Chapter 7

Marginalisation, Triangulated Graphs and Junction Trees

7.1 Functions and Domains

Notation  Let $V = \{X_1, \ldots, X_d\}$ denote the set of random variables, where variable $X_j$ has state space $\mathcal{X}_j = \{x_j^{(1)}, \ldots, x_j^{(k_j)}\}$ for $j = 1, \ldots, d$. Let $\mathcal{X} = \times_{j=1}^d \mathcal{X}_j$ denote the state space for the random vector $\mathbf{X} = (X_1, \ldots, X_d)$. Let $\mathcal{V} = \{1, \ldots, d\}$ denote the indexing set for the variables. For $D \subseteq \mathcal{V}$, where $D = \{j_1, \ldots, j_m\}$, let $\mathcal{X}_D = \times_{j \in D} \mathcal{X}_j$ and let $\mathbf{X}_D = (X_{j_1}, \ldots, X_{j_m})$. Let $\mathbf{x} \in \mathcal{X}$ denote a generic element of $\mathcal{X}$ and let $\mathbf{x}_D = (x_{j_1}, \ldots, x_{j_m}) \in \mathcal{X}_D$, when $\mathbf{x} = (x_1, \ldots, x_d) \in \mathcal{X}$. The notation

$$\mathbf{z}_D = \times_{v \in D} \mathbf{z}_v$$

is used to denote a configuration (or a collection of outcomes) on the nodes in $D$. Furthermore, for any set $W \subseteq V$, let $\mathcal{W}$ denote the indexing set for $W$. The notation $\mathcal{X}_W$ will also be used to denote $\mathcal{X}_W$, $\mathbf{X}_W$ to denote $\times_{W} \mathbf{X}$ and $\mathbf{z}_W$ to denote $\times_{W} \mathbf{z}$. Suppose $D \subseteq W \subseteq \mathcal{V}$ and that $\mathbf{z}_W \in \mathbf{X}_W$. Then, ordering the variables of $W$ so that $\mathcal{X}_W = \mathcal{X}_D \times \mathcal{X}_{W \setminus D}$, the projection of $\mathbf{z}_W$ onto $D$ is defined as the variable $\mathbf{z}_D$ that satisfies

$$\mathbf{z}_D = (\mathbf{z}_D, \mathbf{z}_{W \setminus D}),$$

where the meaning of the notation ‘( )’ is clear from the context. Here $A \setminus B$ denotes the set difference; i.e. the elements in the set $A$ not included in $B$.

Definition 7.1 (Function, Domain). Consider a function $\phi : \mathcal{X}_D \to \mathbb{R}_+$. The space $\mathcal{X}_D$ is known as the domain of the function. If the domain is the state space of a random vector $\mathbf{X}_D$, then $\mathbf{X}_D$ may also be referred to as the domain of the function.

In this setting, a function over a domain $\mathcal{X}_D$ has $\prod_{j \in D} k_j$ entries. For $W \subseteq V$, the domain of a function $\mathcal{X}_W$ may also be denoted by the collection of random variables $W$. 

139
Addition, Multiplication, Division For functions defined on the same domain, addition, multiplication and division are defined pointwise where, by definition,

\[ a(x) = 0, \quad b(x) = 0 \implies \frac{a(x)}{b(x)} = 0. \]

Functions over different domains If function \( \phi_1 \) is defined over domain \( X_{D_1} \) and function \( \phi_2 \) is defined over domain \( X_{D_2} \), then multiplication and division of functions may be defined by first extending both functions to the domain \( X_{D_1 \cup D_2} \).

**Definition 7.2** (Extending the Domain). Let the function \( \phi \) be defined on a domain \( X_D \), where \( D \subseteq \hat{W} \subseteq \hat{V} \). Then \( \phi \), defined over a domain \( X_D \), is extended to the domain \( X_W \) in the following way. For each \( x_W \in X_W \),

\[ \phi(x_W) = \phi(x_D), \]

where \( x_D \) is the projection of \( x_W \) onto \( X_D \), using the definition of \( x_W \) (and hence \( x_W \)) from the beginning of the section, page 139. In other words, the extended function depends on \( x_W \) only through \( x_D \).

Addition, Multiplication and Division of Functions over Different Domains Addition, multiplication and division of functions over different domains is defined as first, extending the domains of definition using Definition 7.2 so that they are defined over the same domain, followed by standard pointwise addition, multiplication or division.

Multiplication of functions may be expressed in the following terms: the product \( \phi_1, \phi_2 \) of functions \( \phi_1 \) and \( \phi_2 \), defined over domains \( X_{D_1} \) and \( X_{D_2} \) is defined as

\[ (\phi_1 \phi_2)(x_{D_1 \cup D_2}) = \phi_1(x_{D_1 \cup D_2}) \phi_2(x_{D_1 \cup D_2}), \]

where \( \phi_1 \) and \( \phi_2 \) have first been extended to \( X_{D_1 \cup D_2} \).

Let \( D_\phi \) denote the index set for the domain variables of a function \( \phi \). Then for two functions \( \phi_1 \) and \( \phi_2 \),

\[ D_{\phi_1 \phi_2} = D_{\phi_1} \cup D_{\phi_2}. \]

Marginalisation The operation of marginalisation is now considered more generally. Let \( U \subseteq W \subseteq V \) and let \( \phi \) be a function defined over \( X_W \). The expression \( \sum_{W \setminus U} \phi \) denotes the margin (or the sum margin) of \( \phi \) over \( X_U \) and is defined for \( x_U \in X_U \) by

\[ \left( \sum_{W \setminus U} \phi \right)(x_U) = \sum_{z \in X_W \setminus X_U} \phi(z, x_U), \]

where the arguments have been rearranged so that those corresponding to \( W \setminus U \) appear first, \( z \in X_W \setminus U \) is the projection of \( (z, x_U) \in X_W \) onto \( X_{W \setminus U} \), and \( x_U \in X_U \), the projection of \( (z, x_U) \in X_W \) onto \( X_U \). The following notation is also used for marginalising a function with domain \( X_W \).
7.1. FUNCTIONS AND DOMAINS

\[ \phi^U = \left( \sum_{W \subseteq U} \phi \right). \]

The marginalisation operation obeys the following rules:

1. The Commutative Law: for any two sets of variables \( U \subset V \) and \( W \subset V \),

\[ (\phi^U)^W = (\phi^W)^U. \]

2. The Distributive Law:

If \( \mathcal{X}_{D_1} \) is the domain of \( \phi_1 \) and \( D_1 \subseteq \hat{V} \), then \( (\phi_1 \phi_2)^{D_1} = \phi_1 (\phi_2)^{D_1} \).

**Definition 7.3** (Charge, Contraction). A charge

\[ \Phi = \{ \phi_1, \ldots, \phi_m \} \]

is defined as a set of functions on \( \mathcal{X} \).

A contraction of a charge, or set of functions is an operation of multiplication of functions, after extending them to \( \mathcal{X} \), that returns the function

\[ \Phi(x) = \prod_{j=1}^{m} \phi_j(x). \]

The same notation is often used to denote the contraction of a charge and of the set of functions (the charge). The context makes it clear which is intended.

**Probability function factorised along a DAG** The joint probability function \( p_{X_1, \ldots, X_d} \) is itself a function, with domain \( \mathcal{X} \). If the joint probability function may be factorised according to a DAG \( \mathcal{G} = (V, D) \), the decomposition is written as

\[ p_{X_1, \ldots, X_d} = \prod_{j=1}^{d} p_{X_j|\Pi_j}. \]

Then for each \( j = 1, \ldots, d \), \( \phi_j \) defined by \( \phi_j = p_{X_j|\Pi_j} \) is a function with domain \( \mathcal{X}_{D_j} = \mathcal{X}_j \times \mathcal{X}_{\Pi_j} \) and \( D_j = \{ j \} \cup \Pi_j \).

