To the Graduate Council:

I am submitting herewith a dissertation written by Robert E. England entitled "Clique Graph Models for Independent Computations." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Michael G. Thomason

M. G. Thomason, Major Professor

We have read this thesis
and recommend its acceptance:

Jean R. S. Blair

Richard Allen

J H Moore

R. C. Gonzalez

Accepted for the Council:

C W Minkel

Vice Provost
and Dean of The Graduate School

CLIQUE GRAPH MODELS FOR INDEPENDENT COMPUTATIONS

A Dissertation

Presented for the

Doctor of Philosophy Degree

The University of Tennessee, Knoxville

Robert E. England

December 1989

To Richard and Dawn.

ACKNOWLEDGEMENTS

Dr. M. G. Thomason and Dr. J. R. S. Blair have provided me with invaluable guidance, encouragement, and support through every step of this research. For their unfailing confidence in me, I am honored and deeply grateful. I appreciate and thank others who served on my committee: Dr. J. H. Poore, Dr. R. C. Gonzalez, and Dr. R. C. Mann.

I thank my dear friends Fami Salkjelsvik, Elizabeth Jones, Norman Sims, and Andy Chen for making my experience here as rewarding personally as it has been intellectually. I also thank Wallace Mayo for her genuine interest in my welfare and for her wise and treasured advice.

ABSTRACT

A good solution to a computing problem that involves massive amounts of data depends heavily on the formulation of a good model of the structure of relationships among the data elements of the problem. If strongly related data can be grouped for similar processing, then the strong relationships should be apparent in the model. If weakly related data can be processed independently, then this also should be apparent in the model. One model that shows structural interrelationships clearly is the clique graph. Clique graphs are used in the development of divide-and-conquer algorithms, sparse matrix factorization schemes, and information propagation in knowledge-based systems, among other important applications.

This dissertation develops new results in clique graphs with emphasis on applications that involve independent processing. A new representation of clique graph structure based on clique intersections, called *exclusive intersections*, is introduced. Templates based on exclusive intersections that describe structural patterns in clique graphs are proposed. An important new method developed by Lauritzen and Spiegelhalter for handling uncertainty in knowledge-based systems uses a clique graph model to which these results are applicable. New results in this dissertation demonstrate how the amount of computation may be reduced and certain processes may be executed independently within the Lauritzen - Spiegelhalter method.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND OVERVIEW

1.1 Introduction

A good solution to a computing problem that involves massive amounts of data depends heavily on the formulation of a good model of the structure of relationships among the data elements of the problem. If strongly related data can be grouped for similar processing, then the strong relationships should be apparent in the model. If weakly related data can be processed independently, then it should be apparent in the model where this is possible.

One commonly used model for the structure of data relationships in computing problems is the *graph*. In a graph, points called nodes represent individual data items and lines called arcs connecting these nodes represent relationships among these items. A model which provides a precise description of the relationships among groups of data items is the *clique graph*. Specifically, a clique graph shows the intersections among the cliques, or completely connected subsets of nodes, of a graph.

Clique graphs have proved to be an important type of model in computing applications where independent processing and parallelism are important. For example, the solution of sparse systems of linear equations [37,39,46,52], the determination of shortest connection networks [48], graph partitioning [38], general parallel algorithm development [6,11,24], the manipulation of probabilistic uncertainty in knowledge-based systems [32], and the improvement of data base organization schemes [4] have each been approached using a clique graph model. Several of these applications involve a flow of information through the clique graph model structure. Methods to facilitate this flow are sought that minimize the total amount of data that must be passed or processed by reducing the amount of redundant information propagated among cliques or by reducing the total distance that data must travel. Considerations to this end have led to much recent exploration of the fundamental properties of clique graphs as abstract structures. Some of the specific areas receiving attention are clique separators, which divide a graph into separate components [20,34,36,61], relaxation labeling and vertex elimination [31,53,54], minimum clique covering [17], and fundamental triangulated graph algorithms [29,40,41].

This dissertation presents new properties of clique graphs and introduces a model based on intersection sets. These results expand general knowledge about clique graphs and can be used in the development of new algorithms that operate on sections of a graph at a time rather than on the nodes and arcs individually. Many existing algorithms can be improved by incorporating the new knowledge in these results. For example, it will be shown in this paper how the search for a maximal spanning tree of a clique graph can be conducted as several smaller searches operating independently within different parts of the graph — an improvement over standard maximal spanning tree algorithms which operate sequentially.

The new *exclusive intersection set* model for the structure of a graph provides a rich base for further exploration of the properties of clique graphs. Among the results developed in this paper using the exclusive intersection set model are rules that allow inferences to be made concerning the emptiness or nonemptiness of cliques and clique intersections in a graph without examining each possible clique or clique intersection explicitly. Also, it is shown that the effects on the clique structure of a graph that occur when nodes and arcs are added or deleted can be precisely predicted if that graph is modeled using exclusive intersection sets.

As an illustration of the utility of the new results in this paper, a specific application is described: the method of Lauritzen and Spiegelhalter [32] is an important new technique for clique-based propagation of probabilities in knowledge-based computation. Results discussed above and others developed specifically for use with this method describe precise conditions for independent processing within the method and thereby offer improvements in both sequential and parallel implementations.

1.2 Probabilities in Knowledge-based Systems

Knowledge-based systems typically involve computations with large amounts of data, much of which may be incomplete or inexact. Various approaches to the problem of dealing with uncertainty in knowledge are currently being investigated, including techniques that involve belief functions [13,43,58], fuzzy reasoning, or Bayesian probabilities [12,15,28,35,45,56,57].

One method of handling uncertainty receiving considerable attention in computer science and related fields is described by Lauritzen and Spiegelhalter in a 1988 paper [32]. This approach combines techniques from graph theory, probability theory, statistics, and expert systems to compute exact Bayesian probabilities based on a highly structured graphical model. In addition to the utility of the method itself, its foundation in rigorous, discrete mathematics has generated and focused interest from a wide variety of disciplines, as evidenced by the large number of comments and contributions immediately following the initial presentation of the techniques in 1988 [32]. The method forms the foundation for the HUGIN core – an extensive project geared toward providing a general framework for handling uncertainty in large systems modeled as networks [1].

Rather than propose a general solution to the problem of handling uncertainty, Lauritzen and Spiegelhalter explore the extent to which a probabilistic model is capable of fulfilling the goals of knowledge-based systems. In the method they describe, expert knowledge is stored in the form of conditional probability tables which implicitly define a Markov network. The probabilities required may be provided by an expert or inferred from relevant data, if available. Conclusions drawn on inferences within the defined model are based on the actual data and on principles from probability and statistical theory. Provisions are made for the incorporation of new evidence into the system. The graphical properties of the underlying Markov network are explicitly used.

The contributions of this dissertation to the method of Lauritzen and Spiegelhalter arise from their use of a clique graph model for the Markov network: results here enable rapid location of the separators in the network, simplify updates to the system model as changes are made to the set of nodes in the network, reduce the total amount of computation required by the method, and specify sections of the method which may be executed independently.

1.3 Outline

The new results developed as part of this work are presented rigorously as theorems and proofs. This dissertation is organized as follows:

Chapter 2 gives basic definitions and introduces newly developed properties of clique graphs. Rules are given to test the validity of a graph as a clique graph. Also presented are distinguishing characteristics of the arcs in a clique graph that may appear as part of a maximal spanning tree of the clique graph. The primary proof technique used in Chapter 2 is to examine the underlying graphical structure implied by a portion of a clique graph to establish whether or not such a structure is indeed possible.

Chapter 3 introduces a new model of graph structure based on the degree to which groups of nodes in the graph are interconnected. In the model, a graph is partitioned into sets of nodes called *exclusive intersection sets* which are subsets of the cliques in the graph. Nodes within an exclusive intersection set share important properties with respect to their relationship to the rest of the graph. Graphs represented in this form may be processed abstractly as a class in the formulation of new general solutions to standard problems in graphical modeling. New algorithms that are able to process groups of nodes at each step rather than individual nodes may be developed for applications such as finding a perfect elimination ordering for the nodes of a graph [54].

Chapter 4 briefly describes the steps of the Lauritzen - Spiegelhalter method, a clique graph model for local calculation of probabilities associated with variables in a knowledge-based system. The method as described by Lauritzen and Spiegelhalter operates strictly sequentially. Results suggested in Chapter 4 and developed in Chapter 5 specify areas where computation during the execution of the method could be done independently, or in some cases completely avoided.

Using results about clique graphs, Chapter 5 gives specific tests within the algorithms of the Lauritzen - Spiegelhalter method to characterize explicitly where computation can be reduced and different phases of the method can be executed independently. These tests, the properties of which are proved rigorously, establish the cases in which probabilities associated with parts of the graphical model are left unchanged or are returned to previously known values by some of the propagation steps in the method. Theorems are derived that explain why these phenomena occur and the conditions under which certain data transformations may be omitted and others may take place concurrently.

Chapter 6 is a discussion of possibilities for future related research.

CHAPTER 2

CLIQUE GRAPHS

2.1 Introduction

This chapter will focus on the development of theorems that describe properties of a special type of graph called a clique graph which diagrams the relationships among completely connected groups of nodes in another graph.

2.2 Concepts from basic graph theory

Many types of problems that arise in computer science are easier to understand and solve if they are represented as diagrams called *graphs*. For this discussion, a graph G is defined to be a pair (V, E), where V is a set of *vertices* or nodes, and E is a set of *edges* or arcs between the nodes. Intuitively, V represents the elements of a problem of interest, and E represents relationships among these elements. If V is the set $\{x_1, x_2, \ldots, x_n\}$, then a typical element of E could be written $x_i x_j$, where $1 \leq i, j \leq n$. Nodes and arcs may or may not be labeled, depending on the nature of the problem represented by the graph.

It is assumed that self-loops such as $x_i x_i$ or multiple arcs between any pair of vertices are not allowed in a graph. Graph G is an *undirected graph* if $x_i x_j$

and x_jx_i are identically the same edge in E; if x_ix_j and x_jx_i are different edges, then G is a *directed graph.* If x_ix_j is an edge in E, then x_i and x_j are *adjacent vertices.* Similarly, any two edges in E that share a vertex, say x_ix_j and x_jx_k, are called *adjacent edges.* A sequence $x_ix_j \ldots x_k$ of distinct adjacent vertices is called a *path* from x_i to x_k. Figure 2.1 shows an undirected graph in which

$$V = \{x_1, x_2, \ldots, x_6\}\,, \text{ and}$$

$$E = \{x_1x_2, x_1x_3, x_2x_4, x_2x_5, x_2x_6, x_3x_4, x_4x_5\}\,.$$

Extensive use of graphs as a tool has lead to the development of many graph theoretic rules and transformations that can help simplify or solve problems that are represented as graphs. Careful examination of a graph often reveals features that determine the applicability of certain of these rules. The features that will be discussed here are not an exhaustive list, but are intended as a basis for the development of theorems that appear later in this paper.

One interesting feature found in the graph in Figure 2.1 is called a *cycle.* A cycle is a path in which the first and last nodes in the sequence are connected by an arc. The cycles in Figure 2.1 are $x_1x_2x_4x_3$, $x_2x_4x_5$, and $x_1x_2x_5x_4x_3$. Note that in the notation for a cycle, the order of the vertices is important to show where the adjacencies lie, and the final edge connecting the first and last vertices written for the cycle is implicit.

An edge between two nonadjacent vertices of a cycle is called a *chord.* For example, in the graph in Figure 2.1, x_2x_4 is a chord of the cycle $x_1x_2x_5x_4x_3$. If every cycle of length four or greater in a graph has a chord, then the graph

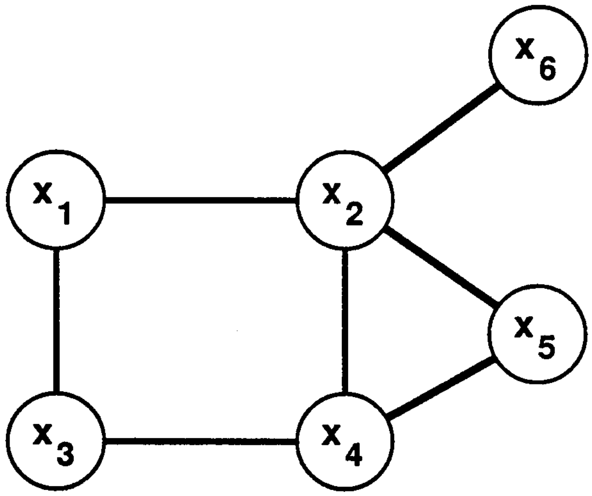


Figure 2.1: An undirected graph.

is said to be *triangulated.* The graph in Figure 2.1 is not triangulated because there is no chord of the cycle $x_1x_2x_4x_3$.

2.3 Clique graphs

A subset of vertices of G is called a *completely connected* set, abbreviated *CCS*, if there is an edge in E between each pair of vertices in the set. A *clique*[1] is a completely connected set of vertices that is not contained in any larger completely connected set. The set of cliques in a graph will usually be denoted as

$$Z = \{C_1, C_2, \ldots, C_m\},$$

where each C_i is a clique in G. For example, the cliques in the graph in Figure 2.1 are

$$\begin{aligned} C_1 &= \{x_1, x_2\} \\ C_2 &= \{x_1, x_3\} \\ C_3 &= \{x_2, x_4, x_5\} \\ C_4 &= \{x_2, x_6\}, \text{ and} \\ C_5 &= \{x_3, x_4\}. \end{aligned}$$

The set $\{x_1, x_2, x_3, x_4\}$ is not a clique since there are no edges x_1x_4 and x_2x_3

[1]Some writers refer to any completely connected set of vertices in a graph as a clique, and *maximal clique* is the term used with the definition of a clique given here.

in E. The set $\{x_2, x_5\}$ is not a clique since it is contained in a larger completely connected set.

Later in this paper, it will be shown how cliques can play an important role in the characterization of a graph by allowing the breakdown of a graph into groups of vertices that have quite similar properties with respect to their relationship to the rest of the graph.

Once the cliques in a graph G have been determined, they can be used as the set of vertices for yet another undirected graph called a *clique graph*, written G_C. The set of arcs E_C of a clique graph G_C contains an arc C_iC_j if and only if the intersection of C_i and C_j in G is not empty; the contents of this intersection is then used as a label for the arc C_iC_j. The clique graph for the graph in Figure 2.1 is shown in Figure 2.2. The *weight* of an arc C_iC_j in G_C is defined to be the number of nodes of G in the label for that arc, and for this the notation $|C_iC_j|$ is used. For clarity, from this point the term *original graph* and the symbol OG will be used to refer to an initial graph on which a corresponding clique graph G_C is based.

A clique graph is a summary of the structure of another graph, and as such there is no guarantee that an arbitrarily labeled clique graph will in fact correspond to any possible original graph, as will be demonstrated by the example following Theorem 2.1. A clique graph G_C is *valid* if a corresponding original graph OG can possibly be constructed. A method for constructing an original graph from a given valid clique graph will be presented in Chapter 3. Based on the definition of a clique graph, rules can be derived which describe valid

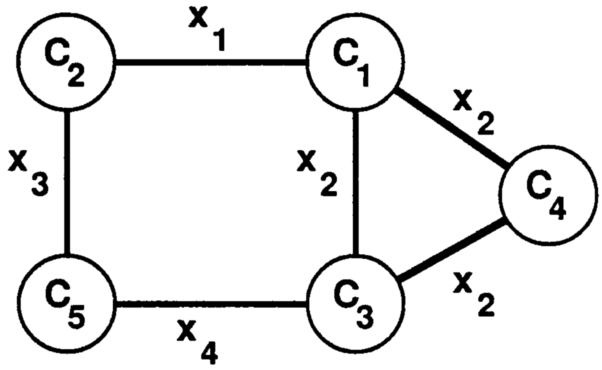


Figure 2.2: A clique graph.

clique graph structures and labeling schemes. The first of these rules are given by the following group of theorems, which deal specifically with *triangles*, or completely connected sets of three vertices, within clique graphs.

2.4 Triangles in clique graphs

For the theorems in this section, we assume the existence of some arbitrary original graph OG with cliques $Z = \{C_1, C_2, \ldots, C_m\}$ and corresponding clique graph G_C. The term *side* will be used to refer to the label on an edge in G_C.

Theorem 2.1 *All three sides of any triangle in G_C must be contained in some clique.*

Proof. Let C_i, C_j, and C_k be the vertices of some triangle in G_C. We must then show that if $x_a, x_b \in (C_iC_j \cup C_iC_k \cup C_jC_k)$, then $x_ax_b \in E$. Note that $[x_a, x_b \in C_A] \Rightarrow [x_ax_b \in E]$ for any clique C_A.

Suppose $x_a \in C_iC_j$. Then $x_b \in (C_iC_j \cup C_iC_k \cup C_jC_k)$ implies at least one of the following:

Case 1:

$$\begin{aligned} [x_b \in C_iC_j] &\Rightarrow [x_a, x_b \in C_i] \\ &\Rightarrow [x_ax_b \in E] \end{aligned}$$

Case 2:

$$\begin{aligned} [x_b \in C_iC_k] &\Rightarrow [x_a, x_b \in C_i] \\ &\Rightarrow [x_ax_b \in E] \end{aligned}$$

Case 3:

$$\begin{aligned} [x_b \in C_jC_k] &\Rightarrow [x_a, x_b \in C_j] \\ &\Rightarrow [x_ax_b \in E] \end{aligned}$$

Arguments are similar for $x_a \in C_iC_k$ and $x_a \in C_jC_k$. Since there exists an edge in E between each pair of vertices in $C_iC_j \cup C_iC_k \cup C_jC_k$, this set of vertices must be contained in some clique. □

Example. If $C_1C_2C_3$ is a triangle in G_C and $C_1C_2 = x_2x_3$, $C_2C_3 = x_4$, and $C_1C_3 = x_1$, then $\{x_1, x_2, x_3, x_4\}$ must be contained in some clique C_a in Z. (See Figure 2.3.)

Clique C_a cannot be equal to C_1, C_2, or C_3 since none of these cliques contains $\{x_1, x_2, x_3, x_4\}$, which must be contained in C_a. Thus, the graph in Figure 2.3 is not a valid clique graph since it does not show clique C_a, which must be present. □

Theorem 2.2 *If one side of a triangle in G_C is contained in another side of the same triangle, then a clique that contains all sides is the node opposite the first side.*

Proof. Suppose C_i, C_j, and C_k are the vertices of some triangle in G_C. Suppose further that $C_iC_j \subseteq C_iC_k$, and that $x \in C_iC_j \cup C_iC_k \cup C_jC_k$. Then we must show $x \in C_k$.

We know x must be contained in at least one of C_iC_j, C_iC_k, or C_jC_k. If $x \in C_iC_k$ or $x \in C_jC_k$ then clearly $x \in C_k$. If $x \in C_iC_j$, then since $C_iC_j \subseteq C_iC_k$, we have $x \in C_iC_k$, and $x \in C_k$. □

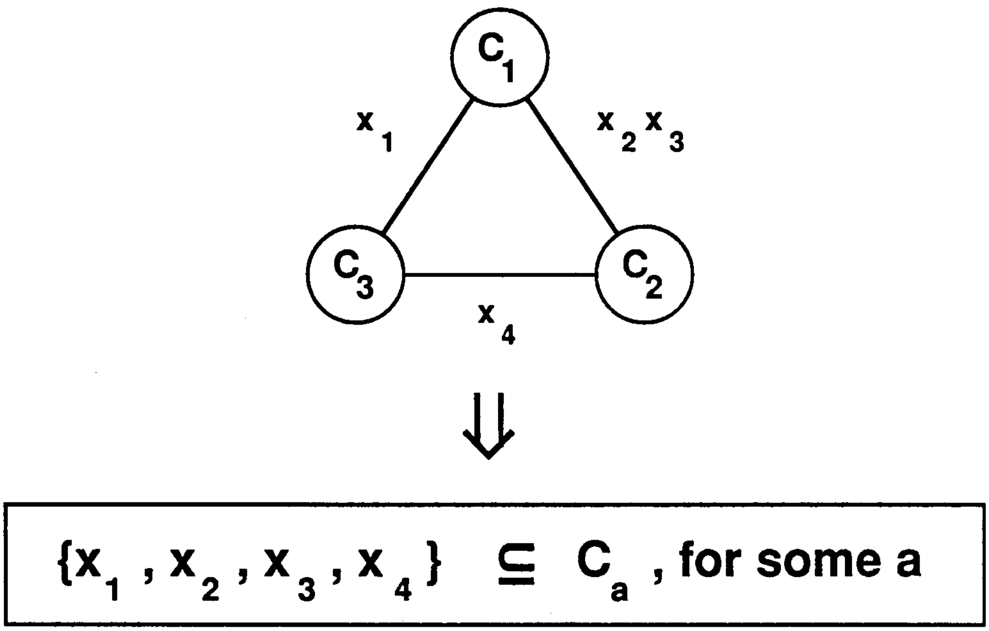


Figure 2.3: All three sides of any triangle in a clique graph must be contained in some clique.

Example. If $C_1C_2C_3$ is a triangle in G_C and $C_1C_2 = x_2x_3$, $C_2C_3 = A$, where A is some set of nodes in OG, and $C_1C_3 = x_1x_2x_3$, then

$$\{x_1, x_2, x_3,\} \cup A$$

must be contained in clique C_3. (See Figure 2.4.) □

Corollary 2.3 *If one side of a triangle in G_C is contained in one of the other sides, then it is contained in both of the other sides.*

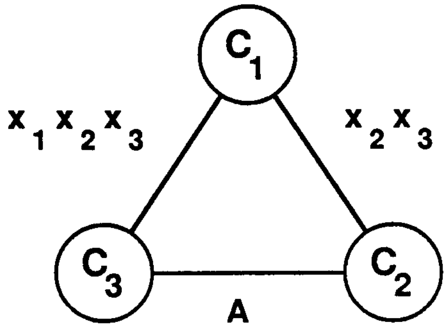
Proof. Given triangle $C_iC_jC_k$ as above with $C_iC_j \subseteq C_iC_k$, then $[x \in C_iC_j] \Rightarrow [x \in C_i \text{ and } x \in C_k]$, by Theorem 2.2. Thus $x \in C_iC_k$. □

Example. If we take triangle $C_1C_2C_3$ as in the example for Theorem 2.2, then it must be that $\{x_2, x_3\}$ is contained in $A = C_2C_3$. (See Figure 2.5.) □

Theorem 2.4 *If no side of a triangle in G_C is contained in another side, then there exists some clique outside the triangle with an arc to each vertex of the triangle, such that the weight of that arc is greater than the weight of either side of the triangle adjacent to that vertex.*

Proof. Suppose there exists some triangle $C_iC_jC_k$ in G_C such that there exist x_a, x_b, and x_c where

$$\begin{aligned} x_a &\in C_iC_j - C_k \\ x_b &\in C_jC_k - C_i \\ x_c &\in C_iC_k - C_j \end{aligned}$$



$$\Downarrow$$

$$\boxed{\{x_1, x_2, x_3\} \cup A \subseteq C_3}$$

Figure 2.4: If one side of a clique graph triangle is contained in another side, then all three sides are contained in the clique opposite the first side.

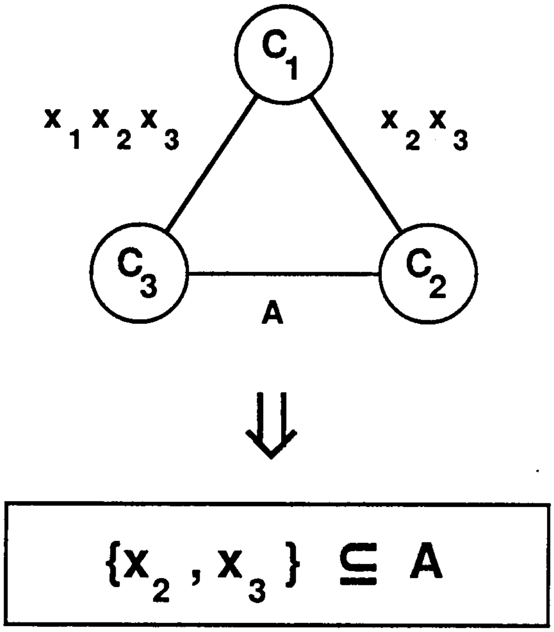


Figure 2.5: If one side of a clique graph triangle is contained in another side, then it is contained in both other sides.

By Theorem 2.1, there exists some clique C_A in G_C such that

$$C_A \supseteq (C_iC_j \cup C_iC_k \cup C_jC_k)$$

Thus we have $x_a, x_b, x_c \in C_A$, but $\{x_a, x_b, x_c\} \not\subseteq C_i, C_j$, or C_k. So $C_A \not\subseteq C_i, C_j$, or C_k. Now $C_iC_k \subseteq C_A$ implies $C_iC_k \subseteq C_AC_i$. Similarly, $C_iC_j \subseteq C_AC_i$. But, by assumption, there exists some $x_a \in C_AC_i$ with $x_a \notin C_iC_k$. Therefore, $C_iC_k \subset C_AC_i$ and $|C_AC_i| > |C_iC_k|$. Similarly, $|C_AC_i| > |C_iC_j|$, so $|C_AC_i| > \max(|C_iC_j|, |C_iC_k|)$.

Arguments are analogous for C_AC_j and C_AC_k. □

Example. If $C_1C_2C_3$ is a triangle in G_C with $C_1C_2 = x_2x_3$, $C_2C_3 = x_4$, and $C_1C_3 = x_1$, then $\{x_1, x_2, x_3, x_4\}$ must be contained in some clique C_a which has arcs in G_C to each of C_1, C_2, and C_3 with weights which are greater than either triangle side adjacent to each of these three cliques, as shown in Figure 2.6. □

Theorem 2.5 *For any side C_iC_j of a triangle $C_iC_jC_k$ whose weight is strictly less than the weight of each of the other two sides of the triangle, there exists some triangle involving C_iC_j such that the other two sides in the triangle properly contain that smallest side.*

Proof. By Theorem 2.1, there exists some clique C_A in G_C such that $C_iC_j \subseteq C_A$ and $C_iC_k \subseteq C_A$. This implies $C_iC_j \subseteq C_iC_A$ and $C_iC_k \subseteq C_iC_A$. But since C_iC_j has minimum weight in triangle $C_iC_jC_k$, we have $|C_iC_j| < |C_iC_k|$, and there must be some vertex $x \in (C_iC_k - C_j)$, with x also in C_iC_A. Thus, $|C_iC_j| <$

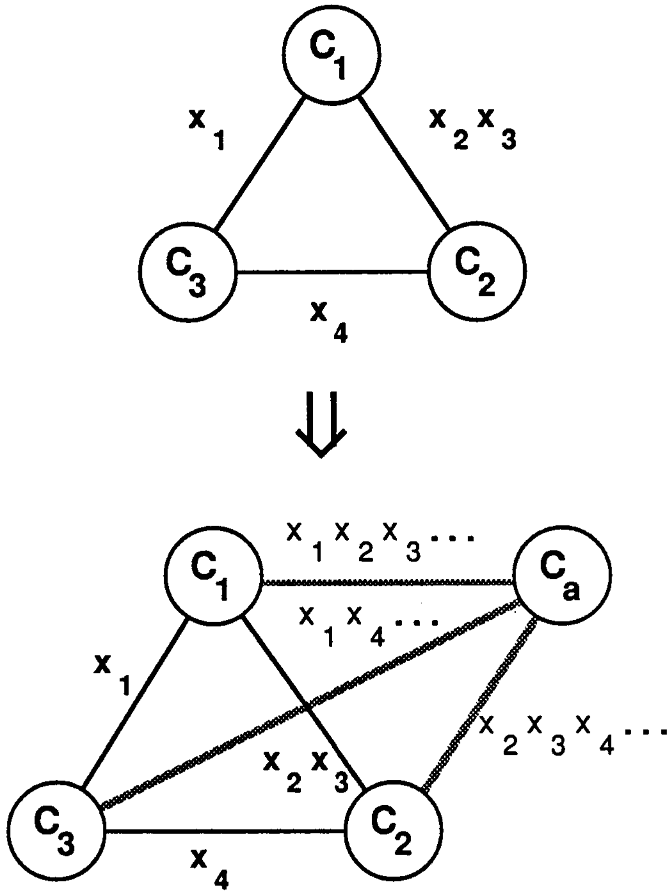


Figure 2.6: If no side of a clique graph triangle is contained in another side, then some clique outside the triangle has an arc to each vertex that contains the sides adjacent to the vertex.

$|C_iC_A|$, which means $C_iC_j \subset C_iC_A$. By a similar argument, $C_iC_j \subset C_jC_A$, so $C_iC_jC_A$ is the triangle we seek. □

Example. In Figure 2.7(a), the smallest weight side C_1C_2 of triangle $C_1C_2C_3$ is contained in another side of the triangle, so side C_1C_2 is contained in both of the other sides of the triangle by Corollary 2.3. In Figure 2.7(b), no side of triangle $C_1C_2C_3$ is contained in another side of the triangle, so all sides of this triangle must be contained in some clique C_a outside the triangle by Theorem 2.4, and the smallest weight side C_1C_2 is properly contained in each of the other two sides of the triangle that C_a forms with C_1 and C_2. □

Corollary 2.6 *For any side of a triangle whose weight is less than or equal to the weight of each of the other two sides of the triangle, there exists some triangle, all of whose sides contain that smallest side.*

Proof. Analogous to the proof of Theorem 2.5. □

The previous set of basic results that describe completely connected sets of three nodes in a clique graph forms the foundation for many of the results in the following section. In Chapter 3, ideas introduced here will be expanded into more general theorems that apply to any size completely connected subset of clique graph nodes. These theorems are a step toward a precise characterization of valid clique graph structure and supply knowledge that can be used to develop rapid algorithms for solving problems that are modeled as clique graphs.

