CHAPTER 4

Triangulated Graphs

1. Introduction

One of the first classes of graphs to be recognized as being perfect was the class of triangulated graphs. Hajnal and Surányi [1958] showed that triangulated graphs satisfy the *perfect property* P_2 (α-perfection), and Berge [1960] proved that they satisfy P_1 (χ-perfection). These two results, in large measure, inspired the conjecture that P_1 and P_2 were equivalent, a statement that we now know to be true (Theorem 3.3). Thus, the study of triangulated graphs can well be thought of as the beginning of the theory of perfect graphs.

We briefly looked at the triangulated graph property in the sneak preview Section 1.3. For completeness' sake, we shall repeat the definition here and mention a few basic properties.

An undirected graph G is called *triangulated* if every cycle of length strictly greater than 3 possesses a chord, that is, an edge joining two nonconsecutive vertices of the cycle. Equivalently, G does not contain an induced subgraph isomorphic to C_n for $n > 3$. Being triangulated is a hereditary property inherited by all the induced subgraphs of G. You may recall from Section 1.3 that the interval graphs constitute a special type of triangulated graph. Thus we have our first example of triangulated graphs.

In the literature, triangulated graphs have also been called *chordal, rigid-circuit, monotone transitive,* and *perfect elimination* graphs.

2. Characterizing Triangulated Graphs

A vertex x of G is called *simplicial* if its adjacency set $\mathrm{Adj}(x)$ induces a complete subgraph of G, i.e., $\mathrm{Adj}(x)$ is a clique (not necessarily maximal).

Dirac [1961], and later Lekkerkerker and Boland [1962], proved that a triangulated graph always has a simplicial vertex (in fact at least two of them), and using this fact Fulkerson and Gross [1965] suggested an iterative procedure to recognize triangulated graphs based on this and the hereditary property. Namely, *repeatedly locate a simplicial vertex and eliminate it from the graph, until either no vertices remain and the graph is triangulated or at some stage no simplicial vertex exists and the graph is not triangulated.* The correctness of this procedure is proved in Theorem 4.1. Let us state things more algebraically.

Let $G = (V, E)$ be an undirected graph and let $\sigma = [v_1, v_2, \ldots, v_n]$ be an ordering of the vertices. We say that σ is a *perfect vertex elimination scheme* (or *perfect scheme*) if each v_i is a simplicial vertex of the induced subgraph $G_{\{v_i, \ldots, v_n\}}$. In other words, each set

$$X_i = \{v_j \in \mathrm{Adj}(v_i) \mid j > i\}$$

is complete. For example, the graph G_1 in Figure 4.1 has a perfect vertex elimination scheme $\sigma = [a, g, b, f, c, e, d]$. It is not unique; in fact G_1 has 96 different perfect elimination schemes. In contrast to this, the graph G_2 has no simplicial vertex, so we cannot even start constructing a perfect scheme—it has none.

A subset $S \subset V$ is a *vertex separator* for nonadjacent vertices a and b (or an *a–b separator*) if the removal of S from the graph separates a and b into distinct connected components. If no proper subset of S is an a–b separator, then S is a *minimal vertex separator* for a and b. Consider again the graphs of Figure 4.1. In G_2, the set $\{y, z\}$ is a minimal vertex separator for p and q, whereas $\{x, y, z\}$ is a minimal vertex separator for p and r. (How is it possible that both are *minimal* vertex separators, yet one is contained in the other?) In G_1, every minimal vertex separator has cardinality 2. This is an unusual phenomenon. However, notice also that the two vertices of such a separator of G_1 are adjacent, in every case. This latter phenomenon actually occurs for all triangulated graphs, as you will see in Theorem 4.1.

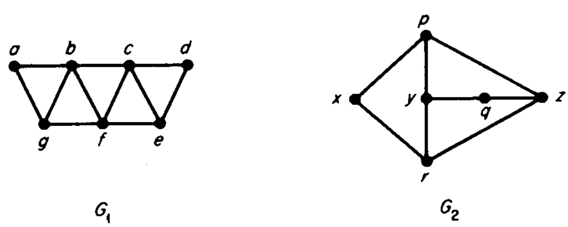


Figure 4.1. Two graphs, one triangulated and one not triangulated.

We now give two characterizations of triangulated graphs, one algorithmic (Fulkerson and Gross [1965]) and the other graph theoretic (Dirac [1961]).

Theorem 4.1. Let G be an undirected graph. The following statements are equivalent:

(i) G is triangulated.
(ii) G has a perfect vertex elimination scheme. Moreover, *any* simplicial vertex can start a perfect scheme.
(iii) Every minimal vertex separator induces a complete subgraph of G.

Proof. (iii) $\Rightarrow$ (i) Let $[a, x, b, y_1, y_2, \ldots, y_k, a]$ $(k \geq 1)$ be a simple cycle of $G = (V, E)$. Any minimal a–b separator must contain vertices x and y_i for some i, so $xy_i \in E$, which is a chord of the cycle.

(i) $\Rightarrow$ (iii) Suppose S is a minimal a–b separator with G_A and G_B being the connected components of G_{V-S} containing a and b, respectively. Since S is minimal, each $x \in S$ is adjacent to some vertex in A and some vertex in B. Therefore, for any pair $x, y \in S$ there exist paths $[x, a_1, \ldots, a_r, y]$ and $[y, b_1, \ldots, b_t, x]$, where each $a_i \in A$ and $b_i \in B$, such that these paths are chosen to be of smallest possible length. It follows that $[x, a_1, \ldots, a_r, y, b_1, \ldots, b_t, x]$ is a simple cycle whose length is at least 4, implying that it must have a chord. But $a_ib_j \notin E$ by the definition of vertex separator, and $a_ia_j \notin E$ and $b_ib_j \notin E$ by the minimality of r and t. Thus, the only possible chord is $xy \in E$. ▮

Remark. It also follows that $r = t = 1$, implying that for all $x, y \in S$ there exist vertices in A and B which are adjacent to both x and y. A stronger result is given in Exercise 12.

Before continuing with the remaining implications, we pause for a message from our lemma department.

Lemma 4.2 (Dirac [1961]). Every triangulated graph $G = (V, E)$ has a simplicial vertex. Moreover, if G is not a clique, then it has two nonadjacent simplicial vertices.

Proof. The lemma is trivial if G is complete. Assume that G has two nonadjacent vertices a and b and that the lemma is true for all graphs with fewer vertices than G. Let S be a minimal vertex separator for a and b with G_A and G_B being the connected components of G_{V-S} containing a and b, respectively.

By induction, either the subgraph G_{A+S} has two nonadjacent simplicial vertices one of which must be in A (since S induces a complete subgraph) or G_{A+S} is itself complete and any vertex of A is simplicial in G_{A+S}. Furthermore, since $\mathrm{Adj}(A) \subseteq A + S$, a simplicial vertex of G_{A+S} in A is simplicial in all of G. Similarly B contains a simplicial vertex of G. This proves the lemma.

We now rejoin the proof of the theorem which is still in progress.

(i) ⇒ (ii) According to the lemma, if G is triangulated, then it has a simplicial vertex, say x. Since $G_{V-\{x\}}$ is triangulated and smaller than G, it has, by induction, a perfect scheme which, when adjoined as a suffix of x, forms a perfect scheme for G.

