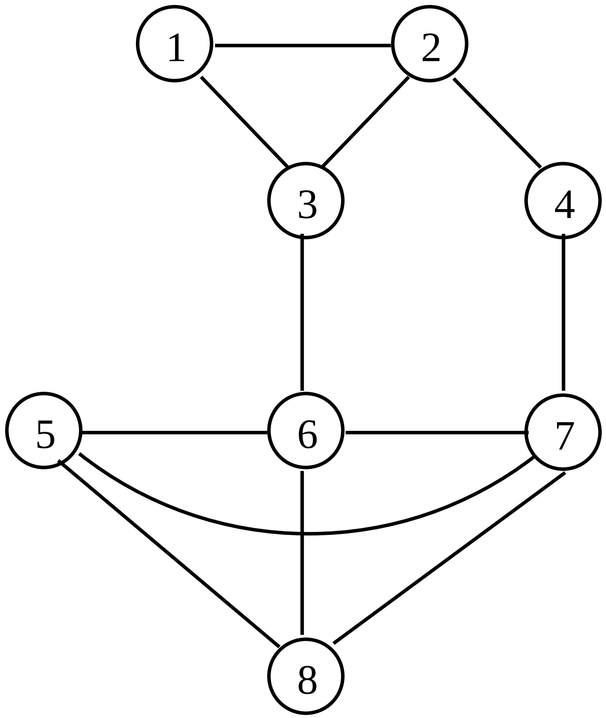


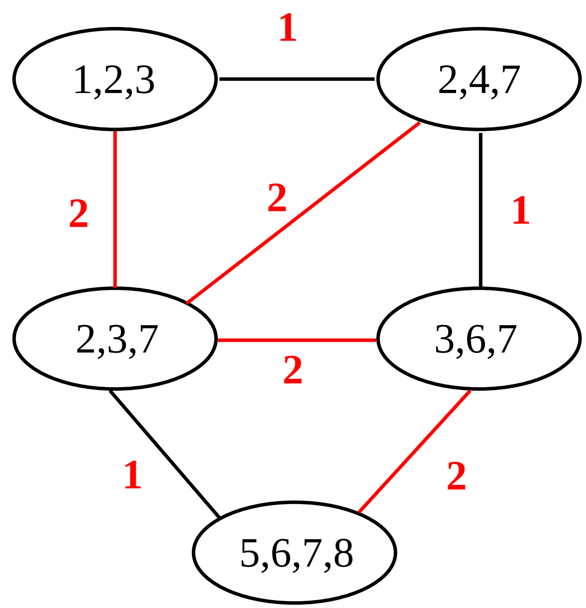
A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs



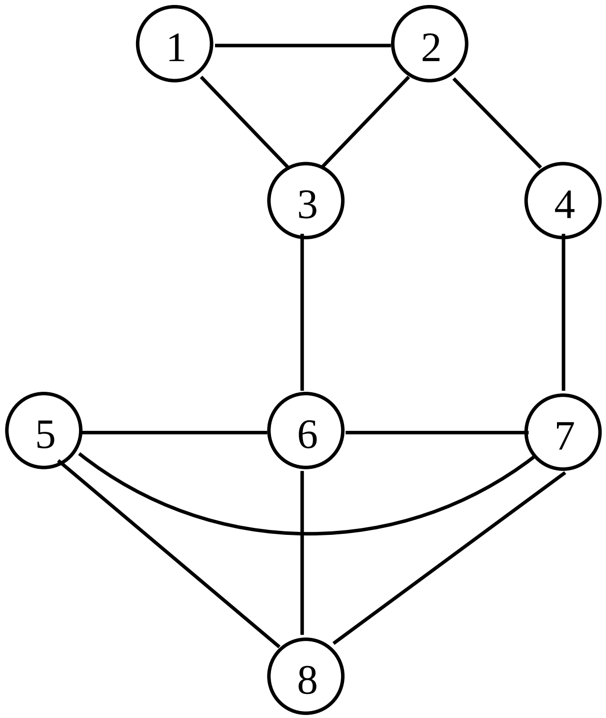


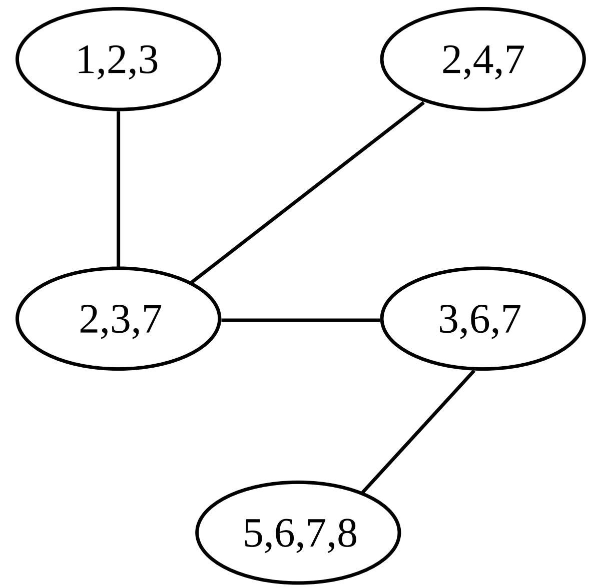
A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs