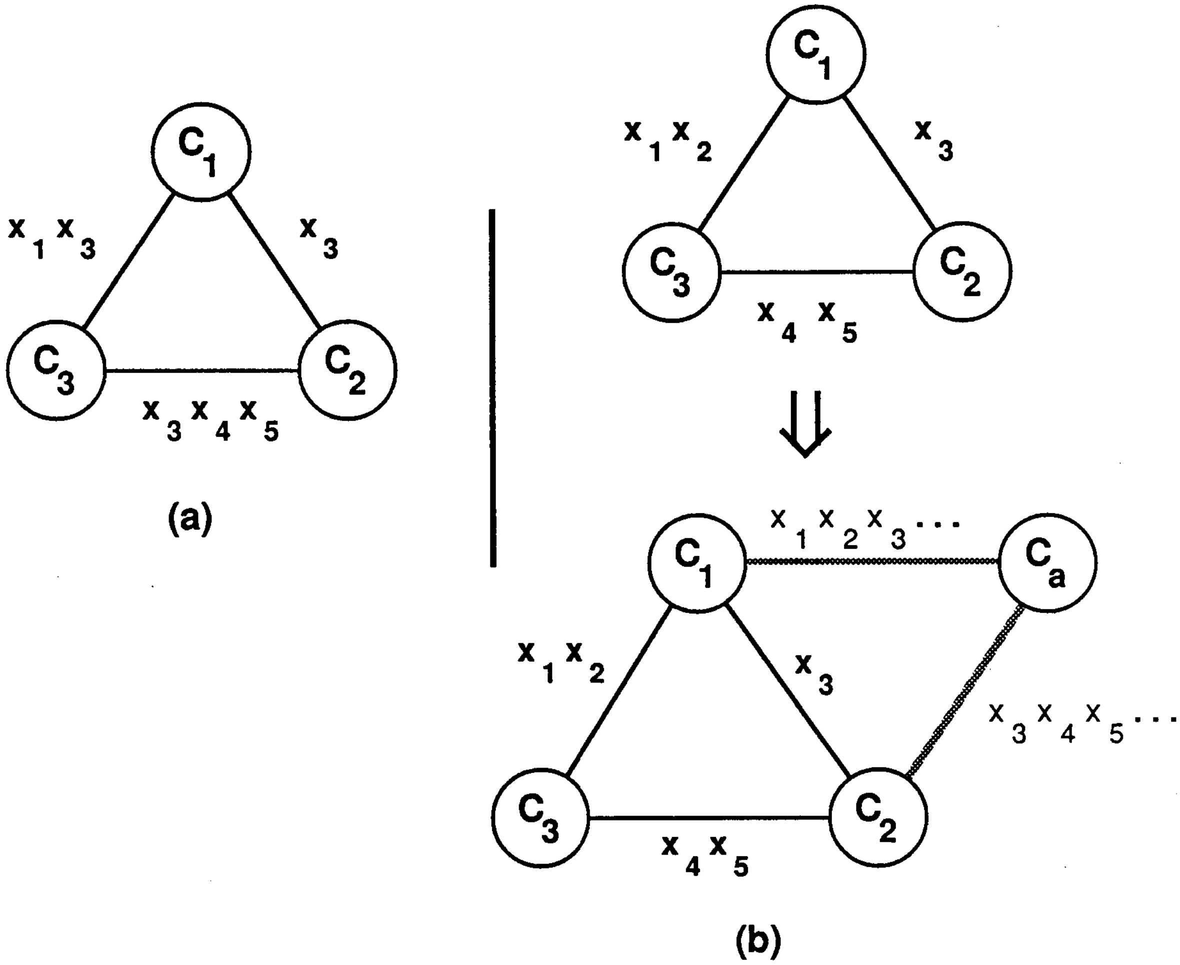


Figure 2.7: The smallest weight side of a clique graph triangle must be a part of some triangle, both of whose sides properly contain that smallest weight side.

2.5 Maximal spanning trees of clique graphs

Given some set of vertices V, a *spanning tree* is a graph in which there is exactly one path between each pair of vertices in V. Note that a spanning tree contains no cycles. Given some graph $OG = (V, E)$ with weighted arcs, then a *maximal spanning tree* of OG is a graph with the same vertices V, and arcs $E_M \subseteq E$ that form a spanning tree such that the total weight of the arcs in E_M is greater than or equal to the total weight of the arcs of any spanning tree of OG. Standard algorithms exist for finding a maximal spanning tree of a graph with weighted arcs [48].

The set of results presented in this section demonstrate that because of special properties of valid clique graphs, certain arcs in a clique graph may automatically be eliminated from consideration for inclusion in any maximal spanning tree of that clique graph. Also, rules are given which allow localization of the search for arcs of a maximal spanning tree in a clique graph by specifying the number of arcs that must be chosen for the tree in each area of the graph.

The term *Clique* (capital C) will be used to refer to a Clique of nodes in clique graph G_C to distinguish it as a special type of *clique* (small c) of nodes in an undirected graph. Also, the symbol $\hat{C}_i$ will represent the ith Clique in a clique graph G_C, while C_i will usually be the ith clique in an original graph OG.

Lemma 2.7 *The intersection of any set of cliques is a* CCS *(Completely Connected Set).*

Proof. The intersection of any set of cliques $\{C_1, C_2, \ldots, C_n\}$ is a subset of, for example, C_1, and any subset of a clique is a CCS. $\square$

Theorem 2.8 *Given any* CCS *in* G_C*, all clique nodes in* G_C *that contain all arc labels of the* CCS *are members of some* G_C *Clique that contains the* CCS.

Proof. Suppose A is a CCS in G_C. Any clique node C_i in G_C that contains all labels of A is connected to each clique node of A since the intersection of C_i with each of these cliques is nonempty. For example, if $C_j \in A$, then C_iC_j contains at least the union of the labels on arcs in A that are adjacent to C_j.

Also, any two clique nodes in G_C that contain all arc labels in A are connected to each other, since their intersection is at least the union of all arc labels in A. These interconnections imply the theorem. $\square$

Corollary 2.9 *Given a* CCS *in* G_C *in the intersection of some set of Cliques* $\{\hat{C}_i, \ldots, \hat{C}_j\}$ *in* G_C*, any node in* G_C *that contains all arc labels in the* CCS *is also in that intersection.*

Proof. Suppose A is a CCS contained in the intersection of $\{\hat{C}_i, \ldots, \hat{C}_j\}$, and that clique C_a contains the label set of A. Since A is contained in each of $\hat{C}_i, \ldots, \hat{C}_j$, we have that C_a is also contained in each of $\hat{C}_i \ldots \hat{C}_j$ by Theorem 2.8. Thus, C_a is in the intersection of $\{\hat{C}_i, \ldots, \hat{C}_j\}$. $\square$

Theorem 2.10 *The arc labels of any chord-free cycle in* G_C *of length greater than or equal to four are pairwise disjoint.*

Proof. Suppose the theorem is false. Then for some chord-free cycle $C_i, \ldots, C_j$, we must be able to find two arcs C_aC_b and C_cC_d, with $a \neq c, d$ and $b \neq c, d$, such that there exists some $x \in C_aC_b$ with $x \in C_cC_d$ also. (See Figure 2.8.) But this implies the existence in G_C of arcs C_aC_c, C_aC_d, C_bC_c, and C_bC_d — at least one of which must be a chord of $C_i, \ldots, C_j$. This is a contradiction. □

The results presented thus far in this chapter apply to all undirected graphs and clique graphs. The theorems that follow apply specifically to triangulated graphs and their corresponding clique graph representations.

Theorem 2.11 *In any cycle in triangulated graph OG, each pair of nodes joined by an arc of the cycle is in some clique with at least one other node of the cycle.*

Proof. Suppose that $x_1, \ldots, x_n$ is a cycle in OG, and that x_j, x_{j+1} are adjacent nodes of this cycle. If chord x_jx_{j+2} is present, then x_j, x_{j+1}, and x_{j+2} are together in some clique since they form a *CCS*. (See Figure 2.9(a).)

If x_jx_{j+2} is not present, let $x_{j+2}, x_{1'}, x_{2'}, \ldots, x_{m'}, x_j$ be the shortest path among nodes of the cycle $x_1, \ldots, x_n$, but not through x_{j+1}, that connects x_j and x_{j+2}. (See Figure 2.9(b).) Because OG is triangulated but arc x_jx_{j+2} is not present, arcs $x_{j+1}x_{1'}, x_{j+1}x_{2'}, \ldots, x_{j+1}x_{m'}$ must be present. Specifically, arcs $x_jx_{j+1}, x_jx_{m'}$, and $x_{j+1}x_{m'}$ are all present, and thus x_j, x_{j+1}, and $x_{m'}$ must be together in some clique. □

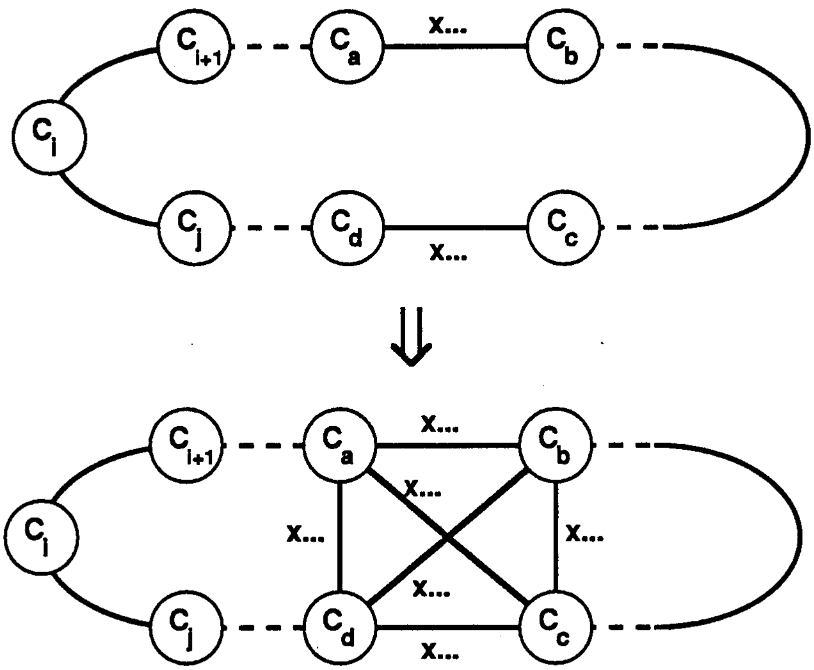


Figure 2.8: If the labels along a cycle in a clique graph are not pairwise disjoint, then the cycle is not chord-free.

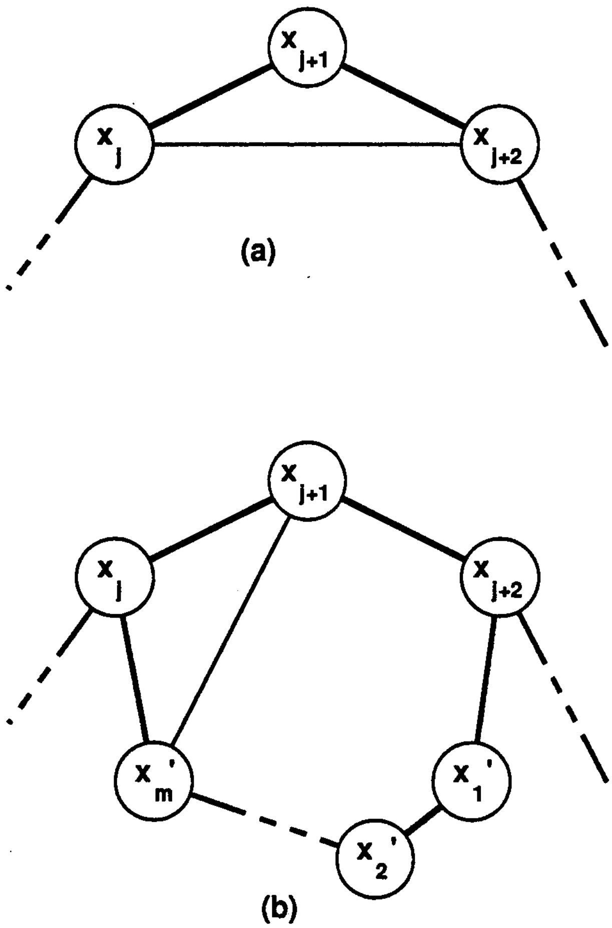


Figure 2.9: Cycle nodes x_j and x_{j+1} must be in a clique with some other cycle node.

Theorem 2.12 *If in a cycle $C_i, C_{i+1}, \ldots, C_j$ in the clique graph G_C of a triangulated graph OG, no side is contained in the union of the other sides, then no arc of the cycle will be included in a maximal spanning tree of G_C.*

Proof. Given the cycle $C_i, C_{i+1}, \ldots, C_j$ in G_C, with no side contained in the union of the other sides, we must be able to find

$$\begin{aligned} x_i &\in C_iC_{i+1} \\ x_{i+1} &\in C_{i+1}C_{i+2} \\ &\vdots \\ x_j &\in C_jC_i \end{aligned}$$

such that each x appears in no other side of the cycle in G_C. Note that we then also have the cycle $x_i, x_{i+1}, \ldots, x_j$ in OG. (See Figure 2.10.)

By way of contradiction, suppose that C_iC_{i+1} is an arc in some maximal spanning tree of G_C. By Theorem 2.11, there must be some clique $C_{a'}$ that contains at least x_i, x_j, and x_a, for some x_a also in the cycle $x_i, x_{i+1}, \ldots, x_j$ in OG. Without loss of generality, let x_a be the lowest numbered such node. Clique $C_{a'}$ with its lowest possible weight arcs to cliques C_i and C_a (also in the cycle in G_C) is shown in Figure 2.11.

Thus we have the cycle $x_i, x_{i+1}, \ldots, x_a$ in OG, and again by Theorem 2.11, there must exist some clique $C_{b'}$ which contains x_i, x_a, and x_b, for some $b < a$, where again we choose x_b to be the lowest numbered such node on the cycle $x_i, \ldots, x_j$ (Figure 2.12).

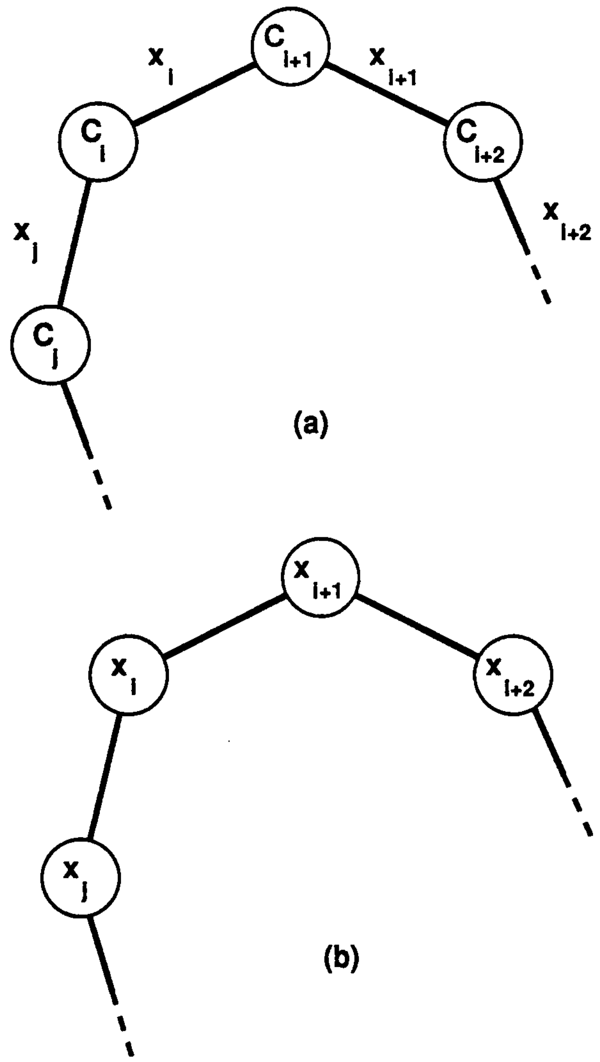


Figure 2.10: The cycle in G_C (a) implies the cycle in OG (b).

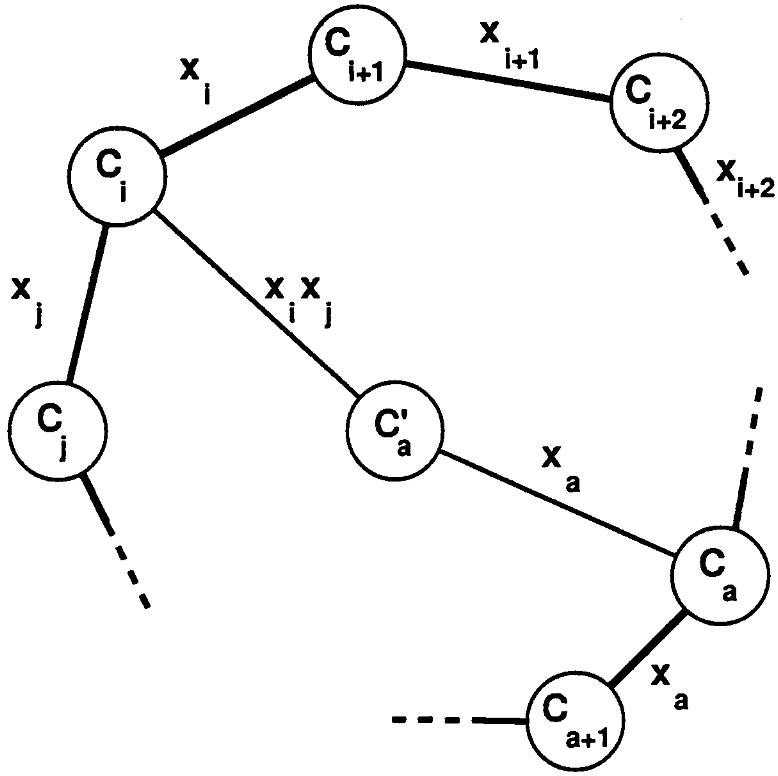


Figure 2.11: Clique $C_{a'}$ contains x_i, x_j, and x_a.

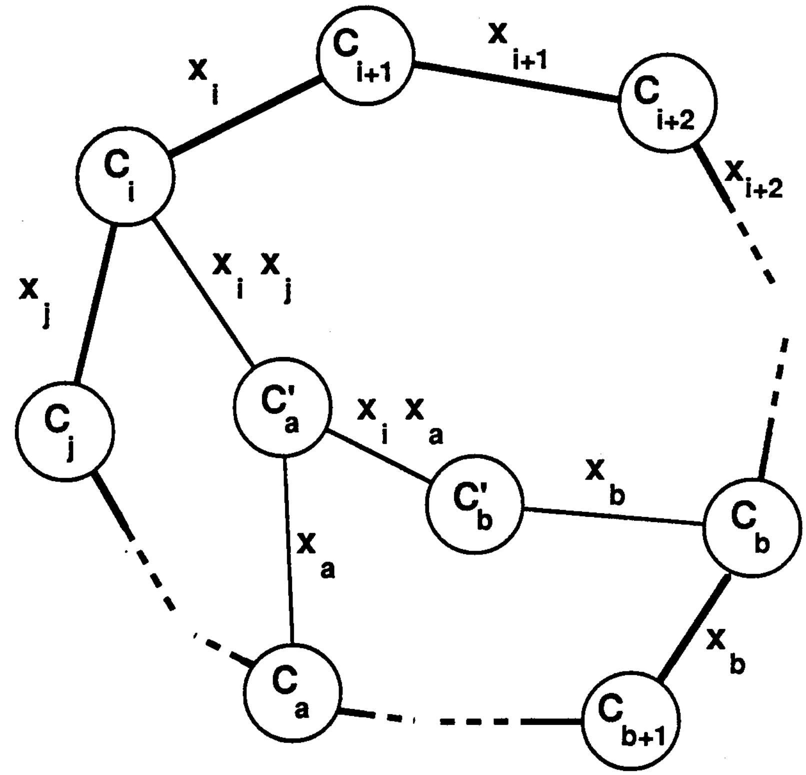


Figure 2.12: Clique $C_{b'}$ contains x_i, x_a, and x_b.

Continuing in this manner, we eventually arrive at some clique $C_{m'}$ that contains x_i, x_{i+1}, and x_m, for some node x_m on the cycle $x_i, \ldots, x_j$ as shown in Figure 2.13.

Since each x selected is found in only one side of the cycle $C_i, C_{i+1}, \ldots, C_j$ in G_C, each of the arcs along the path $C_i, C_{a'}, C_{b'}, \ldots, C_{m'}, C_{i+1}$ contains some x not contained in C_iC_{i+1}. Furthermore, since x_i could be any element of C_iC_{i+1}, and every such x_i is contained in each of the cliques $C_{a'}, C_{b'}, \ldots, C_{m'}$, we have that C_iC_{i+1} is also contained in each arc of the path $C_i, C_{a'}, C_{b'}, \ldots, C_{m'}, C_{i+1}$. Thus, the weight of each arc in this path is strictly greater than the weight of C_iC_{i+1}, and C_iC_{i+1} cannot appear in any maximal spanning tree of G_C. □

Theorem 2.13 *No arc of a chord-free cycle of length greater than or equal to four in the clique graph G_C of a triangulated graph is included in a maximal spanning tree of G_C.*

Proof. By Theorem 2.10, the arc labels along a chord-free cycle are pairwise disjoint, and thus no side is contained in the union of the other sides of the cycle. None of these arcs can be included in a maximal spanning tree of G_C by Theorem 2.12. □

Theorem 2.14 describes a property of cliques and their intersections that is used in the definition of clique trees for which the *running intersection property* holds. (A formal definition for the running intersection property is given in Chapter 4.) This result shows the equivalence of maximal spanning trees of

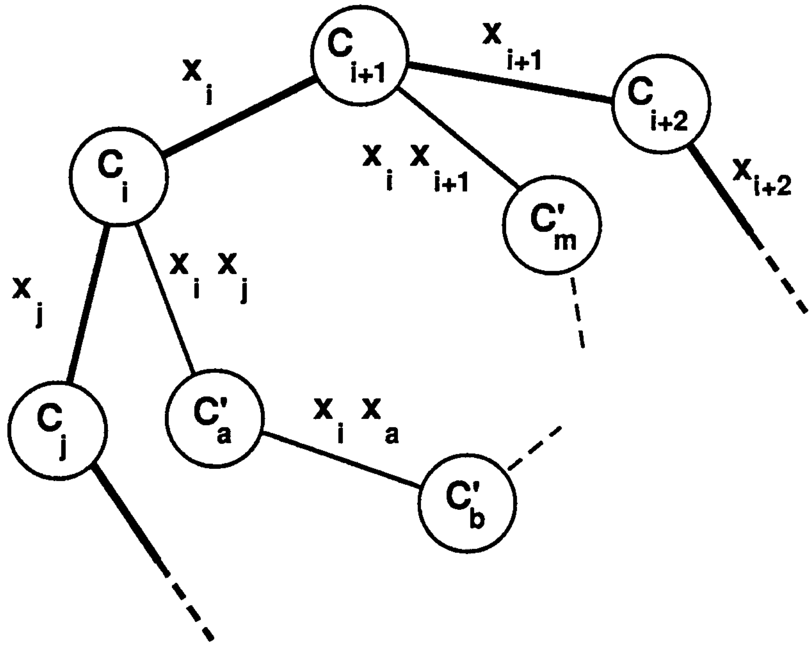


Figure 2.13: Clique $C_{m'}$ contains x_i, x_{i+1}, and x_m.

clique graphs and clique trees that have the running intersection property. An alternative derivation of this equivalence is given in [29].

Theorem 2.14 *If C_aC_b is an arc in the clique graph G_C of a triangulated graph, then every arc in a maximal spanning tree path between clique nodes C_a and C_b contains arc C_aC_b.*

Proof. (By induction)

Base step: Path of one arc:

The maximal spanning tree path from C_a to C_b is exactly the arc C_aC_b and the theorem is trivially true.

Inductive step: Path of several arcs:

Assume the theorem is true for paths of length less than m, and there exists some maximal spanning tree path

$$C_a, C_1, C_2, \ldots, C_{m-1}, C_b$$

of length m. Note that since arc C_aC_b exists in G_C, $C_a, C_1, C_2, \ldots, C_{m-1}, C_b$ is a cycle in G_C.

Case 1: There exists some C_q in the cycle such that $C_aC_b \subseteq C_q$. (See Figure 2.14.)

By the inductive hypothesis, all arcs in the maximal spanning tree path $C_a, C_1, \ldots, C_q$ contain C_aC_q, which contains C_aC_b. Similarly, the arcs in the path $C_q, C_{q+1}, \ldots, C_b$ each contain C_qC_b, which also contains C_aC_b. Thus, each arc in the path $C_a, \ldots, C_q, \ldots, C_b$ contains C_aC_b.

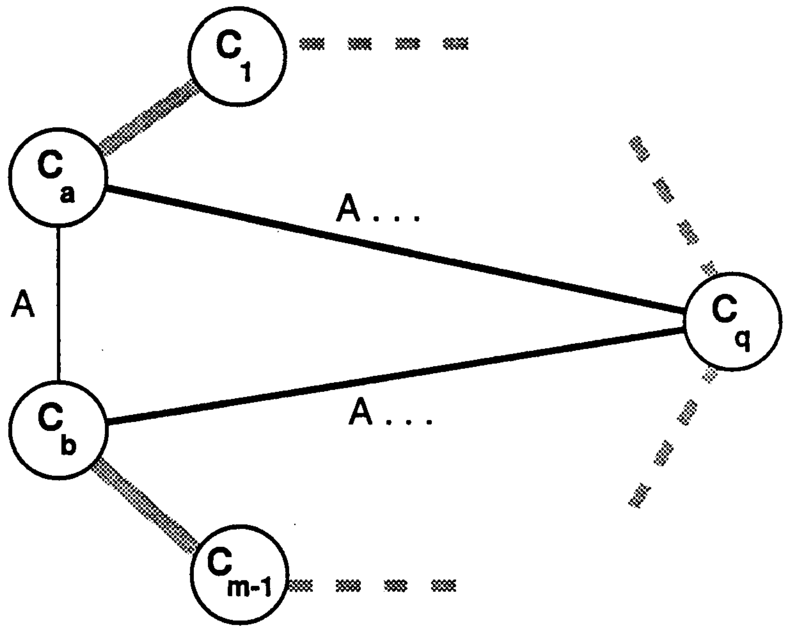


Figure 2.14: Arc C_aC_b is contained in some clique C_q in the maximal spanning tree path.

Case 2: No clique in the cycle contains C_aC_b. (This will lead to a contradiction.)

Suppose that no arc of the cycle $C_a, C_1, C_2, \ldots, C_{m-1}, C_b$ is contained in another arc of the cycle. Then by Theorem 2.12, no arc of the cycle may appear in a maximal spanning tree of G_C, and we have a contradiction.

Suppose then that some side C_jC_{j+1} of the cycle is contained in some other side C_qC_{q+1}, as shown in Figure 2.15. Then we must have $C_jC_{q+1} \supseteq C_jC_{j+1}$ which implies $C_jC_{q+1} = C_jC_{j+1}$; otherwise arc C_jC_{q+1} would have been chosen instead of C_jC_{j+1} in the construction of a maximal spanning tree of G_C.

Since arcs C_jC_{q+1} and C_jC_{j+1} are equal and in the same cycle, we may substitute one for the other in a maximal spanning tree of G_C and still have a maximal spanning tree. (See Figure 2.16.) Substituting C_jC_{q+1} for C_jC_{j+1} in the current tree produces a maximal spanning tree path

$$C_a, C_1, \ldots, C_j, C_{q+1}, \ldots, C_{m-1}, C_b$$

which has length less than m, and thus by the inductive hypothesis, contains C_aC_b in each of its arcs, including C_jC_{q+1}.

Also by the inductive hypothesis, we must have C_jC_{q+1} contained in each of the arcs $C_jC_{j+1}, C_{j+1}C_{j+2}, \ldots, C_qC_{q+1}$. Thus, by transitivity, C_aC_b is contained in each arc of the original maximum spanning tree path $C_a, C_1, \ldots, C_j, C_{j+1}, \ldots, C_q, C_{q+1}, \ldots, C_{m-1}, C_b$, which is a contradiction. □

Corollary 2.15 makes a statement equivalent to Theorem 2.14 but for nodes along a path rather than arcs. This is for convenience in later proofs.

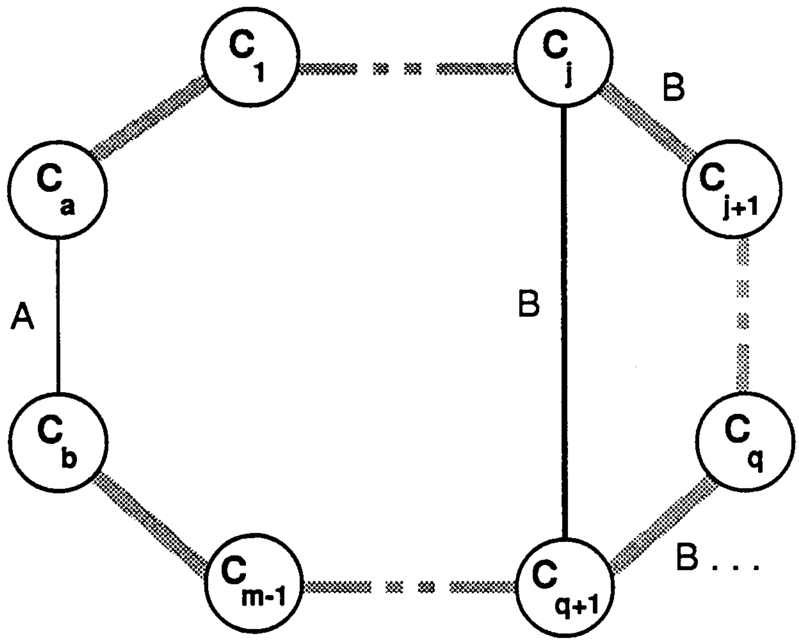


Figure 2.15: Arc C_jC_{j+1} is contained in C_qC_{q+1} in the cycle.

