A CLASS OF PERFECT GRAPHS

© CHINH T. HOANG

SCHOOL OF COMPUTER SCIENCE
McGILL UNIVERSITY

MARCH 1983

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the requirements of the degree of Master of Science.

TABLE OF CONTENTS

		page
	ACKNOWLEDGEMENT	i
	ABSTRACT - RESUME	ii
1.	INTRODUCTION	1
2.	PERFECT GRAPHS	2
3.	PERFECTLY ORDERABLE GRAPHS	6
4.	P_4-FREE GRAPHS	19
5.	P_4-SPARSE GRAPHS	24
	APPENDIX	38
	REFERENCES	40

ACKNOWLEDGEMENT

I would like to thank Professor Vasek Chvátal for introducing the problem to me, for helping me write up this thesis, and most of all for his psychological support.

ABSTRACT

The chordless path with four vertices and three edges is denoted by P_4. A graph is called P_4-sparse if it has no induced subgraph with five vertices and more than one P_4. We shall describe an $O(n^3)$ algorithm for recognizing these graphs, and prove that they are perfect.

RÉSUMÉ

Un chemin sans corde avec quatre sommets et trois arcs est dénoté par P_4. Un graphe est appelé P_4-creux s'il n'a aucun sous-graph induit avec cinq sommets et plus qu'un P_4. Nous prouverons que les graphes P_4-creux sont parfaits, et décrirons un $O(n^3)$ algorithme pour identifier ces graphes.

1. INTRODUCTION

The subject of this thesis belongs to the theory of graphs. We shall use the standard graph-theoretic terminology throughout the text; for the reader's conveniences, all the terms (and their definitions) are listed alphabetically in the Appendix.

In the early 1960's, Berge (1962) introduced the concept of a *perfect graph*. This is a graph in which every induced subgraph has its chromatic number equal to its clique number. Since then, the topic of perfect graph was developed into a rich field. Many classes of perfect graphs, along with their polynomial-time recognition algorithms, have been identified. Yet, nobody has been able to prove the Strong Perfect Graph Conjecture. This conjecture states that the only minimal imperfect graphs are the odd cycles, except for triangles, and the complements of these odd cycles. Moreover, the problem of recognizing perfect graphs (in a polynomial time, of course) remains unsolved.

This thesis is concerned with a class of graphs which will be called *P_4-sparse*. P_4-sparse graphs are graphs in which no two P_4's share more than two vertices. Trivially, these graphs can be recognized in a polynomial time; we shall present a recognition algorithm whose running time is only $O(n^3)$. Our main result shows that P_4-sparse graphs are perfect; this strengthens a result of Lerch (1971,1972), and Seinche (1974), asserting that graphs containing no P_4's are perfect.

In Sections 2, 3, and 4, we discuss background results concerning perfect graphs, perfectly orderable graphs, and P_4-free graphs, respectively. The main original results of this thesis appear in Section 5.

2. PERFECT GRAPHS

The *colouring* (of vertices) of a graph is an assignment of 'colours' to vertices such that every two adjacent vertices always have different colours. The *chromatic number* of a graph is the smallest number of colours that suffice to colour it. A graph is called a *clique* if its vertices are pairwise adjacent. The *clique number* of a graph is the size of the largest clique in this graph. We denote the chromatic number and the clique number of a graph G by $\chi(G)$ and $\omega(G)$, respectively.

The chromatic number of a graph is at least its clique number, since every two adjacent vertices must receive different colours. Berge (1962) defined a perfect graph as a graph in which every induced subgraph H has $\chi(H) = \omega(H)$. At present, no polynomial-time algorithm to recognize perfect graphs is known, although several large classes of perfect graphs, with polynomial-time recognition algorithms, have been found (see Golumbic (1980)).

We define a *cycle* as a sequence of distinct vertices $v_1, v_2, \ldots, v_k$ with the following properties: $v_i v_{i+1}$ is an edge for $i=1,\ldots,k-1$, and $v_1 v_k$ is an edge. A *chord* in a cycle $v_1, v_2, \ldots, v_k$ is an edge $v_i v_j$ other than $v_i v_{i+1}$ $(1 \le i \le k-1)$ or $v_1 v_k$. A chordless cycle is said to have *length* k if it consists of k vertices (and k edges). We denote such a cycle by C_k. The complement $\overline{G}$ of a graph $G=(V,E)$ is the graph (V,E^*) such that $uv \in E^*$ if and only if $uv \notin E$ for all vertices u, v in V. We denote the largest number of pairwise nonadjacent vertices in G by $\alpha(G)$. Note that $\alpha(G) = \omega(\overline{G})$ and $\chi(G) \ge \frac{|V|}{\alpha(G)}$ for any graph $G = (V,E)$.

Consider a graph C_{2k+1}, $k \ge 2$. We have $\omega(C_{2k+1}) = 2$, and it is easy to see that $\chi(C_{2k+1}) = 3$. Let S be the largest set of pairwise nonadjacent vertices in C_{2k+1} so that $|S| = \alpha(C_{2k+1})$. We note that $|S| < k+1$, because each vertex x in S must be followed (in cyclic order) by a vertex x' not in S; thus, $\omega(\overline{C}_{2k+1}) = k$.

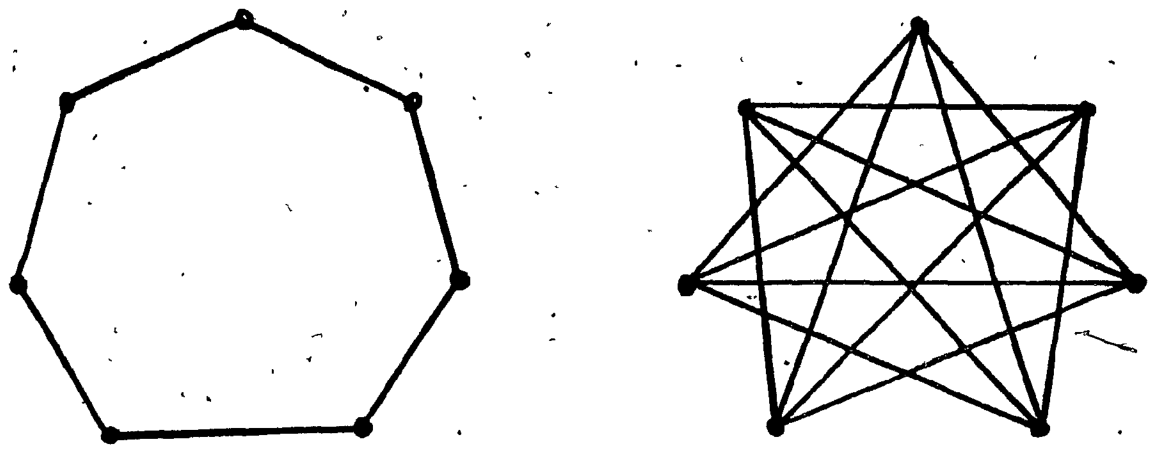


Figure 2.1: the graphs C_7 and $\overline{C}_7$.

But, $\chi(\overline{C}_{2k+1}) \geq \frac{|V|}{\alpha(\overline{C}_{2k+1})} = \frac{2k+1}{2} > k$.

We have $\omega(\overline{C}_{2k+1}) = k$, and $\chi(\overline{C}_{2k+1}) = k+1$. Both C_{2k+1} and $\overline{C}_{2k+1}$ are imperfect.

1. The Strong Perfect Graph Conjecture (Berge (1962))

   The only minimal imperfect graphs are C_{2k+1} and $\overline{C}_{2k+1}$, $k \geq 2$.

2. The Weak Perfect Graph Conjecture (Berge (1962))

   If a graph G is perfect, then its complement $\overline{G}$ is perfect.

The second conjecture was proved by Lovász (1972b). Nowadays, it is called the Perfect Graph Theorem. To see that the Strong Perfect Graph Conjecture implies the Perfect Graph Theorem, consider a perfect graph G. Trivially, G has no induced C_{2k+1} or $\overline{C}_{2k+1}$. Thus, $\overline{G}$ also has no C_{2k+1} or $\overline{C}_{2k+1}$. Now, the Strong Perfect Graph Conjecture implies that $\overline{G}$ is perfect.

Let us define a P_4 as a graph with four vertices a, b, c, d and three edges ab, bc, cd (and no other edges). It is easy to see that the complement of a P_4 is (isomorphic to) a P_4.

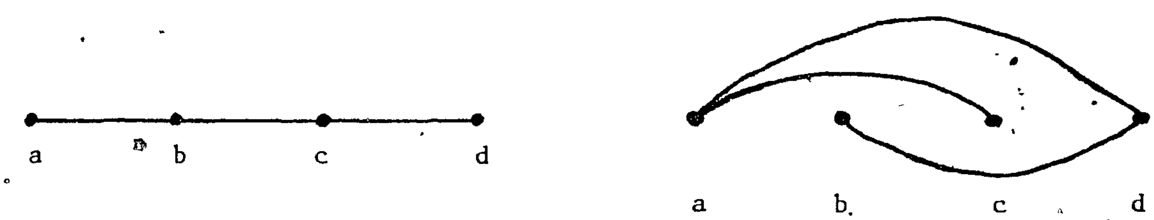


Figure 2.2: a P_4 and its complement.

A graph $G_1 = (V_1,E_1)$ is said to have the *P_4-structure* of a graph $G_2 = (V_2,E_2)$ if there is a bijection f: $V_1 \to V_2$ such that a subset S of V_1 induces a P_4 in G_1 if and only if f(S) induces a P_4 in G_2.

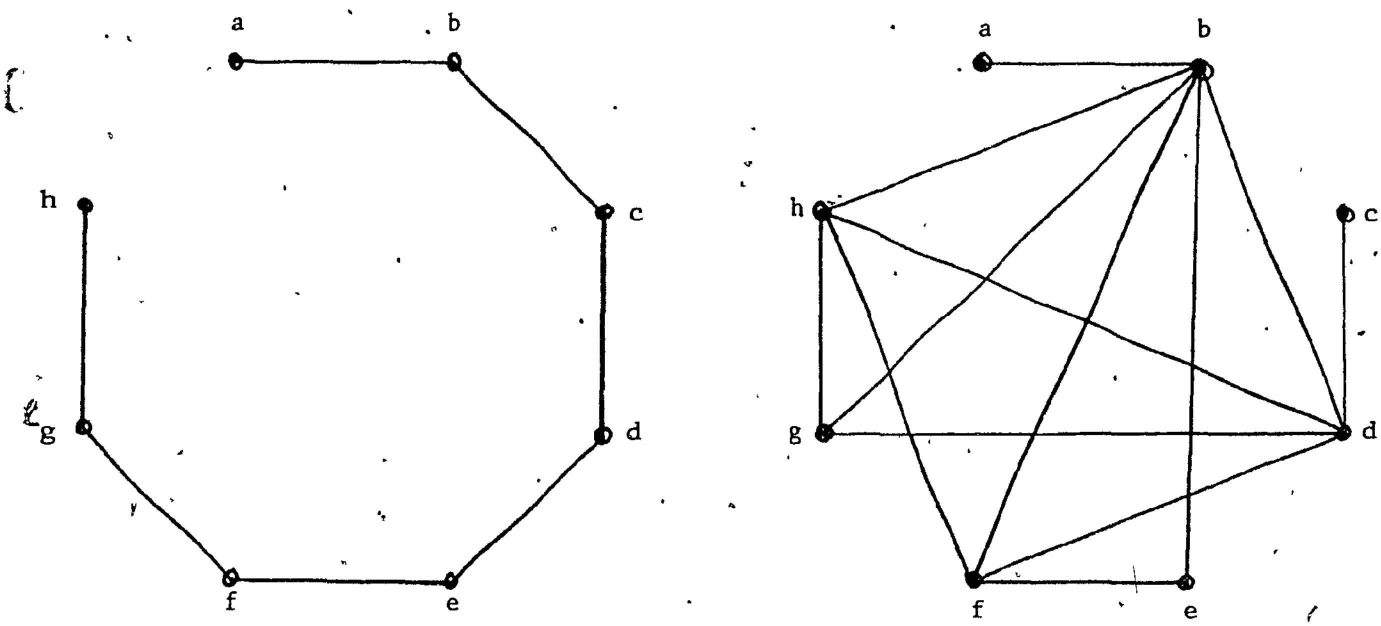


Figure 2.3: two graphs with the same P_4-structure.
(taken from Chvátal (1982))

Chvátal (1982) introduced the notion of P_4-structure and noted that, since a P_4 is self-complementary,

(i) every graph has the P_4-structure of its complement.