**Example 7.4.**

Consider a probability function over six variables that may be factorised along the directed acyclic graph in Figure 7.1. The functions corresponding to the conditional probabilities are

\[ \phi_1 = p_{X_1}, \phi_2 = p_{X_4|X_1}, \phi_3 = p_{X_3|X_1}, \]

\[ \phi_4 = p_{X_4|X_2}, \phi_5 = p_{X_5|X_2, X_3}, \phi_6 = p_{X_6|X_3}. \]

The corresponding domains are
\[ X_{D_1} = X_1, \quad X_{D_2} = X_1 \times X_1, \quad X_{D_3} = X_1 \times X_1 \]
\[ X_{D_4} = X_1 \times X_2, \quad X_{D_5} = X_2 \times X_3, \quad X_{D_6} = X_3 \times X_3. \]

Figure 7.1: A Bayesian Network on 6 variables

**Definition 7.5** (Domain Graph). The domain graph for the set of functions in \( \Phi \) is an undirected graph with the variables as nodes and the links between any pair of variables which are members of the same domain.

Figure 7.2 illustrates the domain graph associated with DAG of Figure 7.1. The domain graph of a DAG is the moral graph, Definition 6.25. The cliques of the moral graph are illustrated in Figure 7.3.

It is clear that the domain graph of a Bayesian network is the moral graph, since by definition all the parents are connected to each other and to the variable.

### 7.2 Marginalisation and Graphical Representations

Let \( \phi_1 \) be a function with domain \( X_{D_1} \) and let \( \phi_2 \) be a function with domain \( X_{D_2} \). Suppose that \( A \subseteq D_1 \cup D_2 \) and their product \( \phi_1 \phi_2 \) is to be marginalised over \( X_A \). If \( A \cap D_1 = \phi \) (the empty set), then
\[ \sum_{\mathcal{A}} \phi_1 \phi_2 = \phi_1 \sum_{\mathcal{A}} \phi_2. \]

In coordinates, let \( \phi_1 \) have domain \( \mathcal{X}_{D_1 \cup D_2} \) and \( \phi_2 \) domain \( \mathcal{X}_{D_3 \cup D_4} \), where \( D_1, D_2, D_3 \) and \( D_4 \) are disjoint. By the distributive law, the marginalisation may be written as

\[
\sum_{x_{D_2}} \phi_1(x_{D_1}, x_{D_3}) \phi_2(x_{D_3}, x_{D_4}) = \phi_1(x_{D_1}, x_{D_3}) \sum_{x_{D_2}} \phi_2(x_{D_2}, x_{D_3}, x_{D_4}).
\]

The function over \( \mathcal{X}_{D_1} \times \mathcal{X}_{D_3} \times \mathcal{X}_{D_4} \) is first marginalised down to a function over \( \mathcal{X}_{D_3} \times \mathcal{X}_{D_4} \). The function is transmitted to the function over \( \mathcal{X}_{D_3} \times \mathcal{X}_{D_4} \), to which it is multiplied. The domains of the two functions to be multiplied have to be extended to \( \mathcal{X}_{D_1} \times \mathcal{X}_{D_3} \times \mathcal{X}_{D_4} \). Using \( X_1, X_2, X_3, X_4 \) to denote the associated domains \( \mathcal{X}_{D_1}, \mathcal{X}_{D_2}, \mathcal{X}_{D_3} \) and \( \mathcal{X}_{D_4} \), the domains under consideration for the operations are illustrated in Figure 7.4. First, the function \( \phi_2 \), defined over \( (X_2, X_3, X_4) \) is considered. This is marginalised to a function over \( (X_3, X_4) \) and is then extended, by multiplying with \( \phi_1 \), to a function over \( (X_1, X_3, X_4) \).

![Figure 7.3: Cliques of the Graph in Figure 7.2](image)

![Figure 7.4: The Distributive Law](image)

**Example 7.6** (Example of a Marginalisation).

Consider the computation for marginalising a contraction of a charge \( \Phi \) defined over a state space \( \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_4 \times \mathcal{X}_5 \) where

\[
\Phi(x) = \phi_1(x_1, x_3, x_5) \phi_2(x_1, x_2) \phi_3(x_3, x_4) \phi_4(x_5, x_6).
\]

More particularly, consider the computation of

\[
\Phi^{(0)} = \sum_{x \in \mathcal{X}} \Phi(x),
\]
where the notation $\Phi^{\mathcal{U}}$ is defined on page 140. With the order of summation: $x_2, x_4, x_6, x_5, x_3, x_1$, the sum may be written (taking sums from right to left) as

$$
\sum_{x_1 \in \mathcal{X}_1} \sum_{x_3 \in \mathcal{X}_3} \sum_{x_5 \in \mathcal{X}_5} \phi_1(x_1, x_3, x_5) \sum_{x_4 \in \mathcal{X}_4} \phi_4(x_5, x_6) \sum_{x_3 \in \mathcal{X}_3} \phi_3(x_3, x_4) \sum_{x_2 \in \mathcal{X}_2} \phi_2(x_1, x_2).
$$

The computation, carried out in this order (right to left), may be represented by the graph in Figure 7.5; a computational tree, according to the distributive law, is given in Figure 7.6.

Recall (page 140) that the operation $\Phi^{\mathcal{U}}(x)$ means marginalising $\Phi$ over all variables not in the set $U$.

**Definition 7.7** (Elimination of a Variable). The variable $X_v$, with index $v \in \mathcal{V} = \mathcal{V} \setminus \mathcal{U}$ is eliminated from $\sum_{\mathcal{X}_\mathcal{V} \setminus \mathcal{U}} \Phi(\mathcal{X}_\mathcal{V} \setminus \mathcal{U})$ by the following procedure, where contraction means multiplying together all the functions in the charge.
1. Let $\Phi_u$ (or $\Phi_{X_u}$) denote the contraction of the functions in $\Phi$ that have $X_u$ in their domain; that is,

$$\Phi_u = \prod_{j \in D_u} \phi_j.$$

2. Let $\phi^{(v)}$ (or $\phi^{(X_v)}$) denote the function $\sum_{x_v \in X_v} \Phi_u$.

3. Find a new set of functions $\Phi^{-v}$ (or $\Phi^{-X_v}$) by setting

$$\Phi^{-v} = (\Phi \cup \{\phi^{(v)}\}) \setminus \Phi_u.$$

This is the definition of $\Phi^{-v}$, also denoted by $\Phi^{-X_v}$.

Those functions that do not contain $X_v$ in their domain have been retained; the others have been multiplied together and then marginalised over $X_v$ (thus eliminating the variable) to give $\phi^{(v)}$. This function has been added to the collection, and all those containing $X_v$ (other than $\phi^{(v)}$) have been removed.

(Note that the notation $\Phi^{-X_v}$ has two meanings: it is used to the collection of functions, and it is also used to denote the contraction of the charge obtained by multiplying together the functions in the collection. The meaning is determined by the context.) Having removed $X_v$, it remains to compute

$$\sum_{x_{W \setminus X_v}} \Phi^{-X_v}(x_U, x_{W \setminus \{X_v\}}).$$

The quantity

$$\Phi^{(v)}(x_U) = \sum_{x_{W \setminus X_v} \in x_{W \setminus U}} \Phi(x_{W \setminus U}, x_U)$$

can be computed through successive elimination of the variables $X_v \in W \setminus U$. The task, of course, is to find a sequence for marginalising the variables such that, at each stage, the variable is to be eliminated from as small a domain as possible. The procedure outlined above may be considered graphically in terms of undirected graphs and their triangulations.

### 7.3 Decomposable Graphs and Node Elimination

Recall the definition of an induced sub graph (Definition 1.8); a subgraph induced by a subset $A \subset V$ is the graph $G_A = (A, E_A)$ where $E_A = E \cap A \times A$. The following definitions are necessary.