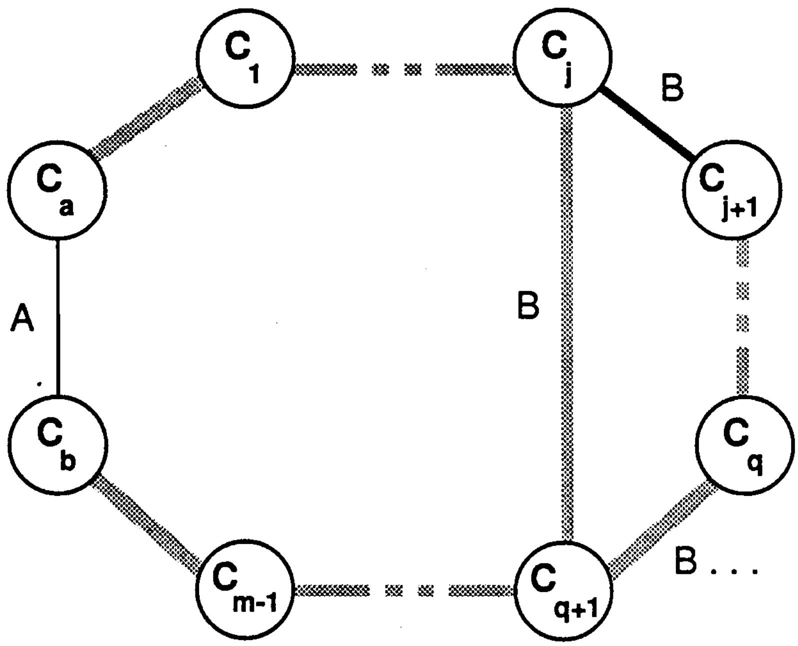


Figure 2.16: Arc C_jC_{q+1} is substituted for C_jC_{j+1} in a maximal spanning tree.

Corollary 2.15 *If C_aC_b is an arc in the clique graph G_C of a triangulated graph, then every node on any maximal spanning tree path between clique nodes C_a and C_b contains arc C_aC_b.*

Proof. Each node in the path contains some arc of the path, which contains C_aC_b by Theorem 2.14. □

The following theorems deal with Cliques of nodes in a clique graph and their relationship to maximal spanning trees of the clique graph.

Theorem 2.16 *If C_aC_b is an arc in the clique graph G_C of a triangulated graph, then every node on any maximal spanning tree path between clique nodes C_a and C_b is contained in some G_C Clique that also contains clique nodes C_a and C_b.*

Proof. Cliques C_a and C_b each contain C_aC_b, as does each clique along any maximal spanning tree path in G_C, by Corollary 2.15. Thus, all of these nodes are in some G_C Clique that contains C_a and C_b, by Theorem 2.8. □

Theorem 2.17 *Given any G_C Clique of order n, any maximal spanning tree of G_C must include $n-1$ arcs from that Clique.*

Proof. Note that a tree of n nodes has $n-1$ arcs. Thus, if more than $n-1$ arcs are chosen from a Clique of order n, then there is some cycle among these arcs within the Clique, which is not legal in a tree.

If fewer than $n-1$ arcs are chosen in the Clique, then some node of the Clique must be connected to the other nodes in the Clique by some maximal spanning tree path outside of the Clique, which is not possible according to Theorem 2.16. Thus, exactly $n-1$ arcs must be chosen. □

Example. Suppose we have a clique graph G_C with nodes that represent the cliques

$$Z = \{C_1, C_2, C_3, C_4, C_5, C_6\}$$

from some original graph, and the arcs of G_C form the Cliques

$$\begin{aligned}\hat{C}_1 &= \{C_1, C_2, C_3, C_4, C_5\},\\ \hat{C}_2 &= \{C_1, C_2, C_3, C_6\},\end{aligned}$$

as shown in Figure 2.17.

The shaded arcs in Figure 2.17(a) could represent a maximal spanning tree of G_C depending on the contents of the cliques in Z. The shaded arcs in Figure 2.17(b), however, cannot be a maximal spanning tree for any clique set Z since by Theorem 2.17, any maximal spanning tree for this graph must include exactly four arcs from $\hat{C}_1$, a clique of order five, and the spanning tree in Figure 2.17(b) uses only three. □

A major result of this chapter, Theorem 2.17 implies that the search for arcs of a maximal spanning tree for a clique graph G_C can be carried out in parallel by conducting searches for smaller maximal spanning trees of $n-1$ arcs in each Clique in G_C of order n. One application for this is that it allows localization in the search for clique separators during the initialization procedure of the Lauritzen - Spiegelhalter method for probability propagation described in Chapter 4.

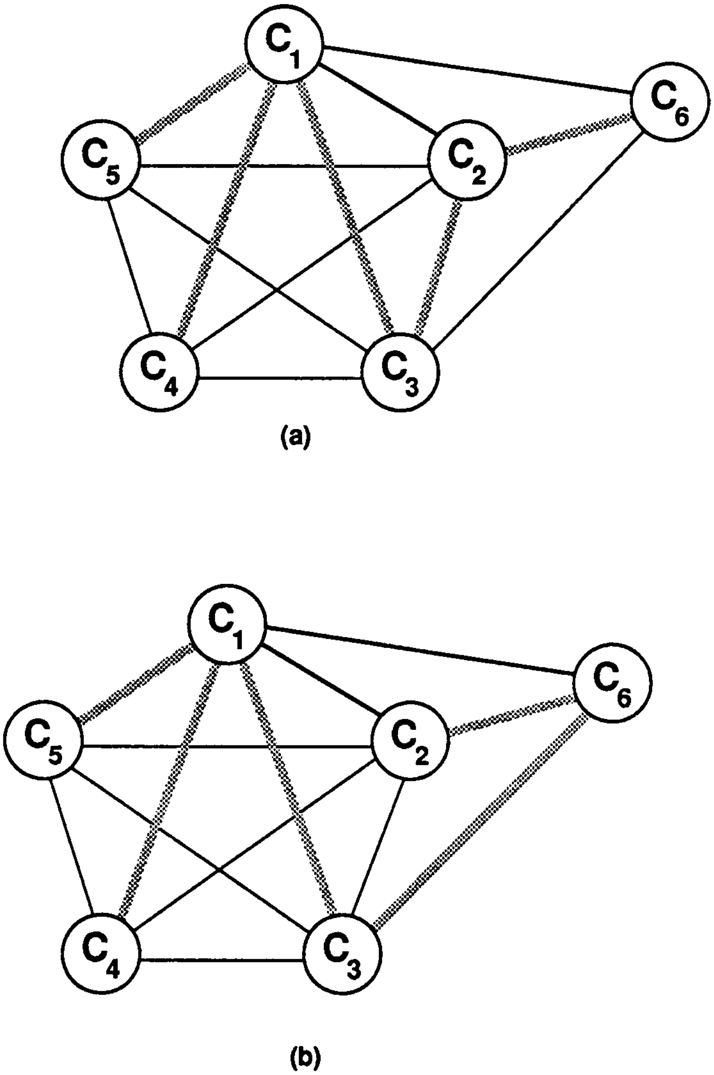


Figure 2.17: The shaded arcs in (a) form a valid maximal spanning tree; the shaded arcs in (b) do not.

CHAPTER 3

EXCLUSIVE INTERSECTION SETS AND CLIQUE TEMPLATES

3.1 Introduction

An exclusive intersection ($\mathcal{XI}$) over a set of cliques is the set of nodes found only in the intersection of those cliques. $\mathcal{XI}$s form partitions of cliques and clique intersections.

Combinations of templates constructed from $\mathcal{XI}$ components may be used to store a precise characterization of a clique graph, allowing fast and local updating of a clique graph as changes in structure are made to the corresponding original graph.

3.2 $\mathcal{XI}$ Components

Definition 3.1 *Given the set $Z = \{C_1, C_2, \ldots, C_n\}$ of cliques in a graph OG, and some subset $\mathcal{A} \subseteq Z$, the* **exclusive intersection** *(written $\mathcal{XI}$) over $\mathcal{A}$ is*

$$\mathcal{XI}(C_i \in \mathcal{A}) = \left[\bigcap_{C_i \in \mathcal{A}} C_i\right] \bigcap \left[\bigcap_{C_j \in (Z-\mathcal{A})} \overline{C}_j\right].$$

The set $\mathcal{A}$ is called the **realm** *of the $\mathcal{XI}$.*

Example. If $Z = \{C_1, C_2, C_3, C_4\}$, and capital letters are used to label $\mathcal{XI}$ sets, then the list of all possible $\mathcal{XI}$s in Z is as follows:

$$
\begin{aligned}
A &= C_1 \cap C_2 \cap C_3 \cap C_4 = \mathcal{XI}(C_1, C_2, C_3, C_4) \\
B &= C_1 \cap C_2 \cap C_3 \cap \overline{C}_4 = \mathcal{XI}(C_1, C_2, C_3) \\
C &= C_1 \cap C_2 \cap \overline{C}_3 \cap C_4 = \mathcal{XI}(C_1, C_2, C_4) \\
D &= C_1 \cap C_2 \cap \overline{C}_3 \cap \overline{C}_4 = \mathcal{XI}(C_1, C_2) \\
E &= C_1 \cap \overline{C}_2 \cap C_3 \cap C_4 = \mathcal{XI}(C_1, C_3, C_4) \\
F &= C_1 \cap \overline{C}_2 \cap C_3 \cap \overline{C}_4 = \mathcal{XI}(C_1, C_3) \\
G &= C_1 \cap \overline{C}_2 \cap \overline{C}_3 \cap C_4 = \mathcal{XI}(C_1, C_4) \\
H &= C_1 \cap \overline{C}_2 \cap \overline{C}_3 \cap \overline{C}_4 = \mathcal{XI}(C_1) \\
I &= \overline{C}_1 \cap C_2 \cap C_3 \cap C_4 = \mathcal{XI}(C_2, C_3, C_4) \\
J &= \overline{C}_1 \cap C_2 \cap C_3 \cap \overline{C}_4 = \mathcal{XI}(C_2, C_3) \\
K &= \overline{C}_1 \cap C_2 \cap \overline{C}_3 \cap C_4 = \mathcal{XI}(C_2, C_4) \\
L &= \overline{C}_1 \cap C_2 \cap \overline{C}_3 \cap \overline{C}_4 = \mathcal{XI}(C_2) \\
M &= \overline{C}_1 \cap \overline{C}_2 \cap C_3 \cap C_4 = \mathcal{XI}(C_3, C_4) \\
N &= \overline{C}_1 \cap \overline{C}_2 \cap C_3 \cap \overline{C}_4 = \mathcal{XI}(C_3) \\
O &= \overline{C}_1 \cap \overline{C}_2 \cap \overline{C}_3 \cap C_4 = \mathcal{XI}(C_4) \\
& \overline{C}_1 \cap \overline{C}_2 \cap \overline{C}_3 \cap \overline{C}_4 = \mathcal{XI}(\emptyset)
\end{aligned}
$$

□

Figure 3.1 is given for reference and shows how $\mathcal{XI}$ components will be represented throughout the remainder of this paper; illustrations composed of clique nodes and $\mathcal{XI}$ components will be called *clique component diagrams.* (It will be shown later that it is not possible for all of the exclusive intersection sets in the above clique component diagram to be nonempty simultaneously.) The $\mathcal{XI}$ set E, for example, is shown as a triangle with sides that connect cliques C_1, C_3, and C_4 — retracing precisely those edges in the clique graph of Z that would include the elements of E in their labels.

Components such as E which have a realm of order greater than one are called *multiclique $\mathcal{XI}$ components.* Combining all elements of the multiclique $\mathcal{XI}$ components that are *adjacent* to any pair of cliques produces the complete clique graph label for the edge connecting those two cliques. The justification for this is given in Theorem 3.1. In the example above, for instance, the label of C_1C_2 would be the concatenation of the elements of exclusive intersection sets A, B, C, and D. The $\mathcal{XI}$ components H, L, N, and O are each exclusive intersections over a single clique, and thus their elements would appear on no arc labels in the clique graph for Z. For consistency, we will still say that the $\mathcal{XI}$ over a single clique is *adjacent* to its corresponding clique node in a clique component diagram.

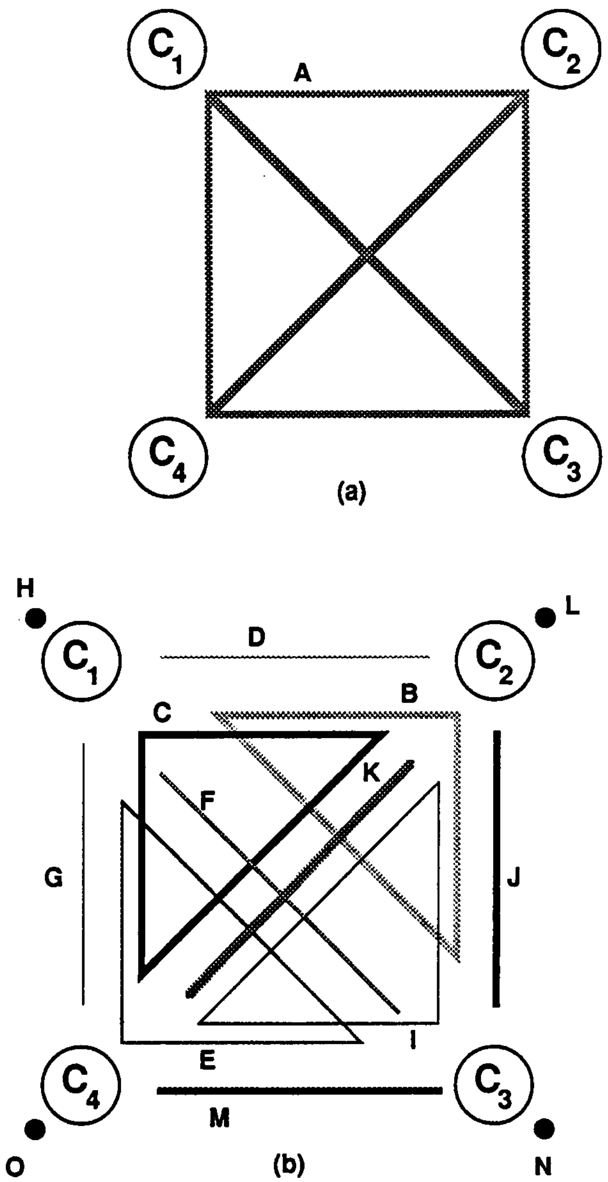


Figure 3.1: Clique components corresponding to all possible exclusive intersection sets in a graph with four cliques: (a) shows the $\mathcal{XI}$ set with realm order four, (b) shows all components with lower order realms.

Theorem 3.1 *Given the set Z of cliques in OG and some subset $A \subseteq Z$, then the intersection of all cliques in A is equal to the union of all $\mathcal{XI}$s in Z whose realms contain A:*

$$\bigcap_{C_i \in A} C_i = \bigcup_{\mathcal{A} \subseteq Z, A \subseteq \mathcal{A}} [\mathcal{XI}(C_j \in \mathcal{A})] .$$

Proof. Let

$$A = \{C_{a_1}, C_{a_2}, \ldots, C_{a_n}\} ,$$

and

$$D = Z - A = \{C_{d_1}, C_{d_2}, \ldots, C_{d_m}\} .$$

Then

$$\begin{aligned}
\bigcap_{C_i \in A} C_i &= C_{a_1} \cap C_{a_2} \cap \cdots \cap C_{a_n} \\
&= [C_{a_1} \cap C_{a_2} \cap \cdots \cap C_{a_n}] \cap Z \\
&= [C_{a_1} \cap C_{a_2} \cap \cdots \cap C_{a_n}] \cap \left[C_{d_1} \cup \overline{C}_{d_1}\right] \\
&= [C_{a_1} \cap \cdots \cap C_{a_n} \cap C_{d_1}] \cup \left[C_{a_1} \cap \cdots \cap C_{a_n} \cap \overline{C}_{d_1}\right] \\
&= \left[[C_{a_1} \cap \cdots \cap C_{a_n} \cap C_{d_1}] \cup \left[C_{a_1} \cap \cdots \cap C_{a_n} \cap \overline{C}_{d_1}\right]\right] \cap \\
&\quad \left[C_{d_2} \cup \overline{C}_{d_2}\right] \\
&= [C_{a_1} \cap \cdots \cap C_{a_n} \cap C_{d_1} \cap C_{d_2}] \cup \\
&\quad \left[C_{a_1} \cap \cdots \cap C_{a_n} \cap \overline{C}_{d_1} \cap C_{d_2}\right] \cup \\
&\quad \left[C_{a_1} \cap \cdots \cap C_{a_n} \cap C_{d_1} \cap \overline{C}_{d_2}\right] \cup \\
&\quad \left[C_{a_1} \cap \cdots \cap C_{a_n} \cap \overline{C}_{d_1} \cap \overline{C}_{d_2}\right] \\
&\quad \vdots
\end{aligned}$$

$$= [C_{a_1} \cap \cdots \cap C_{a_n} \cap C_{d_1} \cap \cdots \cap C_{d_m}] \cup \cdots \cup$$

$$\left[C_{a_1} \cap \cdots \cap C_{a_n} \cap \overline{C}_{d_1} \cap \cdots \cap \overline{C}_{d_m}\right]$$

$$= \bigcup_{\mathcal{A} \subseteq Z, A \subseteq \mathcal{A}} [\mathcal{XI}(C_j \in \mathcal{A})].$$

□

Theorem 3.1 tells how to construct the intersection of a set of cliques from $\mathcal{XI}$ components as was done in the previous example for the label of arc C_1C_2. Theorem 3.2 states further that these $\mathcal{XI}$ components do not overlap; thus, in the example given earlier, we need not be concerned about repeated elements when forming a clique graph arc label by concatenating the elements of the connecting $\mathcal{XI}$ component sets.

Theorem 3.2 *Given the set Z of cliques in OG and any subset $A \subseteq Z$, then the set of all $\mathcal{XI}$s in Z whose realms contain A forms a partition of the nodes in OG that are in the intersection of all cliques in A.*

Proof. By Theorem 3.1, the union of all $\mathcal{XI}$s of the subsets of Z that contain A is equal to the intersection over all cliques in A. To show that these $\mathcal{XI}$s form a partition of the intersection set, we need only show that they are pairwise disjoint.

Suppose we have $a \in \mathcal{XI}(C_i \in \mathcal{A}_j)$ and $b \in \mathcal{XI}(C_i \in \mathcal{A}_k)$, where $\mathcal{A}_j, \mathcal{A}_k \subseteq Z; A \subseteq \mathcal{A}_j, \mathcal{A}_k$; and $\mathcal{A}_j \neq \mathcal{A}_k$. Then $\mathcal{A}_j \neq \mathcal{A}_k$ implies that there exists some $C_p \in \mathcal{A}_j$ such that $C_p \notin \mathcal{A}_k$, or conversely. In the first case, $a \in C_p$ and $b \in \overline{C}_p$, thus $a \neq b$. The second case is similar. □

Using special cases of clique set intersections, it can now be shown that a single clique in OG may be viewed as the union of a set of its $\mathcal{XI}$ components. Also, the set of all $\mathcal{XI}$ components of a graph forms a jigsaw puzzle-like representation of the graph, in which separate pieces are groupings of nodes that are similar in their relationship to the rest of the graph.

Corollary 3.3 *Given the set Z of cliques in OG, and some clique C_k in Z, then the set of all $\mathcal{XI}$s in Z whose realms contain C_k forms a partition of the nodes in C_k.*

Proof. The proof follows directly from Theorem 3.2, letting the set $A = \{C_k\}$.

□

Example. Note that the set of $\mathcal{XI}$s in Z whose realms contain a clique C_i appear as the components adjacent to node C_i in a clique component diagram.

Given the clique set

$$Z = \{C_1, C_2, C_3, C_4\}$$

with the clique component diagram shown in Figure 3.2, each clique in Z may be expressed as the union of partition set elements as follows:

$$\begin{aligned} C_1 &= B \cup C \cup D \cup H, \\ C_2 &= B \cup C \cup D \cup I \cup K, \\ C_3 &= B \cup I \cup N, \\ C_4 &= C \cup I \cup K \cup O. \end{aligned}$$

□

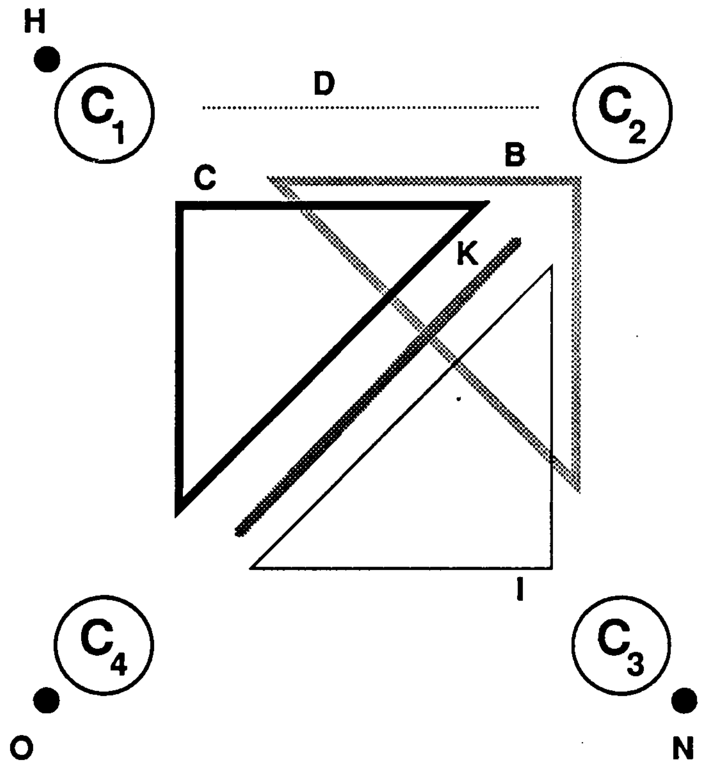


Figure 3.2: Each clique may be expressed as the union of its adjacent components.

Corollary 3.4 *Given the set Z of cliques in OG, then the set of all $\mathcal{XI}$s over subsets of Z forms a partition of the nodes in OG.*

Proof. In Theorem 3.2, let $A = Z$. □

Theorem 3.5 and its corollary show that the set of all $\mathcal{XI}$s that involve some clique C_k is unique to C_k. Thus in a clique component diagram, no two clique nodes may be adjacent to exactly the same set of $\mathcal{XI}$ components.

Theorem 3.5 *Given any clique C_k in OG, no other clique C_j, $j \neq k$, is in the realm of every nonempty $\mathcal{XI}$ that contains C_k.*

Proof. Suppose that such a clique C_j exists. Then by definition of exclusive intersection, $[x \in C_k] \Rightarrow [x \in C_j]$, and thus C_k is not a maximal clique — a contradiction. □

Corollary 3.6 *The intersection over all $\mathcal{XI}$ realms that contain clique C_k is exactly C_k.*

Proof. The intersection set of the $\mathcal{XI}$ realms considered contains C_k by assumption. No other clique C_j may also appear in this intersection by Theorem 3.5. Thus, the intersection set is equal to $\{C_k\}$. □

The principles set forth in the previous set of results can be used to construct a representative original graph OG from a clique component diagram as follows:

1. Put a node x_Y in OG for each distinct $\mathcal{XI}$ component Y in the clique component diagram,

2. Draw an arc between each pair of nodes x_Y and x_W in OG if and only if the realms of $\mathcal{XI}$ components Y and W have a nonempty intersection.

As an example, the realm of each $\mathcal{XI}$ component in the diagram shown in Figure 3.2 is given in Table 3.1 along with the the set of other components with intersecting realms. Using this information, we may then construct the representative original graph shown in Figure 3.3. Since only one node has been placed in OG for each $\mathcal{XI}$ component of the clique component diagram and no sizes were specified for these components, the graph in Figure 3.3 is only one of the possible original graphs that could be constructed to correspond to the diagram in Figure 3.2. It will be shown in Section 3.4 that the number of elements in each nonempty $\mathcal{XI}$ component of a clique component diagram has no effect on the clique structure of the corresponding original graph, and thus each clique component diagram with $\mathcal{XI}$ components of unspecified size represents an entire class of original graphs.

Theorem 3.7 *Given a clique C_k in OG, if every $\mathcal{XI}$ realm of order greater than or equal to two that contains C_k also contains some clique C_j, $j \neq k$, then $\mathcal{XI}(C_k) \neq \emptyset$.*

Proof. Note that $\mathcal{XI}(C_k)$ is the only $\mathcal{XI}$ with a realm of order less than two that contains C_k. By Theorem 3.5, there must be some (nonempty) $\mathcal{XI}$ whose realm contains C_k but not C_j. The assumptions leave $\mathcal{XI}(C_k)$ as the only possible such $\mathcal{XI}$, thus it must be that $\mathcal{XI}(C_k) \neq \emptyset$. □

Component	*Realm*	*Adjacent Components*
B	$\{C_1, C_2, C_3\}$	C, D, H, I, K, N
C	$\{C_1, C_2, C_4\}$	B, D, H, I, K, O
D	$\{C_1, C_2\}$	B, C, H, I, K
H	$\{C_1\}$	B, C, D
I	$\{C_2, C_3, C_4\}$	B, C, D, K, N, O
K	$\{C_2, C_4\}$	B, C, D, I, O
N	$\{C_3\}$	B, I
O	$\{C_4\}$	C, I, K

Table 3.1: Clique component adjacencies.

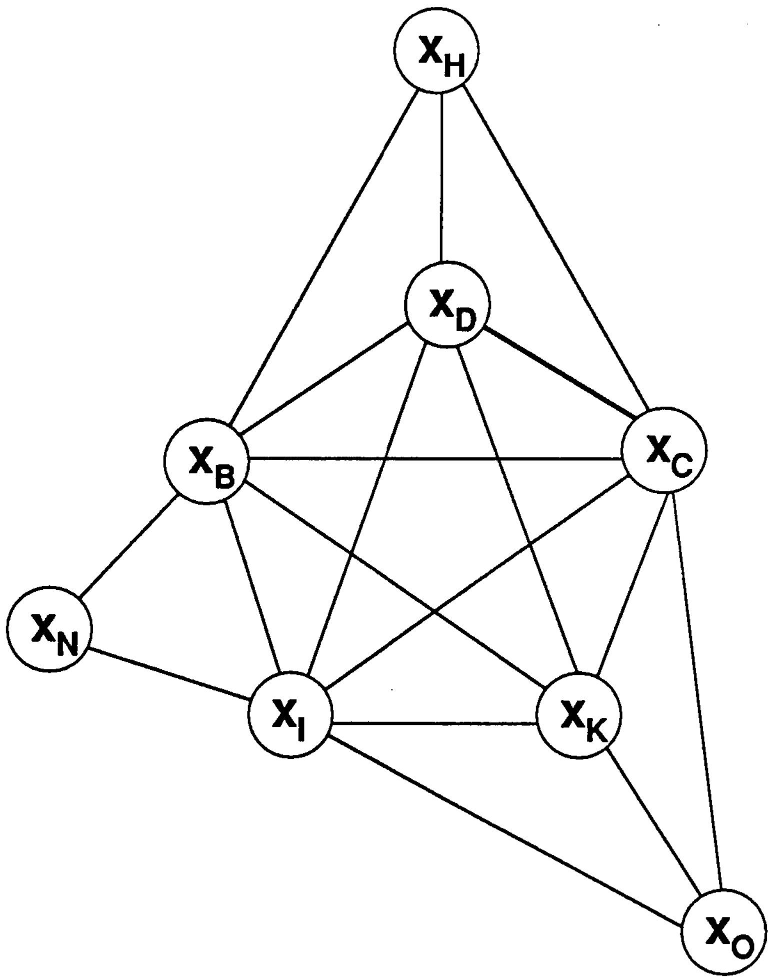


Figure 3.3: An original graph constructed from a clique component diagram.

Example. If the components shown in Figure 3.4 represent the only nonempty multiclique components of some valid clique component diagram, then the exclusive intersection set of each individual clique except for C_4 must be nonempty. □

Theorem 3.7 specifies when an $\mathcal{XI}$ component of a clique must be nonempty: the exclusive intersection over a single clique must be nonempty if that clique would otherwise share all of its $\mathcal{XI}$ components with another clique. This means that in a clique component diagram, the $\mathcal{XI}$ of a single clique C_i must be nonempty if every multiclique $\mathcal{XI}$ component adjacent to C_i is also adjacent to some other clique C_j. [1] Conversely, it is not necessary that a clique own any nodes exclusively if it shares only a proper subset of its nodes with any other single clique. The theorem thus provides a rule which may be used to test an arbitrary clique graph or clique component diagram for validity with no further knowledge of the OG on which these representations are based. Rules of this type could be useful in the development of algorithms based on clique graph representations by allowing testing on randomly produced clique graphs without first generating an underlying original graph.

The next theorem shows that given any set of n cliques in a clique graph, at least one of the $\mathcal{XI}$s over a subset of $n-1$ of those cliques must be empty. This is a generalization of the principle behind Theorem 2.1 which applied only to sets of three cliques.

---

[1]Note that the nodes in $\mathcal{XI}(C_i)$ are adjacent only to other nodes of C_i in the original graph. Nodes with this property are often called *simplicial* nodes in other writing.