(ii) ⇒ (i) Let C be a simple cycle of G and let x be the vertex of C with the smallest index in a perfect scheme. Since $|\mathrm{Adj}(x) \cap C| \geq 2$, the eventual simpliciality of x guarantees a chord in C. ▮

3. Recognizing Triangulated Graphs by Lexicographic Breadth-First Search

From Lemma 4.2 we learned that the Fulkerson–Gross recognition procedure affords us a choice of at least two vertices for each position in constructing a perfect scheme for a triangulated graph. Therefore, we can freely choose a vertex v_n to *avoid* during the whole process, saving it for the last position in a scheme. Similarly, we can pick any vertex v_{n-1} adjacent to v_n to save for the $(n-1)$st position. If we continued in this manner, we would be constructing a scheme *backwards*! This is exactly what Leuker [1974] and Rose and Tarjan [1975] have done in order to give a linear-time algorithm for recognizing triangulated graphs. The version presented in Rose, Tarjan, and Leuker [1976] uses a *lexicographic* breadth-first search in which the usual queue of vertices is replaced by a queue of (unordered) subsets of the vertices which is sometimes refined but never reordered. The method (Figure 4.2) is as follows:

```
   begin
1.   assign the label ∅ to each vertex;
2.   for i ← n to 1 step − 1 do
3.      select: pick an unnumbered vertex v with largest label;
4.         σ(i) ← v; comment This assigns to v the number i.
5.      update: for each unnumbered vertex w ∈ Adj(v) do add i to label(w);
   end
```

Figure 4.2. Algorithm 4.1: Lex BFS.

Algorithm 4.1. Lexicographic breadth-first search.

Input: The adjacency sets of an undirected graph $G = (V, E)$.
Output: An ordering σ of the vertices.
Method: The vertices are numbered from n to 1 in the order that they are selected in line 3. This numbering fixes the positions of an elimination scheme

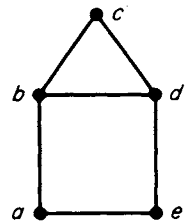


Figure 4.3.

σ. For each vertex x, the *label* of x will consist of a set of numbers listed in decreasing order. The vertices can then be lexicographically ordered according to their labels. (Lexicographic order is just dictionary order, so that $9761 < 985$ and $643 < 6432$.) Ties are broken arbitrarily.

Example. We shall apply Algorithm 4.1 to the graph in Figure 4.3. The vertex a is selected arbitrarily in line 3 during the first pass. The evolution of the labeling and the numbering are illustrated in Figure 4.4. Notice that the final numbering $\sigma = [c, d, e, b, a]$ is a perfect vertex elimination scheme. This is no accident.

For each value of i, let $L_i(x)$ denote the label of x when statement 4 is executed, i.e., when the ith vertex is numbered. Remember, the index is *decremented* at each successive iteration. For example, $L_n(x) = \varnothing$ for all x and $L_{n-1}(x) = \{n\}$ iff $x \in \mathrm{Adj}(\sigma(n))$. The following properties are of prime importance:

(L1) $L_i(x) \leq L_j(x) \qquad (j \leq i)$;
(L2) $L_i(x) < L_i(y) \Rightarrow L_j(x) < L_j(y) \qquad (j < i)$;
(L3) if $\sigma^{-1}(a) < \sigma^{-1}(b) < \sigma^{-1}(c)$ and $c \in \mathrm{Adj}(a) - \mathrm{Adj}(b)$, then there exists a vertex $d \in \mathrm{Adj}(b) - \mathrm{Adj}(a)$ with $\sigma^{-1}(c) < \sigma^{-1}(d)$.

	label	number	label	number	label	number	label	number	label	number	label	number
a	ϕ	–	ϕ	5	ϕ	5	ϕ	5	ϕ	5	ϕ	5
b	ϕ	–	{5}	–	{5}	4	{5}	4	{5}	4	{5}	4
c	ϕ	–	ϕ	–	{4}	–	{4}	–	{4,2}	–	{4,2}	1
d	ϕ	–	ϕ	–	{4}	–	{4,3}	–	{4,3}	2	{4,3}	2
e	ϕ	–	{5}	–	{5,4}	–	{5,4}	3	{5,4}	3	{5,4}	3

Figure 4.4.

Property (L1) says that the label of a vertex may get larger but never smaller as the algorithm proceeds. Property (L2) states that once a vertex gets ahead of another vertex, they stay in that order. Finally, (L3) gives a condition under which there must be a suitable vertex d which was numbered before c (in time) and hence received a larger number.

Lexicographic breadth-first search can be used to recognize triangulated graphs as demonstrated by the next theorem.

Theorem 4.3. An undirected graph $G = (V, E)$ is triangulated if and only if the ordering σ produced by Algorithm 4.1 is a perfect vertex elimination scheme.

Proof. If $|V| = n = 1$, then the proof is trivial. Assume that the theorem is true for all graphs with fewer than n vertices and let σ be the ordering produced by Algorithm 4.1 when applied to a triangulated graph G. By induction, it is sufficient to show that $x = \sigma(1)$ is a simplicial vertex of G.

Suppose x is not simplicial. Choose vertices $x_1, x_2 \in \text{Adj}(x)$ with $x_1 x_2 \notin E$ so that x_2 is as large as possible (with respect to the ordering σ). (Remember, σ increases as you approach the root of the search tree.) Consider the following inductive procedure. Assume we are given vertices $x_1, x_2, \ldots, x_m$ with these properties: for all $i, j > 0$,

(1) $x, x_i \in E \Leftrightarrow i \leq 2$,
(2) $x_i x_j \in E \Leftrightarrow |i - j| = 2$,
(3) $\sigma^{-1}(x_1) < \sigma^{-1}(x_2) < \cdots < \sigma^{-1}(x_m)$,
(4) x_j is the largest vertex (with respect to σ) such that

$$x_{j-2} x_j \in E \qquad \text{but} \qquad x_{j-3} x_j \notin E.$$

(For notational reasons let $x_0 = x$ and $x_{-1} = x_1$.) The situation for $m = 2$ was constructed initially.

The vertices x_{m-2}, x_{m-1}, and x_m satisfy the hypothesis of property (L3) as a, b, and c, respectively. Hence, choose x_{m+1} to be the largest vertex (with respect to σ) larger than x_m which is adjacent to x_{m-1} but not adjacent to x_{m-2}. Now, if x_{m+1} were adjacent to x_{m-3}, then (L3) applied to the vertices x_{m-3}, x_{m-2}, x_{m+1} would imply the existence of a vertex larger than x_{m+1} (hence larger than x_m) which is adjacent to x_{m-2} but not to x_{m-3}, contradicting the maximality of x_m in (4). Therefore x_{m+1} is not adjacent to x_{m-3}. Finally, it follows from (1), (2), and chordality that $x_i x_{m+1} \notin E$ for $i = 0, 1, \ldots, m-4, m$.

Clearly this inductive procedure continues indefinitely, but the graph is finite, a contradiction. Therefore, the vertex x must be simplicial, and the theorem is proved in one direction. The converse follows from Theorem 4.1. ∎

In an unpublished work, Tarjan [1976] has shown another method of searching a graph that can be used to recognize triangulated graphs. It is called *maximum cardinality search* (MCS), and it is defined as follows:

MCS: The vertices are to be numbered from n to 1.
The next vertex to be numbered is always one which is adjacent to the most numbered vertices, ties being broken arbitrarily.

Using an argument similar to the proof of Theorem 4.3, one can show that G is triangulated if and only if every MCS ordering of the vertices is a perfect ellimination scheme. It should be pointed out that there are MCS orderings which cannot be obtained by Lex BFS, there are Lex BFS orderings which are not MCS, and there exist perfect elimination schemes which are neither MCS nor Lex BFS. Exercises 27 and 28 develop some of the results on MCS. Both Lex BFS and MCS are special cases of a general method for finding perfect elimination schemes recently developed by Alan Hoffman and Michel Sakarovich.