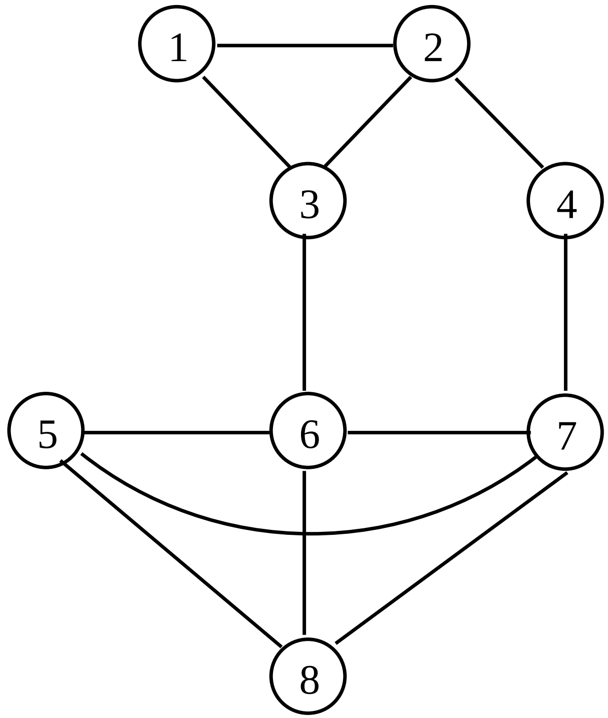


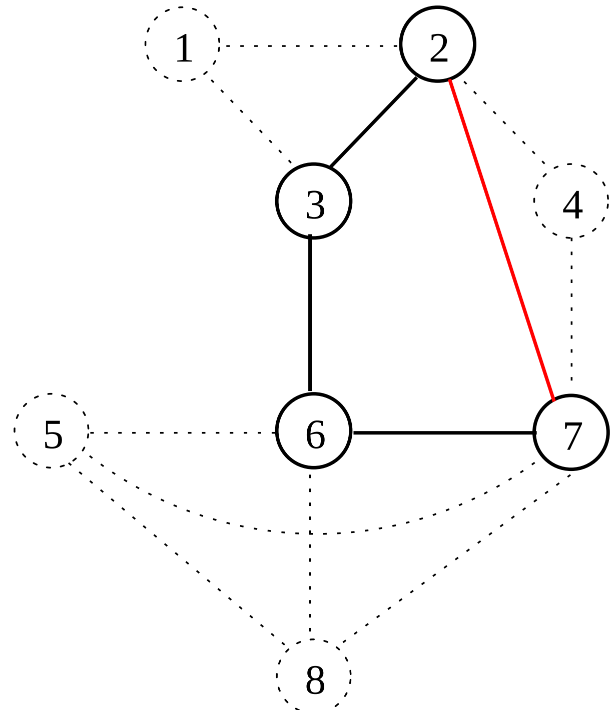
A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs



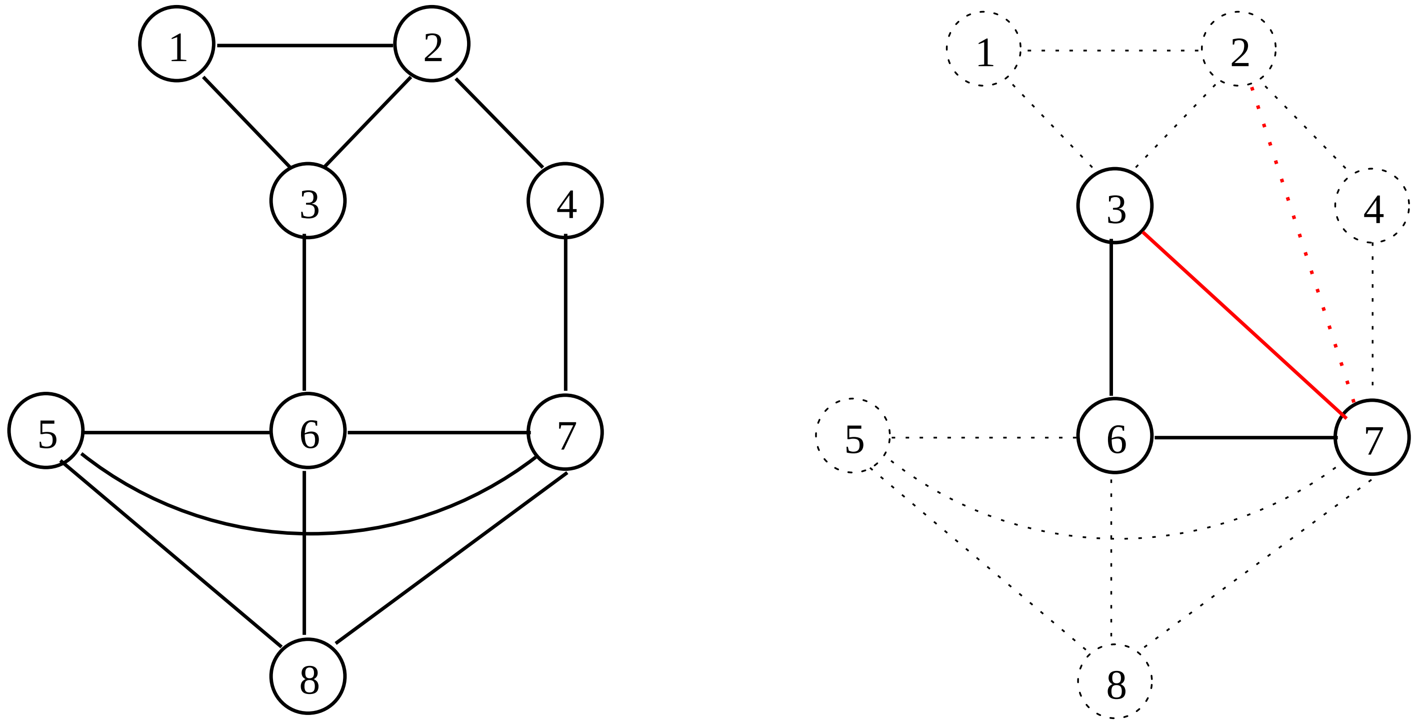


A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs

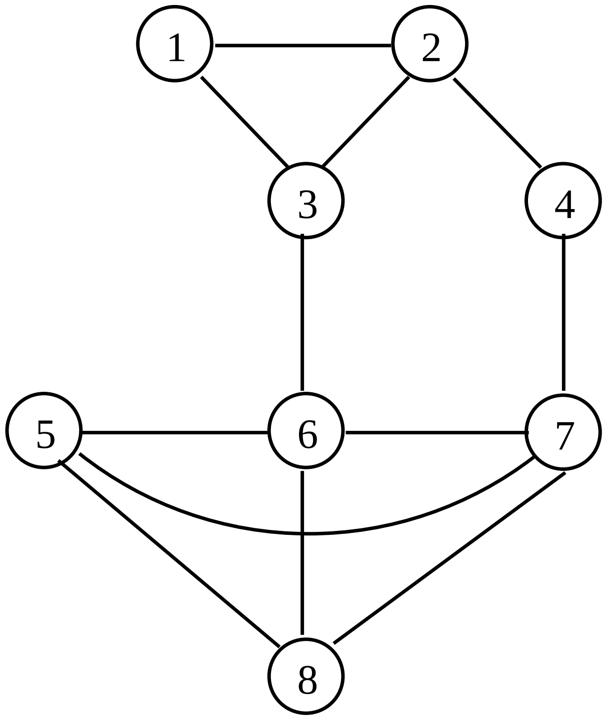


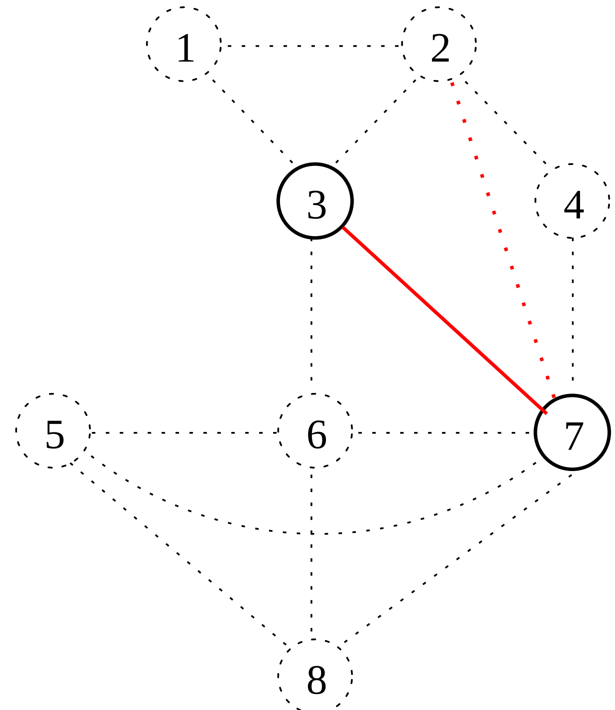
A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs



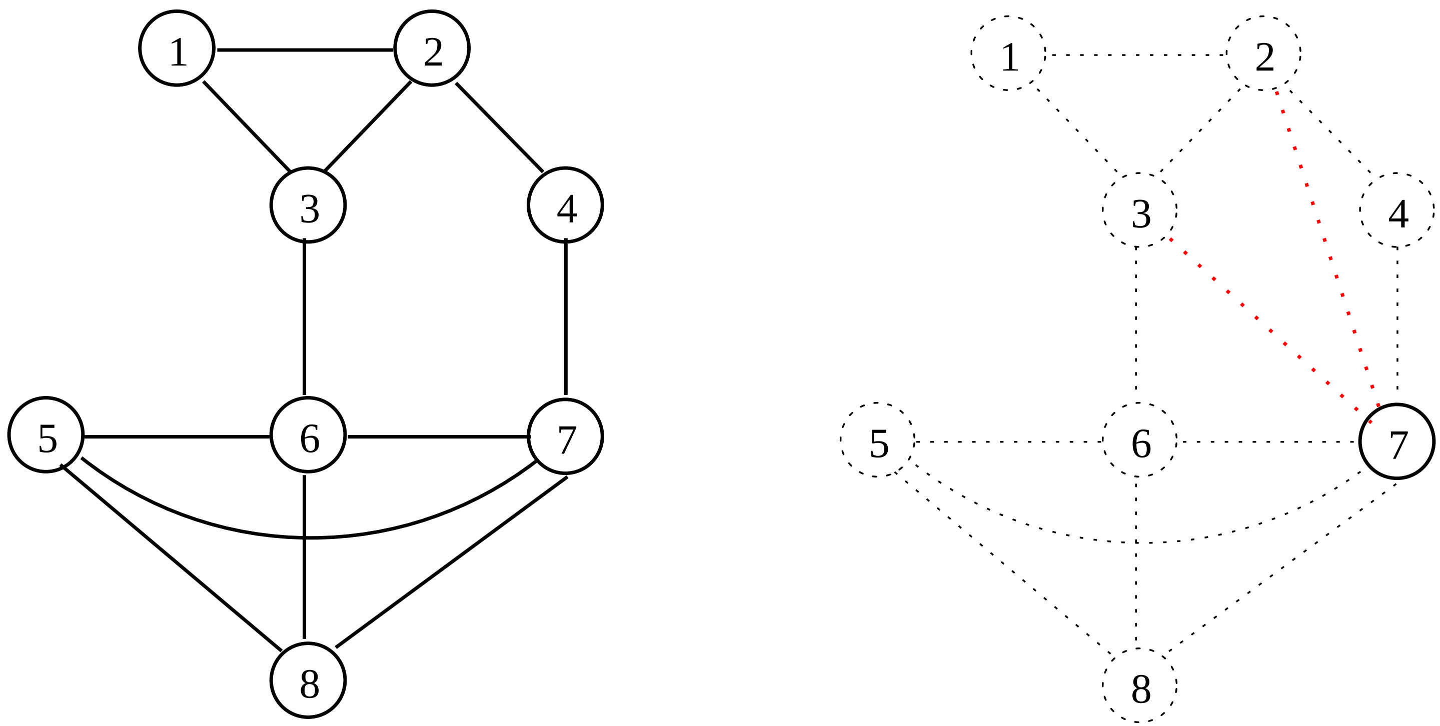


A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs

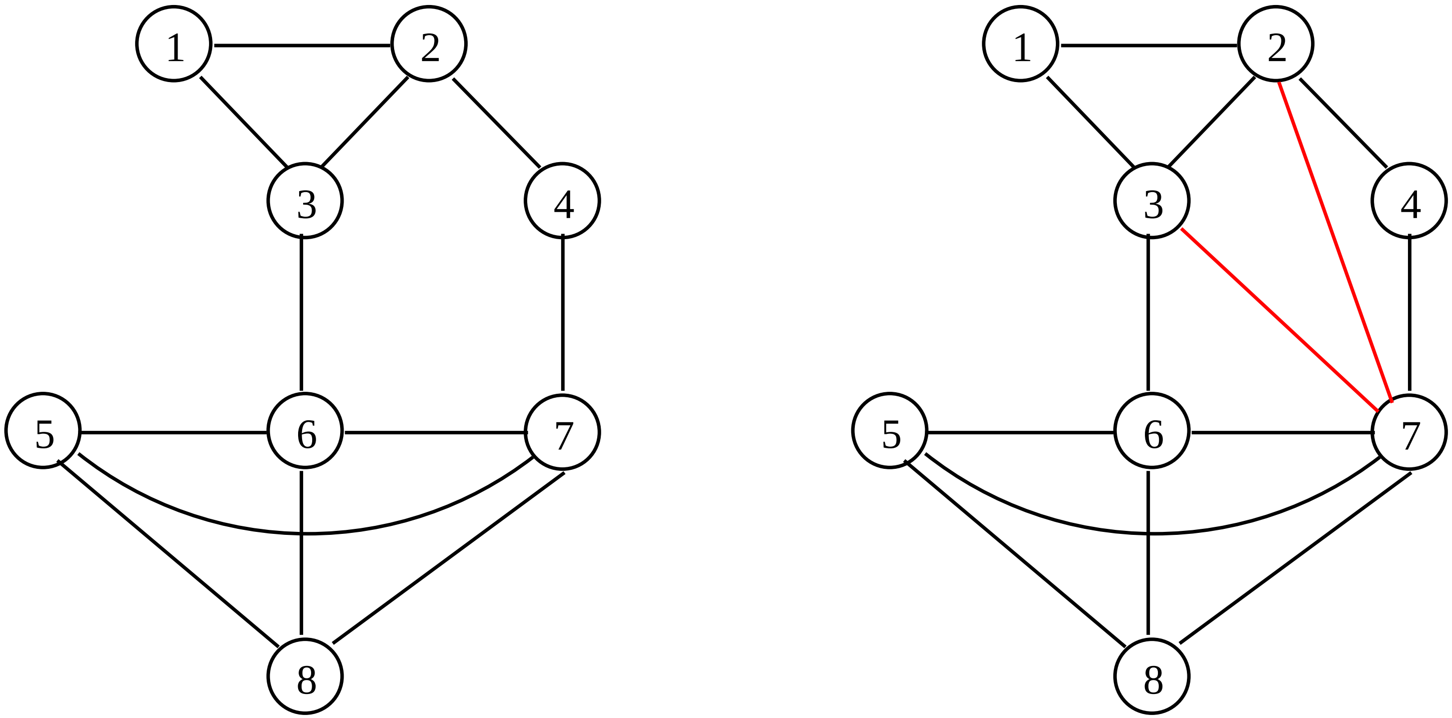


A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs

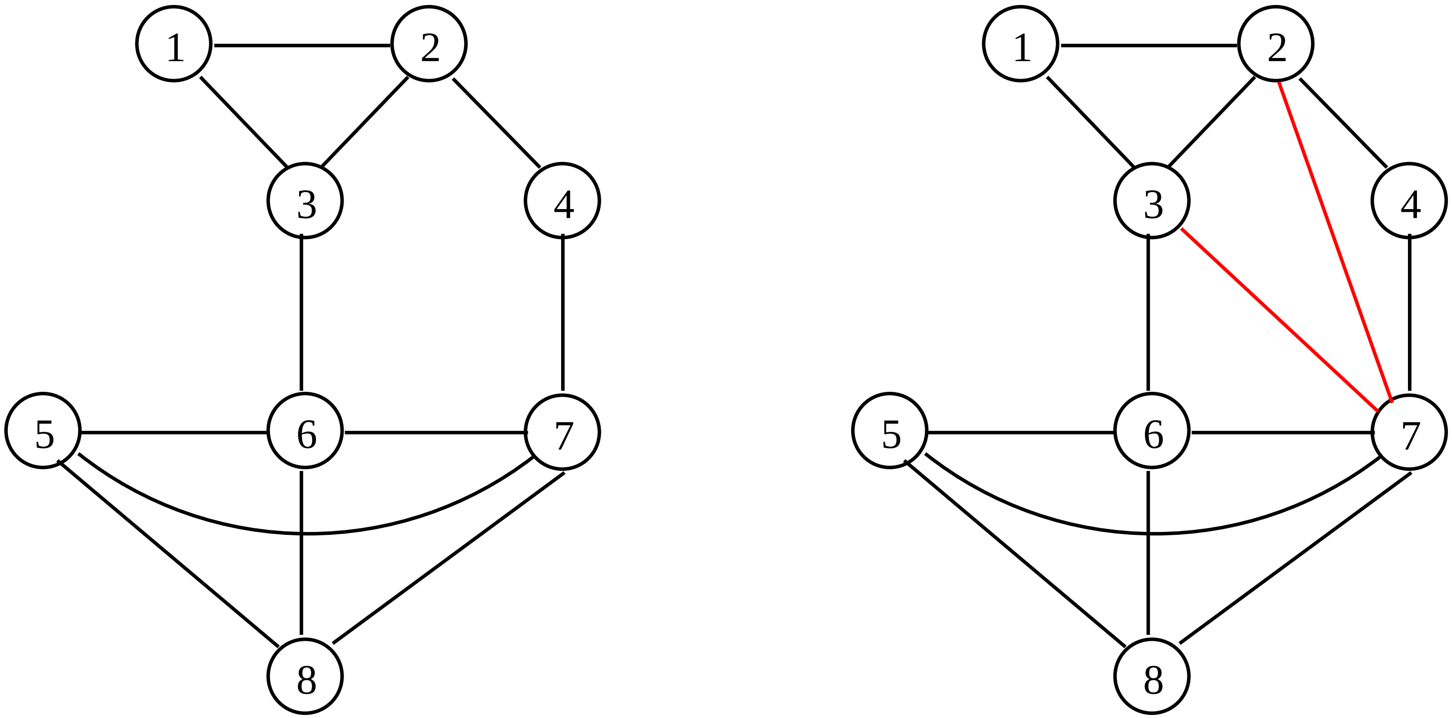


A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