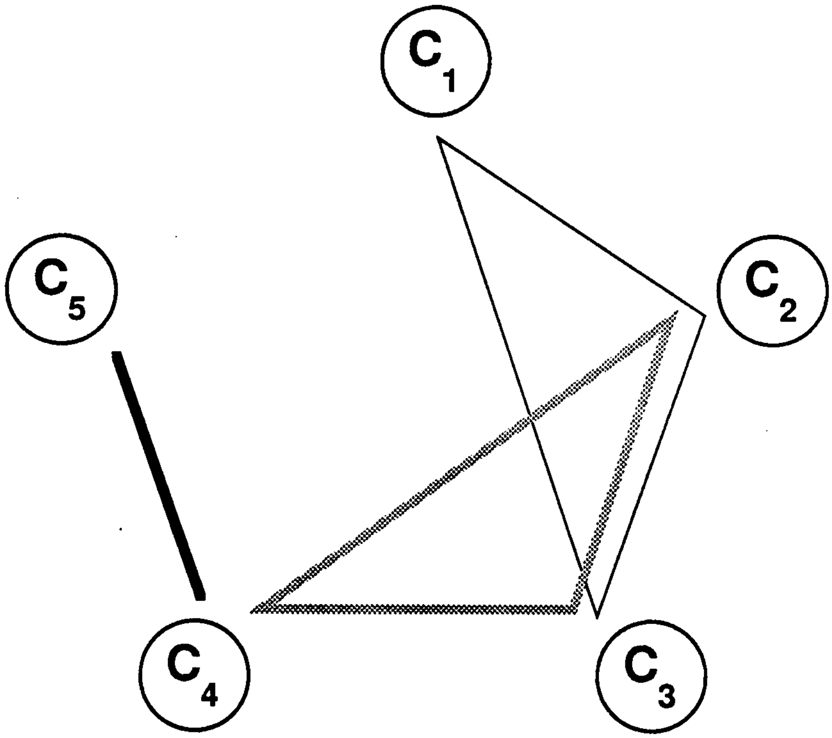


Figure 3.4: The exclusive intersection over each clique alone must be nonempty, except for C_4.

Theorem 3.8 (General Triangle Theorem) *Given a graph OG with cliques*

$$Z = \{C_1, C_2, \ldots, C_n\}, \; n \geq 3,$$

and some subset $A \subseteq Z$, *with* $|A| = m \geq 3$. *Let*

$$B = Z - A,$$

and let

$$\mathbf{A} = \{A\prime | A\prime \subset A, |A\prime| = m - 1\}.$$

Further suppose that we can find some $A_k\prime \in \mathbf{A}$ *such that for every* $A_i\prime \in \mathbf{A}, i \neq k$, *there exists some* $B\prime \subseteq B$ *such that*

$$\mathcal{XI}(C_j \in (A_i\prime \cup B_i\prime)) \neq \emptyset.$$

Then for each $A\prime\prime \subseteq A_k\prime$ *such that* $|A\prime\prime| \geq 2$, *and for any* $B\prime\prime \subseteq B - \bigcap_i B_i\prime$,

$$\mathcal{XI}(C_j \in (A\prime\prime \cup B\prime\prime)) = \emptyset.$$

Proof. Suppose

$$Z = \{C_1, C_2, \ldots, C_m, C_{m+1}, \ldots, C_n\},$$

and, without loss of generality,

$$A = \{C_1, C_2, \ldots, C_m\}.$$

Then

$$\begin{aligned} \mathbf{A} &= \{\{C_2, \ldots, C_m\}, \ldots, \{C_1, \ldots, C_{m-1}\}\} \\ &= \{A_1\prime, \ldots, A_m\prime\}, \end{aligned}$$

and

$$B = \{C_{m+1}, \ldots, C_n\}.$$

Note that $A_{i'} = A - \{C_i\}$.

Suppose now that $A_{1'} = \{C_2, \ldots, C_m\}$ fits the requirements given for $A_{k'}$ in the theorem statement. By assumption, we can find some $B_{i'} \subseteq B$ for each $A_{i'} \in \mathbf{A}, i \neq 1$, so that each of the following exists:

$$\begin{aligned}
x_2 &\in \mathcal{XI}(C_j \in (A_{2'} \cup B_{2'})), \\
x_3 &\in \mathcal{XI}(C_j \in (A_{3'} \cup B_{3'})), \\
&\vdots \\
x_m &\in \mathcal{XI}(C_j \in (A_{m'} \cup B_{m'})).
\end{aligned}$$

Now by way of contradiction, suppose the theorem is false. Then we can find some $A'' \subseteq A_{1'}, |A''| \geq 2$, say

$$A'' = \{C_r, \ldots, C_s\}, \; 2 \leq r < s \leq m,$$

and some $B'' \subseteq B - \bigcap_i B_{i'}$, say

$$B'' = \{C_l, \ldots, C_p\}, \; m+1 \leq l, p \leq n,$$

such that the following also exists:

$$x_1 \in (C_j \in (A'' \cup B'')).$$

Since $x_2, \ldots, x_m \in C_1$, all of $x_2, \ldots, x_m$ are interconnected. Observe that since $r \neq s$, we have either $C_r \neq C_2$ or $C_s \neq C_2$. Thus, either $x_1, x_2 \in C_r$ or

$x_1, x_2 \in C_s$, and x_1 is connected to x_2. By similar arguments, x_1 is connected to each of $x_3, \ldots, x_m$, and thus $\{x_1, \ldots, x_m\}$ is a completely connected set and must be contained in some $C_t \in Z$.

Since $x_i \notin C_i$ for $1 \leq i \leq m$, we have $C_t \notin \{C_1, \ldots, C_m\} = A$. (Note here that if $B = \emptyset$, then $Z = A$, C_t does not exist, and we are finished.)

Suppose $C_t \in B$. Then

$$\begin{aligned} x_2 \in C_t &\Rightarrow C_t \in B_2\prime, \\ x_3 \in C_t &\Rightarrow C_t \in B_3\prime, \\ &\vdots \\ x_m \in C_t &\Rightarrow C_t \in B_m\prime, \end{aligned}$$

and

$$C_t \in \bigcap_{i=2,\ldots,m} B_i\prime.$$

But $[x_1 \in C_t] \Rightarrow [C_t \in B\prime\prime]$, and $B\prime\prime \subseteq B - \bigcap_i B_i\prime$, so $C_t \notin B$.

We now have that $C_t \notin A + B = Z$, so C_t does not exist — a contradiction.

$\square$

Example. As a small example, suppose we have a partial specification of G_C as shown in Figure 3.5, with cliques

$$Z = \{C_1, C_2, C_3\},$$

and we take A to be the only subset of Z of order greater than or equal to 3, which is the entire set of cliques in G_C:

$$A = \{C_1, C_2, C_3\}.$$

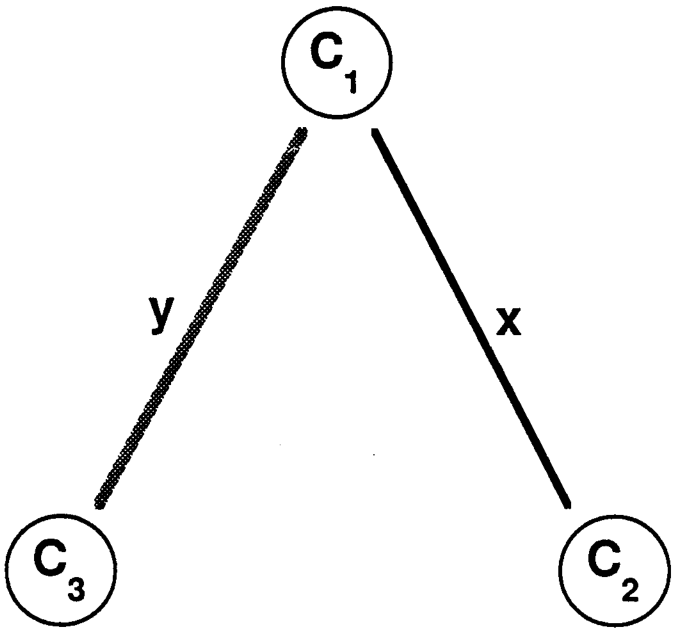


Figure 3.5: A clique component diagram with two components; the third $\mathcal{XI}$ with realm order two must be empty.

Then we have

$$B = Z - A = \emptyset.$$

The subsets of A of order $m - 1 = 2$ are

$$\mathbf{A} = \{\{C_1, C_2\}, \{C_1, C_3\}, \{C_2, C_3\}\}.$$

We must now try to find some element of $\mathbf{A}$ such that each other element of $\mathbf{A}$, combined with some subset of B, produces a set of cliques whose exclusive intersection is nonempty. Examination of G_C reveals that

$$A_{3'} = \{C_2, C_3\}$$

is such an element, and we also have

$$A_{1'} = \{C_1, C_2\}, \; A_{2'} = \{C_1, C_3\}$$

with $B_{i'} = \emptyset$ for all i.

From Figure 3.5 we have

$$\begin{aligned} \mathcal{XI}(A_{1'} \cup B_{1'}) &= \mathcal{XI}(\{C_1, C_2\} \cup \emptyset) \\ &= \mathcal{XI}(\{C_1, C_2\}) \\ &\ni x \\ &\neq \emptyset, \end{aligned}$$

and

$$\begin{aligned} \mathcal{XI}(A_{2'} \cup B_{2'}) &= \mathcal{XI}(\{C_1, C_3\} \cup \emptyset) \\ &= \mathcal{XI}(\{C_1, C_3\}) \end{aligned}$$

$$\ni \; y$$

$$\neq \; \emptyset.$$

The only subset of $A_{3'}$ of order greater than or equal to 2 is

$$A'' = \{C_2, C_3\},$$

and since

$$B'' \subseteq \left(B - \bigcap_i B_{i'}\right) = \emptyset,$$

it must be that

$$B'' = \emptyset.$$

Therefore, according to Theorem 3.8,

$$\begin{aligned} \mathcal{XI}(\{C_2, C_3\} \cup \emptyset) &= \mathcal{XI}(\{C_2, C_3\}) \\ &= \emptyset. \end{aligned}$$

Thus, in a clique component diagram for G_C, there can be no nonempty $\mathcal{XI}$ component adjacent only to clique nodes C_2 and C_3. □

Example. As a somewhat more elaborate example of the application of Theorem 3.8, consider a graph which contains at least the nodes in the exclusive intersection sets shown in the clique component diagram of Figure 3.6.

Here, we have

$$Z = \{C_1, C_2, C_3, C_4, C_5, C_6\},$$

and we will take A to be

$$A = \{C_1, C_2, C_3, C_6\}.$$

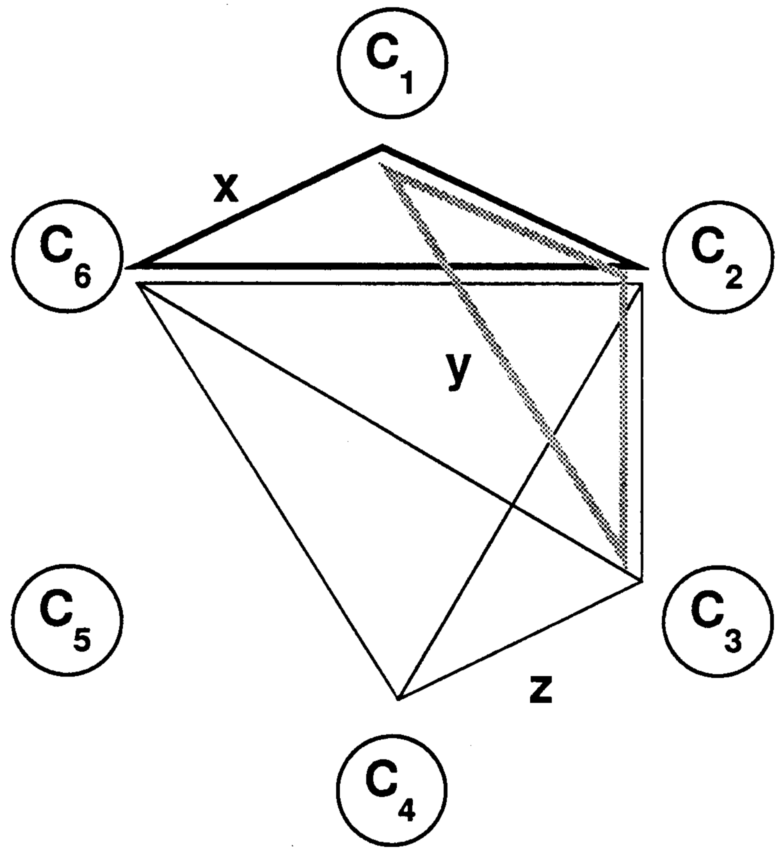


Figure 3.6: A clique component diagram with three components – their presence implies that several other components must be empty.

Thus we also have

$$B = Z - A = \{C_4, C_5\}.$$

The subsets of A of order $m - 1 = 3$ are

$$\mathbf{A} = \{\{C_1, C_2, C_3\}, \{C_1, C_2, C_6\}, \{C_1, C_3, C_6\}, \{C_2, C_3, C_6\}\},$$

and the subset $A_{k'}$ that fits the requirements of the theorem is

$$A_{3'} = \{C_1, C_3, C_6\}.$$

(In general, to find $A_{k'}$ in a clique component diagram, we look for the "missing" $\mathcal{XI}$ component of size $m - 1$ in a set of nodes of size m which has some component adjacent to every other group of $m - 1$ of its nodes.) Now we list the remaining elements of $\mathbf{A}$, each with a corresponding subset of B:

$$\begin{array}{lll} A_{1'} = \{C_1, C_2, C_3\} & A_{2'} = \{C_1, C_2, C_6\} & A_{4'} = \{C_2, C_3, C_6\} \\ \downarrow & \downarrow & \downarrow \\ B_{1'} = \emptyset & B_{2'} = \emptyset & B_{4'} = \{C_4\} \end{array}$$

and from Figure 3.6 we have

$$\begin{aligned} \mathcal{XI}(A_{1'} \cup B_{1'}) &= \mathcal{XI}(\{C_1, C_2, C_3\} \cup \emptyset) \\ &= \mathcal{XI}(\{C_1, C_2, C_3\}) \\ &\ni y \\ &\neq \emptyset, \end{aligned}$$

$$\mathcal{XI}(A_{2'} \cup B_{2'}) = \mathcal{XI}(\{C_1, C_2, C_6\} \cup \emptyset)$$

$$
\begin{aligned}
&= \mathcal{XI}(\{C_1, C_2, C_6\}) \\
&\ni x \\
&\neq \emptyset,
\end{aligned}
$$

and

$$
\begin{aligned}
\mathcal{XI}(A_{4'} \cup B_{4'}) &= \mathcal{XI}(\{C_2, C_3, C_6\} \cup \{C_4\}) \\
&= \mathcal{XI}(\{C_2, C_3, C_4, C_6\}) \\
&\ni z \\
&\neq \emptyset.
\end{aligned}
$$

The subsets of $A_{3'}$ of order greater than or equal to 2 are

$$A_{''} \in \{\{C_1, C_3, C_6,\}, \{C_1, C_3\}, \{C_1, C_6\}, \{C_3, C_6\}\},$$

and

$$
\begin{gathered}
B_{''} \subseteq \left(B - \bigcap_i B_{i'}\right) \Rightarrow \\
B_{''} \in \{\{C_4, C_5\}, \{C_4\}, \{C_5\}, \emptyset\}.
\end{gathered}
$$

Therefore, by Theorem 3.8, each of the following exclusive intersection sets must be empty:

$\mathcal{XI}(\{C_1, C_3, C_4, C_5, C_6\})$	$\mathcal{XI}(\{C_1, C_3, C_4, C_5\})$	$\mathcal{XI}(\{C_1, C_4, C_5, C_6\})$
$\mathcal{XI}(\{C_3, C_4, C_5, C_6\})$	$\mathcal{XI}(\{C_1, C_3, C_4, C_6\})$	$\mathcal{XI}(\{C_1, C_3, C_4\})$
$\mathcal{XI}(\{C_1, C_4, C_6\})$	$\mathcal{XI}(\{C_3, C_4, C_6\})$	$\mathcal{XI}(\{C_1, C_3, C_5, C_6\})$
$\mathcal{XI}(\{C_1, C_3, C_5\})$	$\mathcal{XI}(\{C_1, C_5, C_6\})$	$\mathcal{XI}(\{C_3, C_5, C_6\})$
$\mathcal{XI}(\{C_1, C_3, C_6\})$	$\mathcal{XI}(\{C_1, C_3\})$	$\mathcal{XI}(\{C_1, C_6\})$
$\mathcal{XI}(\{C_3, C_6\})$		

□

The previous two examples demonstrate that Theorem 3.8 is a powerful tool that allows the use of partial knowledge of a clique graph to make inferences about parts of the graph for which there is no empirical information. Results covered thus far in this chapter apply to the $\mathcal{XI}$ components of any graph. In the next section, additional results for exclusive intersections in triangulated graphs are presented.

3.3 $\mathcal{XI}$ Components in Triangulated Graphs

Theorem 3.9 shows that any $\mathcal{XI}$ component whose presence would complete a chord-free cycle of $\mathcal{XI}$ components in the clique component diagram of a triangulated graph must be empty.

Theorem 3.9 (Triangulated Graph Theorem) *Given a triangulated graph OG with cliques*

$$Z = \{C_1, C_2, \ldots, C_n\},$$

and some set of nonempty exclusive intersections

$$\mathcal{XI}(A_1), \mathcal{XI}(A_2), \ldots, \mathcal{XI}(A_m),$$

where each

$$\begin{aligned} A_i &\subset Z, \\ A_i \cap A_{i+1} &\neq \emptyset, \ \textit{for } 1 \le i \le m-1, \end{aligned}$$

$$A_i \cap A_{i+b} = \emptyset, \textit{ for } 1 \le i \le m, |b| < 1, \textit{ and}$$

$$A_i \not\subseteq A_j, \textit{ for } i \ne j.$$

Let

$$B = \{A_1, \ldots, A_k\},$$

$$D = \{A_{k+1}, \ldots, A_m\}$$

for some $1 \le k \le m-1$, *and let*

$$\mathbf{B} = \bigcup_{A_i \in B} A_i,$$

$$\mathbf{D} = \bigcup_{A_i \in D} A_i.$$

If we choose any

$$\mathbf{D}\prime \subseteq (\mathbf{D} - \mathbf{B}), \ \mathbf{D}\prime \ne \emptyset,$$

$$\mathbf{B}\prime \subseteq (\mathbf{B} - \mathbf{D}), \ \mathbf{B}\prime \ne \emptyset,$$

and

$$Z\prime \subseteq Z - (\mathbf{B} \cup \mathbf{D}),$$

and let $\mathbf{X} = \mathbf{D}\prime \cup \mathbf{B}\prime \cup Z\prime$, *then*

$$\mathcal{XI}(\mathbf{X}) = \emptyset.$$

Proof. Suppose the theorem is false. Then, given the assumptions, we can find some $x_X \in \mathcal{XI}(\mathbf{X})$.

Also by assumption, we can find

$$\begin{aligned} x_1 &\in \mathcal{XI}(A_1) \\ x_2 &\in \mathcal{XI}(A_2) \\ &\vdots \\ x_{m-1} &\in \mathcal{XI}(A_{m-1}) \\ x_m &\in \mathcal{XI}(A_m), \end{aligned}$$

and cliques

$$\begin{aligned} C_{\bar{1}} &\in A_1 \cap A_2 \\ C_{\bar{2}} &\in A_2 \cap A_3 \\ &\vdots \\ C_{\overline{m-1}} &\in A_{m-1} \cap A_m. \end{aligned}$$

Now,

$$\begin{aligned} x_1 \in \mathcal{XI}(A_1) &\Rightarrow x_1 \in C_{\bar{1}} \\ x_2 \in \mathcal{XI}(A_2) &\Rightarrow x_2 \in C_{\bar{1}},\ x_2 \in C_{\bar{2}} \\ &\vdots \\ x_{m-1} \in \mathcal{XI}(A_{m-1}) &\Rightarrow x_{m-1} \in C_{\overline{m-2}},\ x_{m-2} \in C_{\overline{m-1}} \\ x_m \in \mathcal{XI}(A_m) &\Rightarrow x_m \in C_{\overline{m-1}}. \end{aligned}$$

Thus, we have arcs $x_1x_2, x_2x_3, \ldots, x_{m-1}x_m$ in E, but no other arcs among $x_1, \ldots, x_m$ since $x_ix_{i+b} \in E$, $|b| > 1$, implies $x_i, x_{i+b} \in C_d$ for some clique $C_d \in A_i \cap A_{i+b}$, but by assumption, $A_i \cap A_{i+b} = \emptyset$ for $|b| > 1$.

Now let $A_H \in B$ be the highest numbered realm in B containing some clique in $\mathbf{B}\prime$, say C_H, and let $A_L \in D$ be the lowest numbered realm in D containing some clique in $\mathbf{D}\prime$, say C_L. Then

$$\begin{aligned} x_X \in \mathcal{XI}(\mathbf{X}) &\Rightarrow x_X \in \mathcal{XI}(\mathbf{D}\prime \cup \mathbf{B}\prime \cup Z\prime) \\ &\Rightarrow x_X \in C_H,\ x_X \in C_L. \end{aligned}$$

Thus, in OG, we have the cycle

$$x_X, x_H, x_{H+1}, \ldots, x_{L-1}, x_L,$$

where $x_H, x_{H+1}, \ldots, x_L$ is a subsequence of $x_1, x_2, \ldots, x_m$.

Since OG is triangulated and there are no other arcs among $x_H, \ldots, x_L$, we must have an arc $x_X x_i$ for each $x_i \in \{x_{H+1}, \ldots, x_{L-1}\}$. Specifically, the arcs $x_X x_k$, $x_k x_{k+1}$, and $x_{k+1} x_X$ all exist. (Recall that $B = \{A_1, \ldots, A_k\}, D = \{A_{k+1}, \ldots, A_m\}$.)

Thus, x_X, x_k, and x_{k+1} are all contained in some clique C_s, which is contained in each of $\mathbf{X}, \mathbf{B},$ and $\mathbf{D}$; that is,

$$C_s \in (\mathbf{X} \cap \mathbf{B} \cap \mathbf{D}).$$

But this intersection is empty since $\mathbf{X}$ contains only cliques from $\mathbf{B}-\mathbf{D}$, $\mathbf{D}-\mathbf{B}$, or $Z-(\mathbf{B} \cup \mathbf{D})$. Thus we have a contradiction, and the theorem must be true. $\square$

Example. For an example of Theorem 3.9, we begin with a triangulated graph with cliques

$$Z = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\}$$

for which there are nonempty exclusive intersections over at least each element of the set

$$\begin{aligned}\mathbf{A} &= \{\{C_1, C_2, C_3\}, \{C_2, C_3, C_4, C_5\}, \\ &\quad \{C_4, C_5, C_6\}, \{C_6, C_7\}\}.\end{aligned}$$

(See Figure 3.7.)

We then divide set $\mathbf{A}$ into the sets

$$B = \{\{C_1, C_2, C_3\}, \{C_2, C_3, C_4, C_5\}\}$$

and

$$D = \{\{C_4, C_5, C_6\}, \{C_6, C_7\}\},$$

and form the composite sets

$$\begin{aligned}\mathbf{B} &= \bigcup_{A_i \in B} A_i \\ &= \{C_1, C_2, C_3, C_4, C_5\}\end{aligned}$$

and

$$\begin{aligned}\mathbf{D} &= \bigcup_{A_i \in D} A_i \\ &= \{C_4, C_5, C_6, C_7\}.\end{aligned}$$

From this we can get

$$\mathbf{B} - \mathbf{D} = \{C_1, C_2, C_3\}$$

and

$$\mathbf{D} - \mathbf{B} = \{C_6, C_7\},$$

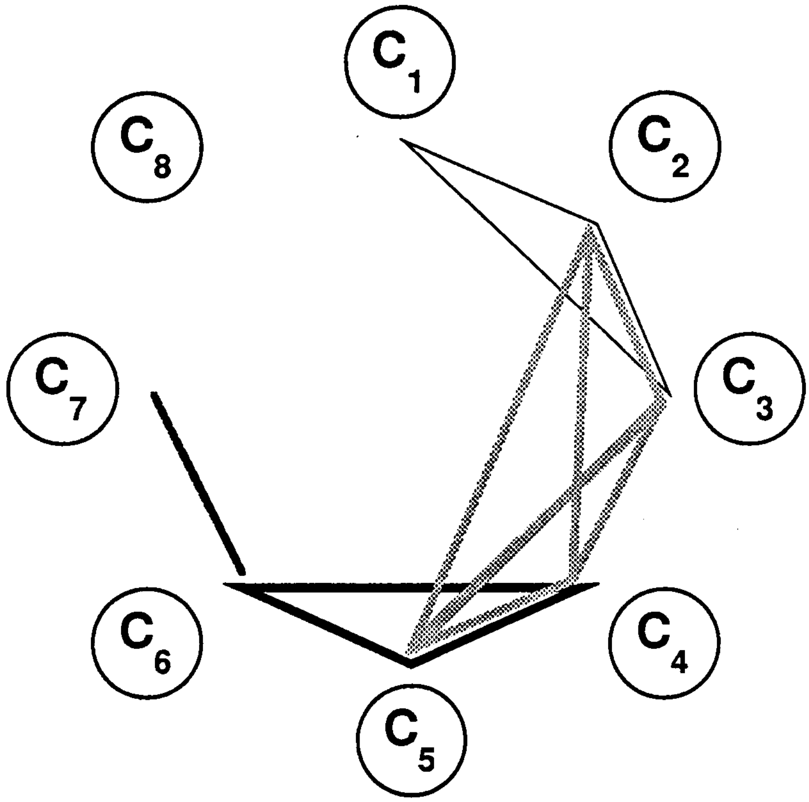


Figure 3.7: A clique component diagram with four components.

and thus we have that

$$\mathbf{B}\prime \in \{\{C_1, C_2, C_3\}, \{C_1, C_2\}, \{C_1, C_3\},$$
$$\{C_2, C_3\}, \{C_1\}, \{C_2\}, \{C_3\}\},$$

$$\mathbf{D}\prime \in \{\{C_6, C_7\}, \{C_6\}, \{C_7\}\},$$

and since the theorem states that we may also choose any $Z\prime \subseteq Z - (\mathbf{B} \cup \mathbf{D})$,

$$Z\prime \in \{\{C_8\}, \emptyset\}.$$

Theorem 3.9 says that if we now choose any $\mathbf{B}\prime$, $\mathbf{D}\prime$, and $Z\prime$ from the sets above and construct the set $\mathbf{X} = \mathbf{D}\prime \cup \mathbf{B}\prime \cup Z\prime$, then $\mathcal{XI}(\mathbf{X})$ must be empty. In this example, there are 7 choices for $\mathbf{B}\prime$, 3 choices for $\mathbf{D}\prime$, and 2 choices for $Z\prime$, so each of the following $7 \times 3 \times 2 = 42$ exclusive intersection sets must be empty:

$\mathcal{XI}(\{C_1, C_2, C_3, C_6, C_7, C_8\})$	$\mathcal{XI}(\{C_1, C_2, C_3, C_6, C_8\})$	$\mathcal{XI}(\{C_1, C_2, C_3, C_7, C_8\})$
$\mathcal{XI}(\{C_1, C_2, C_6, C_7, C_8\})$	$\mathcal{XI}(\{C_1, C_2, C_6, C_8\})$	$\mathcal{XI}(\{C_1, C_2, C_7, C_8\})$
$\mathcal{XI}(\{C_1, C_3, C_6, C_7, C_8\})$	$\mathcal{XI}(\{C_1, C_3, C_6, C_8\})$	$\mathcal{XI}(\{C_1, C_3, C_7, C_8\})$
$\mathcal{XI}(\{C_2, C_3, C_6, C_7, C_8\})$	$\mathcal{XI}(\{C_2, C_3, C_6, C_8\})$	$\mathcal{XI}(\{C_2, C_3, C_7, C_8\})$
$\mathcal{XI}(\{C_1, C_6, C_7, C_8\})$	$\mathcal{XI}(\{C_1, C_6, C_8\})$	$\mathcal{XI}(\{C_1, C_7, C_8\})$
$\mathcal{XI}(\{C_2, C_6, C_7, C_8\})$	$\mathcal{XI}(\{C_2, C_6, C_8\})$	$\mathcal{XI}(\{C_2, C_7, C_8\})$
$\mathcal{XI}(\{C_3, C_6, C_7, C_8\})$	$\mathcal{XI}(\{C_3, C_6, C_8\})$	$\mathcal{XI}(\{C_3, C_7, C_8\})$
$\mathcal{XI}(\{C_1, C_2, C_3, C_6, C_7\})$	$\mathcal{XI}(\{C_1, C_2, C_3, C_6\})$	$\mathcal{XI}(\{C_1, C_2, C_3, C_7\})$
$\mathcal{XI}(\{C_1, C_2, C_6, C_7\})$	$\mathcal{XI}(\{C_1, C_2, C_6\})$	$\mathcal{XI}(\{C_1, C_2, C_7\})$
$\mathcal{XI}(\{C_1, C_3, C_6, C_7\})$	$\mathcal{XI}(\{C_1, C_3, C_6\})$	$\mathcal{XI}(\{C_1, C_3, C_7\})$
$\mathcal{XI}(\{C_2, C_3, C_6, C_7\})$	$\mathcal{XI}(\{C_2, C_3, C_6\})$	$\mathcal{XI}(\{C_2, C_3, C_7\})$
$\mathcal{XI}(\{C_1, C_6, C_7\})$	$\mathcal{XI}(\{C_1, C_6\})$	$\mathcal{XI}(\{C_1, C_7\})$
$\mathcal{XI}(\{C_2, C_6, C_7\})$	$\mathcal{XI}(\{C_2, C_6\})$	$\mathcal{XI}(\{C_2, C_7\})$
$\mathcal{XI}(\{C_3, C_6, C_7\})$	$\mathcal{XI}(\{C_3, C_6\})$	$\mathcal{XI}(\{C_3, C_7\})$

$\square$

Note that Theorem 3.9 applies to triangulated graphs and is more powerful than Theorem 3.8 in its ability to specify empty $\mathcal{XI}$ components of a clique component diagram. For instance, the set $\mathcal{XI}(\{C_1, C_7\})$ in the example above must be empty by Theorem 3.9 but would not be caught by Theorem 3.8; moreover, if it were not specified that the graph used in the example for Theorem 3.9 be triangulated, then it would not necessarily be the case that $\mathcal{XI}(\{C_1, C_7\})$ is empty.

Intuitively, Theorem 3.8 works by locating possible inconsistencies in a clique graph that arise when sets of nodes in the corresponding original graph are erroneously labeled as cliques when they are not in fact maximal. Theorem 3.9 works by not allowing the representation in a clique graph of a configuration that would imply a chord-free cycle of length four or greater in a corresponding original graph.