In addition, he proved that

(ii) the only graphs having the P_4-structure of a C_{2k+1} with $k \geq 2$ are C_{2k+1} itself and its complement.

3. The Semi-Strong Perfect Graph Conjecture (Chvátal (1982))

If a graph G has the P_4-structure of a perfect graph, then G is perfect.

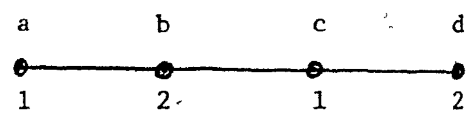
Note that, by (i) the Semi-Strong Perfect Graph Conjecture implies the Weak Perfect Graph Conjecture, and by (ii), the Semi-Strong Perfect Graph Conjecture is implied by the Strong Perfect Graph Conjecture.

3. PERFECTLY ORDERABLE GRAPHS

A natural way of colouring the vertices in a graph is to *order* them in a sequence $v_1, v_2, \ldots, v_n$. Then, scan the sequence from v_1 to v_n and assign to each v_j the smallest positive integer $f(v_j)$ assigned to none of its neighbours v_i with $i<j$. We shall refer to the graph with the linear order on the set of its vertices as an *ordered graph*, and to the procedure of assigning colours to the vertices of an ordered graph as the *greedy procedure*.

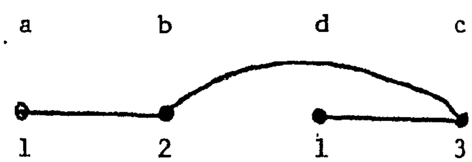
The greedy procedure may not necessarily give the best colouring. Consider the graph P_4 with vertices a,b,c,d and edges ab, bc, and cd, and the following four distinct orderings:

Figure 3.1



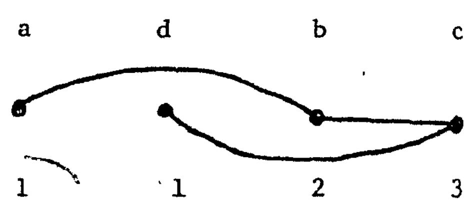
a < b < c < d

Figure 3.2



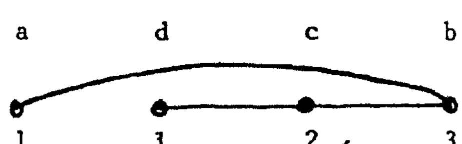
a < b < d < c

Figure 3.3



a < d < b < c

Figure 3.4



a < d < c < b

The greedy procedure produces an optimal colouring of the ordered graph in Figure 3.1, but it does not do so for the ordered graphs in Figures 3.2, 3.3, and 3.4. In particular, the graph in Figure 3.1 has $f(a) = f(c) = 1$, and $f(b) = f(d) = 2$. The graphs in Figures 3.2 and 3.3 have $f(c) = 3$, and the graph in Figure 3.4 has $f(b) = 3$.

FACT 3.1.

For every graph, there is always an ordering on which the greedy procedure produces the optimal colouring.

Proof:

Let G be an unordered graph. Find the optimal colouring of G by 'colours' 1,2,...,k for some k. For each vertex v in G, let g(v) be the colour number assigned to v. Order the vertices of G in a sequence $v_1 < v_2 < \ldots < v_n$ such that $i<j$ whenever $g(v_i) < g(v_j)$. We claim that the colouring f produced by the greedy procedure has $f(v) \leq g(v)$ for any vertex v. Obviously $f(v_1) = g(v_1) = 1$. Consider a vertex v_j, $j>1$, in the sequence. By the induction hypothesis, each vertex v_i with $i<j$ has $f(v_i) \leq g(v_i)$. Consider all neighbours v_i of v_j such that $i<j$. We know that $g(v_i) < g(v_j)$, because if $g(v_i) = g(v_j)$, then v_i is not a neighbour of v_j. Thus, we have $f(v_i) \leq g(v_i) < g(v_j)$ for all neighbours v_i of v_j. Since $f(v_j) \leq 1 + \max f(v_i)$, it follows that $f(v_j) \leq g(v_j)$. The proof is completed. □

An ordered P_4 with vertices a,b,c,d, edges ab, bc, cd such that a<b, d<c is called an *obstruction*. To put it differently, an obstruction is any one of the three ordered graphs in Figures 3.2, 3.3 and 3.4. As in Chvátal (1981), let the *Grundy number* be the largest integer $f(v_i)$ used by

the greedy procedure. A linear order on the set of vertices of a graph will be called:

(i) *admissible* if it creates no obstruction.

(ii) *perfect* if, for each induced subgraph H, the Grundy number of H equals $\chi(H)$.

It is easy to see that every perfect order is admissible. A proof of the converse relies on the following fact.

LEMMA 3.2 (Chvátal (1982))

Let G be a graph and let Q be a clique in G such that each $w \in Q$ has a neighbour $p(w) \notin Q$; let the vertices p(w) be pairwise nonadjacent. If there is an admissible order < such that $p(w) < w$ for all $w \in Q$, then some p(w) is adjacent to all the vertices in Q.

Proof:

By induction on the number of vertices in Q. For each $w \in Q$, the induction hypothesis guarantees the existence of a vertex $w^* \in Q$ such that $p(w^*)$ is adjacent to all the vertices in Q except possibly w. In fact we may assume that $p(w^*)$ is not adjacent to w, for otherwise we are done. Now, it follows that the mapping which assigns w^* to w is one-to-one, and therefore it is onto. In particular, with v standing for that vertex in Q which come first in the admissible order, there are vertices $b,d \in Q$ such that $b^*=v$ and $c^*=b$. But then there is a contradiction: the vertices a,b,c,d with $a=p(b)$ and $d=p(v)$ constitute an obstruction. The proof is completed. □

THEOREM 3.3 (Chvátal (1981))

A linear order of the set of vertices of a graph is perfect if and only if it is admissible.

Proof:

The 'only if' part is trivial; the 'if' part will be proved by induction on the number of vertices. Let G be a graph with an admissible order < of the set of its vertices, and let k stand for the Grundy number of this ordered graph. By virtue of the induction hypothesis, it will suffice to show that the chromatic number of G is at least k. Thus, it will suffice to find k pairwise adjacent vertices in G. For this purpose, consider the smallest i such that there are pairwise adjacent vertices $w_{i+1}, w_{i+2}, \ldots, w_k$ with $f(w_j) = j$ for all j. (Note that i is at most $k-1$, for $k \geq 2$.) If $i=0$, then we have found k pairwise adjacent vertices; otherwise each w_j has a neighbour $p(w_j)$ such that $p(w_j) < w_j$ and $f(p(w_j)) = i$. (To see this, suppose there is a vertex w_j with $f(p(w_j)) \neq i$, then we have $j \leq i$, this is a contradiction). But Lemma 3.2 implies the existence of a vertex v with $f(v) = i$, adjacent to all the vertices w_j, which contradicts the minimality of i. □

A graph is called *perfectly orderable* if it admits an admissible order. Recognizing perfectly orderable graphs in a polynomial time is an open problem. However, Theorem 3.3 tells us that we can recognize perfectly ordered graphs in a polynomial time. (It is sufficient to look for an obstruction in the ordered graph; if this graph has n vertices then it has at most $\binom{n}{4}$ P_4's.)

A property related to perfection has been studied by Berge and Duchet (1982). A *stable set* is a set of pairwise nonadjacent vertices. A graph is called *strongly perfect* if each of its induced subgraphs H contains a stable set meeting all the maximal cliques in H. (Here, as usual, "maximal" is meant with respect to set-inclusion, not size. In particular, a maximal clique is not necessarily largest.)

THEOREM 3.4 (Berge and Duchet (1982))

Strongly perfect graphs are perfect.

Proof:

Let $G = (V,E)$ be a strongly perfect graph.

Using induction on the number of vertices, we only need prove $\chi(G) = \omega(G)$. Let S be a stable set meeting all the maximal cliques in G, H be the subgraph of G induced by V-S. Clearly $\omega(H) = \omega(G)-1$. By the induction hypothesis, H is perfect, and so $\chi(H) = \omega(H)$. We can colour the pairwise nonadjacent vertices in S by an extra colour and have $\chi(G) = \omega(G)$. The proof is completed. □

THEOREM 3.5 (Chvátal (1981))

Every perfectly orderable graph is strongly perfect.

Proof:

It will suffice to find, in an arbitrary graph G with a perfect order $<$, a stable set meeting all the maximal cliques in G. We claim that S can be found by the following algorithm: scan the perfect ordering $v_1, v_2, \ldots v_n$ from v_1 to v_n and place each v_j in S if and only if none of its neighbours v_i $(i<j)$ has been placed in S. Indeed, if the resulting stable set is disjoint from some clique Q, then each $w \in Q$ has a neighbour $p(w)$ in S with $p(w)<w$. But then the lemma 3.2 implies the existence of a vertex $v \in S$ adjacent to all the vertices in Q. Thus, Q is not maximal. The proof is completed. □

By Theorems 3.4 and 3.5, the relationships between the classes of perfect graphs, strongly perfect graphs, and perfectly orderable graphs can be described by the following diagram.