**Definition 7.8** (Complete Graph, Complete Subset). A graph $G$ is complete if every pair of nodes is joined by an undirected edge. That is, for each $(\alpha, \beta) \in V \times V$ with $\alpha \neq \beta$, $(\alpha, \beta) \in E$ and $(\beta, \alpha) \in E$. In other words, $(\alpha, \beta) \in U$, where $U$ denotes the set of undirected edges. A subset of nodes is called complete if it induces a complete sub graph.

**Definition 7.9** (Clique). A clique is a complete sub graph that is maximal with respect to $\subseteq$. In other words, a clique is not a sub graph of any other complete graph.
Definition 7.10 (Simplicial Node). Recall the definition of family, found in Definition 1.5. For an undirected graph, the family of a node $\beta$ is $F(\beta) = \{\beta\} \cup N(\beta)$, where $N(\beta)$ denotes the set of neighbours of $\beta$. A node $\beta$ in an undirected graph is called simplicial if its family $F(\beta)$ is a clique.

This means that, in an undirected graph, a node $\beta$ is simplicial if all its neighbours are neighbours of each other.

Definition 7.11 (Connectedness, Strong Components). Let $\mathcal{G} = (V,E)$ be a simple graph, where $E = U \cup D$. That is, $E$ may contain both directed and undirected edges. Let $\alpha \rightarrow \beta$ denote that there is a path (Definition 1.10) from $\alpha$ to $\beta$. If there is both $\alpha \rightarrow \beta$ and $\beta \rightarrow \alpha$ then $\alpha$ and $\beta$ are said to be connected. This is written:

$$\alpha \leftrightarrow \beta.$$ 

This is clearly an equivalence relation. The equivalence class for $\alpha$ is denoted by $[\alpha]$. In other words, $\beta \in [\alpha]$ if and only if $\beta \leftrightarrow \alpha$. These equivalence classes are called strong components of $\mathcal{G}$.

Note that a graph is connected if between any two nodes there exists a trail (Definition 1.9), but any two nodes $\alpha$ and $\beta$ are only said to be connected if there is path from $\alpha$ to $\beta$ and a path from $\beta$ to $\alpha$, where the definition of a ‘path’ is given in Definition 1.10.

Lemma 7.12. If $\mathcal{G} = (V,E)$ is triangulated, then the induced graph $\mathcal{G}_A$ is also triangulated.

Proof Consider any cycle of length $\geq 4$ in the restricted graph. All the edges connecting these nodes remain. If the cycle possessed a chord in the original graph, the chord remains in the restricted graph. $\square$

Definition 7.13 (Minimal Separator). Let $A \subseteq V$, $B \subseteq V$ and $S \subseteq V$ be three disjoint subsets of $V$. Let $S$ separate $A$ and $B$. The separator $S$ is said to be a minimal separator of $A$ and $B$ if no proper subset of $S$ is itself a separator of $A$ and $B$.

Definition 7.14 (Decomposition, Weak Decomposition). Let $\mathcal{G} = (V,U)$ be an undirected graph. A triple $(A,B,S)$ of disjoint subsets of the node set $V$ of an undirected graph is said to form a decomposition of $\mathcal{G}$ or to decompose $\mathcal{G}$ if

$$V = A \cup B \cup S$$

and

- $S$ separates $A$ from $B$
- $S$ is a complete subset of $V$.

$A$, $B$ or $S$ may be the empty set. If both $A$ and $B$ are non empty, then the decomposition is proper.

A triple $(A,B,S)$ of disjoint subsets of the node set $V$ of an undirected graph is said to form a weak decomposition of $\mathcal{G}$ or to weakly decompose $\mathcal{G}$ if $V = A \cup B \cup S$ and $S$ separates $A$ from $B$. 
A weak decomposition differs from a decomposition in that the separator set $S$ is not necessarily complete. Clearly, every graph can be decomposed to its connected components (Definition 1.9). If the graph is undirected, then the connected components are the strong components (Definition 7.11).

**Definition 7.15** (Decomposable Graph). An undirected graph $G$ is decomposable if either

1. it is complete, or

2. it possesses a proper decomposition $(A,B,S)$ such that both sub graphs $G_{A\cup S}$ and $G_{B\cup S}$ are decomposable.

This is a recursive definition, which is permissible, since the decomposition $(A,B,S)$ is required to be proper, so that $G_{A\cup S}$ and $G_{B\cup S}$ have fewer nodes than the original graph $G$.

**Example 7.16** (Decomposable Graph).

Consider the graph in Figure 7.7. In the first stage, set $S = \{\alpha_3\}$, with $A = \{\alpha_1, \alpha_2\}$ and $B = \{\alpha_4, \alpha_5, \alpha_6\}$. Then $S$ is a clique and $S$ separates $A$ from $B$. Then $A \cup S = \{\alpha_1, \alpha_2, \alpha_3\}$ and $G_{A\cup S}$ is a clique. $B \cup S = \{\alpha_3, \alpha_4, \alpha_5, \alpha_6\}$. The graph $G_{B\cup S}$ is decomposable; take $S_2 = \{\alpha_3, \alpha_5\}$, $A_2 = \{\alpha_4\}$ and $B_2 = \{\alpha_6\}$. Then $G_{A_2\cup S_2}$ and $G_{B_2\cup S_2}$ are cliques.

![Figure 7.7: Example of a Decomposable Graph](image)

**Theorem 7.17.** Let $G = (V,U)$ be an undirected graph. The following conditions 1), 2) and 3) are equivalent.

1. $G$ is decomposable.

2. $G$ is triangulated.

3. For every pair of nodes $(\alpha,\beta) \in V \times V$, their minimal separator is complete.

**Proof of Theorem 7.17:**

1) $\implies$ 2) **Inductive hypothesis:** All undirected decomposable graphs with $n$ nodes or less are triangulated. This is true for one node.

Let $G$ be a decomposable graph with $n + 1$ nodes. There are two alternatives:

- **Either** $G$ is complete, in which case it is triangulated,
Or: by the definition of decomposable, there are three disjoint subsets \(A, B, S\) such that \(S\) is a complete subset, \(S\) separates \(A\) from \(B\), \(V = A \cup B \cup S\) and \(G_{\mathit{AuS}}\) and \(G_{\mathit{BuS}}\) are decomposable. The decomposition is proper, hence \(G_{\mathit{AuS}}\) and \(G_{\mathit{BuS}}\) have less than or equal to \(n\) nodes. Therefore, by the inductive hypothesis \(G_{\mathit{AuS}}\) and \(G_{\mathit{BuS}}\) are triangulated. Therefore, a cycle of length \(\geq 4\) without a chord, will be a cycle from \(A\) which passes through \(B\). By decomposability, \(S\) separates \(A\) from \(B\) and therefore any such cycle must pass \(S\) at least twice. But then this cycle has a chord, since \(S\) is a complete subset. □

Proof of Theorem 7.17: 2) \(\implies\) 3) Assume that \(G = (V, U)\) is an undirected, triangulated graph. Let \(S\) be a minimal separator for two nodes \(\alpha\) and \(\beta\). Let \(A\) denote the set such that \(\alpha \in A\) and \(G_A\) is the largest connected sub-graph of \(G_{\mathit{AaS}}\) such that \(\alpha\) is in the node set. Let \(B = V \setminus (A \cup S)\). For every node \(\gamma \in S\), there is a node \(\tau \in A\) such that \((\gamma, \tau) \in U\) and there is a node \(\sigma \in B\) such that \((\gamma, \sigma) \in U\). Otherwise \(S\) would be a separator for \(\alpha\) and \(\beta\), contradicting the minimality of \(S\). Hence, for any pair \((\gamma, \delta) \in S \times S\), there exist paths \(\gamma, \tau_1, \ldots, \tau_m, \delta\) and \(\gamma, \sigma_1, \ldots, \sigma_n, \delta\) where all the nodes \((\tau_1, \ldots, \tau_m)\) are in \(A\) and all the nodes \((\sigma_1, \ldots, \sigma_n)\) are in \(B\). Then \(\gamma, \tau_1, \ldots, \tau_m, \delta, \sigma_1, \ldots, \sigma_n, \gamma\) is a cycle of length \(\geq 4\) and therefore has a chord. Assume that \(\tau_1, \ldots, \tau_m\) and \(\sigma_1, \ldots, \sigma_n\) have been chosen so that the paths are as short as possible (that is, there is no shorter path from \(\gamma\) to \(\delta\) with all intervening nodes in \(A\) and no shorter path from \(\gamma\) to \(\delta\) with all intervening nodes in \(B\)).