3.4 Allowable $\mathcal{XI}$ Components and Clique Templates

Thus far, the focus of this chapter has been on exclusive intersection components of clique graphs or clique component diagrams which must be either necessarily empty or necessarily nonempty. However, if we are given some set of nonempty $\mathcal{XI}$ components in an arbitrary and possibly incomplete clique component diagram, it will not always be the case that we can use these results to decide deterministically whether or not each other possible $\mathcal{XI}$ set of the diagram is empty. If the emptiness of some component is not required for the validity of a clique component diagram given some knowledge of other components in the diagram, then that component is called an *allowable* $\mathcal{XI}$ component. Furthermore, since $\mathcal{XI}$ components form a partition of the nodes in the original graph, the addition of an allowable $\mathcal{XI}$ component to a clique component diagram does not in any way alter the components already present in the diagram.

Note here that if it is already known that a particular $\mathcal{XI}$ component is nonempty, then that component is clearly allowable. Therefore, in the proofs of the following theorems, we will usually assume that the component under consideration is empty, then show that nodes may be added to the original graph which meet the requirements for membership in the $\mathcal{XI}$ component.

The first theorem states simply that it is always possible for a clique to own some nodes exclusively.

Theorem 3.10 *For any clique C, $\mathcal{XI}(C)$ is allowable.*

Proof. (By construction.)

Assume that $\mathcal{XI}(C)$ is empty. We must now show that a node x_z can be added to OG as an element of $\mathcal{XI}(C)$.

Recall that to be an element of an $\mathcal{XI}$ component, a node must be contained in the intersection of all of the cliques in the realm of the component, but cannot be contained in any other clique.

Let us hypothetically add a new node x_z to OG in which C is a clique, and construct exactly the new arcs $x_z x_i$ in OG for each $x_i \in C$. Since x_z is now connected to each node in C, x_z is an element of C, by definition of a clique.

Suppose that x_z is also in some other clique $C\prime$ in OG. This implies that an arc $x_z x_j$ is present in OG for every $x_j \in C\prime$. But this implies that $C\prime \subseteq C$ since the only arcs in OG involving x_z are of the form $x_z x_i$, with $x_i \in C$. Thus, $C\prime$ is not a clique, and x_z belongs to clique C alone; that is, $x_z \in \mathcal{XI}(C)$. $\square$

The next theorem states that if a set of cliques H has a nonempty exclusive intersection, then all of the nodes in the intersection of any other clique with the cliques in H must be contained in a single clique in H. As later results show, this theorem is very useful for proving additional properties of exclusive intersections.

Theorem 3.11 *If $\mathcal{XI}(H) \neq \emptyset$ for some realm $H \subseteq Z$, then there exists no clique $C\prime \notin H$ such that*

$$C\prime \cap \left[\bigcup_{C_j \in H} C_j\right] = P \neq \emptyset,$$

where $P \not\subseteq C_j$ for any $C_j \in H$.

Proof. Suppose there exists such a clique $C\prime$ and that $x_H \in \mathcal{XI}(H)$. Then arc $x_H x_i$ exists for each $x_i \in C_j$ for each $C_j \in H$. Note that P is a subset of clique $C\prime$, so P is a completely connected set. Furthermore, $P \subseteq \left[\bigcup_{C_j \in H} C_j\right]$ so for each $x_k \in P$, we must have $x_k \in C_j$ for some $C_j \in H$. Thus, arc $x_H x_k$ exists for each $x_k \in P$ and we have that $P \cup \{x_H\}$ is also a completely connected set, so $P \cup \{x_H\}$ must be contained in some clique in Z.

But $x_H \in C_j$ implies $C_j \in H$, and by assumption $P \not\subseteq C_j$ for any $C_j \in H$. Thus the completely connected set $P \cup \{x_H\}$ cannot be contained in any clique, which is a contradiction. □

Theorem 3.12 shows that an allowable exclusive intersection set can be made arbitrarily large by installing new nodes with the proper arc connections in OG. When nodes are added to a nonempty exclusive intersection set, such alterations have no effect on the clique component diagram of OG, and can affect only the labels of existing arcs in the corresponding clique graph — no structural changes in either representation are necessary. Conversely, no structural changes in clique graphs or clique component diagrams can occur as nodes are removed from OG unless the removal of some node causes an exclusive intersection set to become empty.

Theorem 3.12 *Given that $\mathcal{XI}(H)$ is allowable for some realm $H \subseteq Z$, where Z is the set of cliques in a graph OG, if a new node x_z is added to OG along with exactly the arcs $x_z x_i$ for each $x_i \in C_j$ for each $C_j \in H$, then $x_z \in \mathcal{XI}(H)$.*

Proof. By assumption $\mathcal{XI}(H)$ is allowable, so we may also assume that $\mathcal{XI}(H) \neq \emptyset$ for purposes of this argument.

Suppose x_z is added to OG as suggested by the theorem. We must show that $x_z \in \mathcal{XI}(H)$. Since x_z is connected to each node $x_i \in C_j$ for each $C_j \in H$, then $x_z \in C_j$ for each $C_j \in H$ by definition of a clique, and thus

$$x_z \in \bigcap_{C_j \in H} C_j.$$

If we can show that x_z is in no other clique, then we are finished.

Suppose that $x_z \in C\prime$ for some clique $C\prime \notin H$. Since x_z is connected only to other nodes in $\bigcup_{C_j \in H} C_j$ and all nodes of $C\prime$ are connected to x_z, it must be that

$$C\prime \subseteq \bigcup_{C_j \in H} C_j.$$

Thus we have

$$C\prime \cap \left[\bigcup_{C_j \in H} C_j \right] = P \neq \emptyset,$$

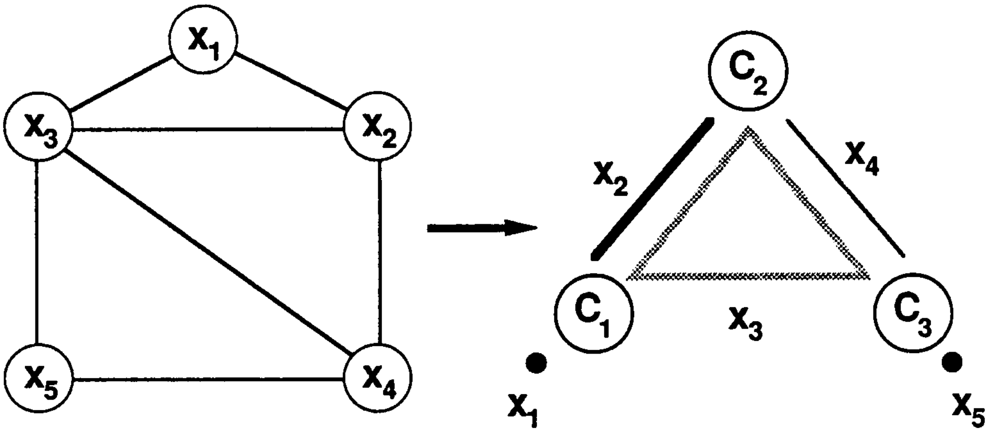
where $P = C\prime$, and $P \not\subseteq C_j$ for any $C_j \in H$, since $P = C\prime$ is a clique not in H.

But by Theorem 3.11, such a $C\prime$ cannot exist. Thus we have a contradiction, and it must be that $x_z \notin C\prime$ for any $C\prime \notin H$, so $x_z \in \mathcal{XI}(H)$. □

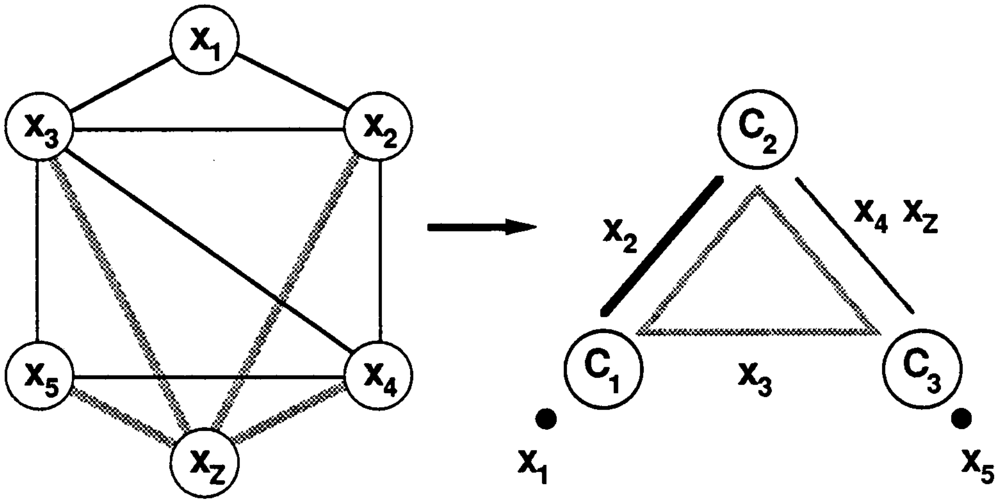
Example. Suppose $Z = \{C_1, C_2, C_3\}$, where $C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_2, x_3, x_4\}$, and $C_3 = \{x_3, x_4, x_5\}$. Then we have the original graph and clique component diagram as shown in Figure 3.8(a). To add new node x_Z to the exclusive intersection of C_2 and C_3, we add it to the original graph along with new arcs to each node in C_2 and C_3, producing the new original graph and clique component diagram shown in Figure 3.8(b). □

Note that the mapping of a labeled clique component diagram to an original graph is an isomorphism. Thus, given the order of each component in any valid clique component diagram, the method for original graph reconstruction given in Section 3.2 can be modified so that subcliques of definite size rather than single nodes are used in OG to represent each $\mathcal{XI}$ component. By doing so, an original graph can be produced which is uniquely determined by the weighted clique component diagram, in contrast to the graph produced in Section 3.2 which was a representative element of a class of graphs based on clique components of unknown size.

Theorems 3.13 and 3.14 show that the set of cliques obtained by taking either the intersection or the union of the realms of two allowable $\mathcal{XI}$ components is itself the realm of an allowable component. These theorems are then rules which may be used to guide the construction of an arbitrary clique graph or clique component diagram while maintaining its validity. As an example of a possible application for these rules, algorithms that operate on undirected



(a)



(b)

Figure 3.8: The new node x_Z is added to the exclusive intersection set of C_2 and C_3.

or triangulated graphs could be tested on a wide variety of structurally distinct graphs by employing Theorems 3.13 and 3.14 to generate and catalog graphs that follow precise compositional descriptions. The efficiency of these algorithms could then be reported for graphs with a clearly defined range of characteristics, providing a detailed projection of the performance of the algorithms in practice.

Theorem 3.13 *Given any pair of allowable $\mathcal{XI}$ components $\mathcal{XI}(A)$ and $\mathcal{XI}(B)$ such that $A \cap B \neq \emptyset$, then $\mathcal{XI}(A \cap B)$ is allowable.*

Proof. (By construction.)

Since $\mathcal{XI}(A)$ and $\mathcal{XI}(B)$ are allowable, assume that they are nonempty. If $\mathcal{XI}(A \cap B)$ is nonempty, then it is also allowable.

Suppose that $\mathcal{XI}(A \cap B)$ is empty, and hypothetically add node x_z and exactly the arcs $x_z x_i$ to OG for each $x_i \in C_j$ for each $C_j \in A \cap B$. Then $x_z \in C_j$ for each $C_j \in A \cap B$ and thus

$$x_z \in \bigcap_{C_j \in A \cap B} C_j.$$

It is then left to show that x_z is not contained in any other clique.

Suppose $x_z \in C'$ for some clique $C' \notin A \cap B$. We must show that this leads to a contradiction. Since x_z is connected only to other nodes in $\bigcup_{C_j \in A \cap B} C_j$ and all nodes of C' are connected to x_z, it must be that

$$C' \subseteq \bigcup_{C_j \in A \cap B} C_j.$$

But this implies

$$C\prime \cap \left[\bigcup_{C_j \in A} C_j \right] = P_A \neq \emptyset,$$

where $P_A = C\prime$. Since $C\prime$ is a clique, and $\mathcal{XI}(A) \neq \emptyset$ by assumption, we must have that $C\prime = P_A = C_A$ for some $C_A \in A$ by Theorem 3.11. By a similar argument, we also find that $C\prime \in B$.

From this we have $C\prime \in A \cap B$, which is a contradiction, so it must be that $x_z \notin C\prime$ for any $C\prime \notin A \cap B$, and therefore

$$x_z \in \mathcal{XI}(A \cap B).$$

□

Example. If we have the partial clique component diagram shown in Figure 3.9, with $\mathcal{XI}(\{C_1, C_2, C_3\})$ and $\mathcal{XI}(\{C_2, C_3, C_4, C_5\})$ both allowable, then $\mathcal{XI}(\{C_2, C_3\})$ must also be allowable. □

Theorem 3.13 applies to the intersection of the realms of allowable $\mathcal{XI}$ components of any graph. A similar theorem for the union of allowable realms, Theorem 3.14, applies only to triangulated graphs. The reason for this will be made clear by an example presented in the discussion following the proof of the theorem.

Theorem 3.14 *Given any pair of allowable exclusive intersection components $\mathcal{XI}(A)$ and $\mathcal{XI}(B)$ in the clique component diagram of a triangulated graph OG such that $A \cap B \neq \emptyset$, then $\mathcal{XI}(A \cup B)$ is allowable.*

Proof. (By construction.)

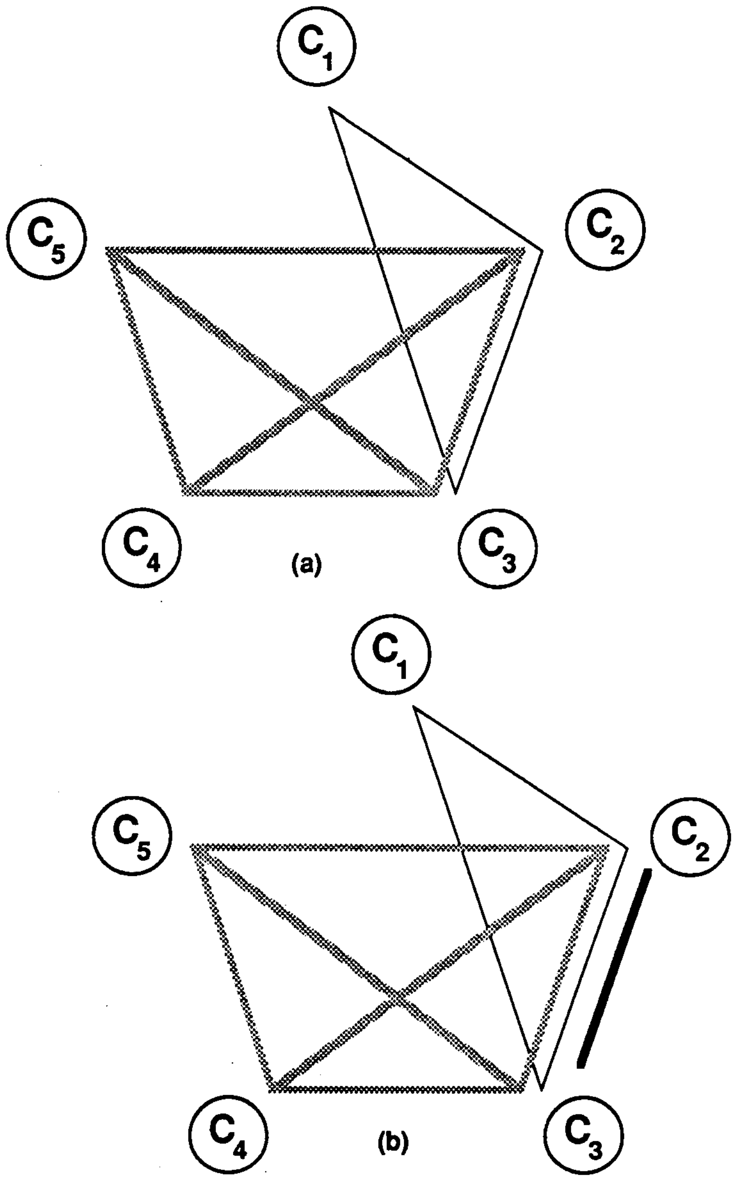


Figure 3.9: The intersection of realms of allowable components is the realm of an allowable component.

Since $\mathcal{XI}(A)$ and $\mathcal{XI}(B)$ are allowable, then so is $\mathcal{XI}(A \cap B)$ by Theorem 3.13, and thus we can assume that each of these components is nonempty. If $\mathcal{XI}(A \cup B)$ is also nonempty, then it is clearly allowable.

Suppose $\mathcal{XI}(A \cup B)$ is empty, and hypothetically add node x_z to OG along with exactly the new arcs $x_z x_i$ for each $x_i \in C_j$, for each $C_j \in A \cup B$. Then we have $x_z \in C_j$ for each $C_j \in A \cup B$ by definition of a clique, and thus

$$x_z \in \bigcap_{C_j \in A \cup B} C_j.$$

We must now show that x_z is contained in no other clique.

Suppose that $x_z \in C\prime$ for some clique $C\prime \notin A \cup B$. Since x_z is connected only to other nodes in $\bigcup_{C_j \in A \cup B} C_j$, and all nodes of $C\prime$ are connected to x_z, it must be that

$$C\prime \subseteq \bigcup_{C_j \in A \cup B} C_j.$$

This gives us that

$$\begin{aligned} C\prime &= C\prime \cap \left[\bigcup_{C_j \in A \cup B} C_j \right] \\ &= C\prime \cap \left[\left[\bigcup_{C_j \in A} C_j \right] \cup \left[\bigcup_{C_j \in B} C_j \right] \right] \\ &= \left[C\prime \cap \left[\bigcup_{C_j \in A} C_j \right] \right] \cup \left[C\prime \cap \left[\bigcup_{C_j \in B} C_j \right] \right]. \end{aligned}$$

If we set

$$P_A = C\prime \cap \left[\bigcup_{C_j \in A} C_j \right] \text{ and } P_B = C\prime \cap \left[\bigcup_{C_j \in B} C_j \right],$$

then we can write

$$C\prime = P_A \cup P_B.$$

If $P_A \subseteq P_B$, then $C\prime = P_B$. But this implies $P_B \not\subseteq C_j$ for any $C_j \in B$ since $P_B = C\prime$ is a clique not in A or B. Thus by Theorem 3.11 together with the assumption that $\mathcal{XI}(B) \neq \emptyset$, we have that $C\prime$ cannot exist. Similarly, it cannot be that $P_B \subseteq P_A$.

Thus there must exist some $x_{\overline{B}} \in P_A - P_B$ which is an element of some $C_A \in A$ but which appears in no clique in B, and some $x_{\overline{A}} \in P_B - P_A$ which is an element of some $C_B \in B$ but which appears in no clique in A. Both are elements of $C\prime$, and therefore arc $x_{\overline{A}}x_{\overline{B}}$ exists.

By assumption, there must also exist some $x_A \in \mathcal{XI}(A)$ and $x_B \in \mathcal{XI}(B)$. Since $A \cap B \neq \emptyset$, there exists some $C_k \in A \cap B$ which contains both x_A and x_B; therefore arc $x_A x_B$ exists.

Now, x_A is in every clique in A, including C_A. Therefore, x_A and $x_{\overline{B}}$ are both contained in C_A, so arc $x_A x_{\overline{B}}$ must exist. Similarly, arc $x_B x_{\overline{A}}$ exists. Thus we have the cycle $x_A x_B x_{\overline{A}} x_{\overline{B}}$ in OG. But arc $x_A x_{\overline{A}}$ cannot exist since x_A is contained only in cliques in A, and $x_{\overline{A}}$ is contained in no clique in A, and any arc in OG must appear in some clique. Similarly, $x_B x_{\overline{B}}$ cannot exist.

This means that $x_A x_B x_{\overline{A}} x_{\overline{B}}$ is a chord-free cycle of length four in OG, which contradicts the assumption that OG is triangulated. Thus there can be no clique $C\prime$ that contains x_z but is not contained in $A \cup B$, so we now have that

$$x_z \in \mathcal{XI}(A \cup B).$$

□

Example. Let us look at an example that involves several applications of Theorem 3.14. Suppose we start with the assumption that $\mathcal{XI}(\{C_1, C_2\})$, $\mathcal{XI}(\{C_1, C_3\})$, and $\mathcal{XI}(\{C_1, C_4\})$ are all allowable in some clique component diagram (see Figure 3.10(a)). Taking the union of the realms of pairs of these components, we get that $\mathcal{XI}(\{C_1, C_2, C_3\})$, $\mathcal{XI}(\{C_1, C_2, C_4\})$, and $\mathcal{XI}(\{C_2, C_3, C_4\})$ are all allowable (Figure 3.10(b)). Taking the union of the realms of all three of the original allowable components, we get that $\mathcal{XI}(\{C_1, C_2, C_3, C_4\})$ is also allowable (Figure 3.10(c)). □

If the exclusive intersection sets of each individual clique were added to this example, we would have a maximal set of $\mathcal{XI}$ components for these four cliques — any additional component would violate one or more of the theorems proved earlier in this chapter. We will call such a set of components, derived by assuming a spanning tree of n cliques of order two $\mathcal{XI}$s then filling in all other allowable components, an *n-Clique template.* Through research beyond that covered in this paper, an algorithm will be developed to generate all templates of a given size, providing a *catalog* of templates for use in the precise description and classification of a graph and to aid in the development of general high level algorithms that operate on the abstract $\mathcal{XI}$ description of the structure of a graph rather than at the level of nodes and arcs. As an example, partial 5-Clique templates based on the set of all structurally different spanning trees of order five are shown in Figure 3.11.

Note that in general, Theorem 3.14 applies only to triangulated graphs. To illustrate this, consider the nontriangulated graph with nodes $\{x_1, x_2, x_3, x_4\}$,

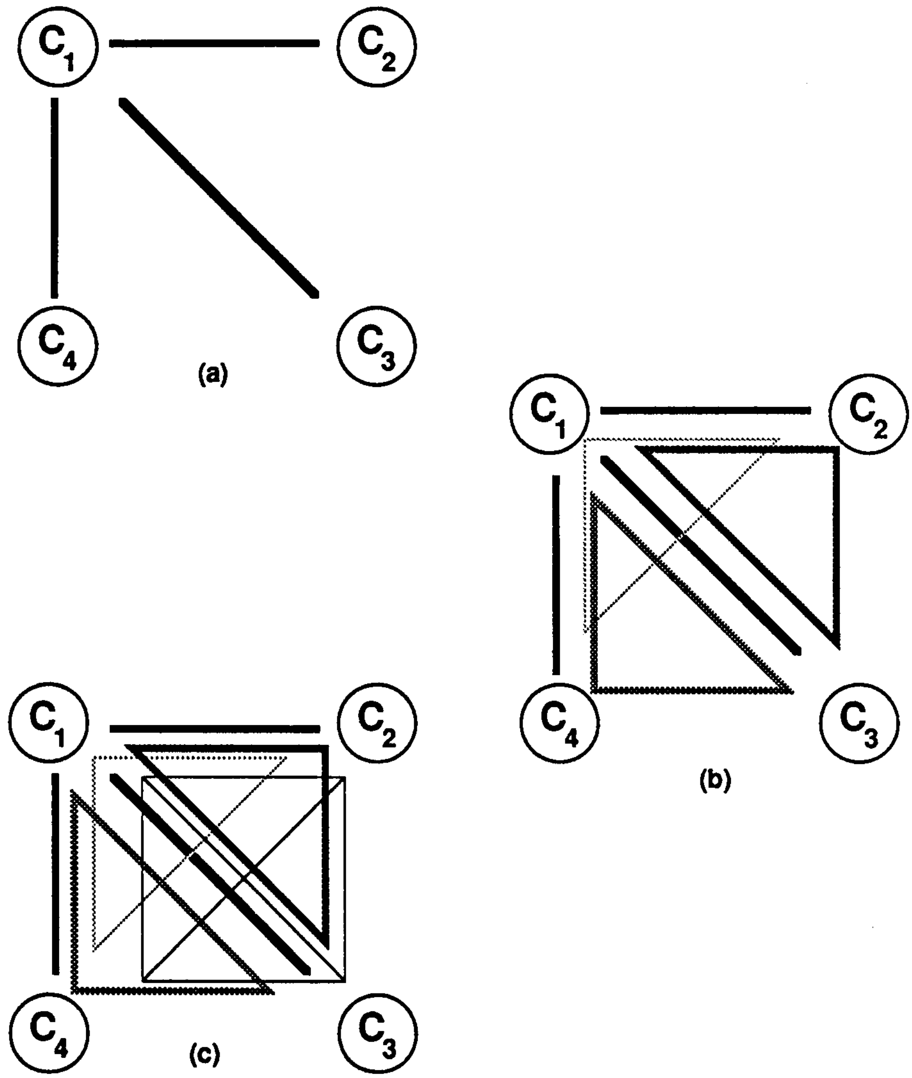


Figure 3.10: The union of realms of allowable components is the realm of an allowable component.

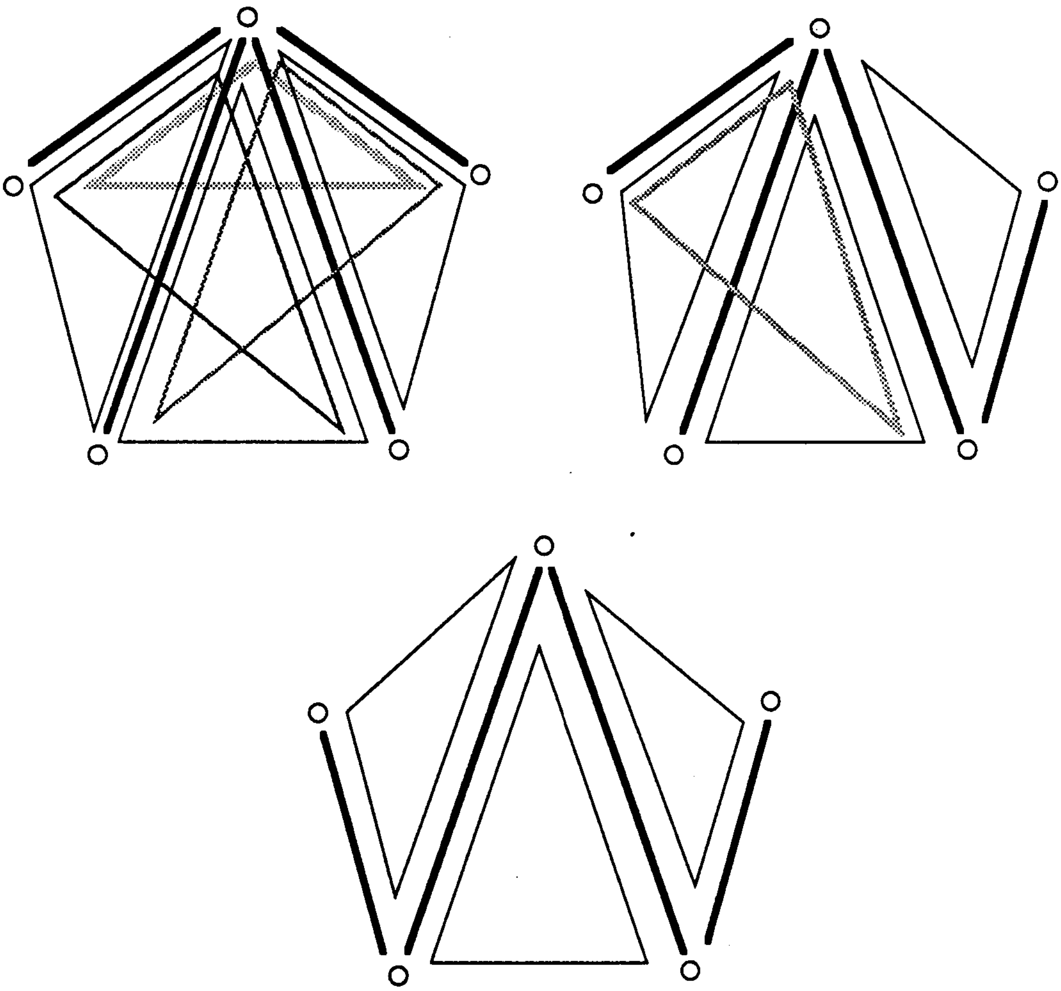


Figure 3.11: Exclusive intersections with realms of order two or three in the three 5-Clique templates.

arcs x_1x_2, x_2x_3, x_3x_4, and x_4x_1, and therefore cliques

$$\begin{aligned} C_1 &= \{x_1, x_2\}, \\ C_2 &= \{x_2, x_3\}, \\ C_3 &= \{x_3, x_4\}, \\ C_4 &= \{x_1, x_4\}. \end{aligned}$$

(See Figure 3.12(a).)

Here we have

$$x_1 \in \mathcal{XI}(\{C_1, C_4\})$$

and

$$x_2 \in \mathcal{XI}(\{C_1, C_2\}),$$

so each of these $\mathcal{XI}$ components is clearly allowable. If Theorem 3.14 held for nontriangulated graphs, then we would have that $\mathcal{XI}(\{C_1, C_2, C_4\})$ must be allowable, and thus we might have some $x_Z \in \mathcal{XI}(\{C_1, C_2, C_4\})$. But as shown in Figure 3.12(b), this is a violation of Theorem 3.8, and x_Z must be in some clique with x_3 and x_4, which are not both present in any of C_1, C_2, or C_4.

