Chapter 1

Introduction

1.1 Background and Motivation

Our mathematical adventure begins with a collection of intervals on the real line. The intervals may have come from an application, for example, they could represent the durations of a set of events on a time line, or fragments of DNA on the genome, or sectors of consecutive elements of a linearly ordered set. Some of the intervals may intersect one another, and others may be disjoint. No matter what they may represent, intervals are familiar to us as mathematical entities. There are many relationships between these intervals that we could study. In this book, we deal mostly with intersection.

When two intervals intersect, we might interpret this positively as their having something important in common, like an opportunity to share information. For example, if each interval represented the time period during which a group of school children would be visiting a science museum, then two groups whose intervals intersect could participate in a joint activity. We might then ask, how many times would we need to flash the new Artificial Bolt of Lightning so that each group would get to see it? Or we might interpret intersection negatively as having a major conflict, like competing for a resource that cannot be shared. For example, in a one television household, when a parent wants to watch the News and at the same time a teenager wants to watch an old movie on a different channel, we have a temporal conflict.

In graph theory, the family of *interval graphs* was introduced to study such problems of intersecting intervals on the line. In this model, each vertex

v in a graph $G = (V, E)$ is associated with an interval I_v, and two vertices are connected by an edge in G if their associated intervals have non-empty intersection. Formally, $uv \in E(G) \iff I_u \cap I_v \neq \emptyset$, for all $u, v \in V(G)$. The graph G is called an interval graph.

In our museum example, there is a well defined minimum number α of how many times that the lightning must be flashed, and it is easy to calculate the number α and an optimal schedule for the flashes. Well, at least it is "easy" for the authors since we have been teaching students about interval graphs for a long time. But it is also "easy" in a computational sense since there are well known linear time algorithms to do this.

But what do you do if the electricity requirements allow only $\alpha - 2$ flashings? Either some of the groups will be disappointed, or they will have to reschedule the time of their visit. Similarly, in our television example, when one spouse wants to watch a game show and the other spouse wants a basketball game, it is fair game to assume that a compromise is needed.

In this book, we will study the class of tolerance graphs, which are a generalization of interval graphs. Tolerance graphs are constructed from intersecting intervals in a manner similar to interval graphs, but putting an edge between two vertices depends on measuring the size of the intersection of their two intervals before declaring that an edge exists. Informally, if both intervals are willing to "tolerate" or ignore the intersection, then no edge is added between their vertices in the graph.

Tolerance graphs were introduced by Golumbic and Monma in [GM82] to generalize some of the well known applications associated with interval graphs. Their original motivation was the need to solve scheduling problems in which resources such as rooms, vehicles, support personnel, etc. may be needed on an exclusive basis, but where a measure of flexibility or tolerance would allow for sharing or relinquishing the resource if a solution is not otherwise possible. Let's look at simple example.

A Motivating Example On a typical morning, six parliamentary or corporate meetings are to convene according to a fixed schedule, where meeting m_i is scheduled for the time interval $I_i = [a_i, b_i]$. Each meeting must be assigned a meeting room. Let us consider the example,

$$I_1 = [8{:}00\text{–}9{:}45], \quad I_2 = [9{:}00\text{–}11{:}30], \quad I_3 = [8{:}30\text{–}11{:}15],$$
$$I_4 = [10{:}00\text{–}11{:}00], \quad I_5 = [10{:}15\text{–}12{:}00], \quad I_6 = [10{:}45\text{–}12{:}30]$$

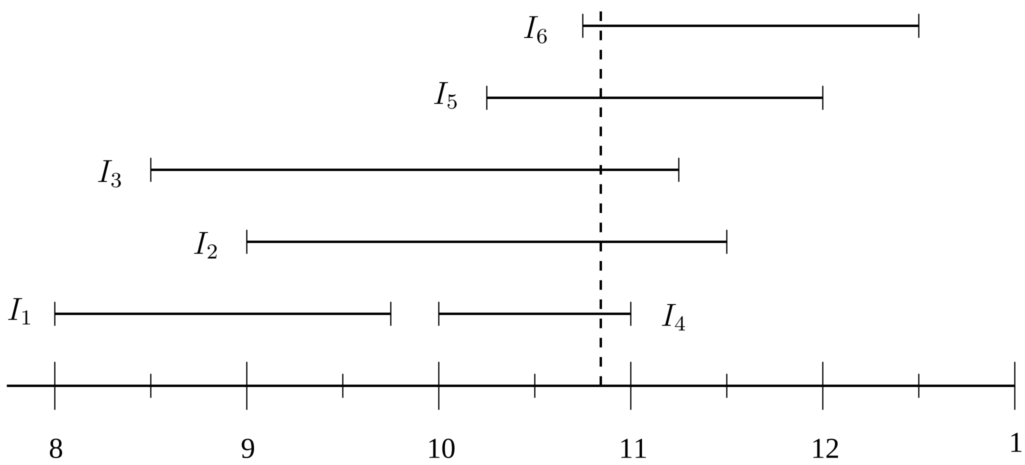

Figure 1.1: A motivating example.