4. The Complexity of Recognizing Triangulated Graphs

Having proved the correctness of Algorithm 4.1, let us now analyze its complexity. We first describe an implementation of Lex BFS, then show that it requires $O(|V| + |E|)$ time. We do not actually calculate the labels, but rather we keep the unnumbered vertices in lexicographic order.

Data Structure

We use a queue Q of sets

$$S_l = \{v \in V \mid \text{label}(v) = l \text{ and } \sigma^{-1}(v) \text{ undefined}\}$$

ordered lexicographically from smallest to largest; each set S_l is represented by a doubly linked list. Initially there is but one set, $S_\phi = V$. Each set S_l has a FLAG initially set at 0. For a vertex w, the array element SET(w) points to $S_{\text{label}(w)}$ and another array gives the address of w in SET(w) for deletion purposes. A list FIX LIST, initially empty, is also used, and simple arrays represent σ and σ^{-1}.

Implementation

Select as v in line 3 any vertex in the last set of Q and delete v from SET(v). Create a new set $S_{l \cdot i}$ for each old set S_l containing an unnumbered vertex

$w \in \mathrm{Adj}(v)$. We delete from S_l all such vertices w and place them in the new set $S_{l \cdot i}$, which is inserted into the queue of sets immediately following S_l. Clearly this method maintains the proper lexicographic ordering without our actually having to calculate the labels. More specifically, *update* can be implemented as follows:

```
5. { for all unnumbered w ∈ Adj(v) do
       begin
         if FLAG(SET(w)) = 0 then
           begin
             Create new set S′ and insert it in Q immediately in back of SET(w);
             FLAG(SET(w)) ← 1; FLAG(S′) ← 0; put a pointer to SET (w) on FIX LIST;
           end
         let S′ be the set immediately in back of SET(w) in Q; delete w from SET(w); add w to S′;
         SET(w) ← S′;
       end
     for each set S on FIX LIST do
       begin
         FLAG(S) ← 0;
         if S is empty then
           delete S from Q;
       end
```

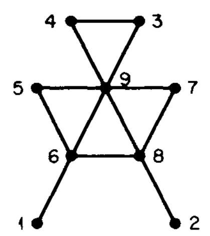
It is easy to verify that, as presented, statement 5 requires $O(|\mathrm{Adj}(v)|)$ time. Consequently, the **for** loop between statements 2 and 5 uses $O(|V| + |E|)$ time. Initializing the data structure including statement 1 takes $O(|V|)$ time. This proves the following result.

Theorem 4.4. Algorithm 4.1 can be implemented to carry out lexicographic breadth-first search on an undirected graph $G = (V, E)$ in $O(|V| + |E|)$ time and space.

Example. Let Q_i denote the queue of sets of unnumbered vertices just before $\sigma(i)$ is defined in Algorithm 4.1. Figure 4.5b gives Q_9, Q_8, and Q_7 for the graph in Figure 4.5a. For convenience, the vertices are identified with their eventual position in σ. Figure 4.5c shows the data structure for Q_7 before the FIX LIST has been emptied and with the implicit labels in parentheses.

In order to use Lex BFS to recognize triangulated graphs, we need an efficient method to test whether or not a given ordering σ of the vertices is a perfect vertex elimination scheme. This is proved by the next algorithm.

Algorithm 4.2. Testing a perfect elimination scheme.
Input. The adjacency sets of an undirected graph $G = (V, E)$ and an ordering σ of V.



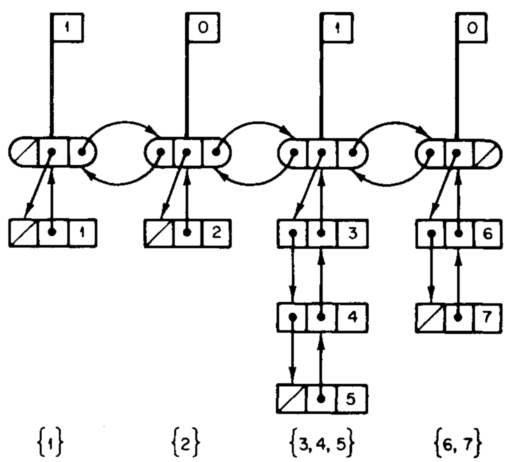
(a)

$$Q_9 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$Q_8 = \{1, 2\} < \{3, 4, 5, 6, 7, 8\}$$

$$Q_7 = \{1\} < \{2\} < \{3, 4, 5\} < \{6, 7\}$$

(b)



(c)

Figure 4.5.

Output. "True" if σ is a perfect vertex elimination scheme and "false" otherwise.

Method. A single call to the procedure PERFECT(σ), given in Figure 4.6. The list $A(u)$ collects all the vertices which will eventually have to be checked for adjacency with u. The actual checking is delayed until the iteration when $u = \sigma(i)$ in lines 8 and 9. This technique is used so that in the $\sigma^{-1}(v)$-th iteration there is no search of Adj(u).

Complexity. Arrays are used for σ and σ^{-1} and lists hold Adj(v) and $A(v)$. Lines 4–7 can be implemented simultaneously in one scan of Adj(v). The **go to** in line 5 will be executed exactly $j - 1$ times, where j is the number of connected components of G. The list $A(u)$ will represent a *set with repetitions.* The test in line 8 simply checks for a vertex w on the list $A(v)$ which is *not*

```
    boolean procedure PERFECT (σ):
    begin
1.    for all vertices v do A(v) ←∅;;
2.    for i ← 1 to n − 1 do
        begin
3.        v ← σ(i);
4.        X ← {x ∈ Adj(v) | σ⁻¹(v) < σ⁻¹(x)};
5.        if X = ∅ then go to 8;;
6.        u ← σ (min {σ⁻¹(x) | x ∈ X});
7.        concatenate X − {u} to A(u);
8.        if A(v) − Adj(v) ≠ ∅ then
9.          return "false";
        end
10.   return "true";
    end
```

Figure 4.6. Procedure to test a perfect vertex elimination scheme.

adjacent to v, can be done in $O(|\mathrm{Adj}(v)| + |\mathrm{A}(v)|)$ time by using an array TEST of size n initially set to all zeros as follows:

```
   ⎧ begin
   ⎪   for w ∈ Adj(v) do TEST(w) ← 1;;
   ⎪   for w ∈ A(v) do
8. ⎨     if TEST(w) = 0 then
   ⎪       return "nonempty";
   ⎪   for w ∈ Adj(v) do TEST(w) ← 0;;
   ⎪   return "empty";
   ⎩ end
```

Thus, the entire algorithm can be performed in time and space proportional to

$$|V| + \sum_{v \in V} |\mathrm{Adj}(v)| + \sum_{u \in V} |A(u)|,$$

where has $A(u)$ is its final value. Now, the middle summand is larger than the last since a given $\mathrm{Adj}(v)$ appears as part of at most one of the lists $A(u)$. Hence, both summands can be replaced by $O(|E|)$. This proves the complexity part of the next theorem.

Theorem 4.5. Algorithm 4.2 correctly tests whether or not an ordering σ of the vertices is a perfect vertex elimination scheme. It can be implemented to run in time and space proportional to $|V| + |E|$.

Proof. The algorithm returns "false" during the $\sigma^{-1}(u)$-th iteration if and only if there exist vertices v, u, w $(\sigma^{-1}(v) < \sigma^{-1}(u) < \sigma^{-1}(w))$, where u is defined in line 4 during the $\sigma^{-1}(v)$-th iteration, and

$$u, w \in \mathrm{Adj}(v) \qquad \text{but } u \text{ is not adjacent to } w.$$

Clearly, if we get "false," then σ is not a perfect elimination scheme.