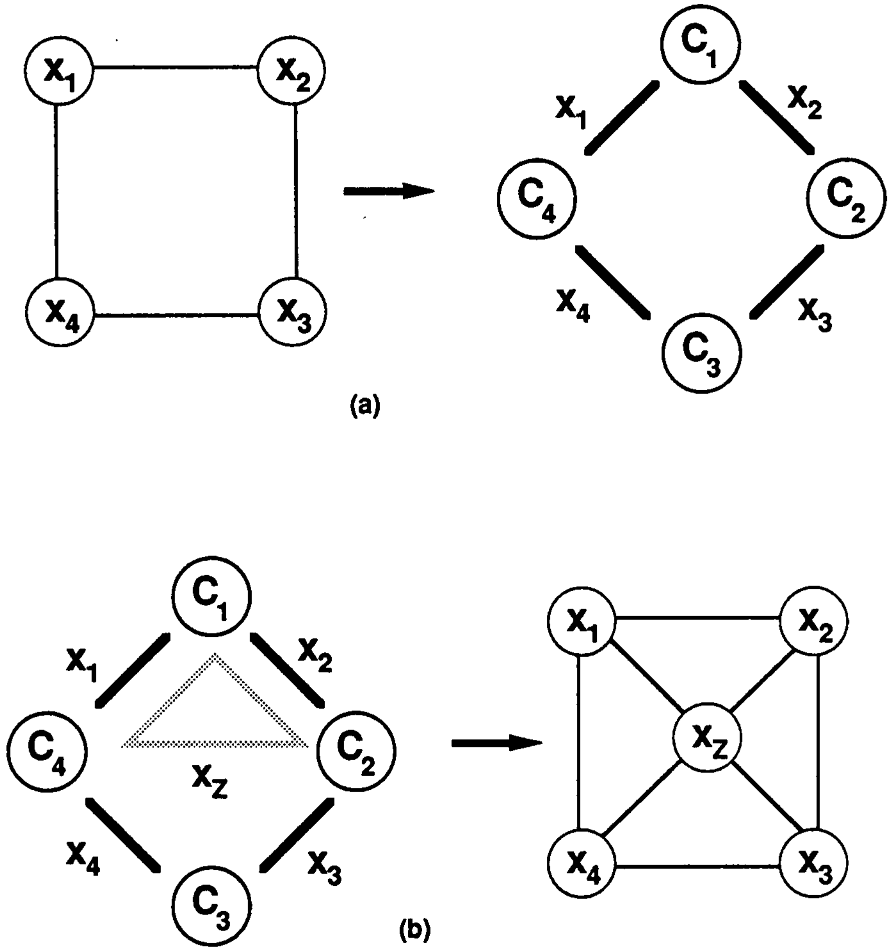


Figure 3.12: The clique component diagram in (a) is valid; the diagram in (b) is not valid.

CHAPTER 4

THE LAURITZEN - SPIEGELHALTER METHOD

4.1 Introduction

Various methods for dealing with uncertainty in artificial intelligence have been reported [1,12,16,27,28,35,43,47,56,59]. This chapter is a review of a method developed primarily by Lauritzen and Spiegelhalter [32] for calculating Bayesian probabilities associated with the variables in a knowledge-based system. The method provides a graph-theoretic approach to solving probabilities of interest in the system and thus is an example of a type of application for which the results of the research presented elsewhere in this paper are particularly useful. Clique theory results from Chapters 2 and 3 can aid in the development of more rapid algorithms for the initialization and updating procedures of the method; results developed specifically to isolate areas of possible parallelism within the method will be covered in Chapter 5.

Other related work has been reported recently in the areas of graph theory [8,11,24,29,38,61,62] and parallelism in knowledge-based systems [23,42,43,44,45].

4.2 Markov Probability Functions

We begin with a brief review of pertinent Bayesian probability theory. Suppose $\Pr(x_1, x_2, \ldots, x_n)$ is a joint probability density function over the discrete variables $x_1, \ldots, x_n$. The chain rule states that we can rewrite the joint density function as

$$\begin{aligned}\Pr(x_1, x_2, \ldots, x_n) &= \Pr(x_1 \mid x_2, \ldots, x_n) \times \Pr(x_2 \mid x_3, \ldots, x_n) \times \ldots \\ &\quad \times \Pr(x_{n-1} \mid x_n) \times \Pr(x_n),\end{aligned}$$

where, for example, $\Pr(x_1 \mid x_2, \ldots, x_n)$ is the conditional probability of x_1 given values of x_2 through x_n, and $\Pr(x_n)$ is the marginal probability of x_n alone. In some cases it may be that the value taken on by variable x_i does not directly depend on all of the values of x_{i+1} through x_n, but only on some smaller subset of x_{i+1} through x_n. For example, if the value of x_2 depends directly only on the values of x_3 and x_5, then the joint density function may be written

$$\begin{aligned}\Pr(x_1, x_2, \ldots, x_n) &= \Pr(x_1 \mid x_2, \ldots, x_n) \times \Pr(x_2 \mid x_3, x_5) \times \ldots \\ &\quad \times \Pr(x_{n-1} \mid x_n) \times \Pr(x_n).\end{aligned}$$

We would say here that x_2 is *conditionally independent* of the other variables in the system, given x_3 and x_5. A joint probability density function that can be written as the product of conditional probabilities representing such reduced dependency relationships is said to have the *Markov property*, and it is functions of this type that will be considered in this discussion.

4.3 Application: Knowledge-based Systems

Probability density functions with the Markov property arise naturally in certain knowledge-based systems comprising large numbers of variables, each variable representing some measurable component of the system model, where the value taken on by each variable is directly influenced by some subset of the remaining variables. Interdependencies among the variables may be represented graphically by a *Markov network* in which each node represents a variable in the system and each directed arc represents an apparent direct dependency relationship as observed in the data or suggested by an expert.

Taking advantage of the limited interdependence of groups of variables as illustrated by the Markov network graph, the goals of our work with this representation are

- to be able to determine quickly the marginal probability density function for one or several variables in the system, and
- to perform rapid calculations of other probabilities of interest inherent in the system, such as the effect on the marginal probability density function of some group of variables, given that values for some other group of variables in the system have been observed.

Considerations to this end include determining how best to divide the Markov network into small groups of variables on which probability calculations can be carried out locally and determining where such calculations can be performed in parallel, as among groups of variables with no explicit interdependencies.

4.4 Markov Network Example: Warm Core Ring Tracking

The example that will be used throughout this discussion is a Markov network based on the analysis of sequences of infra-red satellite image data of a region of the Atlantic Ocean off the US east coast [51,64,65]. Large, elliptically-shaped eddies called *warm-core rings* (WCRs) form in this region from meanders of the Gulf Stream which separate from the Stream and move through the ocean independently of the Stream. After a period of up to roughly one year, a WCR rejoins the Gulf Stream and is no longer visible as a separate entity.

The Markov network used here as an example models the movement of a WCR from the time that it forms until it eventually coalesces with the Gulf Stream (See Figure 4.1). Such a model is useful for making predictions about the current or future location of a WCR in the absence of clear satellite image data because of obstruction by cloud cover. Conditional probabilities used in the model are based on recent statistical analyses of a large number of WCRs over a period of several years [10].

Each variable, or node, in the Markov network represents a set of values from the image data reported in the statistical analyses, and a directed arc in the network represents an implied direct relationship between the values observed for the two variables connected by the arc. The source node of a directed arc is called a *parent* node; the node at the destination of the arc is a *child* of that

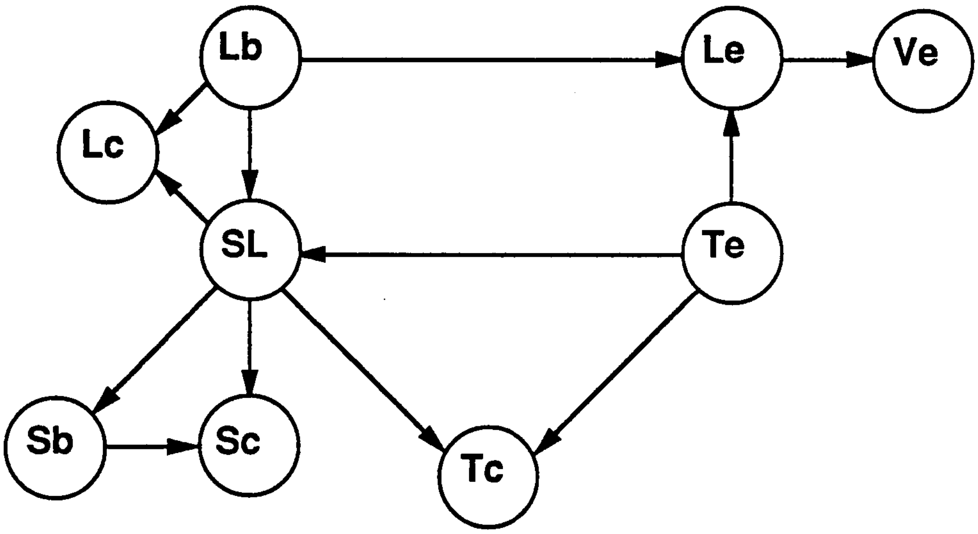


Figure 4.1: An example Markov network: WCR movement.

parent. The variables used in this example are:

Lb: the location of birth of a WCR, i.e., the geographical position of the center of a WCR at the time of its formation, as a pair *(latitude north, longitude west)* in degrees.

Sb: the semi-major axis of a WCR at birth, in kilometers.

Lc: the location of coalescence of a WCR, as a pair *(latitude north, longitude west)* in degrees.

Sc: the semi-major axis of a WCR at coalescence, in kilometers.

Tc: the time of coalescence of a WCR, given as days after birth.

SL: the status of a WCR as short-lived or long-lived, where

$$SL = \begin{cases} \text{short}, & \text{if } Tc < 140, \\ \text{long}, & \text{otherwise.} \end{cases}$$

Te: the age of a WCR in days.

Le: the location of a WCR at time Te, as a pair *(latitude north, longitude west)* in degrees.

Ve: the velocity of a WCR at time Te as a pair *(westward magnitude, northward magnitude)* in centimeters per second.

The graph of the Markov network for this model reflects, for example, that the probability function for the set of values of the variable Tc depends directly only on the probability functions of the variables SL and Te. Thus, in an expression for the joint probability of the entire system as a product of conditional probabilities, we may include the probability of Tc conditioned on SL and Te alone, since no other dependence is given in the experts' study [10]. If a variable appears as a node in the graph with no incoming arcs, such as the variable Lb

in this example, then the marginal probability function for that variable must be given explicitly.

Based on the dependencies represented in the Markov network graph, we may write the joint probability density function for the system as

$$
\begin{aligned}
&\Pr(Lb, Sb, Lc, Sc, Tc, SL, Te, Le, Ve) = \\
&\quad \Pr(Lb) \times \Pr(Sb \mid SL) \times \Pr(Lc \mid Lb, SL) \times \\
&\quad \Pr(Sc \mid Sb, SL) \times \Pr(Tc \mid SL, Te) \times \\
&\quad \Pr(SL \mid Lb, Te) \times \Pr(Te) \times \Pr(Le \mid Lb, Te) \times \\
&\quad \Pr(Ve \mid Le).
\end{aligned}
$$

All of the conditional and marginal probabilities appearing in this expression for the joint density function are obtainable directly from the statistical data recorded. The task is now to use this information to derive other marginal probability functions of interest that are not explicitly given, and to be able to update the entire joint probability density function for the system in the event that specific observed values for some variables in the system become available.

4.5 Description of the Method

4.5.1 Operations on the Graph

The first step in calculating local probability values is to operate directly on the graphical representation of the system, locating within this graph groups of variables with explicit interdependencies. In general, the method for calculating the probability density function for the system as a whole will involve performing calculations on these groups of variables independently, with results from these calculations being passed from group to group as necessary to account for the interdependence among the groups.

Groups are determined as follows: The parent nodes of each node in the graph are *married* by adding undirected arcs between each pair of parents. Then, each directional arc in the graph is replaced with an undirected arc. (It is enough to know that the values taken on by two variables are directly related — the nature of the dependence is unimportant beyond the initial representation of the joint probability density function as the product of conditional probabilities.) If any cycle of length four or greater with no chord now exists in the graph, arbitrarily add chords between nodes in the cycle until no chordless cycle of length four or greater remains. The graph is now said to be *triangulated* (See Figure 4.2).

Note that the process of triangulating a graph in which chordless cycles of length four or greater exist is in general a nondeterministic process, and the choices made from among different possible triangulations may have a significant impact on the efficiency of later calculations in the method. One area of further research, therefore, is to analyze the effects of different triangulation schemes on overall efficiency.

We now locate all maximal, completely connected subgraphs of the modified graph. These groups of nodes are called *cliques* and are the sets of variables

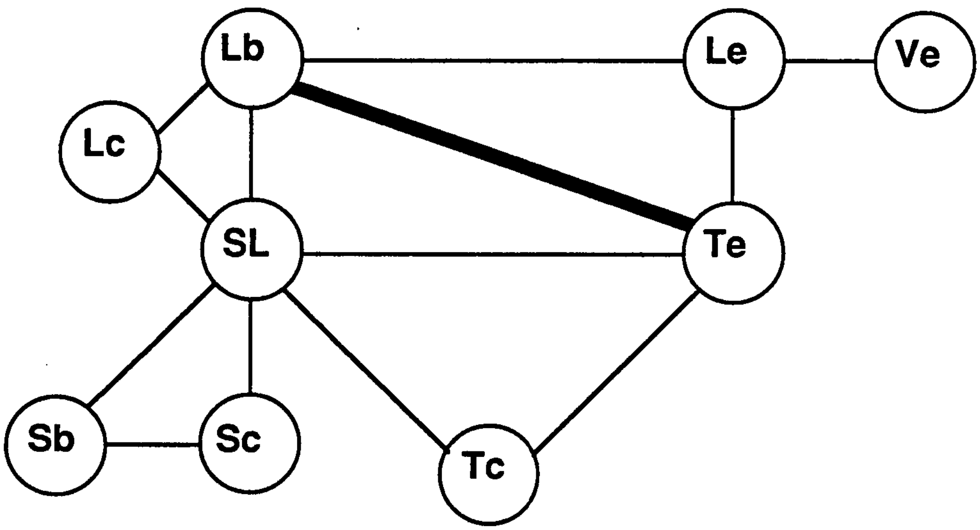


Figure 4.2: The modified graph: parents are married, directions on arcs are dropped, and arcs have been added for triangulation.

that form the basis for the method of local probability calculations. The set of cliques is then ordered so that the following rule holds:

> The intersection of any clique with the union of all cliques preceding it in the ordering, is contained in a single clique, called a *parent* clique.

Such an ordering is said to have the *running intersection property* (RIP). The cliques in the graph in Figure 4.2 are shown in Table 4.1 in one possible RIP ordering, along with the set of possible parents for each clique. Note that a clique may have more than one possible parent and that the first clique in the ordering has no parent.

The intersection of any clique C_i with a parent clique is called the *separator* of the clique, written S_i, and the nodes that are in the clique but not in the separator are called the *residual* of the clique, written R_i. The separator and residual of each clique in the WCR example are also given in Table 4.1. Research has shown that given the set of cliques of a triangulated graph, the set of separators is unique, independent of the RIP clique ordering [7]. If we choose a single parent for each clique that has at least one possible parent, the resulting set of relationships defines a *clique tree.* A clique with no children is a *leaf* of the tree. The three possible clique trees for our example, given the ordering in Table 4.1, are shown in Figure 4.3.

Other work has shown that each clique tree based on a RIP clique ordering for a set of cliques Z corresponds to a maximal spanning tree of the clique graph of Z, where the labels along the arcs of the maximal spanning tree are the clique separators as defined here. Thus, the results of Chapter 2 may be

Clique	*Separator*	*Residual*	*Possible Parents*
$C_1 = \{$ Tc, SL, Te $\}$	$\emptyset$	Tc, SL, Te	-
$C_2 = \{$ Lb, SL, Te $\}$	SL, Te	Lb	C_1
$C_3 = \{$ Lb, Te, Le $\}$	Lb, Te	Le	C_2
$C_4 = \{$ Lb, Lc, SL $\}$	Lb, SL	Lc	C_2
$C_5 = \{$ Sb, Sc, SL $\}$	SL	Sb, Sc	C_1, C_2, C_4
$C_6 = \{$ Le, Ve $\}$	Le	Ve	C_3

Table 4.1: Cliques in the WCR network with their separators and residuals.

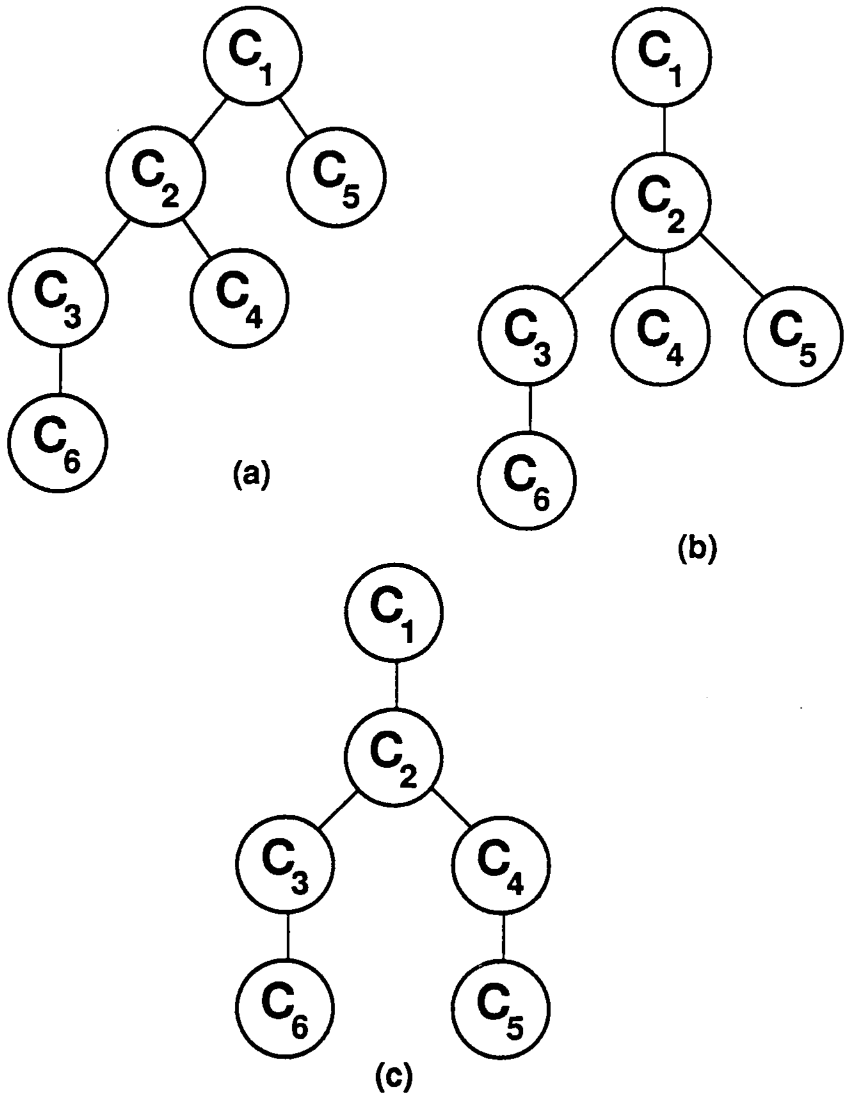


Figure 4.3: Clique trees for the WCR network.

used to conduct the search for separators in parallel: given the clique graph G_C of a system, we know by Theorem 2.17 that any Clique in G_C must contain exactly $n - 1$ arcs that represent separators, where n is the number of nodes in the Clique.

4.5.2 Potential Functions

After some clique tree representation is chosen, each clique C_i is then assigned its own *potential function*, ψ_i, as follows:

1. Choose a term in the product expression for the joint probability density function for the system.

2. Assign this term to a clique that contains all of the variables that appear in the term.

3. Continue such assignments until all terms in the joint probability function expression have been used.

The potential function for each clique is then formed by taking the product of all terms that have been assigned to that clique. If no terms were assigned to some clique, then set the potential function for that clique equal to some arbitrary nonzero constant, such as 1. One possible potential function assignment for this example is as follows:

$$\psi_1 = \Pr(Tc \mid SL, Te) \times \Pr(Te)$$

$$
\begin{aligned}
\psi_2 &= \Pr(SL \mid Lb, Te) \\
\psi_3 &= \Pr(Le \mid Lb, Te) \times \Pr(Lb) \\
\psi_4 &= \Pr(Lc \mid Lb, SL) \\
\psi_5 &= \Pr(Sc \mid Sb, SL) \times \Pr(Sb \mid SL) \\
\psi_6 &= \Pr(Ve \mid Le)
\end{aligned}
$$

Note that this method of constructing clique potential functions is nondeterministic since a term in the joint probability function expression may contain variables which all appear in more than one clique, and thus may be assigned to any one of several cliques. For example, the term $\Pr(Lb)$ could have been assigned to either clique C_2 or clique C_4 rather than clique C_3. In general, there will be a choice of clique potential function assignments for a term whenever the set of variables in that term is contained in a separator. The resulting choice of an initial potential function representation on the set of cliques determines the nature and amount of information that must be passed among the cliques, and may also affect the amount of local computation that must be performed within a single clique. Further exploration of this topic is presented in Chapter 5.

4.5.3 Backward Propagation: Clique Conditionals

The next step is to transform each clique potential function into a conditional probability in terms of its separator and residual only. Once this has been achieved, it will be a simple matter to pass information from a parent clique to a child clique by isolating the probability information that pertains only to the separator of the child — these are the only variables within the parent clique whose values have a direct impact on the values taken on by variables within the child clique. Conversion of potential functions to the form residual-conditioned-on-separator for leaf cliques is facilitated by the following calculation:

$$\begin{aligned}\text{new } \psi_i &= \frac{\psi_i}{\sum_{R_i} \psi_i} \\ &= \Pr(R_i \mid S_i).\end{aligned}$$

(The justification for this step is given in [32].)

The denominator from this calculation, called a *multiplier* and written M_i, is then passed to a parent of the clique, where it is multiplied times the potential function of the parent to form a new, intermediate potential function for the parent. (See Figure 4.4.) Note that in general, the multiplier M_i is an array of real number probability measures, each corresponding to a different combination of possible values of the variables in the separator S_i. When a parent clique C_j has received and processed a multiplier from each of its children, we refer to the new form of its potential function as a *modified potential function*, written $\hat{\psi}_j$, and it is this form that we use in the calculation of still the next form for the potential function which is equal to the probability density function of the residual R_j conditioned on the separator S_j:

$$\begin{aligned}\text{new } \psi_j &= \frac{\hat{\psi}_j}{\sum_{R_j} \hat{\psi}_j} \\ &= \Pr(R_j \mid S_j).\end{aligned}$$

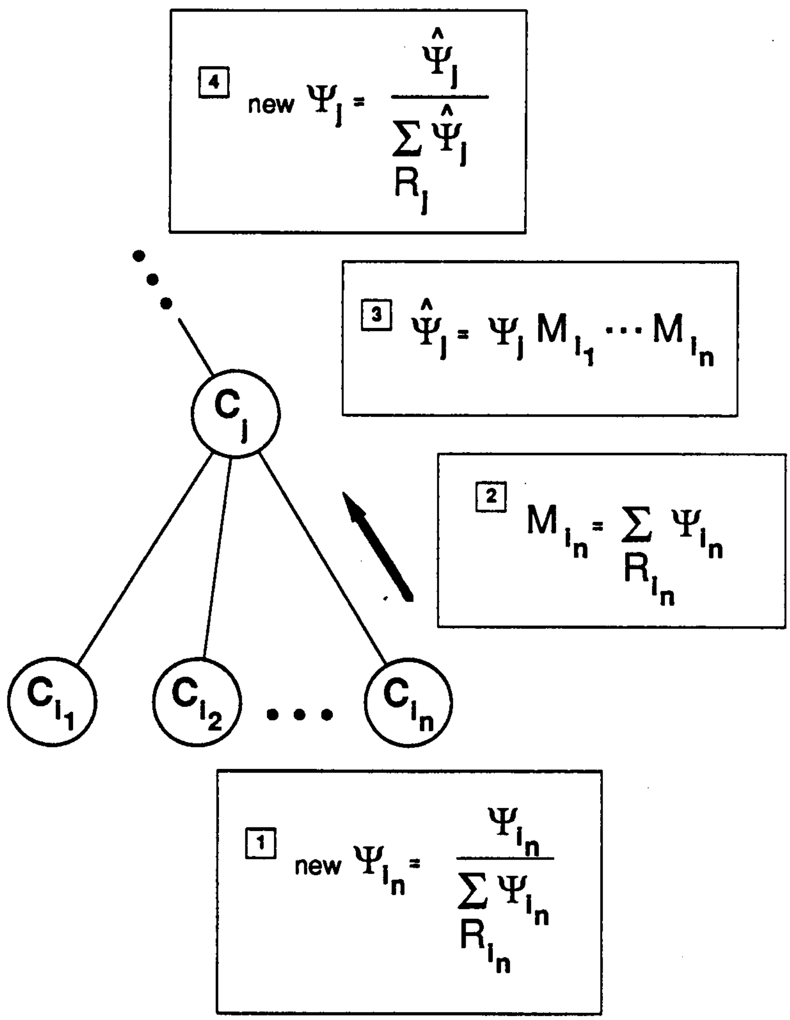


Figure 4.4: Backward propagation in the Lauritzen - Spiegelhalter method.

Since leaf cliques have no children, the modified potential function $\hat{\psi}$ of a leaf clique is equal to its initial potential function ψ, so we may use the above expression as the general rule for this phase of potential function conversion. Thus, the general form for the multiplier passed back from clique C_i is

$$M_i = \sum_{R_i} \hat{\psi}_i.$$

We will call this process of converting potential functions to the form $\Pr(R_i \mid S_i)$ and passing back multipliers the *backward propagation phase* of the Lauritzen - Spiegelhalter method, and we continue the process upward in the clique ordering until the potential functions for all cliques have been transformed to the desired form. Figure 4.5 shows the forms of the potential functions for cliques in the WCR example following backward propagation using the clique tree from Figure 4.3(b).

Note that if the initial potential function at some clique is already equal to the probability of its residual conditioned on its separator, then the backward propagation phase has no effect on the potential functions of that clique or of its parent through that clique. Thus if this condition can be recognized, unnecessary computation can be avoided. An area of further investigation which led to some of the results to be presented in Chapter 5 was to determine which of several possible initial clique potential function representations would result in a lower amount of computation to be performed during this stage of the method.

Another motivation for the research presented here was to determine to what

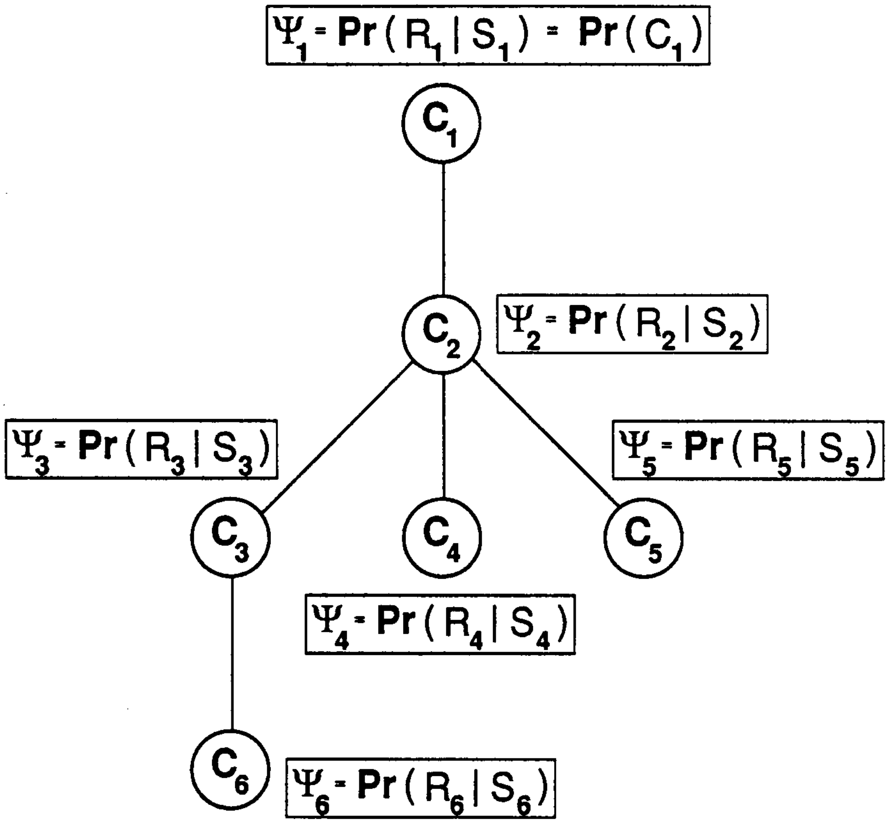


Figure 4.5: Potential functions in a WCR clique tree after backward propagation.

degree the calculations of the new clique potential functions could be carried out in parallel. The calculation of a new potential function for a clique can begin as soon as it has received and incorporated the multipliers from all of its children. Clearly, all cliques with no children can be processed in parallel, as can cliques receiving all ones in the multipliers from their children.

4.5.4 Forward Propagation: Clique Marginals

In the next phase of the Lauritzen - Spiegelhalter method, the potential function at each clique is converted into the marginal probability density function for the variables in the clique. Again, we make a complete pass over the clique tree representation of the system, but this time from top to bottom rather than from the leaves upward. Since the highest clique in the clique ordering, C_1, has no parent and thus no separator, and the backward propagation phase has left each potential function in the form $\Pr(R_i \mid S_i)$, the newly calculated potential function ψ_1 is equal to the probability of the entire set of variables in C_1 conditioned on the empty set. This is simply the marginal probability density function for the set of variables in the clique C_1. For each child of this clique, it is now possible to compute the clique marginal density function as follows:

1. Calculate the marginal of the separator of the child by summing the marginal of the parent clique over all values of variables not in the separator.

2. Multiply the potential function of the child clique by the marginal of its separator.

Since the potential function of each clique is the probability of its residual conditioned on its separator, multiplying the potential function of a clique by the marginal of its separator yields the marginal probability density function for that clique:

$$\begin{aligned}
\psi_i \times \Pr(S_i) &= \Pr(R_i \mid S_i) \times \Pr(S_i) \\
&= \Pr(R_i, S_i) \\
&= \Pr(C_i) \\
&= \mathit{marginal}(C_i).
\end{aligned}$$

Thus, the calculation of the marginal density function of a clique can begin as soon as the marginal density function of a parent clique has been determined. A clique may have several children, each of which can calculate its marginal density function independently of the others.