The chord cannot be of the form \((\tau_i, \tau_j)\) for some \((i, j)\) or \((\sigma_k, \sigma_l)\) for any \((k, l)\) because of the minimality of the lengths of the chosen paths. Therefore, \((\gamma, \delta) \in U\). Therefore, \(\gamma\) and \(\delta\) are adjacent for every pair \((\gamma, \delta) \in S \times S\). It follows that \(S\) is a clique. □

Proof of Theorem 7.17: 3) \(\implies\) 1) If \(G\) is complete, then the result is clear. If \(G\) is not complete, then choose two distinct nodes \((\alpha, \beta) \in V \times V\) that are not adjacent. Let \(S \in V \setminus \{\alpha, \beta\}\) denote the minimal separator for the pair \((\alpha, \beta)\). Let \(A\) denote the node set of the maximal connected component of \(G_{\mathit{AaS}}\) and let \(B = V \setminus (A \cup S)\). Then \((A, B, S)\) provides a decomposition. This procedure can be repeated on \(G_{\mathit{AuS}}\) and \(G_{\mathit{BuS}}\), and repeated recursively, hence the graph is decomposable. □

Definition 7.18 (Perfect Node Elimination Sequence). Let \(V = \{\alpha_1, \ldots, \alpha_d\}\) denote the node set of a graph \(G\). A perfect node elimination sequence of a graph \(G\) is an ordering of the node set \(\{\alpha_1, \ldots, \alpha_d\}\) such that for each \(j\) in \(1 \leq j \leq d - 1\), \(\alpha_j\) is a simplicial node of the sub graph of \(G\) induced by \(\{\alpha_j, \alpha_{j+1}, \ldots, \alpha_d\}\).

Lemma 7.19. Every triangulated graph \(G\) has a simplicial node. Moreover, if \(G\) is not complete, then it has two non adjacent simplicial nodes.

Proof The lemma is trivial if either \(G\) is complete, or else \(G\) has two or three nodes. Assume that \(G\) is not complete. Suppose the result is true for all graphs with fewer nodes than \(G\). Consider two non adjacent nodes \(\alpha\) and \(\beta\). Let \(S\) denote the minimal separator of \(\alpha\) and \(\beta\). Let \(G_A\) denote the largest connected component of \(G_{\mathit{AaS}}\) such that \(\alpha \in A\) and let \(B = V \setminus (A \cup S)\), so that \(\beta \in B\).
By induction, either $G_{A\cup S}$ is complete, or else it has two non adjacent simplicial nodes. Since $G_S$ is complete, it follows that at least one of the two simplicial nodes is in $A$. Such a node is therefore also simplicial in $G$, because none of its neighbours is in $B$.

If $G_{A\cup S}$ is complete, then any node of $A$ is a simplicial node of $G$.

In all cases, there is a simplicial node of $G$ in $A$. Similarly, there is a simplicial node in $B$. These two nodes are then non adjacent simplicial nodes of $G$. 

**Theorem 7.20.** A graph $G$ is triangulated if and only if it has a perfect node elimination sequence.

**Proof.** Suppose that $G$ is triangulated. Assume that every triangulated graph with fewer nodes than $G$ has a perfect elimination sequence. By the previous lemma, $G$ has a simplicial node $\alpha$. Removing $\alpha$ returns a triangulated graph. (Consider any cycle of length $\geq 4$ with a chord. If the cycle remains after the node is removed, then the chord is not removed). By proceeding inductively, it follows that $G$ has a perfect elimination sequence.

Conversely, assume that $G$ has a perfect sequence, say $\{\alpha_1, \ldots, \alpha_d\}$. Consider any cycle of length $\geq 4$. Let $j$ be the first index such that $\alpha_j$ is in the cycle. Let $V(C)$ denote the node set of the cycle and let $V_j = \{\alpha_j, \ldots, \alpha_d\}$. Then $V(C) \in V_j$. Since $\alpha_j$ is simplicial in $G_{V_{j+1}}$, the neighbours of $\alpha_j$ in the cycle are adjacent, hence the cycle has a chord. Therefore $G$ is triangulated.

**Definition 7.21 (Eliminating a Node).** Let $G = (V, E)$ be an undirected graph. A node $\alpha$ is eliminated from an undirected graph $G$ in the following way:

1. For all pairs of neighbours $(\beta, \gamma)$ of $\alpha$ add a link if $G$ does not already contain one. The added links are called fill ins.

2. Remove $\alpha$.

The resulting graph is denoted by $G^{\sim \alpha}$.

For example, consider the graph in Figure 7.8. This graph is already triangulated. But suppose one did not notice this and one decided to eliminate node $\alpha_3$ from the graph in Figure 7.8. The resulting graph is given in Figure 7.9.

![Figure 7.8: Example for Eliminating a Node](image-url)
Figure 7.9: Graph 7.8 with $\alpha_3$ Eliminated

Definition 7.22 (Elimination Sequence). An elimination sequence of $\mathcal{G}$ is a linear ordering of its nodes.

Let $\sigma$ be an elimination sequence and let $\Lambda$ denote the fill ins produced by eliminating a node of $\mathcal{G}$ in the order $\sigma$. Denote by $\mathcal{G}'$ the graph $\mathcal{G}$ extended by $\Lambda$.

Example 7.23.

Consider the graph in Figure 7.8. Suppose the elimination sequence $\alpha_3, \alpha_2, \alpha_4, \alpha_5, \alpha_6$ is employed. Then the fill ins, for each stage, will be $\langle \alpha_1, \alpha_6 \rangle, \langle \alpha_1, \alpha_5 \rangle, \langle \alpha_2, \alpha_6 \rangle, \langle \alpha_5, \alpha_6 \rangle$ for $\alpha_3$, then $\langle \alpha_1, \alpha_4 \rangle, \langle \alpha_4, \alpha_6 \rangle$ for $\alpha_2$. No further fill ins are required. The graph $\mathcal{G}'$ is given in Figure 7.10.

Figure 7.10: $\mathcal{G}'$. Elimination sequence $(\alpha_3, \alpha_2, \alpha_4, \alpha_1, \alpha_5, \alpha_6)$

Definition 7.24 (Elimination sequence, elimination domains). An elimination sequence $\sigma$ is a linear ordering of the set of nodes $V = \{\alpha_1, \ldots, \alpha_d\}$ where for each $\alpha \in \{1, \ldots, d\}$, $\sigma(\alpha)$ denotes the number assigned to variable $X_\alpha$. A node $\beta$ is said to be of higher elimination order than $\alpha$ if $\sigma(\beta) > \sigma(\alpha)$. The elimination domain of a node $\alpha$ is the set of neighbours of $\alpha$ of higher elimination order.
7.4. JUNCTION TREES

In $\mathcal{G}^\sigma$, any node $\alpha$ together with its neighbours of higher elimination order form a complete subset. The neighbours of $\alpha$ of higher elimination order are denoted by $N_{\sigma(\alpha)}$. The sets $N_{\sigma(\alpha)}$ are the elimination domains corresponding to the elimination sequence $\sigma$.