Conversely, suppose σ is not perfect elimination and the algorithm returns "true." Let v be the vertex with $\sigma^{-1}(v)$ largest possible such that $X = \{w \mid w \in \mathrm{Adj}(v) \text{ and } \sigma^{-1}(v) < \sigma^{-1}(w)\}$ is *not* complete. Let u be the vertex of X defined in line 6 during the $\sigma^{-1}(v)$-th iteration, after which (in line 7) $X - \{u\}$ is added to $A(u)$. Since during the $\sigma^{-1}(u)$-th iteration line 9 is not executed,

$$\text{every } x \in X - \{u\} \text{ is adjacent to } u,$$

and

$$\text{every pair } x, y \in X - \{u\} \text{ is adjacent.}$$

The latter statement follows from the maximality of $\sigma^{-1}(v)$. Thus, X is complete, a contradiction. ∎

Corollary 4.6. Triangulated graphs can be recognized in linear time.

Proof. The proof follows from Theorems 4.3–4.5. ∎

5. Triangulated Graphs as Intersection Graphs

We have seen in Chapter 1 that the interval graphs are a proper subclass of the triangulated graphs. This leads naturally to the problem of characterizing triangulated graphs as the intersection graphs of some topological family slightly more general than intervals on a line. In this section we shall show that a graph is triangulated if and only if it is the intersection graph of a family of subtrees of a tree. (See Figure 4.7.)

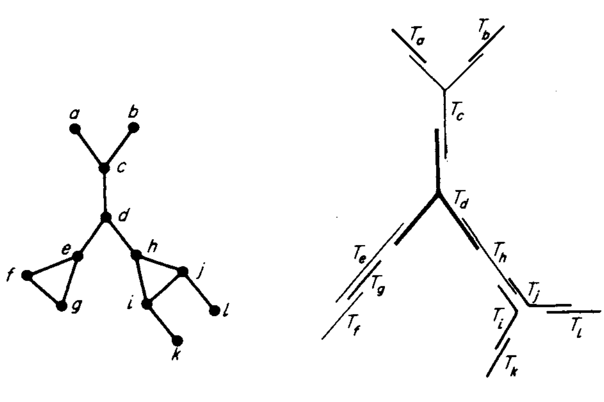


Figure 4.7. A triangulated graph and a subtree representation for it.

A family $\{T_i\}_{i \in I}$ of subsets of a set T is said to satisfy the *Helly property* if $J \subseteq I$ and $T_i \cap T_j \neq \emptyset$ for all $i, j \in J$ implies that $\bigcap_{j \in J} T_j \neq \emptyset$.

If we let T be a tree and let each T_i be a subtree of T, then we can prove the following result.

Proposition 4.7. A family of subtrees of a tree satisfies the Helly property.

Proof. Suppose $T_i \cap T_j \neq \emptyset$ for all $i, j \in J$. Consider three points a, b, c on T. Let S be the set of indices s such that T_s contains at least two of these three points, and let P_1, P_2, P_3 be the simple paths in T connecting a with b, b with c, and a with c, respectively. Since T is a tree, it follows that $P_1 \cap P_2 \cap P_3 \neq \emptyset$, but each $T_s (s \in S)$ contains one of these paths P_i. Therefore,

$$\bigcap_{s \in S} T_s \supseteq P_1 \cap P_2 \cap P_3 \neq \emptyset.$$

The lemma is proved by induction. Let us assume that

$$[T_i \cap T_j \neq \emptyset \qquad \text{for all} \quad i, j \in J] \Rightarrow \bigcap_{j \in J} T_j \neq \emptyset \tag{1}$$

for all index sets J of size $\leq k$. This is certainly true for $k = 2$. Consider a family of subtrees $\{T_{i_1}, \ldots, T_{i_{k+1}}\}$. By the induction hypothesis there exist points a, b, c on T such that

$$a \in \bigcap_{j=1}^{k} T_{i_j}, \qquad b \in \bigcap_{j=2}^{k+1} T_{i_j}, \qquad c \in T_{i_1} \cap T_{i_{k+1}}.$$

Moreover, every T_{i_j} contains at least two of the points a, b, c. Hence, by the preceeding paragraph, $\bigcap_{j=1}^{k+1} T_{i_j} \neq \emptyset$. ∎

Theorem 4.8 (Walter [1972], Gavril [1974a], and Buneman [1974]). Let $G = (V, E)$ be an undirected graph. The following statements are equivalent:

(i) G is a triangulated graph.

(ii) G is the intersection graph of a family of subtrees of a tree.

(iii) There exists a tree $T = (\mathscr{K}, \mathscr{E})$ whose vertex set $\mathscr{K}$ is the set of maximal cliques of G such that each of the induced subgraphs $T_{\mathscr{K}_v} (v \in V)$ is connected (and hence a subtree), where $\mathscr{K}_v$ consists of those maximal cliques which contain v.

Proof. (iii) $\Rightarrow$ (ii) Assume that there exists a tree $T = (\mathscr{K}, \mathscr{E})$ satisfying statement (iii). Let $v, w \in V$. Now

$$vw \in E, \quad v, w \in A \qquad \text{for some clique} \quad A \in \mathscr{K},$$

$$\mathscr{K}_v \cap \mathscr{K}_w \neq \emptyset, \qquad T_{\mathscr{K}_v} \cap T_{\mathscr{K}_w} \neq \emptyset.$$

Thus G is the intersection graph of the family of subtrees $\{T_{\mathscr{K}_v} \mid v \in V\}$ of T.

(ii) $\Rightarrow$ (i) Let $\{T_v\}_{v \in V}$ be a family of subtrees of a tree T such that $vw \in E$ iff $T_v \cap T_w \neq \emptyset$.

Suppose G contains a chordless cycle $[v_0, v_1, \ldots, v_{k-1}, v_0]$ with $k > 3$ corresponding to the sequence of subtrees $T_0, T_1, \ldots, T_{k-1}, T_0$ of the tree T; that is, $T_i \cap T_j \neq \emptyset$ if and only if i and j differ by at most one modulo k. All arithmetic will be done mod k.

Choose a point a_i from $T_i \cap T_{i+1}$ $(i = 0, \ldots, k-1)$. Let b_i be the last common point on the (unique) simple paths from a_i to a_{i-1} and a_i to a_{i+1}. These paths lie in T_i and T_{i+1}, respectively, so that b_i also lies in $T_i \cap T_{i+1}$. Let P_{i+1} be the simple path connecting b_i and b_{i+1}. Clearly $P_i \subseteq T_i$, so $P_i \cap P_j = \emptyset$ for i and j differing by more than 1 mod k. Moreover, $P_i \cap P_{i+1} = \{b_i\}$ for $i = 0, \ldots, k-1$. Thus, $\bigcup_i P_i$ is a simple cycle in T, contradicting the definition of a tree.

(i) $\Rightarrow$ (iii) We prove the implication by induction on the size of G. Assume that the theorem is true for all graphs having fewer vertices than G. If G is complete, then T is a single vertex and the result is trivial. If G is disconnected with components $G_1, \ldots, G_k$, then by induction there exists a corresponding tree T_i satisfying (iii) for each G_i. We connect a point of T_i with a point of T_{i+1} $(i = 1, \ldots, k-1)$ to obtain a tree satisfying (iii) for G.

Let us assume that G is connected but not complete. Choose a simplicial vertex a of G and let $A = \{a\} \cup \mathrm{Adj}(a)$. Clearly, A is a maximal clique of G. Let

$$U = \{u \in A \mid \mathrm{Adj}(u) \subset A\}$$

and

$$Y = A - U.$$

Note that the sets U, Y, and $V - A$ are nonempty since G is connected but not complete. Consider the induced subgraph $G' = G_{V-U}$, which is triangulated and has fewer vertices than G. By induction, let T' be a tree whose vertex set K' is the set of maximal cliques of G' such that for each vertex $v \in V - U$ the set $K'_v = \{X \in K' \mid v \in X\}$ induces a connected subgraph (subtree) of T'.