This process of propagating marginals of separators through the chain of cliques is called the *forward propagation phase* of the Lauritzen - Spiegelhalter method. We continue this process until the marginal density function of each clique has been calculated. A diagram outlining the steps in the process is given in Figure 4.6. Marginals for individual variables in the system can now be computed from the clique marginals by summing over all values of the other variables in a clique that contains the variable of interest.

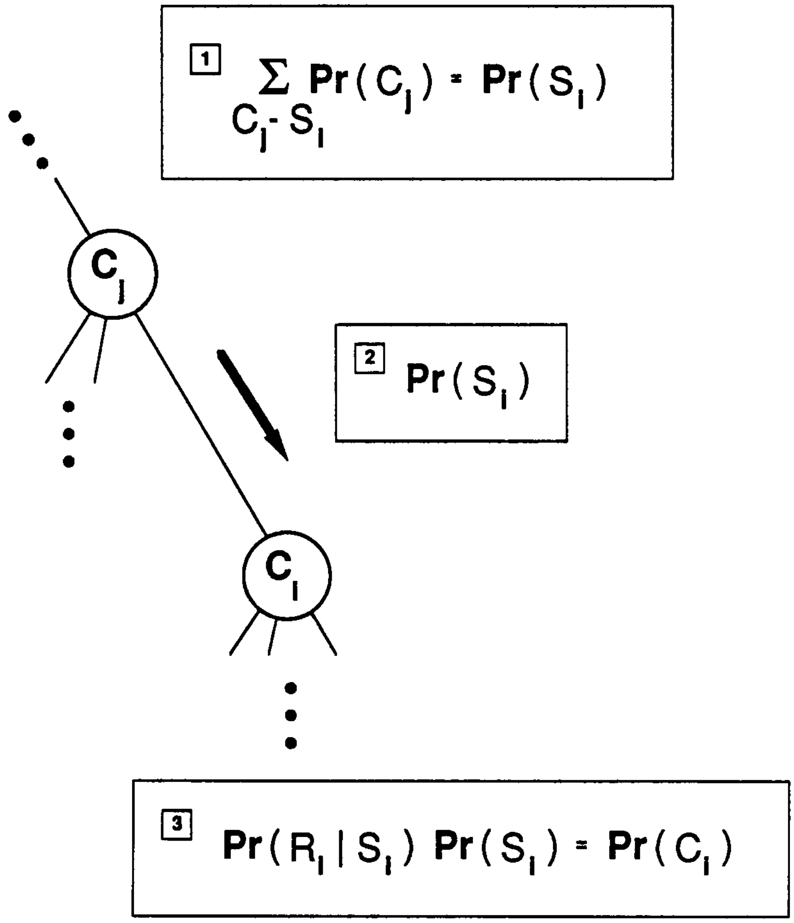


Figure 4.6: Forward propagation in the Lauritzen - Spiegelhalter method.

4.5.5 Absorption of Evidence

When specific values for variables in the network model are observed or hypothesized, the effects of this new evidence must be propagated to the other variables in the network so that new probability functions can be computed which are conditioned on the values observed.

The marginal probability density function of a variable whose value is known is trivial — the probability of the observed value is one, all other probabilities are zero. Thus, nodes corresponding to observed values may be removed from the network (Figure 4.7), reducing the size of the cliques that contained them and possibly altering the basic clique structure of the graph (see Chapter 5). Potential functions for these cliques must then be recomputed based on the newly reduced set of variables. Following this, computation of new probability functions for the nodes remaining in the network can be accomplished by using the backward and forward propagation phases described earlier.

The current method as described by Lauritzen and Spiegelhalter requires the construction of a new clique ordering after evidence has been absorbed, with complete recalculations of separators, residuals, and possible parents. This may not be necessary if a representation of the structure of the original graph of the system is stored as a clique component diagram as described in Chapter 3. It was observed there that the removal of variables from a graph can only affect the clique structure of a graph if such removal causes some $\mathcal{XI}$ clique component of the diagram to become empty.

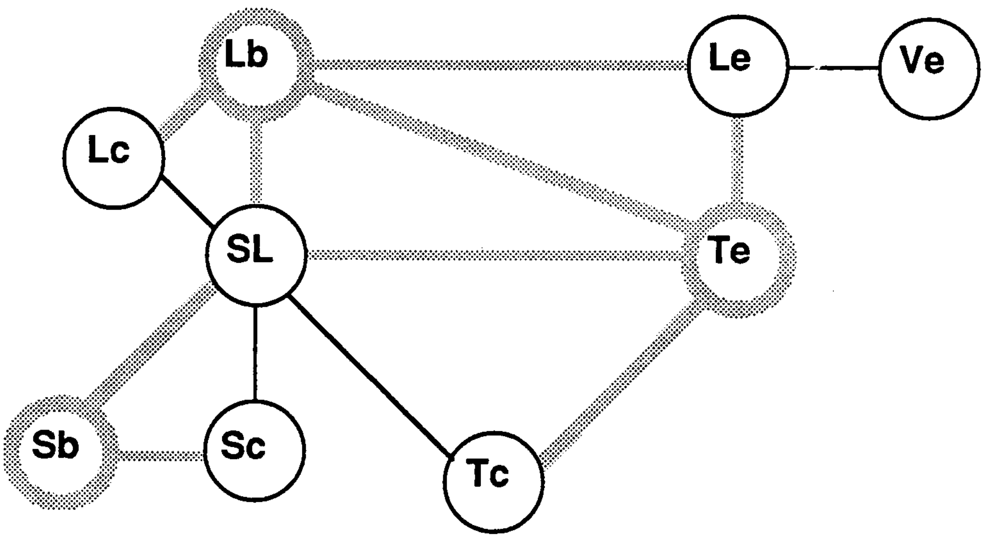


Figure 4.7: Absorption of evidence: values observed for Lb, Te, and Sb.

4.6 Summary

A method developed by Lauritzen and Spiegelhalter has been presented for computing probabilities that measure uncertainties in knowledge-based systems modeled as Markov networks. To illustrate the method, a small example of such a system was given that could be used to predict the position of a warm core ring in the Atlantic Ocean.

Primary areas of research that were discussed were the development of new graph theory, especially pertaining to cliques in graphs and potentials on graphs, and the development of parallel algorithms that make use of the graph theory results.

CHAPTER 5

INDEPENDENT COMPUTATIONS WITHIN THE LAURITZEN - SPIEGELHALTER METHOD

5.1 Introduction

In this chapter, we will take a careful look at the Lauritzen - Spiegelhalter method of local computation of probabilities on graphs to find sections in the algorithms of the method which may be computationally independent, and thus could be executed in parallel. Since the nature and amount of local computation performed in the method depend heavily upon the distribution of data within the network model, we must also specify any restrictions that must be met by an initial mapping of probability data onto the network structure in order to insure that the parts of the algorithms to be performed in parallel are indeed independent.

As we shall see, before the first step of each the backward and forward propagation algorithms which are described by Lauritzen and Spiegelhalter, it is possible to know where these algorithms may be split into separate processes which can be executed simultaneously. To do this, we must arrange for certain conditions to be met by our initial assignments of probability data to the nodes of the clique graph which corresponds to the original network representation of a system. Theorems presented in this chapter describe these conditions and verify that they do indeed render certain sections of the work performed within the Lauritzen - Spiegelhalter method computationally independent.

5.2 Independence in Backward Propagation

The Lauritzen - Spiegelhalter method is composed primarily of two main algorithms which we call the backward propagation phase and the forward propagation phase. The backward phase is executed first.

During the backward propagation phase, the potential function associated with each clique C_i is transformed from its initial form into the probability density function of the variables in the residual R_i conditioned on the variables in the separator S_i.

A key element in the backward propagation phase is the multiplier passed back by a clique. Recall from Chapter 4 that a multiplier in the context of the Lauritzen - Spiegelhalter method is an array of real numbers, each between zero and one inclusive, which is passed from a child node to a parent node in a clique tree representation of the system network. By definition, the multiplier M_i passed back by child clique C_i is equal to the sum of its modified potential function $\hat{\psi}_i$ over all of the values of its residual variables in R_i:

$$M_i = \sum_{R_i} \hat{\psi}_i.$$

Intuitively, the multiplier passed back to a parent clique from clique C_i is a summary of all of the probability information in the clique subtree T_i rooted at C_i that is needed by the remainder of the complete clique tree in order to account for the influence of the variables in the cliques of T_i on the overall joint probability density function of all variables in the system.

The multiplier M_i has a significant effect on the probability density functions associated with the nodes outside of its subtree T_i only if any of the entries in M_i are not equal to one; in this case, the potential function of the parent clique will be altered as the result of the application of the multiplier, and this alteration must take place before the receiving parent can calculate its own multiplier. On the other hand, if M_i comprises only entries that are equal to one, its application to the potential function of a parent clique has no effect, and thus the information it carries is not needed before computations involving the parent clique and the rest of the tree can proceed. In terms of the probabilities of values of variables in the system, this situation may be interpreted as the condition that the probability distribution of no variable outside of the subtree T_i is dependent on the probability distribution of any variable in any clique inside T_i given the representation of the joint probability density function of the entire system implicit in the potential functions of cliques in the network.

Based on these observations, it is desirable to be able to determine when the multiplier passed back by a clique C_i is in fact all ones, or how a multiplier of all ones can be produced at C_i when this does not occur automatically, and thereby know that the processing of a parent clique of C_i during the execution of the backward phase of the Lauritzen - Spiegelhalter method need not wait until processing of clique C_i is completed. Furthermore, if all of the children clique nodes of a parent clique C_j can be induced through judicious initial potential function assignments into having multipliers that are equal to a vector of ones, then the processing of C_j during the backward propagation phase will be completely independent of the portion of the subtree T_j below C_j, and thus clique C_j could be processed at the onset of the backward propagation phase in parallel with the processing of the leaf nodes of the clique tree.

Unless otherwise stated, all theorems in this chapter assume the existence of some clique tree representation of a network system of elemental variables constructed according to the method discussed in Chapter 4. Notation used includes

- the function $parent(C_i)$, which returns the parent clique of C_i in the chosen clique tree representation,
- the set $nodes(T_i)$, which is equal to the union of all cliques in tree T_i, and
- the symbol $\mathbf{1}$, which will be used to represent an array $(1, 1, \ldots, 1)$ of dimension appropriate to the context.

The first group of theorems presented here establishes some basic properties of the residual variables of cliques in a clique tree.

Theorem 5.1 *No variable can be in two different residuals.*

Proof. Suppose we have some $r \in R_i$, $r \in R_j$, with $i \neq j$. Then $C_i \cap C_j \ni r$, and either $r \in S_i$ or $r \in S_j$ by the running intersection property.

But $[r \in S_i] \Rightarrow [r \notin R_i]$ and $[r \in S_j] \Rightarrow [r \notin R_j]$, so either $r \notin R_i$ or $r \notin R_j$. $\square$

Theorem 5.2 *Each variable in separator S_i is in residual R_j for some clique C_j above C_i in the clique tree.*

Proof. Suppose $x \in S_i$. Then $x \in parent(C_i) = C_j$, for some $j < i$, and either $x \in R_j$ or $x \in S_j$.

If $x \in R_j$, we are finished. Otherwise, $x \in S_j$ and thus $x \in parent(C_j) = C_k$, for some $k < j$, and either $x \in R_k$ or $x \in S_k$.

If necessary, we may continue upward in the clique tree in this manner until, at least, $x \in C_1 = R_1$. □

Corollary 5.3 *Each variable is in exactly one residual.*

Proof. Follows directly from the two preceding theorems. □

The following definition allows reference to the aggregate residual for an entire subtree. The accompanying theorem demonstrates that this aggregate residual provides an alternative way of representing the subtree itself.

Definition 5.1 *The* **cumulative residual,** *written $\boldsymbol{R}_i$, of a clique C_i is the union of the residuals of all cliques in the subtree T_i with root C_i:*

$$\boldsymbol{R}_i = \bigcup_{C_j \in T_i} R_j$$

Theorem 5.4 *The union of all residuals in subtree T_i is equal to the union of all cliques in T_i, less S_i:*

$$\boldsymbol{R}_i = nodes(T_i) - S_i$$

Proof. (Induction)

Base step: Tree with one level:

$$
\begin{aligned}
\boldsymbol{R}_i &= \bigcup_{C_j \in T_i} R_j \\
&= R_i \\
&= C_i - S_i \\
&= nodes(T_i) - S_i
\end{aligned}
$$

Inductive step: Tree with several levels:

Assume the theorem is true for level j, and let $i = j + 1$. We must show that the theorem then holds for level i.

Let C_i be any clique on level i, and let $C_{j_1}, C_{j_2}, \ldots, C_{j_n}$ be the children of C_i. Also, for this proof, let $T_{j_k} \equiv nodes(T_{j_k})$.

Then,

$$
\begin{aligned}
\boldsymbol{R}_i &= \bigcup_{C_k \in T_i} R_k \\
&= R_i \cup \boldsymbol{R}_{j_1} \cup \cdots \cup \boldsymbol{R}_{j_n} \\
&= R_i + [T_{j_1} - S_{j_1}] + \cdots + [T_{j_n} - S_{j_n}] \\
&= [C_i - S_i] + [T_{j_1} - S_{j_1}] + \cdots + [T_{j_n} - S_{j_n}] \\
&= C_i \overline{S}_i + T_{j_1} \overline{S}_{j_1} + \cdots + T_{j_n} \overline{S}_{j_n}
\end{aligned}
$$

Since $[S_{j_k} \subset C_i] \Rightarrow [C_i = C_i + S_{j_k}]$, we may write

$$
\boldsymbol{R}_i = [C_i + S_{j_1} + \cdots + S_{j_n}] \overline{S}_i + T_{j_1} \overline{S}_{j_1} + \cdots + T_{j_n} \overline{S}_{j_n}
$$

$$
\begin{aligned}
&= C_i\overline{S}_i + S_{j_1}\overline{S}_i + \cdots + S_{j_n}\overline{S}_i + T_{j_1}\overline{S}_{j_1} + \cdots + T_{j_n}\overline{S}_{j_n} \\
&= C_i\overline{S}_i + S_{j_1}\overline{S}_i + T_{j_1}\overline{S}_{j_1} + \cdots + S_{j_n}\overline{S}_i + T_{j_n}\overline{S}_{j_n}
\end{aligned}
$$

Since $[S_{j_k} \subset T_{j_k}] \Rightarrow [S_{j_k} = T_{j_k}S_{j_k}]$, we have

$$
\begin{aligned}
\boldsymbol{R}_i &= C_i\overline{S}_i + T_{j_1}S_{j_1}\overline{S}_i + T_{j_1}\overline{S}_{j_1} + \cdots \\
&\quad + T_{j_n}S_{j_n}\overline{S}_i + T_{j_n}\overline{S}_{j_n}
\end{aligned}
$$

Also, since $[T_{j_k}S_i \subset S_{j_k}] \Rightarrow [S_{j_k} = S_{j_k} + T_{j_k}S_i]$, we have

$$
\begin{aligned}
\boldsymbol{R}_i &= C_i\overline{S}_i + T_{j_1}S_{j_1}\overline{S}_i + T_{j_1}\left[\overline{S_{j_1} + T_{j_1}S_i}\right] + \cdots \\
&\quad + T_{j_n}S_{j_n}\overline{S}_i + T_{j_n}\left[\overline{S_{j_n} + T_{j_n}S_i}\right] \\
&= C_i\overline{S}_i + T_{j_1}S_{j_1}\overline{S}_i + T_{j_1}\overline{S}_{j_1}\left[\overline{T_{j_1}S_i}\right] + \cdots \\
&\quad + T_{j_n}S_{j_n}\overline{S}_i + T_{j_n}\overline{S}_{j_n}\left[\overline{T_{j_n}S_i}\right] \\
&= C_i\overline{S}_i + T_{j_1}S_{j_1}\overline{S}_i + T_{j_1}\overline{S}_{j_1}\left[\overline{T}_{j_1} + \overline{S}_i\right] + \cdots \\
&\quad + T_{j_n}S_{j_n}\overline{S}_i + T_{j_n}\overline{S}_{j_n}\left[\overline{T}_{j_n} + \overline{S}_i\right] \\
&= C_i\overline{S}_i + T_{j_1}S_{j_1}\overline{S}_i + T_{j_1}\overline{S}_{j_1}\overline{T}_{j_1} + T_{j_1}\overline{S}_{j_1}\overline{S}_i + \cdots \\
&\quad + T_{j_n}S_{j_n}\overline{S}_n + T_{j_n}\overline{S}_{j_n}\overline{T}_{j_n} + T_{j_n}\overline{S}_{j_n}\overline{S}_i \\
&= C_i\overline{S}_i + T_{j_1}S_{j_1}\overline{S}_i + T_{j_1}\overline{S}_{j_1}\overline{S}_i + \cdots + T_{j_n}S_{j_n}\overline{S}_n + T_{j_n}\overline{S}_{j_n}\overline{S}_i \\
&= C_i\overline{S}_i + \left[T_{j_1}S_{j_1}\overline{S}_i + T_{j_1}\overline{S}_{j_1}\overline{S}_i\right] + \cdots + \left[T_{j_n}S_{j_n}\overline{S}_i + T_{j_n}\overline{S}_{j_n}\overline{S}_i\right] \\
&= C_i\overline{S}_i + T_{j_1}\overline{S}_i\left[S_{j_1} + \overline{S}_{j_1}\right] + \cdots + T_{j_n}\overline{S}_i\left[S_{j_n} + \overline{S}_{j_n}\right] \\
&= C_i\overline{S}_i + T_{j_1}\overline{S}_i + \cdots + T_{j_n}\overline{S}_i \\
&= \left[C_i + T_{j_1} + \cdots + T_{j_n}\right]\overline{S}_i \\
&= T_i\overline{S}_i
\end{aligned}
$$

$$= T_i - S_i$$

□

A probability density function for a variable x alone or for x given the values of other variables provides information that is essential for the computation of any other probability measure that involves variable x. Thus, the location within the clique tree of the clique to which such a probability function is assigned during the initial potential function assignments to cliques plays an important role in determining the amount of communication necessary among the cliques that contain x during later phases of the Lauritzen - Spiegelhalter method.

The following theorems provide some basic properties of the type of probability function described above, as well as a range for the location within the clique tree of the clique in which this probability function may appear as a part of a clique potential function.

Definition 5.2 *A variable x is said to be* **substantive** *in an expression if x appears as either*

1. *the only variable in a* marginal *probability density function, for example,* $\Pr(x)$, *or*

2. *the only* dependent *variable in a conditional probability density function, for example,* $\Pr(x|y, z)$.

Each of these is called a **substantive probability function** *for x, abbreviated Spf(x).*

Theorem 5.5 *The sum of any Spf(x) over all possible values of x equals* **1**.

Proof.

Case 1: Spf(x) is a marginal probability density function for x:

$$\begin{aligned}\sum_x \mathrm{Spf}(x) &= \sum_x \Pr(x) \\ &= 1 \\ &= \mathbf{1},\end{aligned}$$

where **1** represents the array with the single entry 1.

Case 2: Spf(x) is a conditional probability density function in which x is a dependent variable:

$$\begin{aligned}\sum_x \mathrm{Spf}(x) &= \sum_x \Pr(x|V),\ x \notin V \\ &= \sum_x \frac{\Pr(x,V)}{\Pr(V)} \\ &= \frac{1}{\Pr(V)} \sum_x \Pr(x,V) \\ &= \frac{1}{\Pr(V)} \Pr(V) \\ &= \mathbf{1},\end{aligned}$$

where **1** represents an array with an entry of 1 for each combination of possible replacement values of variables in V. □

Theorem 5.6 *Each variable is substantive in an initial potential function exactly once.*

Proof. Each variable x that appears as a node in the network must conform to one of the following characterizations:

1. Parent only.

2. Child only.

3. Parent and child.

4. Neither parent nor child.

In cases 1 and 4, a marginal probability density function for x must be given. Since in these cases x would have no parent nodes in the network, x would not appear as a dependent variable in any conditional probability function.

In cases 2 and 3, x has parents in the network. Thus, x will appear as a dependent variable conditioned on these parents in some conditional probability density function. A marginal probability function for x in either of these cases would be redundant in the product expression for the joint probability density function of the entire network, and thus must be left out. □

Theorem 5.7 *If a variable in R_i is not substantive in ψ_i, then it must be substantive in ψ_j for some C_j, a descendant of C_i in the clique tree.*

Proof. Suppose $r \in R_i$ but no probability density function of the form $\Pr(r)$ or $\Pr(r|\ldots)$ appears in ψ_i. By Theorem 5.6 we know that a substantive probability function for r exists, and by the method used to make initial potential function assignments described in Chapter 4, this function must appear in some ψ_j such that $r \in C_j$. Thus, $r \in C_i \cap C_j$, and either $r \in S_i$ or $r \in S_j$ by the running intersection property. But $r \in R_i$, so $r \notin S_i$ and it must be that $r \in S_j$.

By Theorem 5.2, r must be contained in the residual of some clique above C_j in the clique tree, and by corollary 5.3 there is only one such clique. That clique must therefore be C_i, so we have that C_j is a descendent of C_i in the clique tree. □

Just as we defined an aggregate residual for a clique subtree, it is also convenient to have an aggregate potential function for a subtree. These two concepts together will allow us to show the significance of the multiplier passed back by the root clique of a subtree to its parent clique in terms of its contribution to the overall joint probability density function for the entire set of variables in the system.

Definition 5.3 *The* **cumulative potential function,** *written $\boldsymbol{\psi}_i$, of a clique C_i is the product of the initial potential functions assigned to the cliques in the subtree T_i with root C_i:*

$$\boldsymbol{\psi}_i = \prod_{C_j \in T_i} \psi_j$$

Theorem 5.8 *The multiplier M_i passed back by a clique C_i is equal to the sum of the cumulative potential function $\boldsymbol{\psi}_i$ over all values of the cumulative residual $\boldsymbol{R}_i$:*

$$M_i = \sum_{\boldsymbol{R}_i} \boldsymbol{\psi}_i$$

Proof. (Induction)

Base step: Tree with one level:

$$\begin{aligned}
M_i &= \sum_{R_i} \hat{\psi}_i \\
&= \sum_{R_i} \psi_i \\
&= \sum_{\boldsymbol{R}_i} \boldsymbol{\psi}_i
\end{aligned}$$

Inductive step: Tree with several levels:

Assume the theorem is true for level j, and let $i = j + 1$. We must show that the theorem then holds for level i.

Let C_i be any clique on level i, and let $C_{j_1}, C_{j_2}, \ldots, C_{j_n}$ be the children of C_i. Then,

$$\begin{aligned}
M_i &= \sum_{R_i} \hat{\psi}_i \\
&= \sum_{R_i} \psi_i M_{j_1} M_{j_2} \cdots M_{j_n} \\
&= \sum_{R_i} \left[\psi_i \left(\sum_{\boldsymbol{R}_{j_1}} \boldsymbol{\psi}_{j_1} \right) \cdots \left(\sum_{\boldsymbol{R}_{j_n}} \boldsymbol{\psi}_{j_n} \right) \right].
\end{aligned}$$

Note here that $\sum_{\boldsymbol{R}_{j_k}} \boldsymbol{\psi}_{j_k}$ is at most a probability function of variables in S_{j_k}, and $[x \in S_{j_k}] \Rightarrow [x \in C_i] \Rightarrow [x \notin \boldsymbol{R}_{j_m}]$, for any m. So we may write

$$M_i = \sum_{R_i} \left[\psi_i \sum_{\boldsymbol{R}_{j_1} \cdots \boldsymbol{R}_{j_n}} \boldsymbol{\psi}_{j_1} \cdots \boldsymbol{\psi}_{j_n} \right].$$

Also, ψ_i is a function of variables in C_i, and $[x \in C_i] \Rightarrow [x \notin \boldsymbol{R}_{j_m}]$, for any m. Thus,

$$\begin{aligned}
M_i &= \sum_{R_i} \left[\sum_{\boldsymbol{R}_{j_1} \cdots \boldsymbol{R}_{j_n}} \psi_i \boldsymbol{\psi}_{j_1} \cdots \boldsymbol{\psi}_{j_n} \right] \\
&= \sum_{\boldsymbol{R}_i} \boldsymbol{\psi}_i
\end{aligned}$$

□

Theorem 5.9 *ψ_i contains exactly one initial substantive probability function for each r in $\boldsymbol{R}_i$.*

Proof. By Theorem 5.6, we know there can be no more than one substantive probability function for each $r \in \boldsymbol{R}_i$ in ψ_i. To show that a such a function for each $r \in \boldsymbol{R}_i$ is indeed in ψ_i, we need only show that r is substantive in some ψ_j in T_i, the subtree rooted at C_i, since by definition, ψ_i is the product of all ψ_j in T_i.

Suppose $r \in \boldsymbol{R}_i$. Then $r \in R_j$ for some clique C_j in T_i. By Theorem 5.7, if r is not substantive in ψ_j, then it must be substantive in the potential function for some descendent clique of C_j. Since all descendents of C_j are also in T_i, then the substantive probability function for r must be present in ψ_i. □

5.3 Multipliers

In the following theorems, we use basic properties of probability density functions together with restrictions placed on the locations of certain substantive probability functions within the set of clique potential functions to derive a precise description of the conditions under which a multiplier composed only of ones will be passed from a clique back to a parent clique during the backward propagation phase of the Lauritzen - Spiegelhalter method.

Theorem 5.10 *If ψ_i is the product of only the initial substantive probability functions for $\boldsymbol{R}_i$, then $M_i = \mathbf{1}$.*

Proof. Suppose $\boldsymbol{R}_i = \{r_1, r_2, \ldots, r_n\}$, and that $r_1, r_2, \ldots, r_n$ are ordered such that

$$[\Pr(r_i|V) \in \{\text{ initial Spf terms }\}] \Rightarrow [V \subseteq \{r_1, \ldots, r_{i-1}\}].$$

This is always possible since there are no cycles in the original network.

Further suppose that $\psi_i = \mathrm{Spf}_1(r_1)\mathrm{Spf}_2(r_2)\ldots\mathrm{Spf}_n(r_n)$. Then

$$\begin{aligned}
M_i &= \sum_{\boldsymbol{R}_i} \psi_i \\
&= \sum_{r_1 \cdots r_n} \mathrm{Spf}_1(r_1)\mathrm{Spf}_2(r_2)\cdots\mathrm{Spf}_n(r_n) \\
&= \sum_{r_1}\left[\mathrm{Spf}_1(r_1) \sum_{r_2 \cdots r_n} \mathrm{Spf}_2(r_2)\cdots\mathrm{Spf}_n(r_n)\right] \\
&\ \ \vdots \\
&= \sum_{r_1}\left[\mathrm{Spf}_1(r_1)\sum_{r_2}\left[\mathrm{Spf}_2(r_2)\cdots\sum_{r_{n-1}}\left[\mathrm{Spf}_{n-1}(r_{n-1})\sum_{r_n}\mathrm{Spf}_n(r_n)\right]\cdots\right]\right] \\
&= \sum_{r_1}\left[\mathrm{Spf}_1(r_1)\sum_{r_2}\left[\mathrm{Spf}_2(r_2)\cdots\sum_{r_{n-1}}\left[\mathrm{Spf}_{n-1}(r_{n-1})\mathbf{1}\right]\cdots\right]\right] \\
&= \sum_{r_1}\left[\mathrm{Spf}_1(r_1)\sum_{r_2}\left[\mathrm{Spf}_2(r_2)\cdots\sum_{r_{n-2}}\left[\mathrm{Spf}_{n-2}(r_{n-2})\mathbf{1}\right]\cdots\right]\right] \\
&\ \ \vdots \\
&= \mathbf{1}
\end{aligned}$$

□

Theorem 5.11 *If x is substantive in ψ_i and $x \notin \boldsymbol{R}_i$, then $x \in S_i$.*

Proof.

$$
\begin{aligned}
[x \text{ is substantive in } \psi_i] &\Rightarrow [x \in C_j, \text{ for some } C_j \text{ in } T_i] \\
&\Rightarrow [x \in nodes(T_i)] \\
&\Rightarrow [x \in \boldsymbol{R}_i + S_i]\,, \text{ by Theorem 5.4.}
\end{aligned}
$$

But $x \notin \boldsymbol{R}_i$, so $x \in S_i$. □

We note here that the expression $A < \mathbf{1}$ indicates that there exists at least one entry of array A which is less than one.