- $|\mathrm{fa}(X)|$ (fill-in-size).

Triangulation of graphs

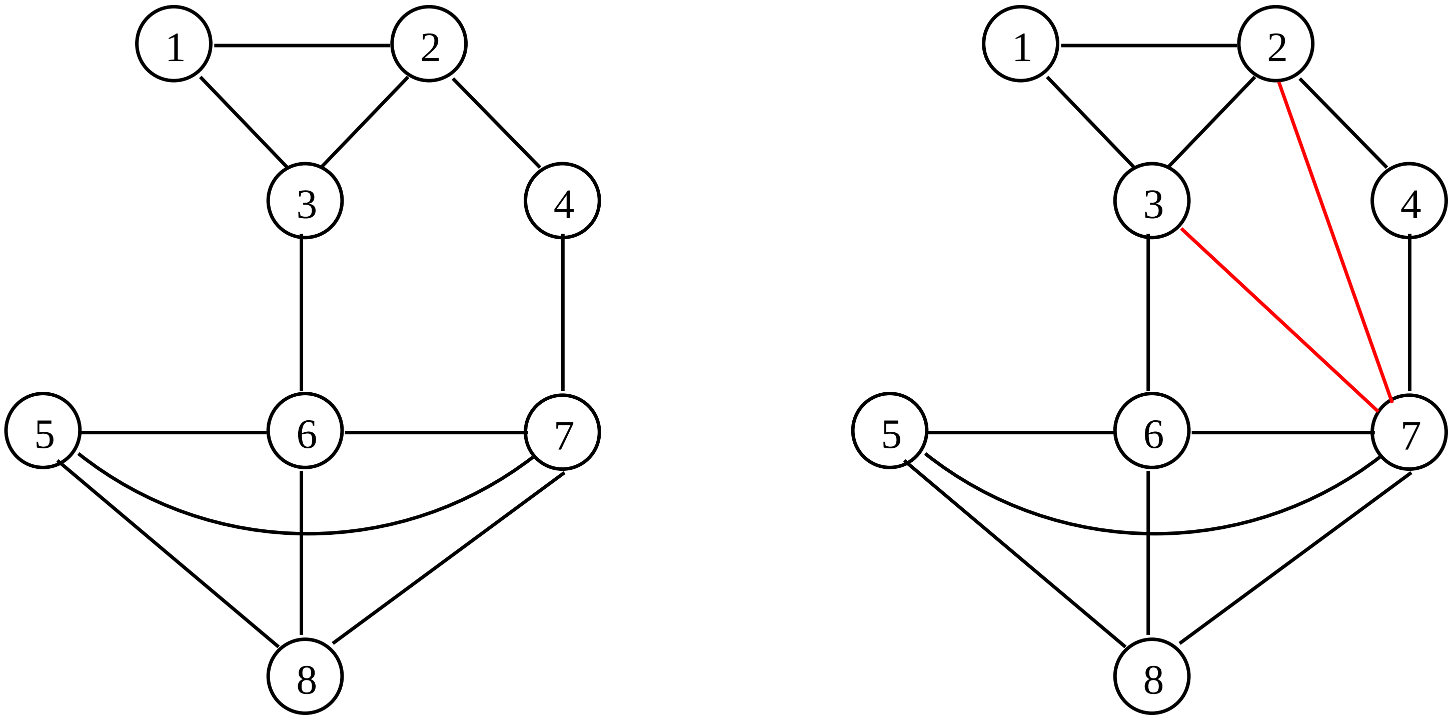


A heuristic

Repeatedly eliminate a simplicial node, and if this is not possible eliminate a node minimizing ($\mathrm{fa}(X)$ are noneliminated neighbors of X):