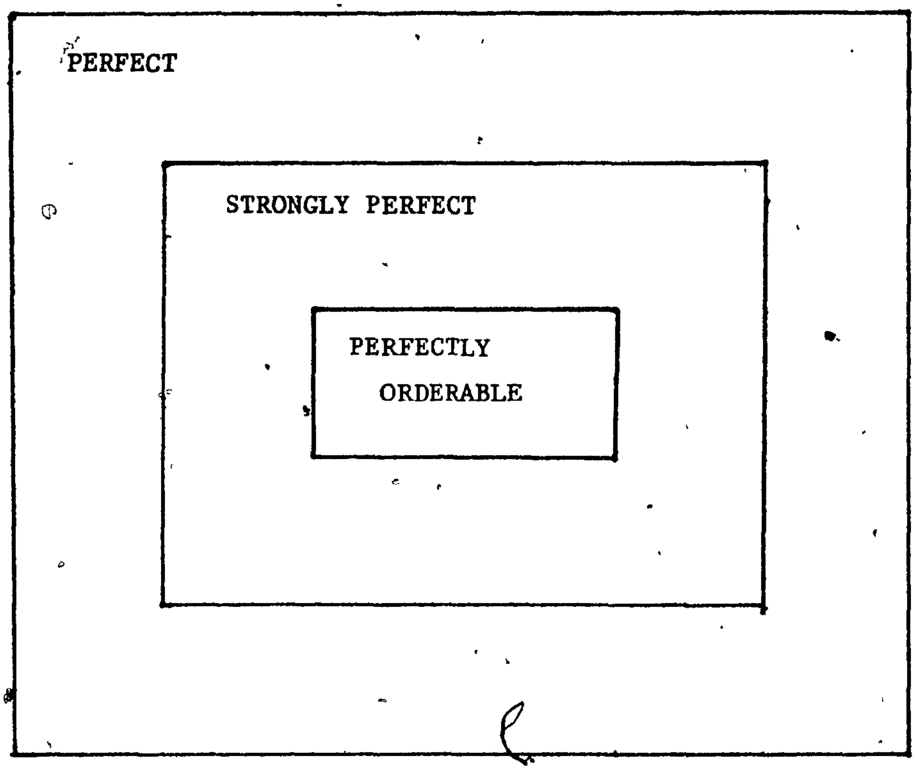


Figure 3.5

We are going to show that both inclusions are strict.

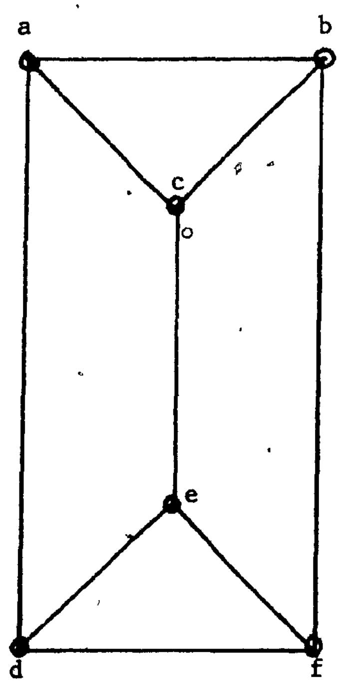


Figure 3.6: the graph $\overline{C}_6$.

First, let us prove that the graph $\overline{C}_6$ in figure 3.6 (taken from Berge and Duchet (1982)) is perfect but not strongly perfect. We shall prove that:

(1) Every proper induced subgraph is perfectly orderable

(2) $\chi(G) = \omega(G) = 3$

(3) G is not strongly perfect.

By symmetry, we only need prove the graph H induced by vertices b,c,d,e, and f are perfectly orderable to establish (1). Consider two adjacent vertices u and v, we shall represent the relation u<v by an edge directed from u to v. If an ordered graph has an obstruction, then it must have a subgraph isomorphic to the graph in Figure 3.7.

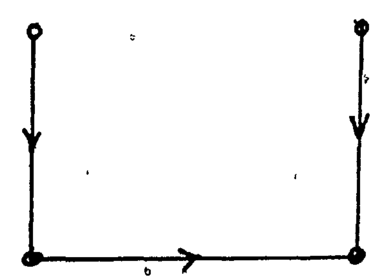


Figure 3.7: an obstruction.

Now, Figure 3.8 shows that the graph H is perfectly orderable.

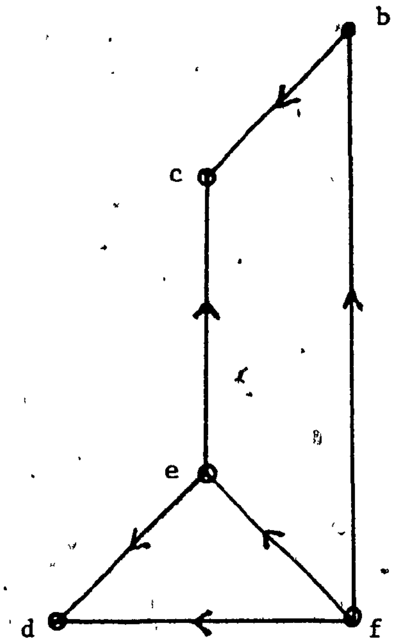


Figure 3.8.

To establish (2), we only need colour vertices a, e with colour 1, vertices c, f with colour 2, and vertices b, d with colour 3. It follows from (1) and (2) that $\overline{C}_6$ is perfect.

Let S be the largest stable set in G. It is easy to see that S has at most one vertex in {a,b,c} and at most one vertex in {d,e,f} . Thus $|S|=2$. But then the existence of maximal cliques ad, ec, bf shows that S can not meet all maximal cliques in G. (3) is established.

Secondly, let us prove that the graph G = (V,E) in Figure 3.9 (taken from Chvátal (1981)) is strongly perfect but not perfectly orderable. We shall establish:

(4) Every induced subgraph of G is perfectly orderable.

(5) There is a stable set meeting all maximal cliques in G.

(6) G is not perfectly orderable.

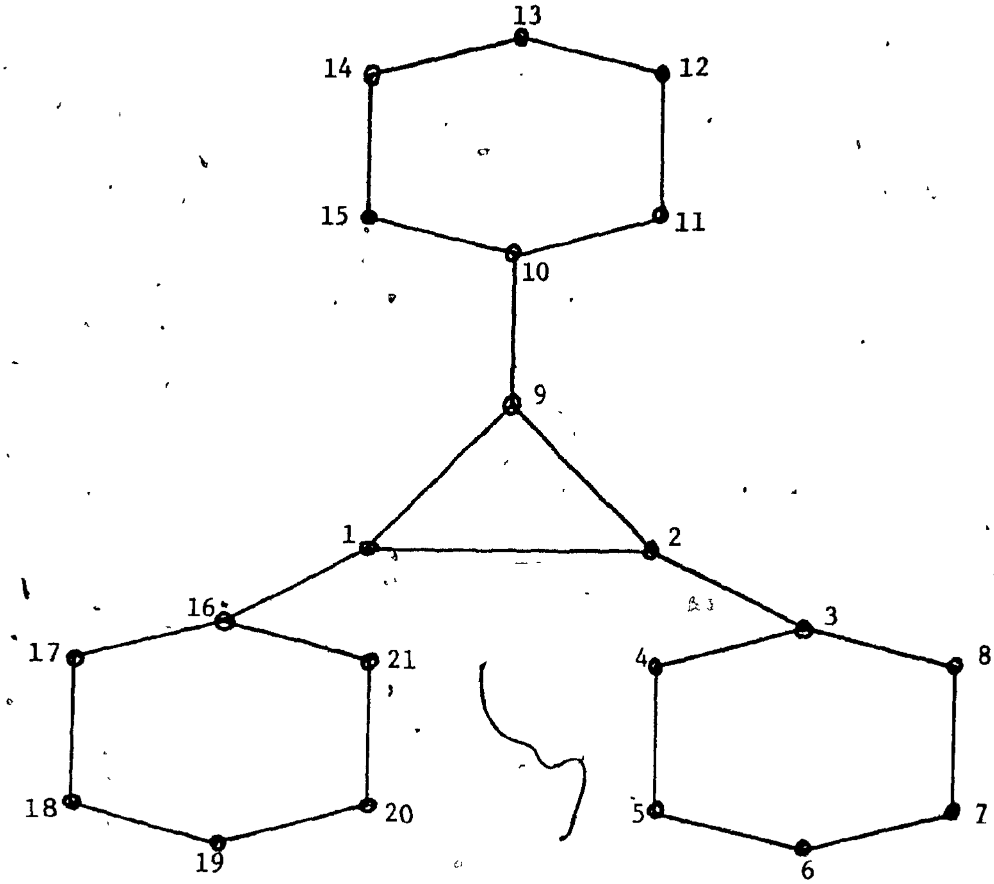


Figure 3.9

To establish (5), it is sufficient to show the stable set S = {8,6,4,2,16,18,20,10,12,14}. To establish (4), we only need show the five graphs induced by V-{15}, V-{14}, V-{13}, V-{10}, and V-{9} are perfectly orderable. Figures 3.10-3.14 show that the above graphs are perfectly orderable.

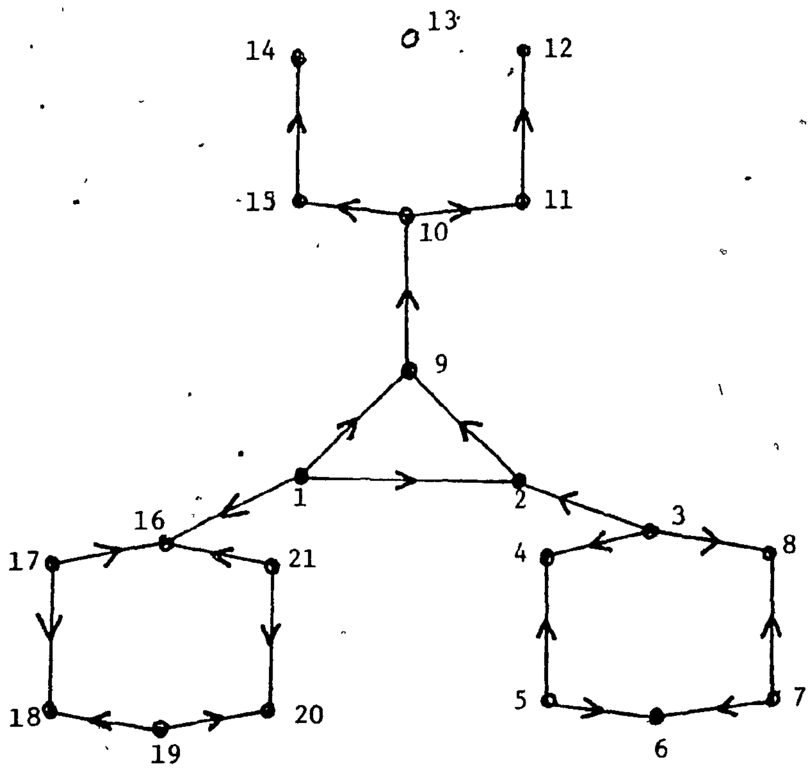


Figure 3.10: the perfectly orderable graph induced by V - {13}.

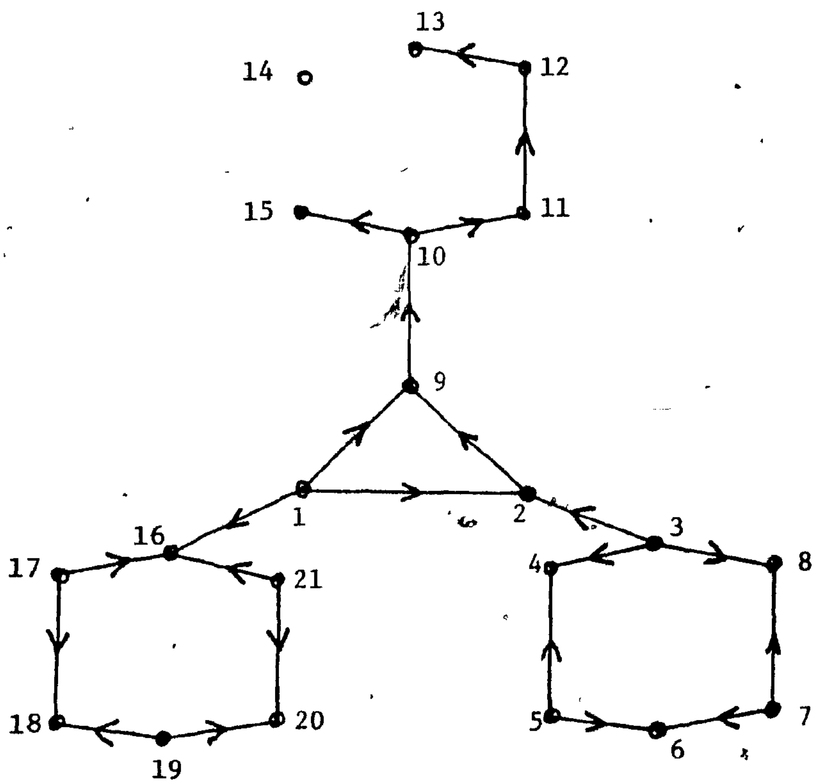


Figure 3.11: the perfectly orderable graph induced by V - {14}.