An efficient algorithm clearly tries to minimise the number of fill ins. If possible, one should find an elimination sequence that does not introduce fill ins.

**Proposition 7.25.** All cliques in a $\mathcal{G}^\sigma$ are a $N_{\sigma(\alpha)}$ for some $\alpha \in V$.

**Proof** Let $C$ be a clique in $\mathcal{G}^\sigma$ and let $\alpha$ be a variable in $C$ of the lowest elimination order. Then $C = N_{\sigma(\alpha)}$. \(\square\)

An efficient algorithm ought to find an elimination sequence for the domain graph that yields cliques of minimal total size.

The following proposition is clear.

**Proposition 7.26.** Any $\mathcal{G}^\sigma$ is a triangulation of $\mathcal{G}$.

**Proof** By construction, the elimination sequence $\sigma$ for graph $\mathcal{G}^\sigma$ does not require any fill-ins. \(\square\)

Recall that a graph is triangulated if and only if it has an elimination sequence without fill ins. This is equivalent to the statement that an undirected graph is triangulated if and only if all nodes can be eliminated by successively eliminating a node $\alpha$ such that the family $F_\alpha = \{\alpha\} \cup N_\alpha$ is complete. From the definition, such a node $\alpha$ is a simplicial node.

**Marginalisation and triangulation of graphs** Let $\mathcal{G} = (V, U)$ be an undirected graph, where $V = \{X_1, \ldots, X_d\}$. Recall the definition of the domain graph (Definition 7.5) and note that the cliques of the domain graph are the domains of the functions of the charge. Recall Definition 7.7, which describes the procedure for eliminating a variable in a marginalisation. When a node $X_\alpha$ is eliminated from the graph $\mathcal{G}$, the resulting graph is denoted by $\mathcal{G}^{-X_\alpha}$. Graphically, the procedure described in Definition 7.7 is the same as Definition 7.21, eliminating a node. If $\mathcal{G}$ is the domain graph for a set of functions $\Phi$, then it is clear from Definition 7.21 that the graph $\mathcal{G}^{-X_\alpha}$ is the domain graph for the set of functions $\Phi^{-X_\alpha}$. Therefore, if the domain graph is triangulated, there is a perfect elimination sequence; there is an order for eliminating the variables that, at each stage, the elimination domain corresponds to a clique in the current domain graph.

7.4 Junction Trees

Decomposable graphs provide the basis for one of the key methods for updating a probability distribution described in terms of a Bayesian network. The DAG is moralised and then triangulated using the most efficient triangulation algorithms available. The triangulated graph is then decomposed and
organised to form a *junction tree*, which supports a effective algorithms. The purpose of this section is to define junction trees and to show how to construct them. They provide a key tool for updating a Bayesian network.

**Definition 7.27** (Junction Trees). Let $\mathcal{C}$ be a collection of subsets of a finite set $V$ and $T$ be a tree with $\mathcal{C}$ as its node set. Then $T$ is said to be a junction tree (or join tree) if any intersection $C_1 \cap C_2$ of a pair $C_1, C_2$ of sets in $\mathcal{C}$ is contained in every node on the unique path in $T$ between $C_1$ and $C_2$. Let $\mathcal{G}$ be an undirected graph and $\mathcal{C}$ the family of its cliques. If $T$ is a junction tree with $\mathcal{C}$ as its node set, then $T$ is known as junction tree for the graph $\mathcal{G}$.

**Theorem 7.28.** There exists a junction tree $T$ of cliques for the graph $\mathcal{G}$ if and only if $\mathcal{G}$ is decomposable.

**Proof** The proof is by construction; a sequence is established in the following way. Firstly, a *simplicial node* $\alpha$ is chosen; $F_\alpha$ is therefore a clique. The algorithm continues by choosing nodes from $F_\alpha$ that only have neighbours in $F_\alpha$. The set of nodes $F_\alpha$ is labelled $C_1$ and the set of those nodes in $F_\alpha$ that have neighbours not in $F_\alpha$ is labelled $S_1$. This set is a *separator*.

Now remove the nodes in $F_\alpha$ that do not have neighbours outside $F_\alpha$ and name the new graph $\mathcal{G}'$. Choose a new node $\alpha$ in the graph $\mathcal{G}'$ such that $F_\alpha$ is a clique. Repeat the process, with the index $j$, where $j$ is the previous index, plus 1.

When the parts have been established (as indicated in the diagram below), each separator $S_i$ is then connected to a clique $S_j$ with $j > i$ and such that $S_i \subset C_j$. This is always possible, because $S_i$ is a *complete* set and, in the elimination sequence described above, the first point of $S_i$ is eliminated when dealing with a clique of index greater than $i$.

It is necessary to prove that the structure constructed is a tree and that it has the junction tree property.

Firstly, each clique has at most one parent, so there are not multiple paths. The structure is therefore a tree.

To prove the *junction tree* condition, consider two cliques, $C_i$ and $C_j$ with $i > j$ and let $\alpha$ be a member of both. There is a *unique* path between $C_i$ and $C_j$.

Because $\alpha$ is not eliminated when dealing with $C_j$, it is a member of $S_j$. By construction, it is also a member of the child of $C_j$, say $C_k$. Arguing similarly, it is also a member of the child of $C_k$ and, by induction it is also a member of $C_i$ and, of course, all the separators in between.

**Example 7.29.**

Consider the directed acyclic graph in Figure 7.11. The corresponding moral graph is given in Figure 7.12.

An appropriate elimination sequence for this moral graph is

$$(\alpha_8, \alpha_7, \alpha_4, \alpha_9, \alpha_2, \alpha_3, \alpha_1, \alpha_6).$$

There are two fill-ins; these are $\{\alpha_1, \alpha_5\}$ corresponding to the elimination of $\alpha_2$ and $\{\alpha_1, \alpha_6\}$, corresponding to the elimination of $\alpha_3$. The corresponding triangulated graph is given in Figure 7.13.
7.4. JUNCTION TREES

The junction tree construction may be applied. The cliques and separators, with the labels resulting from the diagram, are shown in Figure 7.14 and put together to form the junction tree, or join tree, shown in Figure 7.15.

Later, when using the algorithm for updating, it will be useful to designate one node as the root.

**Definition 7.30** (Rooted Tree). A rooted tree $T$ is a tree graph with a designated node $p$ called the root. A leaf of a tree is a node that is joined to at most one other node.

**Notes** The material is standard from algorithmic graph theory. See, for example, [44]. The proof of Theorem 7.17 follows the lines of Cowell, David, Lauritzen and Spiegelhalter in [29]. Exercises 1 to 7 of Section 7.5 are based on the article [117].
Figure 7.13: The triangulated graph corresponding to Figure 7.12

Figure 7.14: The Cliques and Separators from Figure 7.13
Figure 7.15: A Junction Tree (or join tree) constructed from the triangulated graph in Figure 7.13

Figure 7.16: Illustration of a Rooted Tree
7.5 Exercises: A Recursive Method for Learning a DAG

Recall the definition of weak decomposition (Definition 7.14). The results of the following exercise set are steps in the proof that the recursive algorithm given below returns the essential graph of a faithful DAG if there exists a faithful graphical representation. The algorithm assumes that independence statements $\alpha \perp \beta | S$ for variables $\alpha, \beta$ and subsets $S \subseteq V \setminus \{\alpha, \beta\}$ can be verified. The algorithm proceeds in the following way.

- An undirected graph is constructed. This is the graph $\mathcal{G} = (V, U)$ where $(\alpha, \beta) \in U$ if and only if $\alpha \not\perp \beta | V \setminus \{\alpha, \beta\}$. By Exercise 6 page 133, the graph constructed in this way is the moral graph of a faithful DAG.

- Find a weak decomposition (Definition 7.14) $(A, B, S)$ of the moral graph, if such a decomposition exists.