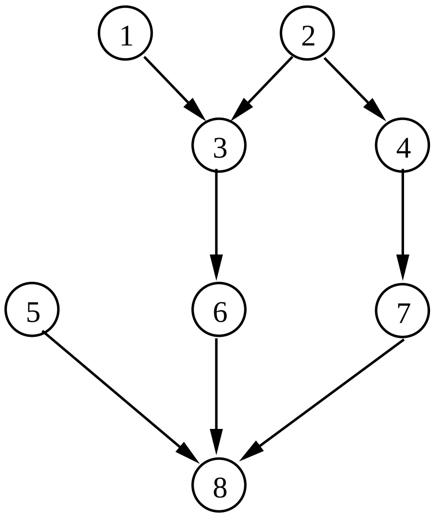
- $|\mathrm{fa}(X)|$ (fill-in-size).
- $|\,\mathrm{sp}(\mathrm{fa}(X))\,|$ (clique-size).
- $\sum_{\{Y,Z\}:\{Y,Z\}\in \mathrm{nb}(X) \wedge Z \notin \mathrm{nb}(Y)} |\,\mathrm{sp}(\{Y,Z\})\,|$ (fill-in-weight)

Triangulation of graphs



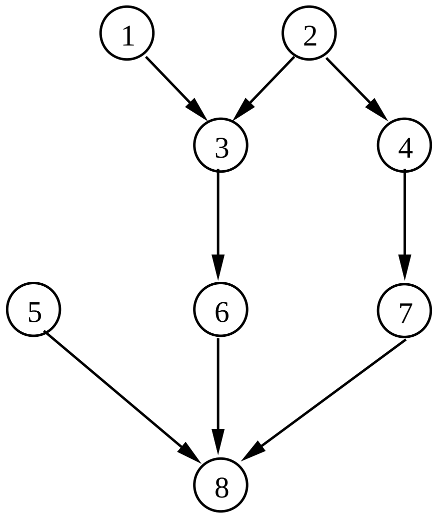
Usually there are many possible triangulations and finding the best one is NP-hard!