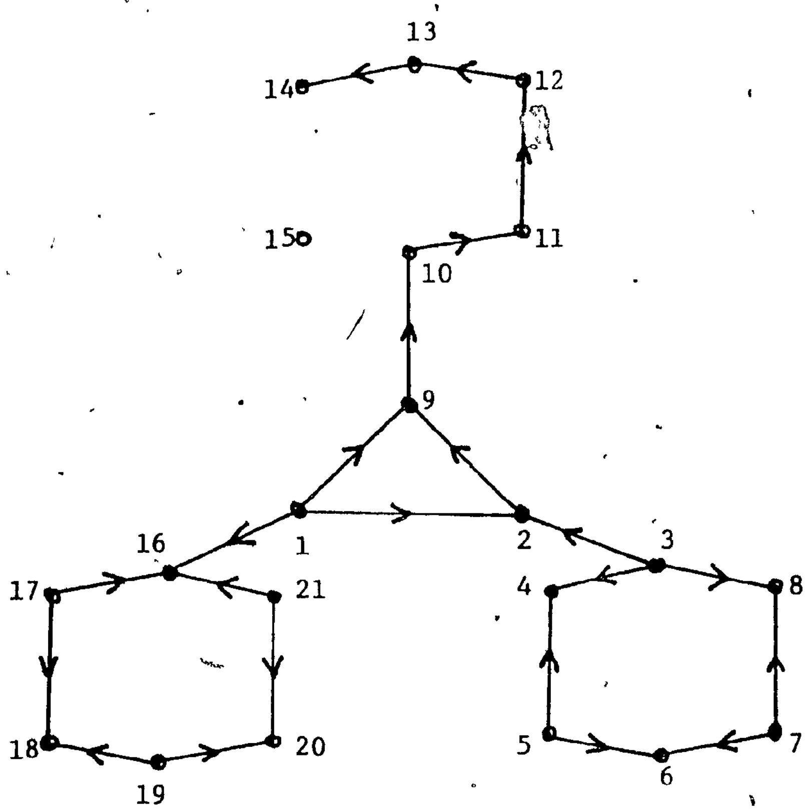


Figure 3.12: the perfectly orderable graph induced by V - {15}.

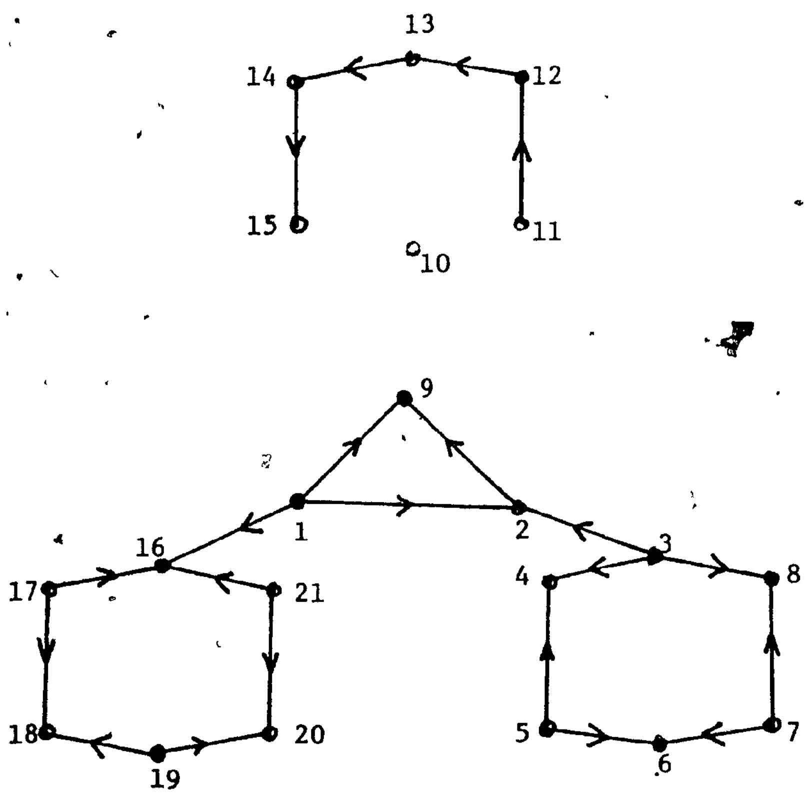


Figure 3.13: the perfectly orderable graph induced by V - {10}.

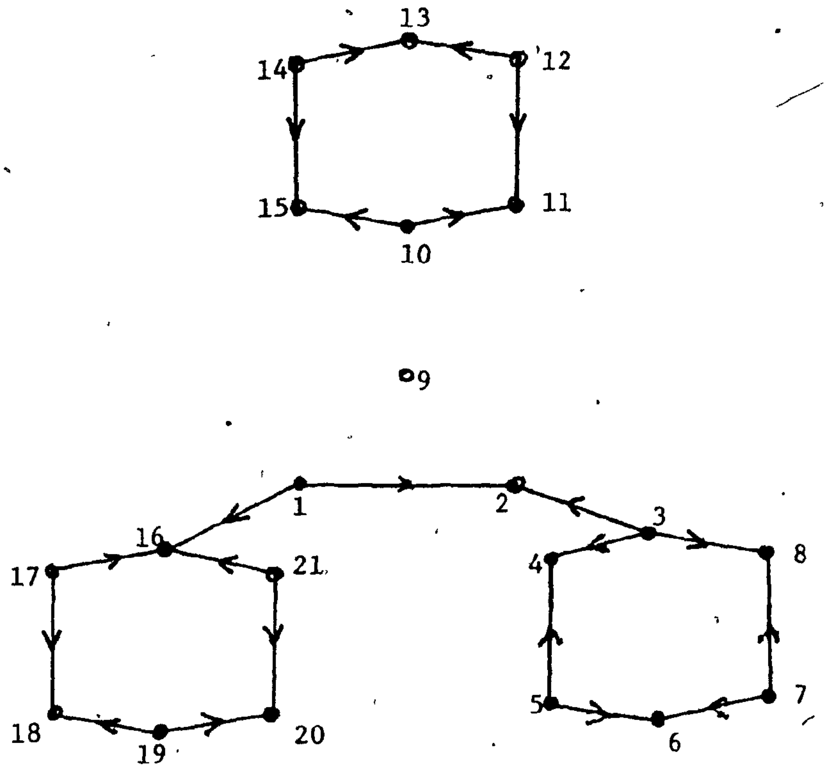


Figure 3.14: the perfectly orderable graph induced by V - {9}.

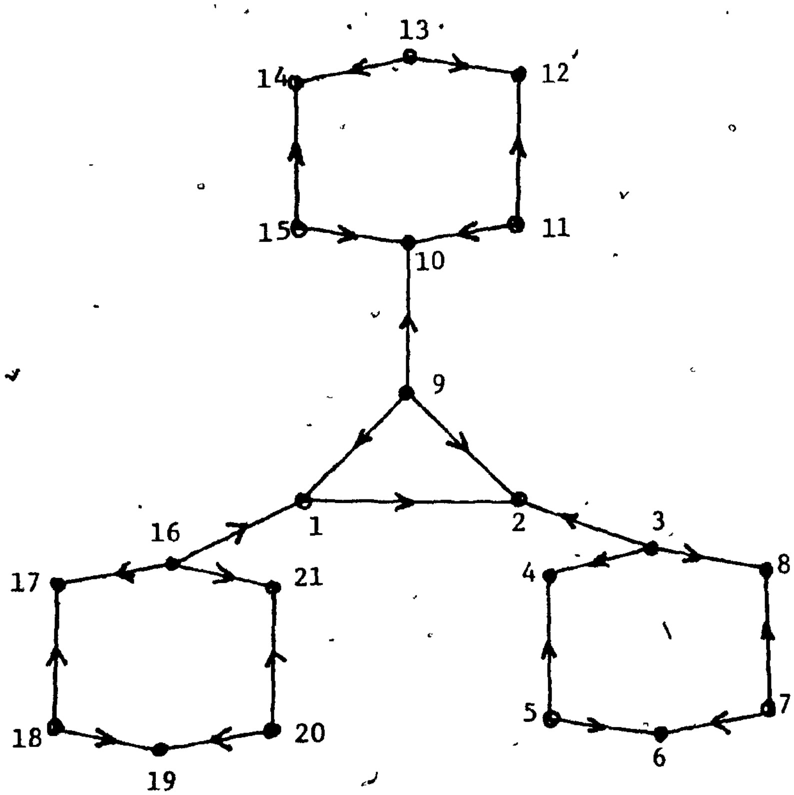


Figure 3.15: G is not perfectly orderable.

Now, we only need establish (6). Without loss of généralité, we can set 1<2. This forces the relations: 3<4, 5<6, 7<8, 3<2, 9<10, 11<12, 13<14, 15<10, 9<1, 16<17, 18<19, 20<21, 16<1. But the vertices 16,1,2,3 constitute an obstruction. G is strongly perfect but not perfectly orderable.

4. P_4-FREE GRAPHS

We shall call a graph *P_4-free* if it has no induced P_4. P_4-free graphs have been studied by many people; terms synonymous with "P_4-free graphs" include cographs (Corneil, Lerchs, Stewart Burlingham (1981)), D*-graphs (Jung (1978)), and HD or Hereditary Dacey graph (Sumner (1974)). In the early 1970's, Lerchs (1971, 1972) studied the structional and algorithmic properties of P_4-free graphs. His work was extended by Stewart (1978), who developed an $O(n^2)$ recognition algorithm for P_4-free graphs. Lerchs (1971, 1972) and Seinche (1974) independently proved that P_4-free graphs are perfect.

LEMMA 4.1: (Seinche (1974))

If a graph G is P_4-free, then either G or $\overline{G}$ is disconnected.

Proof:

Let $G=(V,E)$ be a P_4-free graph.