- For each $\alpha \in A \setminus S$ and $\beta \in B \setminus S$, set $S_{\alpha, \beta} = S$, the separator of $\alpha, \beta$.

- Construct $\mathcal{G}_{AuS}'$ and $\mathcal{G}_{BuS}'$, where for each $\gamma, \delta \in A \cup S$, $(\gamma, \delta) \in U_{AuS}'$ if and only if $\gamma \not\perp \delta | (A \cup S) \setminus \{\gamma, \delta\}$, similarly for $\mathcal{G}_{BuS}'$. These are the independence graphs for $A \cup S$ and $B \cup S$ respectively.

- Find weak decompositions of $\mathcal{G}_{AuS}'$ and $\mathcal{G}_{BuS}'$, the independence graphs associated with these weak decompositions and continue recursively until it is not possible to decompose any of these pieces further. At each stage, if $W \subseteq V$ is decomposed into $A', B', S'$, set $S_{\alpha, \beta} = S'$ for each $\alpha \in A', \beta \in B'$.

Before the assembly stage, the following additional stage is carried out on the pieces that cannot be decomposed further, when it is known a priori that the distribution has a faithful graphical representation:

- For each clique $A$ in the decomposition and each pair $\{\alpha, \beta\} \subseteq A$, check whether there is a subset $S \subseteq A \setminus \{\alpha, \beta\}$ such that $\alpha \perp \beta | S$. If there is, then remove the edge $\{\alpha, \beta\}$ and let $S_{\alpha, \beta} = S$, the sep-set of $\{\alpha, \beta\}$.

From these pieces, the DAG is constructed as follows:

- Two sub-skeletons $L_{AuS} = (A \cup S, U_{AuS})$ and $L_{BuS} = (B \cup S, U_{BuS})$ are combined to form

$$L_{AuBuS} = (A \cup B \cup S, U_{AuBuS})$$

where

$$U_{AuBuS} = U_{AuS} \cup U_{BuS} \setminus \{\{\alpha, \beta\}|\alpha, \beta \in S, \ (\alpha, \beta) \notin U_{AuS} \cap U_{BuS}\}.$$  

- This is done recursively until all the pieces have been added.

- For each separator $S_{\alpha, \beta}$, orient a vee-structure $(\alpha, \gamma, \beta)$ as an immorality $\alpha \rightarrow \gamma \leftarrow \beta$ if $\gamma \notin S_{\alpha, \beta}$.

- Orient the compelled edges (strongly protected edges).
7.5. **EXERCISES: A RECURSIVE METHOD FOR LEARNING A DAG**

### 7.5.1 Exercises

The first 7 exercises are of a theoretical nature; these are the main results proved in Xie-Geng [117] to verify the algorithm. In the remaining exercises the algorithm is applied.

1. Let $G = (V, D)$ be a directed acyclic graph. Let $\alpha, \beta \in V$. Let $F = G^n_{\text{An}(\{\alpha, \beta\} \cup S)}$ where $\text{An}(W)$ denotes the set $W$ together with all nodes that are ancestor nodes in $G$ for any node in $W$. First, the subgraph is taken, then it is moralised. Prove that $S$ separates $\alpha$ and $\beta$ in $F$ if and only if $\alpha \! \perp \! \beta | \text{G} S$.

2. Let $G = (V, D)$ and let $S \subset V$. Prove that two nodes $\{\alpha, \beta\}$ are $d$-separated by $S$ if and only if they are $d$-separated by $\text{an}(\{\alpha, \beta\}) \cap S$, where $\text{an}(W) = \text{An}(W) \setminus W$.

3. Let $G = (V, D)$ be a DAG and suppose that $\rho$ is a trail between two non-adjacent vertices $\alpha$ and $\beta$. Prove that if there are any nodes in $\rho$ that are not in $\text{An}(\{\alpha, \beta\})$, then the trail $\rho$ is blocked by any subset $S \subseteq \text{an}(\{\alpha, \beta\})$.

4. Let $G = (V, D)$ be a DAG. Suppose that $A \! \perp \! B | \text{G} S$ for three subsets $A, B, S \subset V$. Let $\alpha \in A$ and $\beta \in A \cup S$. Prove that $\alpha \! \perp \! \beta | \text{G} R$ for some $R \subset A \cup B \cup S$ if and only if $\alpha \! \perp \! \beta | \text{G} R'$ for a subset $R' \subset A \cup S$.

5. Prove that two non-adjacent nodes $\alpha$ and $\beta$ in a directed acyclic graph $G = (V, D)$ are $d$-separated by a set $S \subset V$ if and only if for any sequence $\lambda = (\alpha, \lambda_1, \ldots, \lambda_{n-1}, \beta)$ (where the same node can appear more than once) with edges between each consecutive pair

- either $\lambda$ contains a chain or a fork connection such that the chain node or fork node is in $S$ or
- $\lambda$ contains a collider connection such that the collider node is not in $S$ and has no descendant in $S$.

A sequence $\lambda$ with edges between each consecutive pair that satisfies this property is said to be blocked by $S$.

6. Let $G = (V, D)$ be a DAG and suppose that $A, B, S \subset V$ such that $A \! \perp \! B | \text{G} S$. Let $\alpha, \beta \in S$. Prove that there is a subset $R \subseteq A \cup B \cup S$ such that $\alpha \! \perp \! \beta | \text{G} R$ if and only if either there is a subset $R' \subseteq A \cup S$ or there is a subset $R' \subseteq B \cup S$ such that $\alpha \! \perp \! \beta | \text{G} R'$.

7. (a) Assuming that there is a faithful graphical representation for the distribution, prove that the skeleton of the graph returned by the algorithm is the skeleton of a faithful DAG.

(b) Assuming that there is a faithful graphical representation for the distribution, prove that the immoralities of the graph returned by the algorithm are those of a faithful DAG.

8. Consider a probability distribution for which the DAG in Figure 7.17 is faithful. Outline the steps of the algorithm to locate the DAG.
Figure 7.17: Structure learning example
7.6 Short Answers

1. Assume that there is a path from \( \alpha \) to \( \beta \) in \( F \) that has no nodes in \( S \). Then a trail from \( \alpha \) to \( \beta \) in \( G \) may be found by taking the directed edge in \( G \) if it corresponds to an edge in \( F \) or two edges to form a collider if there is no corresponding edge in \( F \); the two directed edges corresponding to the immorality that was removed when the graph was moralised.

If the collider node, or any of its descendants is in \( S \), then the node is \( S \)-active. Assume that there is one collider \( \gamma \) that is not \( S \)-active. Then each parent node (they are both in \( F \)) is either an ancestor of \( \alpha \) or an ancestor of \( \beta \) and hence the collider node is either an ancestor of \( \alpha \) or an ancestor of \( \beta \). It follows that there is a directed path from that node to \( \alpha \) or \( \beta \) that does not pass through \( S \). Assume that it is \( \alpha \) and consider the trail between \( \alpha \) and \( \beta \) with the part between \( \alpha \) and \( \gamma \) replaced by this directed path from \( \gamma \) to \( \alpha \).

Proceeding inductively, a trail can be constructed such that the only colliders are \( S \)-active and there are no other nodes in \( S \) on the trail. It follows that \( \alpha \parallel \beta | S \).

Now assume that all paths from \( \alpha \) to \( \beta \) in \( F \) have at least one node in \( S \). Consider any trail in \( G \) between \( \alpha \) and \( \beta \). The skeleton of any trail that has only fork or chain connections is in \( F \) and hence has a node in \( S \). Consider any trail in \( G \) and consider the \( S \)-active collider connections. In \( H \), there is an undirected edge \( (X, Y) \) for any collider connection \( (X, Z, Y) \) such that \( Z \) is \( S \)-active. If the trail has nodes not in \( \text{An}(\{\alpha, \beta\} \cup S) \), then it clearly has a collider that is uninstantiated and has no descendants in \( S \). If all the nodes of the trail are in \( \text{An}(\{\alpha, \beta\} \cup S) \), then since the undirected path in \( H \) formed by taking the directed edge \( (X, Y) \) instead of \( (X, Z), (Z, Y) \) has a node in \( S \), it follows that the original trail has a fork or chain node in \( S \) and hence is blocked.