Summary

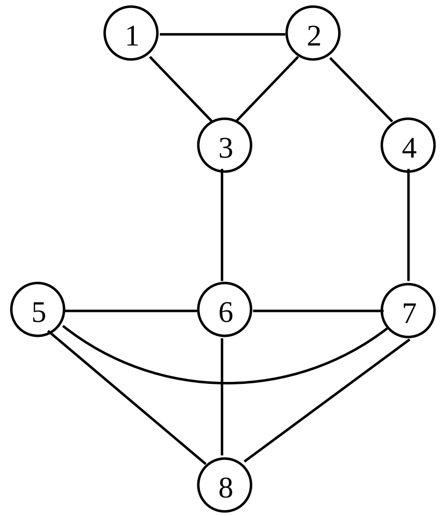


Bayesian network

Summary

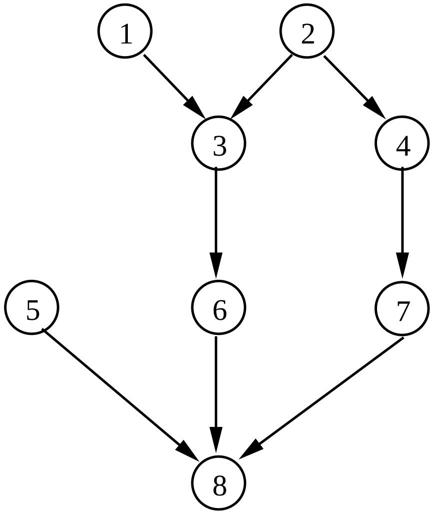


Bayesian network

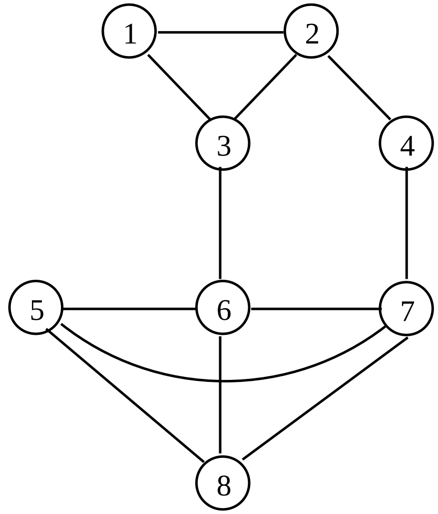


Moral graph

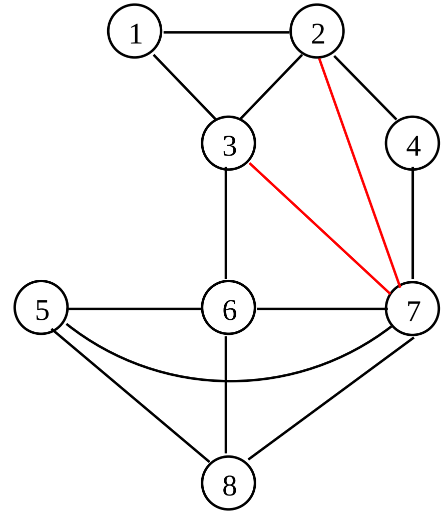
Summary



Bayesian network

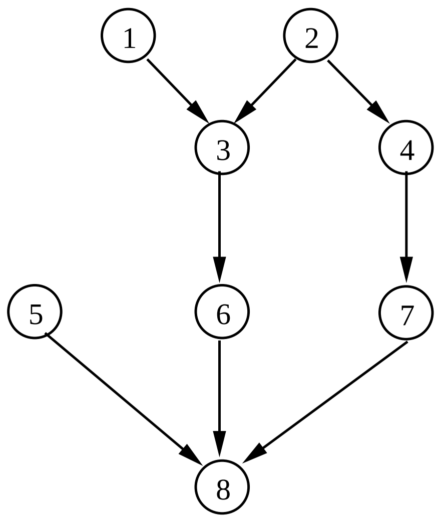


Moral graph

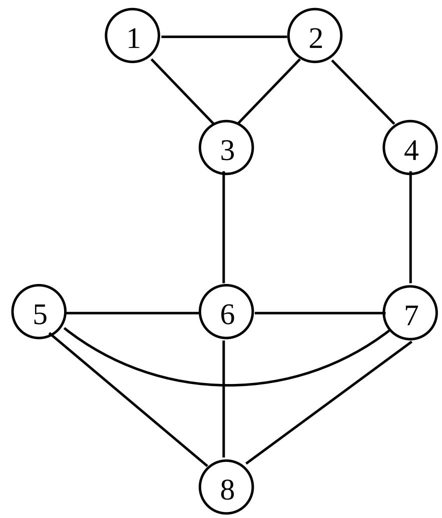


Triangulated graph

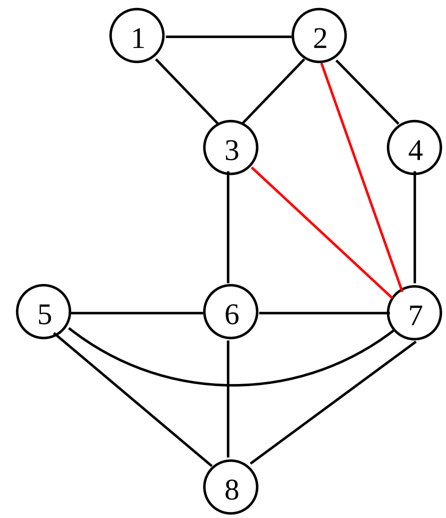
Summary



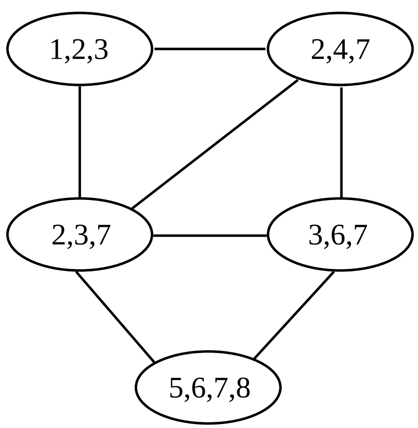
Bayesian network



Moral graph

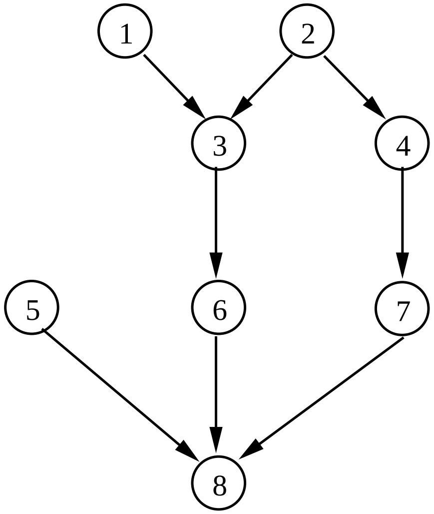


Triangulated graph

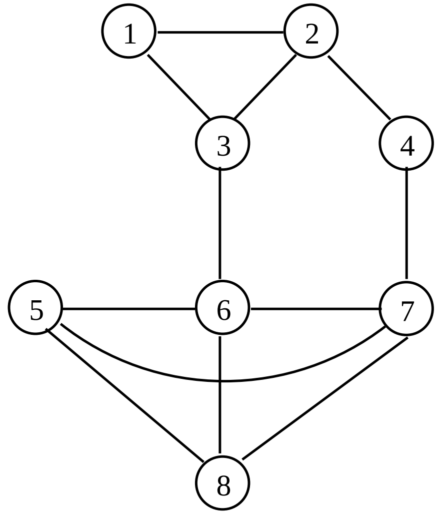


Join graph

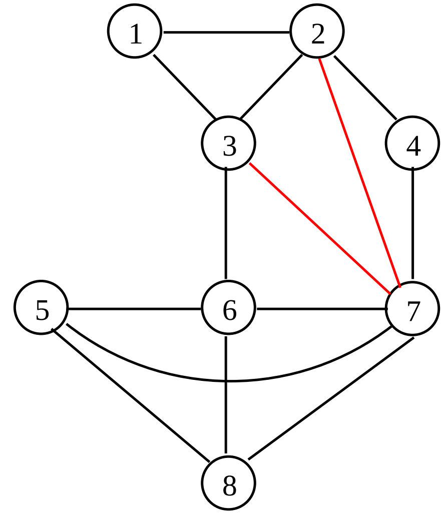
Summary



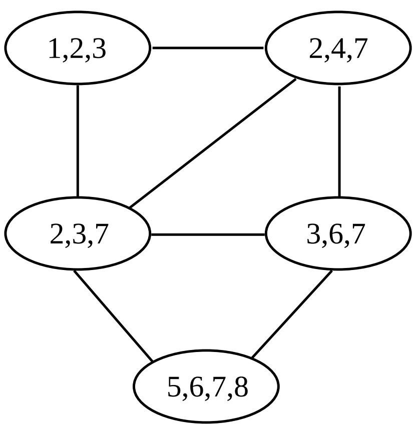
Bayesian network



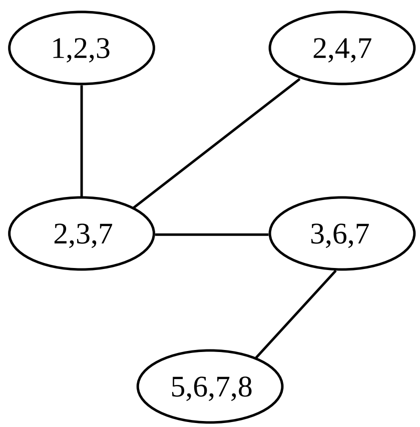
Moral graph



Triangulated graph

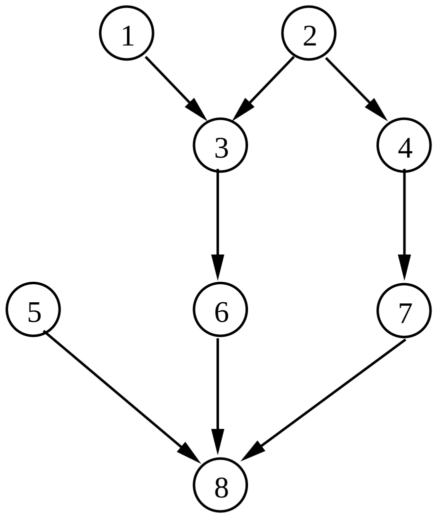


Join graph

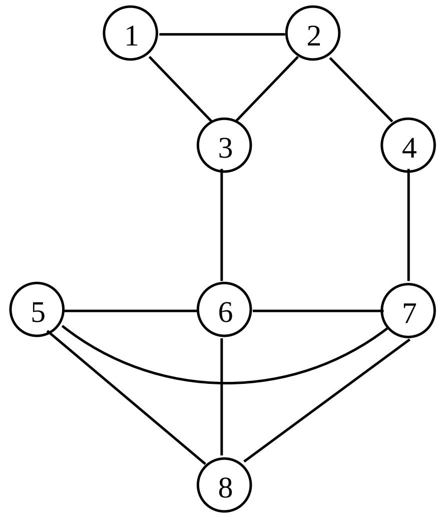


Join tree

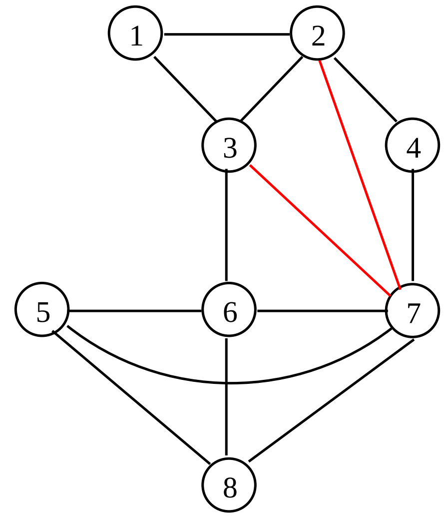
Summary



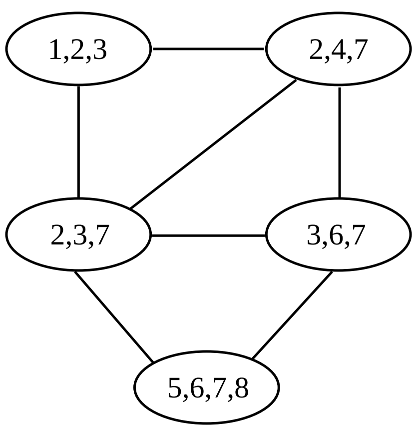
Bayesian network



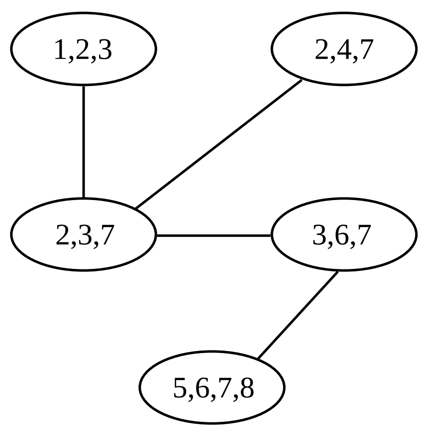
Moral graph



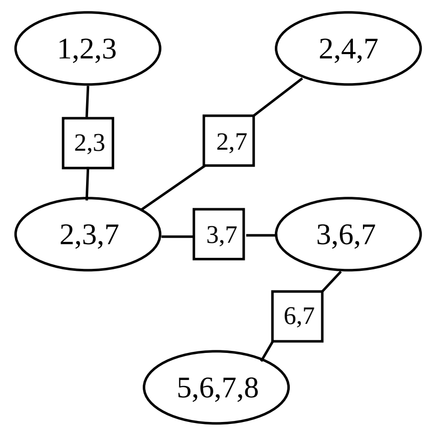
Triangulated graph



Join graph



Join tree



Junction tree

Exact Inference

Summary

Given: Bayesian network BN for $P(\mathbf{V})$