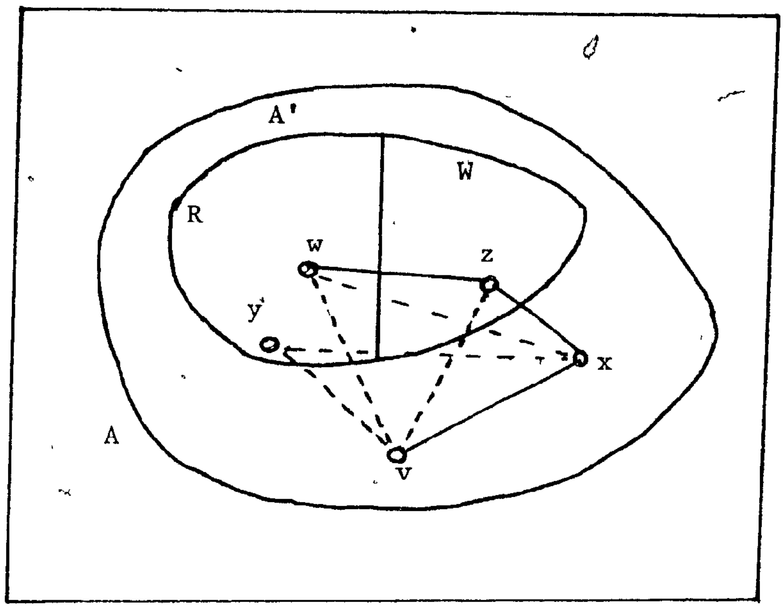
Suppose both G and $\overline{G}$ are connected. Let A be the smallest induced subgraph of G such that A has at least two vertices and such that A and $\overline{A}$ are both connected. Let x be a vertex such that its removal would disconnect A (we can always interchange G and $\overline{G}$, so that this is the case). Since $\overline{A}$ is connected, there is a vertex y in A-x such that $xy \notin E$. Let A' be the connected component of A-x that includes y. Let us partition the set of vertices in A' into disjoint sets R and W such that

(i) $u \in R$ if $ux \notin E$

(ii) $u \in W$ if $ux \in E$

Since A is connected, there is a vertex v outside $A' \cup \{x\}$ such that $vx \in E$; note that $vu \notin E$ for any vertex u in A'. Since A' is connected, there is a path P from y to x; but the only edges leaving A' are edges from

W to x, this path must include vertices w in R, z in W such that $zw \in E$. But the vertices v,x,z,w and edges vx, xz, zw form a P_4. The proof is completed. □



THEOREM 4.2: (Seinche (1974))

Every P_4-free graph is perfect.

Proof:

By induction on the number of vertices. Let $G = (V,E)$ be a P_4-free graph. Using the induction hypothesis, we only need prove that $\chi(G) = \omega(G)$. If G is disconnected, then by the induction hypothesis, each component Q of G has $\chi(Q) = \omega(Q)$. Since $\chi(G) = \max \chi(Q)$ and $\omega(G) = \max \omega(Q)$, it follows that $\chi(G) = \omega(G)$.

If G is connected then (by Lemma 4.1) $\overline{G}$ is not. Let $\overline{C}_1, \overline{C}_2, \ldots, \overline{C}_k$ be the components of $\overline{G}$. In G, we have $xy \in E$ for any choice of vertices x in C_i, y in C_j, $i \neq j$. By induction, each subgraph C_i of G has $\chi(C_i) = \omega(C_i)$. But $\omega(G) = \Sigma\, \omega(C_i)$ and $\chi(G) = \Sigma\, \chi(C_i)$. Thus, $\omega(G) = \chi(G)$. The proof is completed. □

After Seinche submitted his proof for publication, he was informed that P_4-free graphs can be obtained from a single vertex by repeated doubling of one vertex with or without joining the two doubles (Lovász (1972a)). Trivially, P_4-free graphs are perfectly orderable. (If any linear order is imposed on the vertices of a P_4-free graph, then no obstruction is created, simply because P_4-free graphs have no induced P_4.)

Let G_1 and G_2 be two disjoint graphs. The graph obtained from G_1 and G_2 by adding all edges joining vertices of G_1 to vertices of G_2 is sometimes called the *join* of G_1 and G_2 and denoted by $G_1 + G_2$. The graph obtained from G_1 and G_2 by not adding any extra edge is called the *union* of G_1 and G_2 and denoted by $G_1 \cup G_2$.

A rooted tree is called a *cotree* if

(i) Every internal node, except possibly the root, has at least two children.

(ii) The root is always labeled 'one'.

(iii) The children of a node labeled 'one' are labeled 'zero' and the children of a node labeled 'zero' are labeled 'one'.

A cotree T is said to *represent* a graph G if there is a bijection between nodes of T and certain induced subgraph of G such that:

(i) The leaves of T are one-to-one correspondence with the one-point subgraphs of G.

(ii) The root T corresponds to G itself.

(iii) If an internal node is labeled 'zero', then it represents the union of all subgraphs represented by its children.

(iv) If an internal node is labeled 'one', then it represents the join of all subgraphs represented by its children.

Each P_4-free graph can be represented by a cotree because it is either the union or the join of smaller P_4-free graphs.

Lorna Stewart (-Burlingham) (1978) designed an $O(n^2)$ algorithm which, given an arbitrary graph G, finds either a cotree representing G (thus establishing that G is P_4-free) or a P_4 in G. (Her algorithm is based on an O(n) procedure which, given any graph G together with a cotree representing some G–v, finds either a cotree representing G or a P_4 in G. Of course, this P_4 must include v.)

Figure 4.1 illustrates how a P_4-free graph can be represented by a cotree.

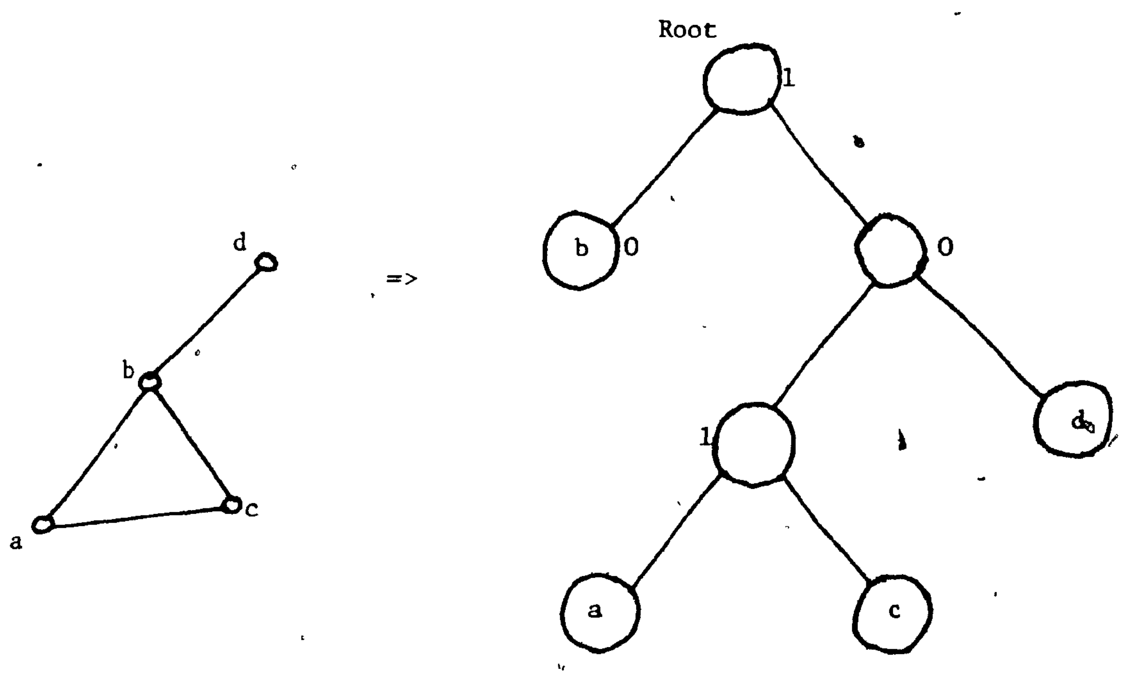


Figure 4.1

Note that if a graph is disconnected then the root has only one child. (See Figure 4.2.)

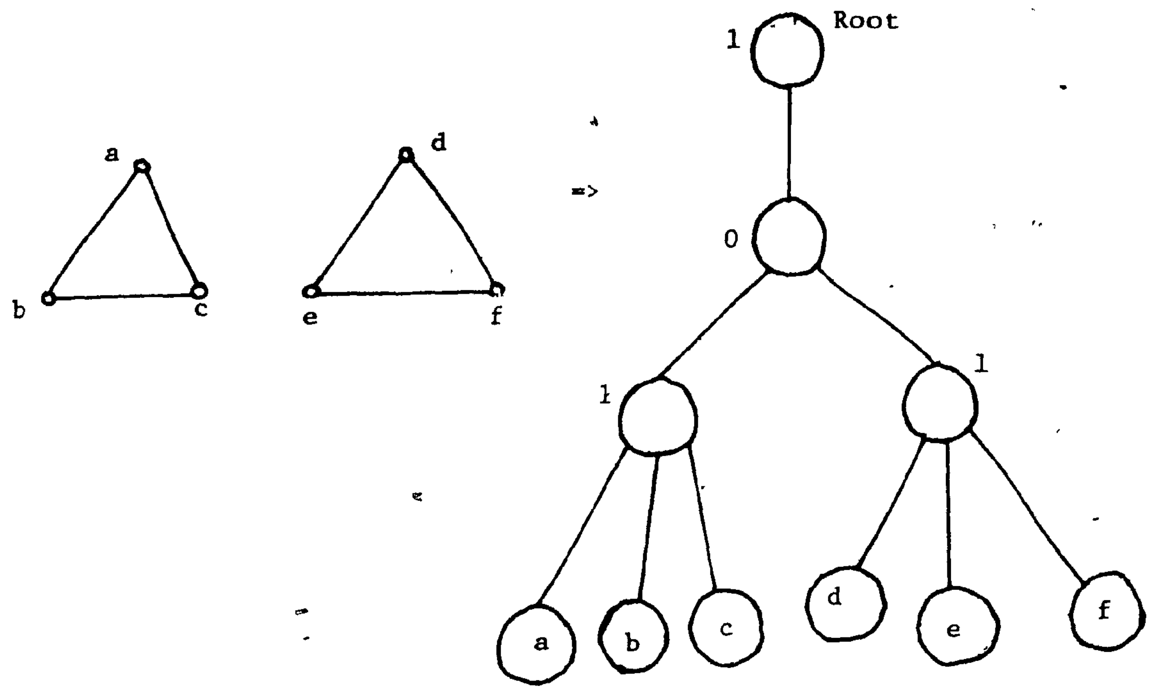


Figure 4.2

We shall assume that we can write a procedure called STEWART (G) to implement Stewart's algorithm. Given a graph G = (V,E) with $|V| = n$, STEWART (G) either finds a P_4 in G, or constructs a cotree representing G in $O(n^2)$ steps; thus establishing that G is P_4-free. This procedure STEWART (G) will be used in the next Section.

5. P_4-SPARSE GRAPHS

Let C be the class of graphs that have five vertices and at least two induced P_4's. *P_4-sparse* graphs are graphs that have no induced subgraphs that belong to C.