2. Set \( S' = \text{an}(\{\alpha, \beta\}) \cap S \). Since \( S \supseteq S' \), it follows that if \( \alpha \parallel \beta | S' \) then (trivially) there is a subset \( R \in S \) such that \( \alpha \parallel \beta | G_R \).

Now suppose that \( \alpha \parallel \beta | G_{S'} \). From Exercise 1, there is a path \( \rho \) connecting \( \alpha \) and \( \beta \) in \( G_{\text{An}(\{\alpha, \beta\})} \) that does not contain any vertex of \( S' \) and hence that \( \rho \) does not contain any vertex in \( S \setminus \{\alpha, \beta\} \).

Suppose that \( \alpha \) and \( \beta \) are \( d \)-separated by \( S_0 \subseteq S \). Since \( \text{an}(\{\alpha, \beta\}) \cap S_0 \subseteq S' \), it follows that \( \rho \) does not contain any vertex in \( \text{an}(\{\alpha, \beta\}) \cap S_0 \) and hence, by Exercise 1, \( \alpha \parallel \beta | G_{S_0} \). It follows that if there is a subset \( R \subseteq S \) such that \( \alpha \parallel \beta | G_R \), then \( \alpha \parallel \beta | \text{an}(\{\alpha, \beta\}) \cap S \).

3. It is clear that such a trail contains a collider connection, where the collider node is not in \( \text{An}(\{\alpha, \beta\}) \) and hence the node does not in \( \text{an}(\{\alpha, \beta\}) \), nor does it have a descendant in this set.

4. Since \( A \cup B \cup S \supseteq A \cup S \), it follows trivially that existence of a suitable subset of \( A \cup S \) implies existence of a suitable subset of \( A \cup B \cup S \).

To prove that existence of a subset in \( A \cup B \cup S \) implies existence of a subset in \( A \cup S \), assume that \( \alpha \) and \( \delta \) are two vertices in \( A \) and \( A \cup S \) respectively, that are \( d \)-separated by a subset of \( A \cup B \cup S \). Let

\[ S' = (\text{an}(\{\alpha\}) \cup \text{an}(\{\delta\}) \cap (A \cup S) \]
By Exercise 2, it is sufficient to show that $S'$ blocks every trail $\rho$ between $\alpha$ and $\delta$. There are two cases:

- $\rho$ not contained completely in $\text{An}(\{\alpha, \delta\})$
- $\rho$ contained completely in $\text{An}(\{\alpha, \delta\}).$

By Exercise 3, in the first case, $\rho$ is blocked by $S'$ since $S' \in \text{an}(\{\alpha\}) \cup \text{an}(\{\delta\}).$

For the second case, $A \perp B|_G S$ implies that $\{\alpha\} \cup (S' \cap A) \perp \beta|_S$ for each $\beta \in B$ and hence (using Exercise 2 page 19) that $\alpha \perp \beta|_G (S' \cap A) \cup S$. Since $S' \subseteq A \cup S$, it follows that

$$\alpha \perp \beta|_G (S' \cup S).$$

Now suppose there is a trail $\rho$ contained in $\text{An}(\{\alpha, \delta\})$ between $\alpha$ and $\delta$ that is not blocked by $S'$. Let $W = S' \cup S$. Then $W$ blocks $\rho$. There is therefore at least one node in $\rho$ that is in $W \setminus S'$. Note that $W \in B$. Let $\gamma \in W \setminus S'$ denote the first node on the trail $\rho$, starting from $\alpha$, that is in $W \setminus S'$. Let $\rho'$ denote the sub-trail of $\rho$ between $\alpha$ and $\gamma$. Since $\rho$ is not blocked by $S'$, neither is $\rho'$. Since $\gamma$ is the only node of $\rho'$ that is in $B$, it follows that if $\rho'$ is $S'$ active, it is also $W$ active and hence $\alpha \not\perp \gamma|_G (S' \cup S)$, which is a contradiction.

5. If every sequence satisfies these properties, then clearly it satisfies these properties for every trail and hence, from the definition, $\alpha \not\perp \beta|_G S$.

If $\alpha \perp \beta|_G S$, then consider any such sequence of nodes. Take a subsequence by removing the loops so that any node appears at most once. This is a trail. Since $d$-separation holds, the trail has the property listed. The property therefore holds for the original sequence.

6. It is clear that if there is a set $R' \subseteq A \cup S$ or $R' \subseteq B \cup S$, then $R = R' \subseteq A \cup B \cup S$ satisfies the criterion.

Now suppose there is a set $\hat{R} \subseteq A \cup B \cup S$ such that $\alpha \perp \beta|_G \hat{R}$ and let $\gamma_1, \gamma_2 \in S$ such that $\gamma_1 \perp \gamma_2|_G \hat{R}$. By Exercise 2, $\gamma_1 \perp \gamma_2|_G R$ where

$$R = (\text{an}(\gamma_1) \cup \text{an}(\gamma_2)) \cap (A \cup B \cup S).$$

Suppose that $\gamma_2$ is not an ancestor of $\gamma_1$. This can be done without loss of generality, by exchanging the roles of $\gamma_1$ and $\gamma_2$ if necessary.

Let

$$R_1 = (\text{an}(\gamma_1) \cup \text{an}(\gamma_2)) \cap (A \cup S).$$

$$R_2 = (\text{an}(\gamma_1) \cup \text{an}(\gamma_2)) \cap (B \cup S).$$

To prove that $R_1$ or $R_2$ $d$-separate $\gamma_1$ and $\gamma_2$, it is sufficient to show that for two trails $\rho_1$ in $A \cup S$ and $\rho_2$ in $B \cup S$ either $\rho_1$ is $R_1$ active, or $\rho_2$ is $R_2$ active, or both.

Consider the two cases separately:

- One of the trails $\rho_2$ is not completely contained in $\text{An}(\{\gamma_1, \gamma_2\})$
7.6. SHORT ANSWERS

- both trails $\gamma_1$ and $\gamma_2$ are contained in $\text{An}(\{\gamma_1, \gamma_2\})$.

For the first case, since both $R_1$ and $R_2$ are subsets of $\text{an}(\gamma_1) \cup \text{an}(\gamma_2)$, it follows from Exercise 3 that $\rho_j$ is blocked by both $R_1$ and $R_2$.

Now consider the second case. Suppose that $\rho_1$ is $R_1$ active and $\rho_2$ is $R_2$ active. Both $\rho_1$ and $\rho_2$ are blocked by $R = R_1 \cup R_2$. It follows that $\rho_1$ has a node in $R \setminus R_1$ and $\rho_2$ has a node in $R \setminus R_2$.

Let $\delta_1$ and $\delta_2$ denote the nodes on $\rho_1$ and $\rho_2$ respectively that are closest to $\gamma_1$. As with the previous exercise, $\gamma_1 \in R \setminus R_1 \in B$ and $\gamma_2 \in R \setminus R_2 \in A$. Let $\rho'_1$ and $\rho'_2$ denote the subtrails of $\rho_1$ and $\rho_2$ respectively between $\gamma_1 \leftrightarrow \delta_1$, and $\gamma_1 \leftrightarrow \delta_2$ respectively. Note that $\rho'_1$ is $R_1$ active, and $\rho'_2$ is $R_2$ active. Connecting at $\gamma_1$ gives a sequence $\rho'$ between $\delta_1$ and $\delta_2$ through $\gamma_1$. Note that $\rho'$ may not be a trail, since there may be repeated nodes.