Remark. Either $K = K' + \{A\} - \{Y\}$ or $K = K' + \{A\}$ depending upon whether or not Y is a maximal clique of G'.

Let B be a maximal clique of G' containing Y.

Case 1. If $B = Y$, then we obtain T from T' by renaming B, A.

Case 2. If $B \neq Y$, then we obtain T from T' by connecting the new vertex A to B.

In either case, $K_u = \{A\}$ for all u in U and $K_v = K'_v$ for all v in $V - A$, each of which induces a subtree of T. We need only worry about the sets K_y $(y \in Y)$.

In case 1, $K_y = K'_y + \{A\} - \{B\}$, which induces the same subtree as K'_y since only names were changed. In case 2, $K_y = K'_y + \{A\}$, which clearly induces a subtree.

Thus, we have constructed the required tree T and the proof of the theorem is complete. ∎

Buneman [1972, 1974] discusses the application of the subtree intersection model in constructing evolutionary trees and in certain other classificatory problems.

An undirected graph $G = (V, E)$ is called a *path graph* if it is the intersection graph of a family of paths in a tree. Renz [1970] showed that G is a path graph if and only if G is triangulated and G is the intersection graph of a family $\mathscr{F}$ of paths in an undirected graph such that $\mathscr{F}$ satisfies the Helly property. Gavril [1978] presented an efficient algorithm for recognizing path graphs; he also proved a theorem for path graphs analogous to the equivalence of (ii) and (iii) in Theorem 4.8 (see Exercise 26).

6. Triangulated Graphs Are Perfect

Occasionally, the minimum graph coloring problem and the maximum clique problem can be simplified using the *principle of separation into pieces* (Berge [1973, p. 329]). This method is described in the following theorem and its proof. In particular, it is applicable to triangulated graphs.

Theorem 4.9. Let S be a vertex separator of a connected undirected graph $G = (V, E)$, and let $G_{A_1}, G_{A_2}, \ldots, G_{A_t}$ be the connected components of G_{V-S}. If S is a clique (not necessarily maximal), then

$$\chi(G) = \max_i \chi(G_{S+A_i})$$

and

$$\omega(G) = \max_i \omega(G_{S+A_i}).$$

Proof. Clearly $\chi(G) \geq \chi(G_{S+A_i})$ for each i, so $\chi(G) \geq k = \max_i \chi(G_{S+A_i})$. In fact, G can be colored using exactly k colors. First color G_S, then independently extend the coloring to each *piece* G_{S+A_i}. This composite will be a coloring of G. Thus, $\chi(G) = k$.

Next, certainly $\omega(G) \geq \omega(G_{S+A_i})$ for each i, so $\omega(G) \geq \max_i \omega(G_{S+A_i}) = m$. Let X be a maximum clique of G, i.e., $|X| = \omega(G)$. It is impossible that two vertices of X lie in G_{A_i} and G_{A_j} $(i \neq j)$ since the vertices are connected. Thus, X lies wholly in one of the pieces, say G_{S+A_r}. Hence, $m \geq \omega(G_{S+A_r}) \geq |X| = \omega(G)$.Therefore, $\omega(G) = m$. ▮

Corollary 4.10. Let S be a separating set of a connected undirected graph $G = (V, E)$, and let $G_{A_1}, G_{A_2}, \ldots, G_{A_t}$ be the connected components of G_{V-S}. If S is a clique, and if each subgraph G_{S+A_i} is perfect, then G is perfect.

Proof. Assume that the result is true for all graphs with fewer vertices than G. It suffices to show that $\chi(G) = \omega(G)$. Using Theorem 4.9 and the fact that each graph G_{S+A_i} is perfect, we have

$$\chi(G) = \max_i \chi(G_{S+A_i}) = \max_i \omega(G_{S+A_i}) = \omega(G). \quad ▮$$

We are now ready to state the main result.

Theorem 4.11 (Berge [1960], Hajnal and Surányi [1958]). Every triangulated graph is perfect.

Proof. Let G be a triangulated graph, and assume that the theorem is true for all graphs having fewer vertices than G. We may assume that G is connected, for otherwise we consider each component individually. If G is complete, then G is certainly perfect. If G is not complete, then let S be a minimal vertex separator for some pair of nonadjacent vertices. By Theorem 4.1, S is a clique. Moreover, by the induction hypothesis, each of the (triangulated) subgraphs G_{S+A_i}, as defined in Corollary 4.10, is perfect. Thus, by Corollary 4.10, G is perfect. ▮

Remark. The proofs in this section used only the perfect graph property (P_1) (Berge [1960]). Historically, however, until Theorem 3.3 was proved, the arguments had to be carried out for property (P_2) as well (Hajnal and Surányi [1958]).

Let $\mathscr{G}$ denote the class of all undirected graphs satisfying the property that every odd cycle of length greater than or equal to 5 has at least two chords. Clearly, every triangulated graph is in $\mathscr{G}$. Our ultimate goal in the remainder of this section is to prove that the graphs in $\mathscr{G}$ are perfect. The technique used to show this will be constructive in the following sense: Given a k-coloring of a graph $G \in \mathscr{G}$, we will show how to reduce it into an ω-coloring of G, where $k \geq \omega = \omega(G)$, by performing a sequence of color interchanges called *switchings*.

Let G be an undirected graph which has been properly colored. An (α, β)-*chain* in G is a chain whose vertices alternate between the colors α and β. Let

$G_{\alpha\beta}$ denote the subgraph induced by the vertices of G which are colored α or β. An $\langle\alpha, \beta\rangle$ *switch with respect to* G consists of the following operation:

Either interchange the colors in a nontrivial connected component of $G_{\alpha\beta}$ and leave all other colors unchanged, or recolor all isolated vertices of $G_{\alpha\beta}$ using β and leave all other colors unchanged.

Note that the result of an $\langle\alpha, \beta\rangle$ switch with respect to G is again a proper coloring of G.

Lemma 4.12. Let $G \in \mathscr{G}$ be properly colored, and let x be any vertex of G. Let vertices $y, z \in \mathrm{Adj}(x)$ be colored α and β, respectively, with $\alpha \neq \beta$. If y and z are linked by an (α, β)-chain in G, then they are linked by an (α, β)-chain in $G_{\mathrm{Adj}(x)}$.

Proof. Let $\mu = [y = x_0, x_1, x_2, \ldots, x_l = z]$ be an $[\alpha, \beta]$ chain in G of minimum length between y and z. Clearly, l must be odd. We claim that $\{x_0, x_1, x_2, \ldots, x_l\} \subseteq \mathrm{Adj}(x)$.

The claim is certainly true if $l = 1$. Let us assume that $l \geq 3$ and that the claim is true for all minimum (α, β)-chains of odd length strictly less than l. Now, the cycle $\bar{\mu} = [x, x_0, x_1, \ldots, x_l, x]$ has odd length $l + 2 \geq 5$, and all of its chords must have x as an endpoint since a chord between an α vertex and a β vertex of μ would give a shorter chain. Therefore, every subchain $\mu[x_s, x_t] = [x_s, \ldots, x_t]$ of μ is a minimum (α, β)-chain, and since $G \in \mathscr{G}$ the cycle $\bar{\mu}$ has at least two chords, xx_i and xx_j $(i < j)$.