The class C has seven pairwise nonisomorphic graphs. Let $H=(V,E)$ be a graph in C with vertices a,b,c,d,e such that vertices a,b,c,d and edges ab, bc, cd form a P_4. We can describe H as this P_4 and:

(i) edge ea. (edges eb, ec, ed being not in E, see Figure 5.1)

(ii) edge eb. (edges ea, ec, ed being not in E, see Figure 5.3)

(iii) edges ea, eb. (edges ec, ed being not in E, see Figure 5.5)

(iv) edges ea, ec. (edges eb, ed being not in E, see Figure 5.6)

(v) edges ea, ed. (edges eb, ec being not in E, see Figure 5.7)

(vi) edges ea, eb, ec. (edge ed being not in E, see Figure 5.4)

(vii) edges ea, ec, ed. (edge eb being not in E, see Figure 5.2)

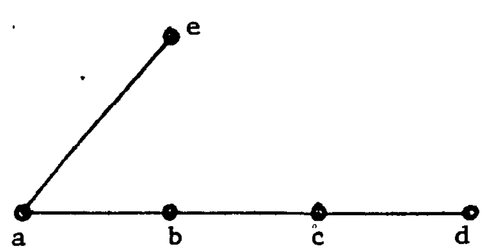


Figure 5.1

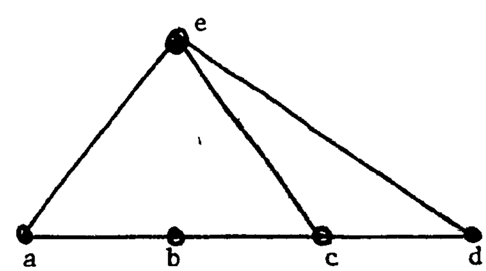


Figure 5.2

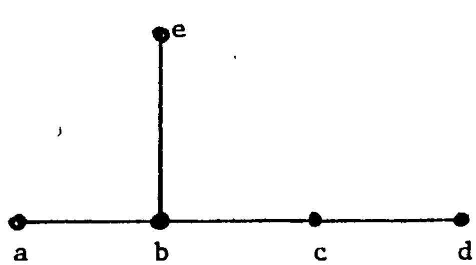


Figure 5.3

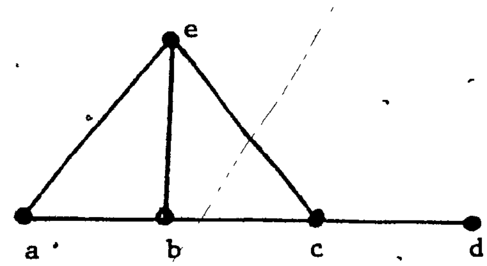


Figure 5.4

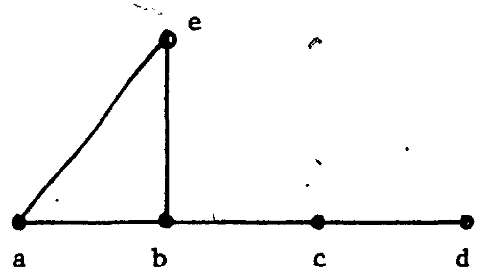


Figure 5.5

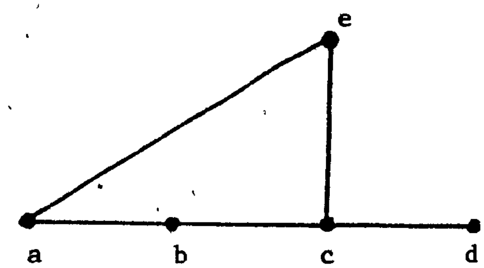


Figure 5.6

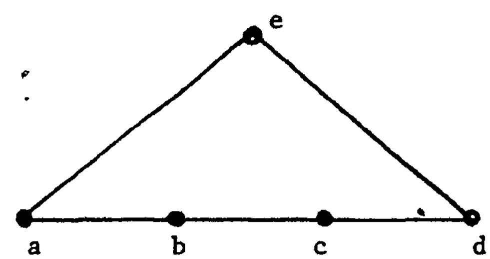


Figure 5.7

The seven figures 5.1–5.7 show all graphs of $\mathcal{C}$. It is easy to see that the complement of the graph in Figure 5.1 is the graph in Figure 5.2. Similarly, the graphs in Figure 5.3 and 5.4 are complements of the graphs in Figure 5.5 and 5.6 respectively. The graph in Figure 5.7 is a C_5, and $C_5 = \overline{C}_5$.

Let G be a P_4-sparse graph. Consider a P_4 in G and an arbitrary vertex x not in this P_4. The vertex x has one of the following properties:

(i) x is not adjacent to all vertices of the P_4.

(ii) x is adjacent to all vertices of the P_4.

(iii) x is adjacent to two 'middle' vertices of the P_4, and non-adjacent to the two 'end' vertices.

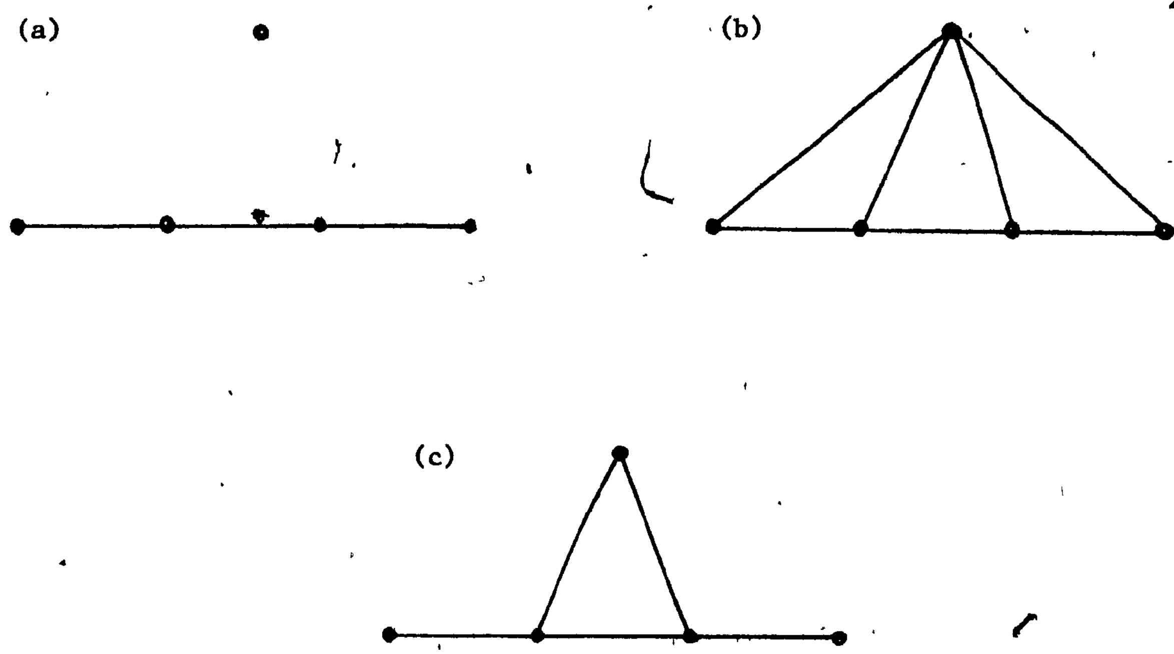


Figure 5.8: a P_4 and a vertex.

The graph shown in Figure 5.8a has the graph in Figure 5.8b as its complement. The graph in Figure 5.8c is isomorphic to its complement.

Let G = (V,E) be a graph. A set Y of vertices will be called *homogenous* if $2 \le |Y| < |V|$ and if there are no vertices u,v,w such that $u \notin Y$, $v,w \in Y$, and $uv \in E$, $uw \notin E$. (Note that Y is homogenous in G if and only if it is homogenous in $\overline{G}$.)

A graph G = (V,E) will be called a *turtle* if its vertices can be labeled $a_1,a_2,\dots,a_k$, $b_1,b_2,\dots,b_k$ or $t,a_1,a_2,\dots,a_k$, $b_1,b_2,\dots,b_k$ such that:

(i) $a_ia_j \notin E$ for all i and j

(ii) $b_ib_j \in E$ for all i and j

(iii) $a_ib_j \in E$ if and only if i=j

(iv) If t is present, then we have $ta_i \notin E$, $tb_j \in E$ for all i and j.

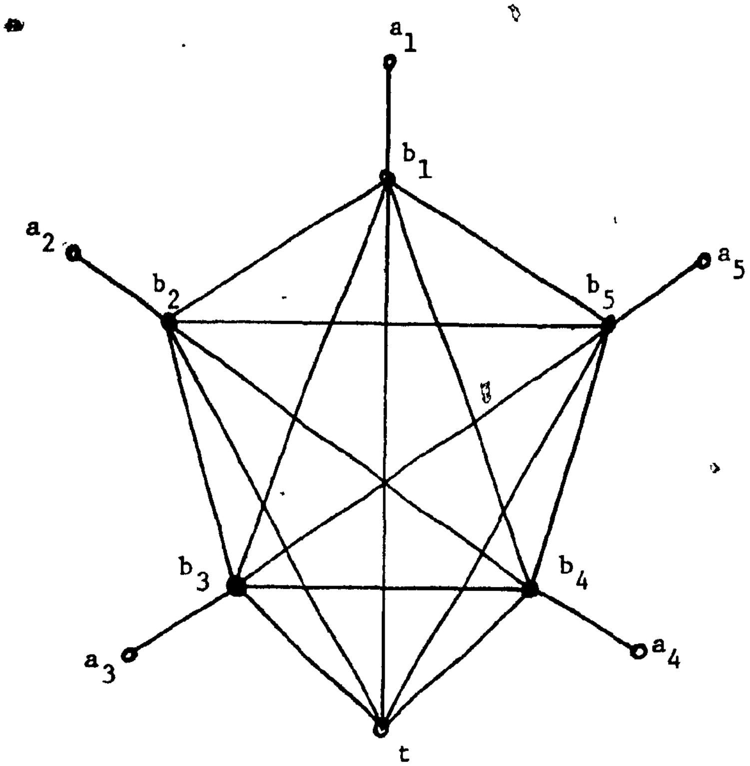


Figure 5.9: a turtle with k = 5 (and t).

We now describe a procedure RECOGNIZE which shall be used to determine whether a graph is P_4-sparse. Given a graph G, the procedure RECOGNIZE attempts to find the offending subgraph H that belongs to $\mathcal{C}$; in case of failure, it shows that either G has a homogenous set Y, or G (or $\overline{G}$) is a turtle.

RECOGNIZE terminates in step 1 if G is P_4-free (in which case it returns a cotree representing G). It terminates in one of the steps 2,3,5 if G has the subgraph H in $\mathcal{C}$. For the remaining cases, RECOGNIZE terminates in steps 6, or 7, if G has a homogenous set, else it terminates in step 8, showing that G is a turtle, or the complement of a turtle.

Assume that G has a P_4. Let vertices and edges of this P_4 be a_1,b_1,b_2,a_2 and edges a_1b_1,b_1b_2,b_2a_2. RECOGNIZE partitions the remaining vertices into disjoint sets P,Q,R,T as followed: for each vertex u

(i) $u \in P$ if u is adjacent to all four vertices a_1,b_1,b_2,a_2.

(ii) $u \in Q$ if u is nonadjacent to all four vertices a_1,b_1,b_2,a_2.

(iii) $u \in R$ if u is adjacent to b_1,b_2, and nonadjacent to a_1,a_2.

(iv) $u \in T$ if $u \notin P \cup Q \cup R$.

If T is nonempty, then there is a graph H with vertices a_1,b_1,b_2,a_2 and $u \in T$. We may assume T is empty.