Lemma 5.12 *Any substantive probability function for x takes on a value less than $\mathbf{1}$ for some replacement value of x.*

Proof. Suppose $\mathrm{Spf}(x)$ is a substantive probability function for x, and the replacement set for x is $\{a_1, a_2, \ldots, a_n\}$. Then, by Theorem 5.5,

$$
\begin{aligned}
\left[\sum_{x=a_1\ldots a_n} \mathrm{Spf}(x) = \mathbf{1}\right] &\Rightarrow \left[\mathrm{Spf}(a_1) + \sum_{x=a_2\ldots a_n} \mathrm{Spf}(x) = \mathbf{1}\right] \\
&\Rightarrow \left[\mathrm{Spf}(a_1) = \mathbf{1} - \sum_{x=a_2\ldots a_n} \mathrm{Spf}(x)\right]
\end{aligned}
$$

If $\mathrm{Spf}(a_1) < \mathbf{1}$ we are finished. If $\mathrm{Spf}(a_1) = \mathbf{1}$, then since Spf is a probability density function, we must have $\mathrm{Spf}(a_i) \geq \mathbf{0}$ for each a_i, and thus $\mathrm{Spf}(x) < \mathbf{1}$ for $x = a_2, \ldots, a_n$. □

Theorem 5.13 *Let $X = \{x | x$ is substantive in ψ_i and $x \notin \boldsymbol{R}_i\}$. If $X \neq \emptyset$, then $M_i < \mathbf{1}$.*

Proof. Suppose $X \neq \emptyset$. If we write $X = \{x_1, \ldots, x_n\}$ and $\boldsymbol{R}_i = \{r_1, \ldots, r_m\}$, then by Theorem 5.9,

$$\begin{aligned} M_i &= \sum_{\boldsymbol{R}_i} \psi_i \\ &= \sum_{\boldsymbol{R}_i} \mathrm{Spf}_{x_1}(x_1) \cdots \mathrm{Spf}_{x_n}(x_n) \mathrm{Spf}_{r_1}(r_1) \cdots \mathrm{Spf}_{r_m}(r_m), \end{aligned}$$

By lemma 5.12, there exists some replacement value $\overline{x}_1$ for x_1 such that $\mathrm{Spf}_{x_1}(\overline{x}_1) < \mathbf{1}$. Since it must be that $\mathrm{Spf}_h(h) \leq \mathbf{1}$ for $h \in \{x_2, \ldots, x_n\} \cup \boldsymbol{R}_i$, we have, for any sets of replacement values $\{\overline{x}_2, \ldots, \overline{x}_n\}$ and $\{\overline{r}_1, \ldots, \overline{r}_m\}$,

$$\mathrm{Spf}_{x_1}(\overline{x}_1) \cdots \mathrm{Spf}_{x_n}(\overline{x}_n) \mathrm{Spf}_{r_1}(\overline{r}_1) \cdots \mathrm{Spf}_{r_m}(\overline{r}_m) < 1.$$

But this is one of the entries in M_i. Thus, $M_i < \mathbf{1}$. □

Theorem 5.14 *The multiplier M_i passed back by a clique C_i is equal to $\mathbf{1}$ if and only if no variable in separator S_i is substantive in ψ_i:*

$$[M_i = \mathbf{1}] \Leftrightarrow [[x \text{ is substantive in } \psi_i] \Rightarrow [x \notin S_i]].$$

Proof. (IF:) Suppose that no variable in S_i is substantive in ψ_i. Then by Theorem 5.11, any variable that is substantive in ψ_i must be in $\boldsymbol{R}_i$. By Theorem 5.9, ψ_i contains a substantive probability function for each $r \in \boldsymbol{R}_i$, and thus by Theorem 5.10, $M_i = \mathbf{1}$.

(ONLY IF:) Suppose that some $x \in S_i$ is substantive in ψ_i. By Theorem 5.13, if x is substantive in ψ_i but $x \notin \boldsymbol{R}_i$, then $M_i \neq \mathbf{1}$. Since $S_i \cap \boldsymbol{R}_i = \emptyset$, we have $x \notin \boldsymbol{R}_i$ and thus $M_i \neq \mathbf{1}$. □

5.4 Independence in Forward Propagation

In contrast to the role of the multiplier M_i associated with clique C_i which embodies all of the probability information which must be communicated from subtree T_i rooted at clique C_i to the cliques outside of the subtree, the marginal probability density function of the nodes in C_i, $marginal(C_i)$, contains the probability information from outside that is required by the cliques inside subtree T_i. Information flows from the root of each subtree to its descendents during the forward propagation phase of the Lauritzen - Spiegelhalter method. As each clique C_i is processed by the forward propagation algorithm, its potential function is transformed into the marginal probability density function for the variables in C_i, after which probability information can be passed to each of the children of C_i in the clique tree.

In some cases, it may be that the potential function associated with a clique may be equal to the marginal probability density function for that clique even before its processing during the forward propagation phase. If this condition can be arranged or recognized, then the forward propagation phase can proceed from that clique immediately and in parallel with the backward propagation phase processing that may still be occurring in other parts of the clique tree.

The goal of the theorems in this section is to provide a precise description of the conditions under which the potential function at a clique will be equal to the marginal probability density function for the variables in that clique before the clique has been processed as part of the forward propagation phase of the Lauritzen - Spiegelhalter method.

Lemma 5.15 *The modified potential function $\hat{\psi}_i$ is equal to the sum of the cumulative potential function $\boldsymbol{\psi}_i$ over all values of the variables that are in the cumulative residual $\boldsymbol{R}_i$ but not in residual R_i:*

$$\hat{\psi}_i = \sum_{\boldsymbol{R}_i - R_i} \boldsymbol{\psi}_i.$$

Proof. Recall that by definition, the multiplier passed back from a clique is given by

$$M_i = \sum_{R_i} \hat{\psi}_i$$

and by Theorem 5.8,

$$M_i = \sum_{\boldsymbol{R}_i} \boldsymbol{\psi}_i$$

.

Thus,

$$\begin{aligned} \sum_{R_i} \hat{\psi}_i &= \sum_{\boldsymbol{R}_i} \boldsymbol{\psi}_i \\ &= \sum_{R_i} \sum_{\boldsymbol{R}_i - R_i} \boldsymbol{\psi}_i \end{aligned}$$

and since these sums are vectors which must match entry for entry, it must be that

$$\hat{\psi}_i = \sum_{\boldsymbol{R}_i - R_i} \boldsymbol{\psi}_i$$

. □

Theorem 5.16 *The modified potential function $\hat{\psi}_i$ is equal to the marginal probability density function of the variables in clique C_i if and only if every node in subtree T_i is substantive in the cumulative potential function ψ_i:*

$$\left[\hat{\psi}_i = \text{marginal}(C_i)\right] \Leftrightarrow [[x \in nodes(T_i)] \Rightarrow [x \text{ is substantive in } \psi_i]].$$

Proof. (IF:) Note that $nodes(T_i) = S_i \cup R_i \cup (\boldsymbol{R}_i - R_i)$. Assume that each $x \in nodes(T_i)$ is substantive in ψ_i. By Theorem 5.11, no other variables can be substantive in ψ_i. Also, according to the method used for the assignment of initial potential functions to cliques, no variable outside of $nodes(T_i)$ can appear in any probability function expression in ψ_i. Thus, ψ_i is a product expansion for the marginal probability of all $x \in nodes(T_i)$ in terms of the initial probability function expressions, and we have

$$\begin{aligned}\psi_i &= \Pr(nodes(T_i)) \\ &= \Pr(S_i, R_i, \boldsymbol{R}_i - R_i)\end{aligned}$$

and

$$\begin{aligned}\hat{\psi}_i &= \sum_{\boldsymbol{R}_i - R_i} \psi_i \\ &= \sum_{\boldsymbol{R}_i - R_i} \Pr(S_i, R_i, \boldsymbol{R}_i - R_i) \\ &= \Pr(S_i, R_i)\end{aligned}$$

which is $marginal(C_i)$.

(ONLY IF:) Given the assumptions and argument above, $\hat{\psi}_i$ is exactly $marginal(C_i)$. Since the representation of $marginal(C_i)$ as a product of initial substantive probability functions is unique, any deletions from the set of substantive variables in ψ_i also alters $\hat{\psi}_i$, and thus we would have $\hat{\psi}_i \neq marginal(C_i)$. □

Note that for $s \in S_i$, we must have s substantive in ψ_i in order for $\hat{\psi}_i$ to be equal to the marginal of clique C_i — it is not necessary that s be substantive in the initial potential function ψ_i.

5.5 Example

In this section, we will use the theorems developed earlier in this chapter to show that the amount of computation required to process the probabilities in the WCR example presented in Chapter 4 can be reduced significantly through careful initial potential function assignments to the cliques in the clique tree representation of the WCR network.

Recall from Chapter 4 that the initial potential function assignments to the cliques in the WCR clique graph were as follows:

$$\begin{aligned}
\psi_1 &= \Pr(Tc \mid SL, Te) \times \Pr(Te) \\
\psi_2 &= \Pr(SL \mid Lb, Te) \\
\psi_3 &= \Pr(Le \mid Lb, Te) \times \Pr(Lb) \\
\psi_4 &= \Pr(Lc \mid Lb, SL) \\
\psi_5 &= \Pr(Sc \mid Sb, SL) \times \Pr(Sb \mid SL) \\
\psi_6 &= \Pr(Ve \mid Le)
\end{aligned}$$

Assume that the clique tree representation chosen for the system is that shown in Figure 4.3(b). Figure 5.1 shows this tree with each clique node labeled with its variables from the original WCR network; separator variables are italicized for each clique. Also shown are the variables which are substantive in the potential function for each clique, given the initial potential function assignments above.

For this set of assignments, we have by Theorem 5.14 that the multipliers passed back by cliques C_4, C_5, and C_6 during the backward propagation phase of the Lauritzen - Spiegelhalter method are each equal to an array of all ones since none of these cliques has a separator variable which is substantive in its initial potential function. This implies that during backward propagation, these three leaf cliques can be processed in parallel, but clique C_2 must wait until C_3 has been processed since M_3 is not all ones, and similarly C_1 must wait for a multiplier from C_2.

Using Theorem 5.16, we see that no clique other than the root C_1 can have a modified potential function equal to the marginal probability density function of its variables because no other subtree T_i has all substantive probability function terms for the variables in the separator S_i assigned to potential functions of cliques at or below the root C_i. Thus, forward propagation must begin with clique C_1, proceed to clique C_2, then cliques C_3, C_4, and C_5 can be processed in parallel, and finally clique C_6 — a complete downward pass of the clique tree.

Suppose now that we instead make initial potential function assignments as follows:

$$\psi_1 \quad = \quad \Pr(Tc \mid SL, Te)$$

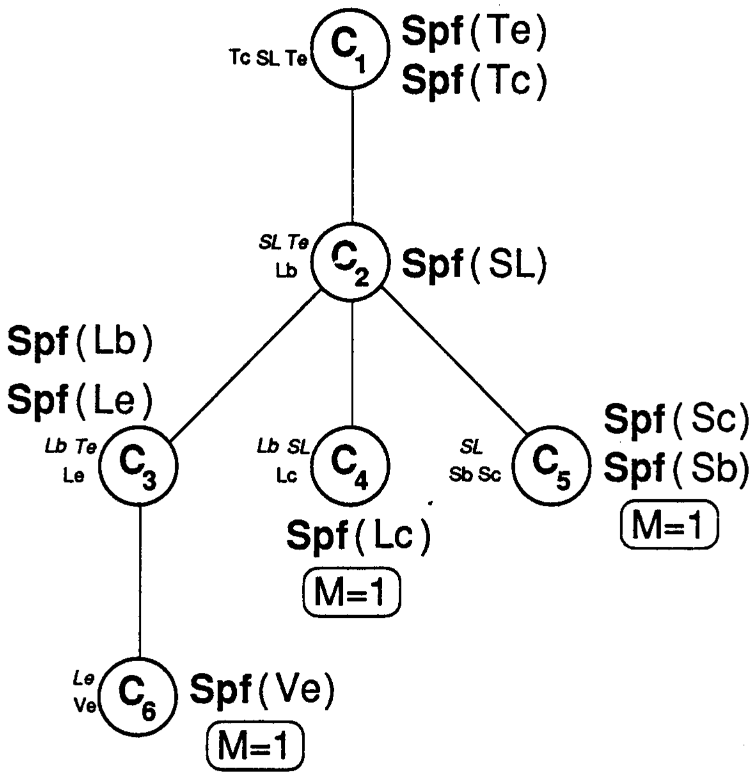


Figure 5.1: A WCR clique tree with initial potential function assignments.

$$\psi_2 = \Pr(SL \mid Lb, Te) \times \Pr(Lb) \times \Pr(Te)$$

$$\psi_3 = \Pr(Le \mid Lb, Te)$$

$$\psi_4 = \Pr(Lc \mid Lb, SL)$$

$$\psi_5 = \Pr(Sc \mid Sb, SL) \times \Pr(Sb \mid SL)$$

$$\psi_6 = \Pr(Ve \mid Le)$$

(See Figure 5.2.) Here, we have used the knowledge that each of the substantive terms Spf(Lb) and Spf(Te) contain only variables that appear in some separator, and thus there is more than one choice for the clique potential function to which each could be assigned. Furthermore, all substantive terms for variables in separator S_2 have been assigned to potential functions of cliques within subtree T_2 (so that the modified potential function $\hat{\psi}_2$ will equal the marginal probability density function of C_2), and we have avoided assigning to any other clique a substantive term for any of its separator variables (so that the multipliers passed back by these cliques will be all ones). Thus, backward propagation in this case involves solely the passing of multiplier M_2 from clique C_2 to clique C_1 since all multipliers below C_2 are ones, and forward propagation may start from C_2 rather than C_1 since C_2 has its marginal density function initially.

If movement of probability information along one arc of the clique tree during either the forward or backward propagation phases of the method is assumed to require one time unit, then we may summarize the time differences between the two sets of initial potential function assignments described above as shown in Table 5.1. In this table, arrows to the left represent backward

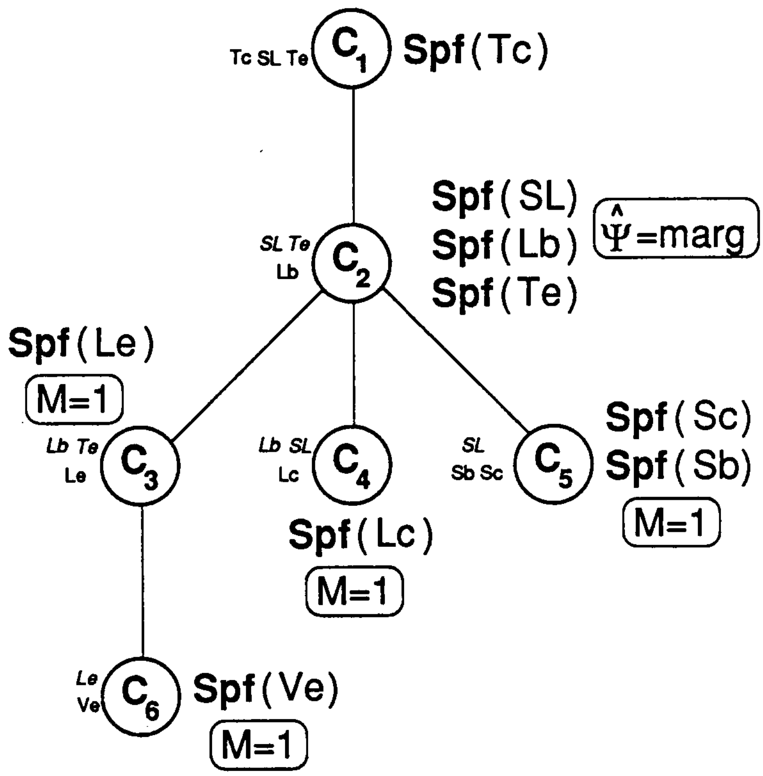


Figure 5.2: A WCR clique tree with better initial potential function assignments.

Time	*Assignment 1*	*Assignment 2*
1	$C_2 \leftarrow C_3$	$C_1 \leftarrow C_2 \rightarrow \{C_3, C_4, C_5\}$
2	$C_1 \leftarrow C_2$	$C_3 \rightarrow C_6$
3	$C_1 \rightarrow C_2$	
4	$C_2 \rightarrow \{C_3, C_4, C_5\}$	
5	$C_3 \rightarrow C_6$	
Total units used:	7	5

Table 5.1: Time units used by two sets of initial potential function assignments.

propagation operations, right arrows are forward propagation operations. The leftmost column represents clock time in units; operations that appear on the same line may be performed in parallel since the clique passing information during that operation has all of the information from other cliques that it requires.

CHAPTER 6

CONCLUDING REMARKS

6.1 Synopsis

New properties of clique graphs have been presented:

- Clique graphs can be tested for validity independently of a corresponding original graph.
- An exact number of arcs for a maximal spanning tree is present in each Clique in a clique graph.
- The information in a clique graph can be conveniently represented as a set of *exclusive intersections*, which partition the corresponding original graph.

An application in knowledge-based systems based on a clique graph model has been described. Based on original results, areas allowing independent computation in the application are determined.

6.2 Future Work

The results in Chapter 2 could be applied to the development of a parallel algorithm for finding maximal spanning trees in clique graphs. Rules for valid clique graph structures could be unified into a comprehensive test method that would rapidly determine the validity of an arbitrarily constructed clique graph.

In the development of the exclusive intersection set model for graphs, a well-defined clique template production algorithm would facilitate the generation of a complete catalog of possible clique structures so that pattern matching techniques could be used to map an application graph model onto an abstract graph with known properties. Precise descriptions of intersections of Clique templates, including the specification of the components which may be involved, have yet to be developed. These will enable the construction of general exclusive intersection templates for graph structures composed of several Cliques. The constructive technique of building clique graph templates from allowable exclusive intersection components can be combined with the component elimination technique implicit in the triangulated graph theorems of Chapter 2 to formulate a rapid classification scheme for clique graphs. Practical algorithms that process sets of nodes represented by exclusive intersection components could be developed for standard graphical model applications for which current algorithms process each node individually.

In the Lauritzen - Spiegelhalter method, use of an exclusive intersection clique component diagram to model the structure of the Markov network of probability variables would allow for more rapid updating of the network as new evidence becomes available and variables are added to or removed from the network. Areas of nondeterminism remaining in the method include the choice of arcs added to the initial Markov network model to triangulate the graph. Selections made here have a significant impact on the sizes of cliques in the graph, which affect the load balance for independent processes and the total amount of computation required. The choice of a clique tree representation can affect the applicability of the rules given in Chapter 5 for independent computations. Guidelines that direct the selection of a clique tree representation to maximize the possibility of independent computations based on the rules in Chapter 5 would be useful.

On the technical level, provisions could be made for the storage of clique marginal density functions separate from the clique potential functions within the clique tree structure to allow for more rapid and direct updating of probabilities. Storage-efficient handling of sparse conditional probability data tables within the model is another area for further research.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] S. K. Andersen, F. V. Jensen, K. G. Olesen, "The HUGIN core – preliminary considerations on systems for fast manipulation of probabilities," in *Proc. AI – Workshop on Inductive Reasoning*, Aalborg, Denmark, April 1987.

[2] S. Andreassen, F. V. Jensen, S. K. Andersen, B. Falck, U. Kjaerulff, M. Woldbye, A. R. Sorensen, A. Rosenfalck, F. Jensen, "MUNIN – an expert EMG assistant," Dept. of Medical Informatics and Image Analysis, Aalborg University, February 16, 1988.

[3] A. Barr, E. A. Feigenbaum, *Handbook of Artificial Intelligence*, 1 and 2, Kaufmann, Los Altos, 1981.

[4] C. Beeri, R. Fagin, D. Maier, M. Yannakakis, "On the desirability of acyclic data base schemes," *Journal of the ACM*, vol. 30, no. 3, pp. 479-513, July 1983.

[5] C. Berge, *Graphs and Hypergraphs*, North-Holland, Amsterdam, 1973.

[6] F. Berman, "On mapping parallel algorithms into parallel architectures," *Journal of Parallel and Distributed Computing*, vol. 4, pp.439-458, 1987.

[7] J. R. S. Blair, R. E. England, M. G. Thomason, "Cliques and their separators in triangulated graphs," Technical Report No. CS-73-88,Dept. of Computer Science, Univ. of Tennessee, Knoxville TN, August 1988.

[8] J. R. S. Blair, S. S. Ravi, "On clique graphs of undirected graphs," SUNY at Albany Technical Report, Albany NY, 1988.

[9] P. L. Bogler, "Shafer-Dempster reasoning with applications to multisensor target identification systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-17, no. 6, pp. 968-977, November/December 1987.

[10] O. B. Brown, P. C. Cornillon, S. R. Emmerson, H. M. Carle, "Gulf Stream warm rings: a statistical study of their behavior," *Deep Sea Research*, vol. 33, nos. 11/12, pp.1459-1473, 1986.

[11] P. Chaudhuri, "Fast parallel graph searching with applications," BIT 28, pp. 2-18, 1988.

[12] P. Cheeseman, "A method of computing generalized Bayesian probability values for expert systems," in *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pp. 198-202, Karlsruhe, 1983.

[13] G. F. Cooper, "Expert systems based on belief networks – current research directions," Stanford Univ. Memo, Stanford CA, August 1987.

[14] A. Dawish, A. K. Jain, "A rule based approach for visual pattern recognition," *IEEE Trans. PAMI*, vol. PAMI-10, pp. 56-68, 1988.

[15] R. O. Duda, P. E. Hart, N. J. Nilsson, "Subjective Bayesian methods for rule-based inference systems," *Proceedings for 1976 National Computer Conference*, vol. 45, pp. 1075-1082, Arlington VA, 1976.

[16] H. Farreny, H. Prade, "Default and inexact reasoning with possibility degrees," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-16, no. 2, pp. 270-276, March/April 1986.

[17] F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph," *SIAM Journal of Computing*, vol. 1, no. 2, pp. 180-187, June 1972.

[18] F. Gavril, "Algorithms on clique separable graphs," *Discrete Mathematics*, vol. 19, pp. 159-165, 1977.

[19] M. R. Genesereth, N. J. Nilsson, "Reasoning with uncertain beliefs," in *Logical Foundations of Artificial Intelligence*, pp. 177-206, Morgan Kaufmann Publishers, Los Altos CA, 1987.

[20] J. R. Gilbert, D. J. Rose, A. Edenbrandt, "A separator theorem for chordal graphs," *SIAM Jour. Alg. Disc. Meth.*, vol. 5, no. 3, pp. 306-313, September 1984.

[21] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, London, 1980.

[22] M. C. Golumbic, R. E. Jamison, "Edge and vertex intersection of paths in a tree," *Discrete Mathematics*, vol. 55, pp. 151-159, 1985.

[23] A. Gupta, C. Forgy, A. Newell, R. Wedig, "Parallel algorithms and architectures for rule-based systems," *Proc. 13th Annual Int'l Sym. Comp. Arch.*, pp.28-37, Tokyo, Japan, June 2-5, 1986.

[24] C. W. Ho, R. C. T. Lee, "Efficient parallel algorithms for finding maximal cliques, clique trees, and minimum coloring on chordal graphs," *Information Processing Letters*, vol. 28, pp. 301-309, Elsevier Science Publishers B. V. (North-Holland), 1988.

[25] J. E. Hopcroft, R. E. Tarjan, "Dividing a graph into triconnected components," *SIAM Journal of Computing*, vol. 2, no. 3, pp. 135-158, September 1973.

[26] E. J. Horvitz, J. S. Breese, M. Henrion, "Decision theory in expert systems and artificial intelligence," *Journal of Approximate Reasoning*, July 1988.

[27] R. C. Hughes, J. N. Maksym, "Acoustic signal interpretation: reasoning with non-specific and uncertain information," *Pattern Recognition*, vol. 18, no. 6, pp. 475-483, 1985.

[28] J. Ihara, "Extension of conditional probability and measures of belief and disbelief in a hypothesis based on uncertain evidence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, pp. 561-568, July 1987.

[29] F. V. Jensen, "Junction trees and decomposable hypergraphs," JUDEX Technical Report, Aalborg Denmark, 1988.

[30] L. N. Kanal, J. F. Lemmer, eds., *Uncertainty in Artificial Intelligence*, North-Holland (Elsevier), Amsterdam, 1986.

[31] J. Kittler, "Relaxation labelling," in *Pattern Recognition Theory and Applications*, ed. P. A. Devijver and J. Kittler, pp. 99-108, Springer-Verlag, Berlin, 1987.

[32] S. L. Lauritzen, D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Soc.*, vol. 50, no. 2, pp. 157-224, 1988.

[33] J. G. Lewis, B. W. Peyton, "A fast implementation of the Jess and Kees algorithm," *Talk given at ORNL*, June 1988.

[34] R. J. Lipton, R. E. Tarjan, "A separator theorem for planar graphs," *SIAM Jour. Appl. Math.*, vol. 36, no. 2, pp. 177-189, April 1979.

[35] D. V. Lindley, "The probability approach to the treatment of uncertainty in artificial intelligence and expert systems", *Statistical Science*, vol. 3, pp. 17-24, 1987.

[36] R. J. Lipton, R. E. Tarjan, "Applications of a planar separator theorem," *SIAM Jour. Computing*, vol. 9, no. 3, pp. 615-627, August 1980.

[37] J. W. H. Liu, "Equivalent sparse matrix reordering by elimination tree rotations," York Univ. Technical Report No. CS-86-12, Dept. of Computer Science, North York, Ontario, Canada, November 1986.

[38] J. W. H. Liu, "A graph partitioning algorithm by node separators," York Univ. Technical Report No. CS-88-01, Dept. of Computer Science, North York, Ontario, Canada, February 1988.

[39] J. W. H. Liu, "The role of elimination trees in sparse factorization," York Univ. Technical Report No. CS-87-12, Dept. of Computer Science, North York, Ontario, Canada, October 1987.

[40] J. W. H. Liu, A. Mirzaian, "A linear reordering algorithm for parallel pivoting of chordal graphs," York Univ. Technical Report No. CS-87-02, Dept. of Computer Science, North York, Ontario, Canada, October 1987.

[41] J. Naor, M. Naor, A. A. Schaffer, "Fast parallel algorithms for chordal graphs," *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pp. 355-364, 1987.

[42] J. Pearl, "Distributed revision of composite beliefs," *Artificial Intelligence*, vol. 33, pp. 173-215, 1987.

[43] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artificial Intelligence*, vol. 29, pp. 241-288, 1986.

[44] J. Pearl, "On evidential reasoning in a hierarchy of hypotheses," *Artificial Intelligence*, vol. 28, pp. 9-15, 1986.

[45] J. Pearl, "Reverend Bayes on inference engines: a distributed hierarchical approach," *Proc. AAAI National Conference on AI*, pp. 133-136, Pittsburgh, 1982.

[46] B. Peyton, "Some applications of clique trees to the solution of sparse linear systems," PhD Thesis, Clemson Univ., Clemson SC, 1986.

[47] H. Prade, "A computational approach to approximate and plausible reasoning with applications to expert systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 3, pp. 260-283, May 1985.

[48] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, pp. 1389-1401, 1957.

[49] H. E. Rauch, "Probability concepts for an expert system used for data fusion," *The AI Magazine*, pp. 55-60, Fall 1984.

[50] A. Rege, A. M. Agogino, "Topological framework for representing and solving probabilistic inference problems in expert systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, pp. 402-414, May/June 1988.

[51] P. L. Richardson, "Gulf Stream rings," in *Eddies in Marine Science*, ed. A. R. Robinson, pp. 19-64, Springer-Verlag, Berlin, 1983.

[52] D. J. Rose, "A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations," in *Graph Theory and Computing*, ed. R. Read, pp. 183-217, Academic Press, New York, 1973.

[53] D. J. Rose, "Triangulated graphs and the elimination process," *Journal of Mathematical Analysis and Applications*, vol. 32, pp. 597-609, 1970.

[54] D. J. Rose, R. E. Tarjan, G. S. Lueker, "Algorithmic aspects of vertex elimination on graphs," *SIAM Jour. of Computing*, vol. 5, no. 2, pp. 266-283, June 1976.

[55] R. D. Shachter, "Evaluating influence diagrams," *Operations Research*, vol. 34, no. 6, pp. 871-882, November/December 1986.

[56] G. Shafer, "Probability judgement in artificial intelligence and expert systems," *Statistical Science*, vol. 2, no. 1, pp. 3-44, 1987.

[57] G. Shafer, P. P. Shenoy, "Bayesian and belief-function propagation," Univ. of Kansas Working Paper, Laurence KS, April 1988.

[58] P. P. Shenoy, G. Shafer, "Propagating belief functions with local computations," *IEEE Expert*, Fall 1986.

[59] D. J. Spiegelhalter, "A statistical view of uncertainty in expert systems," in *Artificial Intelligence and Statistics*, ed. W. A. Gale, pp. 17-55, Addison-Wesley, New York, 1986.

[60] D. J. Spiegelhalter, "Coherent evidence propagation in expert systems," *Statistician*, vol. 36, pp. 201-210, 1987.

[61] R. E. Tarjan, "Decomposition by clique separators," *Discrete Mathematics*, vol. 55, no. 2, pp. 221-232, October 1985.

[62] R. E. Tarjan, M. Yannakakis, "Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs," *SIAM Journal of Computing*, vol. 13, no. 3, pp. 566-579, August 1984.

[63] M. G. Thomason, "Structural methods in pattern analysis," in *NATO Advanced Study Institute*, ed. J. Kittler, F, vol. 30, pp. 307-321, Springer-Verlag, Berlin, 1987.

[64] M. G. Thomason, "Knowledge-based analysis of satellite oceanographic images," *Int'l Jour. Intell. Sys.*, to appear.

[65] M. G. Thomason, R. E. Blake, "Development of an expert system for interpretation of oceanographic images," NORDA Technical Report, June 1986.

[66] M. G. Thomason, G. S. Cole, "Parallel computation in dynamic programming for optimal string-to-network alignments," Tech. Report CS-88-74, Computer Science Dept., Univ. of Tennessee, August, 1988.

[67] A. Y. Wu, "Embedding of tree networks into hypercubes," *Journal of Parallel and Distributed Computing*, vol. 2, pp. 238-249, 1985.

[68] M. Yannakakis, "Computing the minimum fill-in is NP-complete," *SIAM Journal of Alg. Desc. Meth.*, vol. 2, no. 1, pp. 77-79, March 1981.

[69] Z. Zhang, M. Simaan, "A rule-based interpretation system for segmentation of seismic images," *Jour. Pat. Recog.*, vol. 20, pp. 45-53, 1987.

VITA

Robert E. England was born in Wynne, Arkansas on February 25, 1957. He attended elementary school in Mammoth Spring, Arkansas, secondary school in Henderson, Tennessee, and graduated from Chester County High School in June 1975. The following September he entered Southwestern At Memphis, and in June 1979 he received a Bachelor of Arts degree in Mathematics.

After teaching high school mathematics for three years in Violet Hill, Arkansas, he accepted a teaching assistantship at The University of Tennessee, Knoxville in September 1982. He was awarded the Master of Science degree with a major in Computer Science in June 1985.

For two years he worked as a computer research scientist at the Chr. Michelsen Institute in Bergen, Norway, after which he accepted a research assistantship at The University of Tennessee, Knoxville in September 1987. He received the Doctor of Philosophy degree with a major in Computer Science in December 1989.

The author is a member of Phi Beta Kappa.