If $\mu[x_0, x_i]$, $\mu[x_i, x_j]$, and $\mu[x_j, x_l]$ all have odd length, then applying the induction hypothesis to each of them we obtain $\{x_0, x_1, \ldots, x_l\} \subseteq \mathrm{Adj}(x)$. Otherwise, at least one of $\mu[x_0, x_i]$ or $\mu[x_j, x_l]$ has even length. Without loss of generality, assume that $\mu[x_0, x_i]$ has even length so that $\mu[x_i, x_l]$ has odd length. By induction, $\{x_i, x_{i+1}, \ldots, x_l\} \subseteq \mathrm{Adj}(x)$. In particular, $x_{i+1} \in \mathrm{Adj}(x)$, so $\mu[x_0, x_{i+1}]$ has odd length and by induction $\{x_0, x_1, \ldots, x_{i+1}\} \subseteq \mathrm{Adj}(x)$. This proves the claim. ∎

Let $G' = G_{\mathrm{Adj}(x)}$. Lemma 4.12 says that a nontrivial connected component of $G_{\alpha\beta}$ contains only one nontrivial connected component of $G'_{\alpha\beta}$ or only isolated α vertices of $G'_{\alpha\beta}$ or only isolated β vertices of $G'_{\alpha\beta}$.

Lemma 4.13. Let f be a proper coloring of a graph $G \in \mathscr{G}$, and let x be a vertex of G colored γ. Let $f_{G'}$ be the restriction of f to the subgraph G' induced by those vertices adjacent to x whose colors are from some arbitrary subset Q of colors with $\gamma \notin Q$. If $f_{G'}$ can be transformed into a coloring g' of G' by a sequence of switchings with respect to G' (using colors from Q), then f can be transformed into a coloring f' of G by a sequence of switchings with respect to G such that $f'_{G'} = g'$.

Proof. It is sufficient to consider the case of a single $\langle\alpha, \beta\rangle$ switch with respect to G', where $\alpha, \beta \neq \gamma$. Suppose that a connected component $H'_{\alpha\beta}$ of $G'_{\alpha\beta}$ was switched. If $H'_{\alpha\beta}$ is nontrivial, then by Lemma 4.12 the same result could be obtained by switching the component of $G_{\alpha\beta}$, containing $H'_{\alpha\beta}$. If $H'_{\alpha\beta}$ has only one vertex, then all isolated vertices of $G'_{\alpha\beta}$ were switched to β. In this case the same result could be obtained by switching all nontrivial components of $G_{\alpha\beta}$ which contain isolated α vertices of $G'_{\alpha\beta}$ plus switching all isolated vertices of $G_{\alpha\beta}$ to β. ∎

Theorem 4.14 (Meyniel [1976]). Let $G \in \mathscr{G}$ and let f be a k-coloring of G. Then there exists a q-coloring g of G with $q = \chi(G)$ which is obtainable from f by a sequence of switchings with respect to G.

Proof. The theorem is obviously true for graphs with one vertex. Assume that the theorem is true for all graphs with fewer vertices than G.

Consider a k-coloring f of G using the colors $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ with $k > q = \chi(G)$. Choose a vertex x with color $\alpha \neq \alpha_1, \alpha_2, \ldots, \alpha_q$; if there is none, the proof is finished. Let G' be the subgraph induced by the vertices colored $\alpha_1, \alpha_2, \ldots, \alpha_q$ and adjacent to x. Clearly,

$$q' = \chi(G') \leq \chi(G_{\mathrm{Adj}(x)}) \leq q - 1.$$

Since $G' \in \mathscr{G}$, the induction hypothesis implies that there exists a q'-coloring g' of G' which is obtainable from $f_{G'}$ by a sequence of switchings with respect to G'. By Lemma 4.13, g' can also be obtained from f by a sequence of switchings with respect to G. After performing this sequence of switchings, we can recolor x with one of the colors $\alpha_1, \alpha_2, \ldots, \alpha_q$ which is unused by g' (since $q' \leq q - 1$). Thus, we have enlarged the set of vertices colored $\alpha_1, \alpha_2, \ldots, \alpha_q$. Repeating this process until all vertices of G are colored $\alpha_1, \alpha_2, \ldots, \alpha_q$ will yield a minimum coloring. ∎

We are now ready to show that the graphs in $\mathscr{G}$ are perfect. Gallai [1962] originally proved the case where each odd cycle has two noncrossing chords; a shorter proof appeared in Surányi [1968]. The case where each odd cycle has two crossing chords was proved by Olaru [1969] (see Sachs [1970]). The general case, as presented here, is due to Meyniel [1976].

Theorem 4.15. If G is an undirected graph such that every odd cycle has two chords, then G is perfect.

Proof. Let $G \in \mathscr{G}$ with $\chi(G) = q$, and let H be an induced subgraph of G satisfying

$$\chi(H) = q,$$
$$\chi(H - x) = q - 1 \qquad \text{for every vertex } x \text{ of } H.$$

Choose a vertex x of H and a $(q - 1)$-coloring f of $H - x$, and let H' be the subgraph induced by $\mathrm{Adj}_H(x)$. If H' were $(q - 2)$-colorable, then by Theorem 4.14 f restricted to H' could be transformed into a $(q - 2)$-coloring of H' by a sequence of switchings with respect to H'. Then by Lemma 4.13 there would exist a $(q - 1)$-coloring of $H - x$ using $q - 2$ colors for $\mathrm{Adj}_H(x)$. But this would imply that $\chi(H) = q - 1$, a contradiction.

Therefore, $\{x\} \cup \mathrm{Adj}_H(x)$ is not $(q - 1)$-colorable, and hence it must be the entire vertex set of H. Since this argument holds for all x, it follows that H is a q-clique. Thus, $\chi(G) = \omega(G) = q$. In like manner, $\chi(G') = \omega(G')$ for all induced subgraphs G' of G since being in $\mathscr{G}$ is a hereditary property. Thus G is perfect. ∎

7. Fast Algorithms for the COLORING, CLIQUE, STABLE SET, and CLIQUE-COVER Problems on Triangulated Graphs

Let $G = (V, E)$ be a triangulated graph, and let σ be a perfect elimination scheme for G. It was first pointed out by Fulkerson and Gross [1965] that every maximal clique was of the form $\{v\} \cup X_v$ where

$$X_v = \{x \in \mathrm{Adj}(v) \mid \sigma^{-1}(v) < \sigma^{-1}(x)\}.$$

This elementary fact is easily shown. By the definition of σ, each $\{v\} \cup X_v$ is complete. Let w be the first vertex in σ contained in an arbitrary maximal clique A; then $A = \{w\} \cup X_w$. Therefore, we have the following result.

Proposition 4.16 (Fulkerson and Gross [1965]). A triangulated graph on n vertices has at most n maximal cliques, with equality if and only if the graph has no edges.

It is easy enough to modify Algorithm 4.2 to print out each set $\{v\} \cup X_v$. However, some of these will not be maximal, and we would like to filter them out. The mechanism that we employ is the observation that $\{u\} \cup X_u$ is *not* maximal iff for some i, in line 7 of Algorithm 4.2, X_u is concatenated to $A(u)$ (Exercise 13). The modified algorithm is as follows:

Algorithm 4.3. Chromatic number and maximal cliques of a trangulated graph.

Input: The adjacency sets of a triangulated graph G and a perfect elimination scheme σ.
Output: All maximal cliques of G and the chromatic number $\chi(G)$.

Method: A single call to the procedure CLIQUES(σ) given in Figure 4.8. The number $S(v)$ indicates the size of the largest set that would have been concatenated to $A(v)$ in Algorithm 4.2. A careful comparison will reveal that Algorithm 4.3 is a modification of Algorithm 4.2.

Theorem 4.17. Algorithm 4.3 correctly calculates the chromatic number and all maximal cliques of a triangulated graph $G = (V, E)$ in $O(|V| + |E|)$ time.