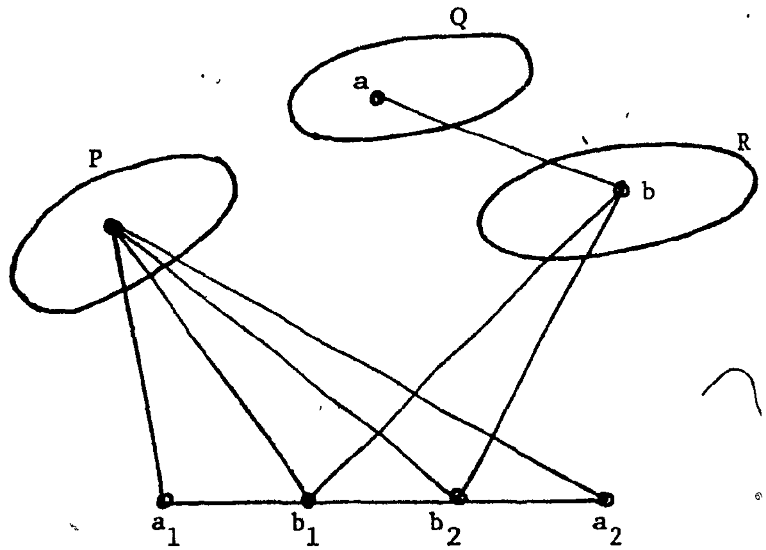


Figure 5.10: identifying a maximal turtle.

If there are vertices $b \in R$, $a \in Q$ such that $ab \in E$ (see Figure 5.10), then RECOGNIZE extends a_1,b_1,b_2,a_2 into a maximal turtle (step 3); during this process, it may find an induced subgraph H in G or $\overline{G}$ (in which case, it stops. For example, if there is a vertex u in P such that $ua, ub \notin E$, then the graph H has vertices a,b,b_1,b_2,u.). If we have $ab \notin E$ for any choice of vertices b in R, a in Q, then we have one of the two following cases:

case 1: if all vertices in P are adjacent to all vertices in R, then the set $Y = R \cup \{a_1,b_1,b_2,a_2\}$ is homogeneous *case 2: there are nonadjacent vertices u in P, v in R. In this case, we get a bigger turtle by complementing the graph G.

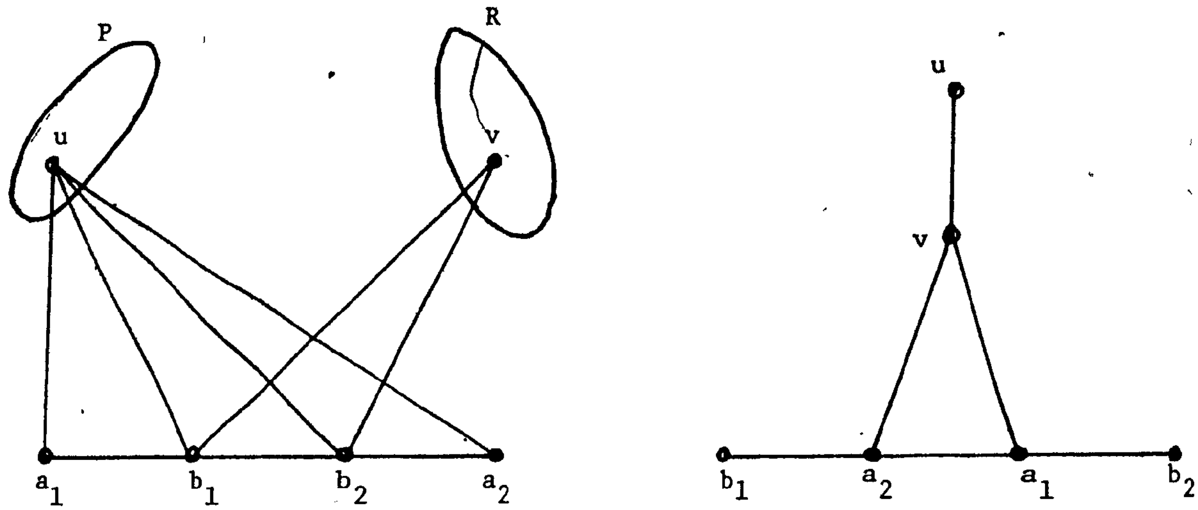


Figure 5.11: getting a bigger turtle by complementing.

From Figure 5.11, it is clear that the subgraph of $\overline{G}$ induced by vertices u,v,a_1,b_1,b_2,a_2 is a turtle with k=3 (Step 4).

RECOGNIZE(G):

Input: a graph G = (V,E) with $|V|=n$.

Output: one of the following: a subgraph H, a homogeneous set Y, a turtle, a complement of a turtle, a cotree representing the graph G.

1. Call STEWART(G). If a cotree is returned, then stop; else choose vertices a_1,b_1,b_2,a_2 such that $a_1b_1,b_1b_2,b_2a_2 \in E$, $a_1b_2,a_1a_2,b_1a_2 \notin E$, and set k=2.

2. Set

$u \in P$ if $ua_1, ua_2 \in E$ and $ub_1, ub_2 \in E$

$u \in Q$ if $ua_1, ua_2 \notin E$ and $ub_1, ub_2 \notin E$

$u \in R$ if $ua_1, ua_2 \notin E$ and $ub_1, ub_2 \in E$

If some vertex w^* other than a_1, b_1, b_2, a_2 lies outside P,Q,R, then return the subgraph H induced by a_1, b_1, b_2, a_2 and w^* and stop.

3. As long as there are adjacent vertices $a \in Q$, and $b \in R$, repeat the following operations:

3.1 If some $w^* \in P$ has $w^* a \notin E$ or $w^* b \notin \epsilon$ (or both) then return the subgraph H induced by a_1, b_1, b, a and w^*, and stop.

3.2 If some $w^* \in Q$ has $w^*a \in E$ or $w^*b \in E$ (or both) then return the subgraph H induced by a_1, b_1, b, a, and w^* and stop.

3.3 If some $w^* \in R$ has $w^*a \in E$ or $w^*b \notin E$ (or both) then return the subgraph H induced by a_1, b_1, b, a and w^*, and stop.

3.4 Delete a from Q, delete b from R, set $a_{k+1} = a$, $b_{k+1} = b$, and replace k by k+1.

4. If k=2 and some $u \in P$ is nonadjacent to some $v \in R$ then set

$x \leftarrow a_1$, $y \leftarrow b_1$, $z \leftarrow b_2$, $t \leftarrow a_2$,

$a_1 \leftarrow y$, $b_1 \leftarrow t$, $b_2 \leftarrow x$, $a_2 \leftarrow z$,

Replace G by $\overline{G}$, interchange P and Q, and return to step 3. (Note that a=u, and b=v have just become available.)

5. If $k \geq 3$ and some $u \in P$ is nonadjacent to some $v \in R$, then return the subgraph H induced by a_1, u, b_2, v, and b_3, and stop.

6. If $P \cup Q \neq \emptyset$, then set $\overline{Y} = \{a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_k\} \cup R$, return the homogenous set Y and stop.

7. If $|R| \geq 2$ then set Y = R. Return the homogenous set Y, and stop.

8. G or $\overline{G}$ is a turtle. Return this turtle, and stop.

We shall assume, as usual, that G is represented by its adjacency lists (see, for instance, Aho, Hopcroft, Ullman (1974)).

As noted in Section 4, the running time of STEWART(G) is $O(n^2)$, and so Step 1 can be executed in $O(n^2)$ steps. Execution of Step 2 involves scanning the adjacency lists of a_1,b_1,b_2,a_2, taking only O(n) steps.

Having executed Step 2, we may form a list of all edges ab such that $a \in Q$, $b \in R$ in $O(n^2)$ steps; each execution of the loop in Step 3 begins by removing an arbitrary item ab from this list. (Since Q and R shrink throughout the run of the algorithm, we may find that the item ab just removed from the list no longer has $a \in Q$, $b \in R$. In that case, we simply move on to the next item on the list.) With each execution of the loop in Setp 3, the algorithm either terminates or else Q and R shrink by one vertex each. Hence the loop is executed only O(n) times; each of its executions takes only O(n) steps (in particular, the conditions on w* can be tested by scanning the adjacency list of w*).

If the loop in Step 3 is executed at least once then $k \geq 3$ after the execution of Step 3, and so Step 4 is not executed at all. On the other hand, if Step 4 is executed then its execution is followed by an execution of the loop in Step 3, where k=2 is replaced by k=3. Hence Step 3 and Step 4 are executed at most once. Even a crude implementation of Step 4 takes only $O(n^2)$ steps. Each of Steps 5 - 8 is executed at most once. A straightforward implementation of Step 5 takes $O(n^2)$ steps; straightforward implementations of Steps 6 - 8 take O(n) steps. Therefore, the time complexity of procedure RECOGNIZE is $O(n^2)$.

We now describe a procedure DETERMINE which, in $O(n^3)$ steps, determines whether a graph G is P_4-sparse. The procedure DETERMINE may call procedure RECOGNIZE n times.

DETERMINE(G):

Input: a graph $G = (V,E)$ with $|V|=n$.

Output: a message saying whether G is P_4-sparse.

1. Call RECOGNIZE(G).
2. If a homogeneous set Y is returned, then:
   - 2.1 If there is a P_4 with one vertex in Y and three vertices not in Y, then go to step 5.
   - 2.2 Let G_Y and G_W be the subgraphs induced by Y and V−Y respectively. Call DETERMINE(G_Y) and DETERMINE(G_W). If both G_Y and G_W are P_4-sparse, then go to step 6, else go to step 5.
3. ~~If a turtle or a~~ cotree is returned, then go to step 6.
4. If a subgraph H is returned, then go to step 5.
5. Return the message 'G is not P_4-sparse' , and stop.
6. Return the message 'G is P_4-sparse' , and stop.

It is easy to see that Step 2 is executed at most $\frac{n}{2}$ times. Substep 2.1 can be tested in $O(n^2)$ steps. We partition the vertices of G into sets A,B,Y as follows. The set Y is the homogeneous set returned by RECOGNIZE(G). For each vertex u in V −Y, we set $u \in B$ if u has a neighbour in Y, else we set $u \in A$. Let G_A be the subgraph of G induced by A, and $\overline{G}_B$ be the subgraph of $\overline{G}$ induced by B. If there is a component F of G_A (or F of $\overline{G}_B$) such that $|F| \geq 2$, and F is not homogeneous in G, then return the message 'G is not P_4-sparse' (this means that there is a P_4 with vertices a,b,c,d and edges $ab,bc,cd \in E$, edges $ac,ad,bd \notin E$ such that we have either (i) $a \in A, b,d \in B, c \in Y$ if $F \subseteq \overline{G}_B$, or (ii) $a,b \in A$, $c \in B$, $d \in Y$ if $F \subseteq G_A$).

RECOGNIZE shows that if a graph G is P_4-sparse, then either

(i) G has a homogenous set, or

(ii) G or $\overline{G}$ is a turtle.

LEMMA 5.1

Let G be a graph with a homogenous set Y. If there is a P_4 with at least one vertex in Y and at least one vertex not in Y, then this P_4 has precisely one vertex in Y and three vertices not in Y. Furthermore, if such a P_4 is present, then G is not P_4-sparse.

Proof:

Since Y is homogenous, the set of vertices outside Y can be partitioned into disjoint sets A, B such that, for each vertex u, we have

(i) $u \in A$ if $ux \notin E$ whenever $u \notin Y$, $x \in Y$

(ii) $u \in B$ if $ux \in E$ whenever $u \notin Y$, $x \in Y$

If there is one P_4 with at least one vertex in Y and at least one vertex not in Y, then this P_4 has at least one vertex in B. Thus, such a P_4 can have only one vertex in Y. So, its vertices can be enumerated as a,b,c,d such that we have either $a \in A$, $b,d \in B$, $c \in Y$, or $a,b \in A$, $c \in B$, $d \in Y$. Since $|Y| \geq 2$, there is a vertex e in Y such that a,b,c,d,e are vertices of a graph H in $\mathcal{C}$. The proof is completed. □

THEOREM 5.2:

Every P_4-sparse graph is perfectly orderable.