Any node that is not a collider node in $\rho'_1$, since it is in $\text{an}(\gamma_1) \cup \text{an}(\gamma_2)$ and since neither $\rho_1$ nor $\rho'_1$ are blocked by $R_1$, is not in $R_1 \cup S$. Similarly $S$ does not contain any collider node on $\rho'_2$. Therefore, except perhaps for $\gamma_1$, $\rho'$ does not have any collider connections where the collider node is in $S$.

Let $\nu_1$ denote the neighbour of $\gamma_1$ on $\rho'_1$. Since $\nu_1 \in \text{an}(\gamma_1) \cup \text{an}(\gamma_2)$ and it is not $\gamma_2$, it is an ancestor of $\gamma_1$ or $\gamma_2$. If the orientation is $\gamma_1 \rightarrow \nu_1$, then $\gamma_2$ is an ancestor of $\gamma_1$, contradicting the assumption. Therefore the edge is oriented $\nu_1 \rightarrow \gamma_1$. Similarly, for $\nu_2$ a neighbour of $\gamma_1$ on $\rho'_2$. It follows that $(\nu_1, \gamma_1, \nu_2)$ is a collider on $\rho'$. Therefore $S$ does not contain any nodes on $\rho'$ that are not collider nodes on the trail.

Consider any collider node $c$ in $\rho'_j$ (that is, the centre of a collider connection in $\rho'_j$). It is either in $R_j$ or else has a descendant in $R_j$. Since $c \in \text{an}(\gamma_1) \cup \text{an}(\gamma_2)$, it follows that $\gamma_1 \in S$ or $\gamma_2 \in S$ is a descendant of $c$. Since $\gamma_1 \in S$, it follows that each collider node in $\rho'$ is either in $S$ or has a descendant in $S$.

It follows that $\delta_1 \notin \delta_2|_G S$, contradicting $A \nparallel B|_G S$. It follows that either $\gamma_1 \nparallel \gamma_2|_G R_1$ or $\gamma_1 \nparallel \gamma_2|_G R_2$.

7. The result of Theorem 1.23 page 14, stating that a DAG $G = (V, D)$ has an edge between $\alpha$ and $\beta$ in $D$ if and only if $\alpha \nparallel \beta|_G S$ for any subset $S$, is used crucially here, together with the definition of ‘faithful’, that conditional independence statements and $d$-separation statements are equivalent.

(a) Suppose that $G_{A \cup C}$ and $G_{B \cup C}$ are faithful for the distributions over $A \cup C$ and $B \cup C$ respectively and are combined according to the rules given to give $G_{A \cup B \cup C}$. The results of exercises 4 and 6 may be used to establish the $d$-separation properties.

For any $\alpha \in A$ and $\beta \in B$, $S_{\alpha \beta} = C$ and therefore there is no edge $\alpha \sim \beta$ in a faithful DAG for the distribution over $A \cup B \cup C$. Following the reconstruction, there is no edge in $G_{A \cup B \cup C}$.

For $\alpha, \beta \in C$, the reconstruction has an edge $\alpha \sim \beta$ in $G_{A \cup B \cup C}$ if and only if there are edges in both $G_{A \cup C}$ and $G_{B \cup C}$. Exercise 6 states that if $G(A \cup B \cup C)$ is a DAG over the variables $A \cup B \cup C$ and there is a set $R \subseteq A \cup B \cup C$ such that $\alpha \nparallel \beta|_{G(A \cup B \cup C)} R$ if and only if either
there is a set $R' \subseteq A \cup C$ such that $\alpha \perp \beta |_{\mathcal{G}(A \cup C)} R'$ or there is a set $R' \subseteq B \cup C$ such that $\alpha \perp \beta |_{\mathcal{G}(B \cup C)} R'$. Therefore, $\mathcal{G}(A \cup B \cup C)$ is a faithful graph for $p_{A \cup B \cup C}$ then its skeleton contains an edge $\alpha \sim \beta$ between two variables in $C$ if and only if both $\mathcal{G}_{A \cup C}$ and $\mathcal{G}_{B \cup C}$ contain the edge.

For $\alpha \in A$ and $\beta \in C$, the graph $\mathcal{G}_{A \cup B \cup C}$ in the reconstruction has an edge $\alpha \sim \beta$ if and only if there is an edge $\alpha \sim \beta$ in the graph $\mathcal{G}_{A \cup C}$. Exercise 4 states that if $\mathcal{G}(A \cup B \cup C)$ is a DAG over the variables $A \cup B \cup C$ then there is a set $R \subseteq A \cup B \cup C$ such that $\alpha \perp \beta |_{\mathcal{G}(A \cup B \cup C)} R$ if and only if there is a set $R' \subseteq A \cup C$ such that $\alpha \perp \beta |_{\mathcal{G}(A \cup C)} R'$. It follows that if $\mathcal{G}(A \cup B \cup C)$ is faithful for $p_{A \cup B \cup C}$ then its skeleton contains an edge $\alpha \sim \beta$ between two variables $\alpha \in A$ and $\beta \in C$ if and only if $\mathcal{G}(A \cup C)$ contains an edge between $\alpha$ and $\beta$.

(b) If $\alpha, \beta \in V$, then any vee-structure $\alpha \prec \gamma \prec \beta$ such that $\gamma \notin S_{\alpha \beta}$ is an immorality, hence the immoralities are correct.

8. The moral graph is given in Figure 7.18.

Decomposition into $\{A, C, D\}$ and $\{B, C, D, E, F, G, H\}$; $A \perp \{B, E, F, G, H\}$; $\mathcal{G}(A \cup B \cup C) \{C, D\}$. Consider $\{A, C, D\}$. The graph is; $A \perp D \perp C$, since for variables $\{A, D, C\}$ $C \perp D|A$. This is decomposed further into $\{A, D\}$ and $\{A, C\}; C \perp D|A$; this decomposition is complete.

Now consider $\{B, C, D, E, F, G, H\}$. Then $B \perp \{D, F, H, G\}|\{C, E\}$; $\{B, C, E\}$ remains a clique at this stage, since $B \perp C|E$.

The other part is $\{C, D, E, F, G, H\}$. $E \perp \{D, F, H\}|\{C, G\}$, so it is decomposed into $\{C, D, F, G, H\}$ and $\{C, E, G\}$. $\{C, E, G\}$ remains a clique at this stage, since $C \perp G|E$.

For $\{C, D, F, G, H\}$, $C \perp G|\{D, F, H\}$, so the graph does not contain the edge $C \perp G$.

Then $G \perp \{C, F, D\}|H$, so decompose into $\{G, H\}$ and $\{C, D, F, H\}$.

Now consider $\{C, D, F, H\}$ and decompose into $\{C, D, F\}$ and $\{H, D, F\}; C \perp H|\{D, F\}$. Since $C \perp D|F$, this remains a clique. Since $D \perp H|F$, this remains a clique.
Finally, for \( \{C,D,F\} \) and \( \{D,F,H\} \), the edge \( C - D \) is not removed; \( C \not\perp D \) and \( C \not\perp D|F \). The edge \( D - H \) is removed, with separation set (sep set) \( \phi \), since \( D \perp H \) (with no instantiated notes, there is an open collider in each trail in the original DAG).

For the final stage of the ‘deconstruction’ phase, edge \( B - C \) is removed because \( B \perp C \), with sep set \( S_{BC} = \phi \).

For the reconstruction, these are put together, using the rule that \( G_{AuBuC} \) has an edge between two variables in \( C \) if and only if both \( G_{AuC} \) and \( G_{BuC} \). At this stage, the edge \( C - D \) is removed from the final graph, since at the earlier stage \( S_{CD} = \{A\} \). Similarly, \( C - G \) is removed because \( S_{CG} = \{D,F,H\} \).

Vee structures \((\alpha,\gamma,\beta)\) are immoralities if and only if \( \gamma \not\perp S_{\alpha,\beta} \). This gives the essential graph of Figure 7.19.