The proof is similar to that of Theorem 4.5.

Next we tackle the problem of finding the stability number $\alpha(G)$ of a triangulated graph. Better yet, since G is perfect, let us demand that we produce both a stable set and clique cover of size $\alpha(G)$. A solution is given by Gavril.

Let σ be a perfect elimination scheme for $G = (V, E)$. We define inductively a sequence of vertices $y_1, y_2, \ldots, y_t$ in the following manner: $y_1 = \sigma(1)$; y_i is the first vertex in σ which follows y_{i-1} and which is not in $X_{y_1} \cup X_{y_2} \cup \cdots \cup X_{y_{i-1}}$; all vertices following y_t are in $X_{y_1} \cup \cdots \cup X_{y_t}$. Hence

$$V = \{y_1, y_2, \ldots, y_t\} \cup X_{y_1} \cup \cdots \cup X_{y_t}.$$

The following theorem applies.

Theorem 4.18 (Gavril [1972]). The set $\{y_1, y_2, \ldots, y_t\}$ is a maximum stable set of G, and the collection of sets $Y_i = \{y_i\} \cup X_{y_i}$ $(i = 1, 2, \ldots, t)$ comprises a minimum clique cover of G.

```
     procedure CLIQUES (σ):
     begin
1.     χ ← 1;
2.     for all vertices v do S(v) ← 0;;
3.     for i ← 1 to n do
         begin
4.         v ← σ(i);
5.         X ← {x ∈ Adj(v) | σ⁻¹(v) < σ⁻¹(x)};
6.         if Adj(v) = ∅ then print {v};;
7.         if X = ∅ then go to 13;;
8.         u ← σ(min{σ⁻¹(x) | x ∈ X});
9.         S(u) ← max{S(u), |X| − 1};
10.        if S(v) < |X| then do
             begin
11.            print {v} ∪ X;
12.            χ = max{χ, 1+ |X|};
             end
13.      end
14.    print "The chromatic number is",χ;
     end
```

Figure 4.8. Procedure to list all maximal cliques of a triangulated graph, given a perfect elimination scheme.

Proof. The set $\{y_1, y_2, \ldots, y_t\}$ is stable since if $y_j y_i \in E$ for $j < i$, then $y_i \in X_{y_j}$, which cannot be. Thus $\alpha(G) \geq t$. On the other hand, each of the sets $Y_i = \{y_i\} \cup X_{y_i}$ is a clique, and so $\{Y_1, \ldots, Y_t\}$ is a clique cover of G. Thus, $\alpha(G) = k(G) = t$, and we have produced the desired maximum stable set and minimum clique cover. ∎

Implementing this procedure to run efficiently is a straightforward exercise and is left for the reader (Exercise 25). For a treatment of the maximum weighted stable set problem, see Frank [1976].

EXERCISES

1. Show that for $n \geq 5$ the graph $\bar{C}_n$ is not triangulated.
2. Using Theorem 4.1, condition (iii), prove that every interval graph is triangulated. What is the interpretation of a separator in an interval representation of a graph?
3. Prove properties (L1)–(L3) of lexicographic breadth-first search (Section 4.3).
4. Apply Algorithm 4.1 to the graph in Figure 3.3 by arbitrarily selecting the vertex of degree 2 in line 3 during the first pass of the algorithm. (i) What is the perfect scheme you get? (ii) Find a perfect scheme of G which cannot possibly arise from Algorithm 4.1.

The class of undirected graphs known as *k-trees* is defined recursively as follows: A k-tree on k vertices consists of a clique on k vertices (k-clique); given any k-tree T_n on n vertices, we construct a k-tree on $n + 1$ vertices by adjoining a new vertex x_{n+1} to T_n, which is made adjacent to each vertex of some k-clique of T_n and nonadjacent to the remaining $n - k$ vertices. Notice that a 1-tree is just a tree in the usual sense, and that a k-tree has at least k vertices. Exercises 5–7 below are due to Rose [1974]. Harary and Palmer [1968] discuss 2-trees.
5. Show that a k-tree has a perfect vertex elimination scheme and is therefore triangulated. Give an example of a triangulated graph which is not a k-tree for any k.
6. Prove the following result: An undirected graph $G = (V, E)$ is a k tree if and only if

(i) G is connected,
(ii) G has a k-clique but no $(k + 2)$-clique, and
(iii) every minimal vertex separator of G is a k-clique.

7. Let $G = (V, E)$ be a triangulated graph which has a k-clique but no $(k + 2)$-clique. Prove that $\|E\| \leq k|V| - \frac{1}{2}k(k + 1)$ with equality holding if and only if G is a k-tree.

8. Show that every 3-tree is planar.

9. Let G be an undirected graph and let H be constructed as follows. The vertices of H correspond to the edges of G, and two vertices of H are adjacent if their corresponding edges form two sides of a triangle in G. Prove that G is a 2-tree if and only if H is a cactus of triangles.

10. Show that every vertex of a minimal x–y separator is adjacent to some vertex in each of the connected components containing x and y, respectively.

11. Let S be a minimal x–y separator of a connected graph G. Show that every path in G from x to y contains a member of S and that every $s \in S$ is contained in some path μ from x to y which involves no other element of S, that is, $\mu \cap S = \{s\}$.

12. Prove the following: For any minimal vertex separator S of a triangulated graph $G = (V, E)$, there exists a vertex c in each connected component of G_{V-S} such that $S \subseteq \mathrm{Adj}(c)$. (Hint: Prove the inclusion for each subset $X \subseteq S$ using induction.)

13. Program Algorithms 4.1 and 4.2 using the data structures suggested and test some graphs for the triangulated graph property.

14. Give a representation of the graph in Figure 4.5a as intersecting subtrees of a tree.

15. Prove that G is triangulated if and only if G is the intersection graph of a family $\mathscr{T}$ of subtrees of a tree where no member of $\mathscr{T}$ contains another member of $\mathscr{T}$ (Gavril [1974a]).

16. Give an algorithm which constructs for any triangulated graph G a collection of subtrees of a tree whose intersection graph is isomorphic to G.

17. Prove the following: H is a tree if and only if every family of paths in H satisfies the Helly property.

18. Prove the following theorem of Renz [1970]: G is the intersection graph of a family of paths in a tree iff G is triangulated and is the intersection graph of a family of arcs of a graph satisfying the Helly property.

19. Using the Helly property for subtrees of a tree, show directly that (ii) implies (iii) in Theorem 4.8. (Hint: for each clique A of the intersection graph, paint the subtree corresponding to the intersection of all members of A red and paint the remainder of the tree green. What does it look like when you collapse each red piece to a point?)

20. Prove Corollary 4.10 using the perfect graph property (P_2) instead of (P_1).

21. The line graph $L(G)$ of G is defined to be the undirected graph whose vertices correspond to the edges of G, and two vertices of $L(G)$ are joined by an edge if and only if they correspond to adjacent edges in G. Prove that G is triangulated if and only if $L(G)$ is triangulated.

22. Prove that Algorithm 4.3 correctly calculates the chromatic number and all maximal cliques of a triangulated graph.

23. Let σ be a perfect vertex elimination scheme for a triangulated graph $G = (V, E)$. Let $H = (V, F)$ be an orientation of G, where $xy \in F$ iff $\sigma^{-1}(x) < \sigma^{-1}(y)$. Show that H is acyclic. Let τ be any topological sorting of H. Show that τ is also a perfect elimination scheme for G.

24. Prove that a height function h (see Chapter 2, Exercise 8) of the acyclic oriented graph H defined in the preceding exercise is a minimum coloring of the triangulated graph G. Thus, a triangulated graph can be colored with a minimum number of colors in time proportional to its size.