Proof:

By induction on the number of vertices. Let G=(V,E) be a P_4-sparse graph.

Case 1: G is a turtle.

If G is a turtle, then we have the perfect order $t<b_1<b_2<\ldots<b_k<a_1<a_2<\ldots<a_k$. This order has no obstructions, since any P_4 must have vertices a_i, b_i, b_m, a_m and edges a_ib_i, b_ib_m, b_ma_m (and we have $b_i<a_i$, $b_m<a_m$). This case is settled.

Case 2: G is the complement of a turtle.

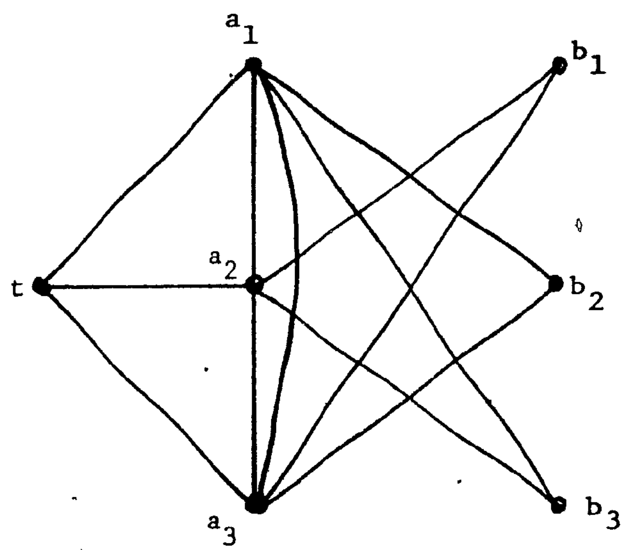


Figure 5.12: the complement of a turtle with k = 3 (and t).

If G is the complement of a turtle, then the vertices of G can be enumerated as $t, a_1, \ldots a_k, b_1, \ldots, b_k$ such that

(i) $a_ia_j \in E$ for all i and j

(ii) $b_ib_j \notin E$ for all i and j

(iii) $ta_i \in E$ and $tb_j \notin E$ for all i and j (if t is present).

(iv) $a_ib_j \in E$ if and only if $i \neq j$.

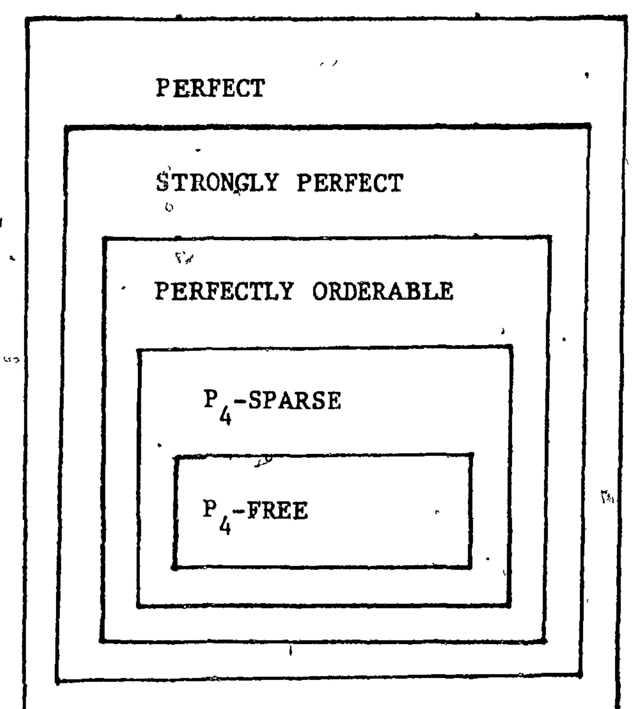
The sequence $t < a_1 < a_2 < \ldots < a_k < b_1 < b_2 < \ldots < b_k$ has no obstruction, since any P_4 must have vertices b_i, a_j, a_m, b_n, edges $b_i a_j, a_j a_m, a_m b_n$ (and we have $a_j < b_i$ for all i and j). This case is settled.

Case 3: G has a homogenous set.

Let G_Y and G_W be the subgraphs of G induced by Y and V-Y respectively. By the induction hypothesis, G_Y and G_W are both perfectly orderable. Let the perfect orders of G_Y and G_W be $y_1 < y_2 < \ldots < y_r$ and $w_1 < w_2 < \ldots < w_s$ respectively. We order the vertices in G in a sequence $y_1 < y_2 < \ldots < y_r < w_1 < w_2 < \ldots < w_s$. This order is perfect, since Lemma 5.1 guarantees that G has no P_4 with a vertex in Y, and a vertex not in Y. The proof is completed. □

COROLLARY 5.3: P_4-sparse graphs are strongly perfect. □

By Theorem 5.2, the relationships between the classes of perfect graphs, strongly perfect graphs, perfectly orderable graphs, P_4-free graphs, and P_4-sparse graphs can be described by the following diagram.



To show that all inclusions are strict, we only need show that some perfectly orderable graphs are not P_4-sparse. (The other inclusions had been proved strict in previous Sections.)

Consider the graph C_6 shown in Figure 5.13

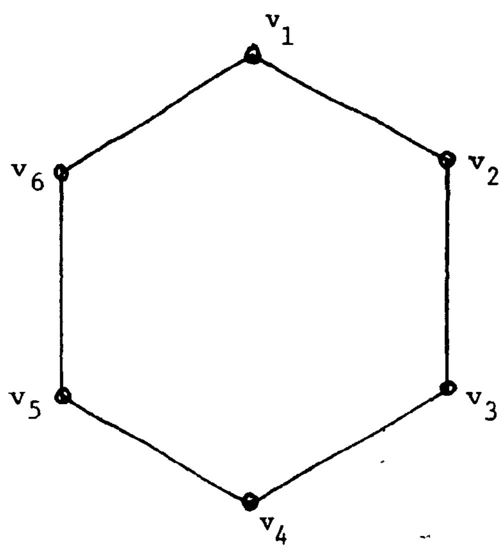


Figure 5.13

Figure 5.14 shows that C_6 is perfectly orderable.

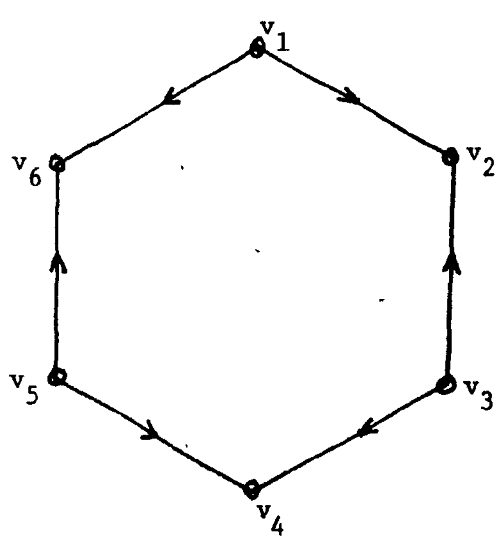


Figure 5.14

However, the subgraph induced by vertices v_1, v_2, v_3, v_4, v_5 belongs to the class C. So, C_6 is not P_4-sparse.

APPENDIX

Adjacent : two vertices are adjacent if and only if they are joined by an edge.

Bijection : a mapping one-to-one and onto.

Chord : a chord in a cycle $v_1, v_2, \ldots, v_k$ is an edge $v_i v_j$ other than $v_i v_{i+1}$ $(1 \le i \le k)$ or $v_1 v_k$.

Chromatic number : the smallest number of colours that suffice to colour a graph.

Clique : a set of pairwise adjacent vertices.

Clique number : the number of vertices of the largest clique in a graph.

Colouring : an assignment of 'colours' to vertices such that adjacent vertices always have different colours.

Complement : the complement of a graph $G = (V,E)$ is denoted by $\overline{G} = (V,E')$ with the same set of vertices, and the set E' of edges such that for any two vertices x,y in V, we have $xy \in E'$ if and only if $xy \notin E$.

Connected : a graph is connected if there is at least a path between any two vertices.

Cutset : a set of vertices such that its removal would disconnect a connected graph.

Cycle : a cycle is a path from a vertex x to a vertex y with the edge xy.

Edge : see Graph.

Graph : An ordered pair (V,E) such that V is a set and E is a set of two-point subset of V. The elements of V are called

Graph (cont.)	:	vertices and the elements of E are called edges.
Induced subgraph	:	a graph $H = (V_H, E_H)$ is an induced subgraph of a graph $G = (V,E)$ if $V_H \leq V$ and for each edge xy in E, we have $xy \in E_H$ if and only if both x and y are in V_H.
Neighbour	:	a vertex x is a neighbour of vertex y if x and y are adjacent.
Path	:	a sequence of distinct vertices $v_1, v_2, \ldots, v_n$ such that $v_i v_{i+1} \in E$ $(i \leq 1 \leq n-1)$.
Stable set	:	a set of pairwise nonadjacent vertices.
Vertex	:	see Graph.

REFERENCES

AHO, A.V., J.E. HOPCROFT, and J.D. ULLMAN (1974). The Design and Analysis of Computer Algorithms, Addison-Wesley.

BERGE, C. (1962). Sur une conjeture relative au problème des codes optimaux, comm. 13 ème assembleé générale de l'URSI, Tokyo.

BERGE, C. (1973). Graphs and hypergraphs, North-Holland, Amsterdam.

BERGE, C. and V. CHVÁTAL (1983). Topics on Perfect Graphs, North-Holland, Amsterdam. (to appear).

BERGE, C. and P. DUCHET (1982). Strongly perfect graphs. To appear in Berge and Chvátal (1983).

CHVÁTAL, V. (1982). A semi-strong perfect graph conjecture, School of Computer Science, McGill University, Technical Report 82.1. To appear in Berge and Chvátal (1983).

CHVÁTAL, V. (1981). Perfectly orderable graphs, School of Computer Science, McGill University, Technical Report 81.28. To appear in Berge and Chvátal (1983).

CORNEIL, D.G., H. LERCHS, and L. STEWART-BURLINGHAM (1981). Complement reducible graphs, Discrete Applied Math. 3: 163-174.

GOLUMBIC, M.L. (1980). Algorithmic Graph-Theory and Perfect Graphs, Academic Press, New York.

JUNG, H.A. (1978). On a class of posets and the corresponding comparability graphs, J. Combin. Theory B24: 125-133.

LERCHS, H. (1971). On cliques and kernels, Dept. of Comp. Sci., University of Toronto.

LERCHS, H. (1972). On the clique-kernel structure of graphs, Dept. of Comp. Sci., University of Toronto.

LOVÁSZ, L. (1972a). Normal hypergraphs and the perfect graph conjecture, Discrete Math. 2: 253-267.

LOVÁSZ, L. (1972b). A characterization of perfect graphs, J. Combin. Theory B13: 95-98.

SEINSCHE, D. (1974). On a property of the class of n-colorable graphs, J. Combin. Theory B46: 191-193.

STEWART, L. (1978). Cographs, a class of tree representable graphs, M.Sc. Thesis, Dept. of Comp. Sci., University of Toronto, TR 126/78.

SUMNER, D.P. (1974). Dacey graphs, J. Australian Math. Soc. 18 (4), 492-502.