Exact Inference

Summary

Given: Bayesian network BN for $P(\mathbf{V})$

$\Downarrow$ Moralize

Domain graph G for BN (also called the moral graph)

Exact Inference

Summary

Given: Bayesian network BN for $P(\mathbf{V})$

$\Downarrow$ Moralize

Domain graph G for BN (also called the moral graph)

$\Downarrow$ Triangulate

Triangulated graph for G

Exact Inference

Summary

Given: Bayesian network BN for $P(\mathbf{V})$

$\Downarrow$ Moralize

Domain graph G for BN (also called the moral graph)

$\Downarrow$ Triangulate

Triangulated graph for G

$\Downarrow$ Find cliques

Join graph

Exact Inference

Summary

Given: Bayesian network BN for $P(\mathbf{V})$

$\Downarrow$ Moralize

Domain graph G for BN (also called the moral graph)

$\Downarrow$ Triangulate

Triangulated graph for G

$\Downarrow$ Find cliques

Join graph

$\Downarrow$ Find maximal spanning tree

Join tree

Exact Inference

Summary

Given: Bayesian network BN for $P(\mathbf{V})$

$\Downarrow$ Moralize

Domain graph G for BN (also called the moral graph)

$\Downarrow$ Triangulate

Triangulated graph for G

$\Downarrow$ Find cliques

Join graph

$\Downarrow$ Find maximal spanning tree

Join tree

$\Downarrow$

Perform belief updating by message passing.