25. Modify Algorithm 4.3 so that, in addition, it prints out a maximum stable set and prints an asterisk next to those cliques which together comprise a minimum clique cover.

26. Prove the following: $G = (V, E)$ is a path graph if and only if there exists a tree T whose vertex set is $\mathscr{K}$ (the maximal cliques of G) such that for all $v \in V$, the induced subgraph $T_{\mathscr{K}_v}$ is a path in T. ($\mathscr{K}_v$ denotes the set of maximal cliques which contain v.) (Gavril [1978].)

27. Let $G = (V, E)$ be an undirected graph, and let $\sigma = [v_1, v_2, \ldots, v_n]$ be an ordering of V. Consider the following property:

(T): If $\sigma^{-1}(u) < \sigma^{-1}(v) < \sigma^{-1}(w)$ and $w \in \mathrm{Adj}(u) - \mathrm{Adj}(v)$, then there exists an x such that $\sigma^{-1}(v) < \sigma^{-1}(x)$ and $x \in \mathrm{Adj}(v) - \mathrm{Adj}(w)$.

Prove that if G is a triangulated graph and σ satisfies (T), then σ is a perfect elimination scheme for G (Tarjan [1976]).)

28. (i) Prove that any MCS order, as defined at the end of Section 4.3, satisfies property (T) from the preceding exercise.

(ii) Give an implementation of MCS to recognize triangulated graphs in $O(n + e)$ time. (Hint. To achieve linearity you may wish to link together all unnumbered vertices which are currently adjacent to the same number of numbered vertices (Tarjan [1976]).)

29. An undirected graph is called *i-triangulated* if every odd cycle with more than three vertices has a set of chords which form with the cycle a planar graph whose unbounded face is the exterior of the cycle and whose bounded faces are all triangles. Prove that a graph is i-triangulated if and only if every cycle of odd length k has $k - 3$ chords that do not cross one another (Gallai [1962]).

Bibliography

Berge, Claude

[1960] Les problèmes de colorations en théorie des graphs, *Publ. Inst. Statist. Univ. Paris* **9**, 123–160.

[1973] "Graphs and Hypergraphs," Chapter 16. North-Holland, Amsterdam. MR50 #9640.

Buneman, Peter

[1972] The recovery of trees from measures of dissimilarity, "Mathematics in the Archaeological and Historical Sciences," pp. 387–395. Edinburgh Univ. Press, Edinburgh.

[1974] A characterization of rigid circuit graphs, *Discrete Math.* **9**, 205–212. MR50 #9686.

Cantalupi, Gabriella Tazzi, and Zucchetti, Bianca Ricetti

[1972] Singrammi triangolati sferici, *Inst. Lombarado Accad. Sci. Lett. Rend. A* **106**, 697–703. MR48 #8287.

Which connected triangulated graphs can be drawn on the sphere?

Dirac, G. A.

[1961] On rigid circuit graphs, *Abh. Math. Sem. Univ. Hamburg* **25**, 71–76. MR24 #57.

Frank, Andraś

[1976] Some polynomial algorithms for certain graphs and hypergraphs, *Proc. 5th British Combin. Conf.*, Congressus Numerantium No. XV, Utilitas Math., Winnipeg, MR53 #13500.

Fulkerson, D. R., and Gross, O. A.

[1965] Incidence matrices and interval graphs, *Pacific J. Math.* **15**, 835–855. MR32 #3881.

Gallai, Tibor

[1962] Graphen mit triangulierbaren ungeraden Vielecken, *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **7**, 3–36. MR26 #3039.

Gavril, Fanica

[1972] Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph, *SIAM J. Comput.* **1**, 180–187. MR48 #5922.

[1974a] The intersection graphs of subtrees in trees are exactly the chordal graphs, *J. Combin. Theory B* **16**, 47–56. MR48 #10868.

[1974b] An algorithm for testing chordality of graphs, *Inform. Process. Lett.* **3**, 110–112. MR52 #9671.

[1977] Algorithms on clique separable graphs, *Discrete Math.* **19**, 159–165.

[1978] A recognition algorithm for the intersection graphs of paths in trees, *Discrete Math.* **23**, 211–227.

Hajnal, Andras, and Surányi, Janos

[1958] Über die Auflösung von Graphen in vollständige Teilgraphen, *Ann. Univ. Sci. Budapest Eötvös. Sect. Math.* **1**, 113–121. MR21 #1944.

Harary, Frank, and Palmer, Edgar M.

[1968] On acyclic simplicial complexes, *Mathematika* **15**, 115–122. MR37 #3936.

Lekkerkerker, C. G., and Boland, J. Ch.

[1962] Representation of a finite graph by a set of intervals on the real line, *Fund. Math.* **51**, 45–64. MR25 #2596.

Leuker, George S.

[1974] Structured breadth first search and chordal graphs, Princeton Univ. Tech. Rep. TR-158.

Meyniel, H.

[1976] On the perfect graph conjecture, *Discrete Math.* **16**, 339–342. MR55 #12568.

Ohtsuki, Tatsuo

[1976] A fast algorithm for finding an optimal ordering for vertex elimination on a graph, *SIAM J. Comput.* **5**, 133–145. MR52 #13515.

Ohtsuki, Tatsuo, Cheung, L. K., and Fujisawa, T.

[1976] Minimal triangulation of a graph and optimal pivoting order in a sparse matrix. *J. Math. Anal. Appl.* **54**, 622–633.

Olaru, Elefterie
[1969] Über die Überdeckung von Graphen mit Cliquen, *Wiss. Z. Tech. Hochsch. Ilmenau* **15**, 115–121. MR43 #3162.
Renz, P. L.
[1970] Intersection representations of graphs by arcs, *Pacific J. Math.* **34**, 501–510. MR42 #5839.
Rose, Donald J.
[1970] Triangulated graphs and the elimination process, *J. Math. Anal. Appl.* **32**, 597–609. MR42 #5840.
[1972] A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations, *in* "Graph Theory and Computing," (Ronald C. Read, ed.), pp. 183–217. Academic Press, New York. MR49 #6579.
[1974] On simple characterizations of k-trees, *Discrete Math* **7**, 317–322. MR49 #101.
Rose, Donald J., and Tarjan, Robert Endre
[1975] Algorithmic aspects of vertex elimination. *Proc. 7th Annu. ACM Symp. Theory Comput.*, 245–254. MR56 #7320.
[1978] Algorithmic aspects of vertex elimination of directed graphs, *SIAM J. Appl. Math.* **34**, 176–197.
Rose, Donald J., Tarjan, Robert Endre, and Leuker, George S.
[1976] Algorithmic aspects of vertex elimination on graphs, *SIAM J. Comput.* **5**, 266–283. MR53 #12077.
Sachs, Horst
[1970] On the Berge conjecture concerning perfect graphs, *in* "Combinatorial Structures and their Applications," pp. 377–384. Gordon & Breach, New York. MR42 #7549.
Surányi, L.
[1968] The covering of graphs by cliques, *Studia Sci. Math. Hungar.* **3**, 345–349. MR38 #76.
Tarjan, Robert Endre
[1976] Maximum cardinality search and chordal graphs. Stanford Univ. Unpublished Lecture Notes CS 259.
Trotter, William Thomas, Jr.
[1971] A note on triangulated graphs, *Notices Amer. Math. Soc.* **18**, 1045 (A).
Wagon, Stanley
[1978] Infinite triangulated graphs, *Discrete Math* **22**, 183–189.
Walter, J. R.
[1972] Representations of rigid cycle graphs, Ph.D. thesis, Wayne State Univ.
[1978] Representations of chordal graphs as subtrees of a tree, *J. Graph Theory* **2**, 265–267.