In our example, meeting m_1 could use the same room as either m_4 or m_5 or m_6 since its time interval I_1 does not intersect with the time intervals I_4, I_5 or I_6. Being very strict with these intervals, we see that at 10:50 five rooms are needed simultaneously, (see Figure 1.1). But suppose there are only four meeting rooms! Should we cancel one of the meetings? Probably not. Rather, we should try to identify some flexibility in these time constraints which may allow us to find an acceptable assignment of rooms.

The tolerance graph model, which we will formally define below, provides a mechanism for associating a numerical tolerance to each meeting to indicate the degree of its flexibility in allowing some intersection with other intervals. In this way, it may be possible to give an assignment of rooms to all the meetings by sharing the room for a short period or by moving the start or finish time. In our example, if both I_4 and I_6 were willing to tolerate an overlap of more than 15 minutes, then there would be a four room solution.

Resource assignment problems of this nature arise in many contexts: motorcycles for delivering express mail (or pizza), nurses for operating rooms, waterfront space for picnics, ovens for warming a caterer's dishes, etc. In a real world situation, some meetings or deliveries may indeed have strict deadlines which must be met, while others may be more flexible. By taking these tolerances into account, solutions can often be found which would otherwise not exist under the strict constraints. There would be a great benefit to having algorithmic methods for automatically resolving such conflicts.

This example, and the discussion on intersecting intervals, briefly motivates the topic of our book. The volume and scope of research in this area has expanded significantly both from the mathematical and algorithmic points of view. Many special families of graphs and ordered sets will be encountered along the way. Each will depend on the specific tolerance model being discussed.

In this chapter, we will provide the formal definition of a tolerance graph and give some elementary properties. We will also give a brief review of many of the important families of graphs which are related in some way to tolerance graphs.

1.2 Intersection Graphs and Interval Graphs

Let $\mathcal{F}$ be a collection of sets. The *intersection graph* of $\mathcal{F}$ is the graph obtained by assigning a distinct vertex to each set in $\mathcal{F}$ and joining two vertices by an edge precisely when their corresponding sets have a nonempty intersection. When the types of sets allowed in $\mathcal{F}$ is limited, interesting classes of graphs result.

Most important to us will be the *interval graphs* which arise when the sets in $\mathcal{F}$ are intervals in the real line, that is, a graph $G = (V, E)$ is an *interval graph* if each vertex $v \in V$ can be assigned a real interval I_v so that $xy \in E \iff I_x \cap I_y \neq \emptyset$. The set of intervals $\{I_v \mid v \in V\}$ is an *interval graph representation* of G.

Interval graphs are important for their applications to scheduling problems, microbiology, and VLSI circuit design. In our previous motivating example (Figure 1.1), the intervals represented fixed time slots for a set of meetings which needed to be assigned rooms. The interval graph for this example is shown in Figure 1.2. Finding a consistent assignment of rooms can be viewed as a *coloring problem* on the interval graph, where the meeting rooms are the colors and adjacent vertices must be assigned different colors. There are efficient algorithms for coloring the vertices of an interval graph using a minimum number of colors [Gol80]. In our example, there cannot be a solution with four rooms since the interval graph has a *clique* (or *complete subgraph*) of size five. Indeed, the only subsets that could be colored by the same color in this example are $\{1, 4\}$ or $\{1, 5\}$ or $\{1, 6\}$. A *stable set* (or

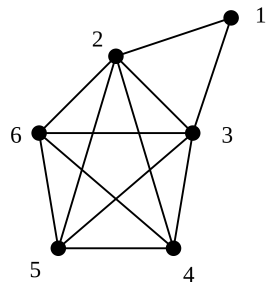

Figure 1.2: The interval graph for our motivating example.

independent set) is a subset of vertices no two of which are connected by an edge. Here there is no stable set larger than size 2.

In this book, we also consider other families of intersection graphs, such as *trapezoid graphs* and *parallelogram graphs* which are intersection graphs of trapezoids (resp. parallelograms) having two of their sides on two fixed parallel lines. Later in this chapter, we discuss permutation graphs which can be interpreted as intersection graphs of line segments in a matching diagram. Also, in Chapter 11, we present a variety of intersection graphs involving subtrees and paths in trees.

All of these families of intersection graphs satisfy the *hereditary property*, namely, if a graph $G = (V, E)$ is the intersection graph of a certain type (e.g., intervals, trapezoids, etc.), then every *induced subgraph* G_X of G is also an intersection graph of that same type, where $V(G_X) = X \subseteq V(G)$ and $E(G_X) = \{uv \in E(G) \,|\, u, v \in X\}$.

1.3 Tolerance Graphs: Definitions and Examples

A graph $G = (V, E)$ is a *tolerance graph* if each vertex $v \in V$ can be assigned a closed interval I_v and a tolerance $t_v \in \mathbf{R}^+$ so that $xy \in E$ if and only if $|I_x \cap I_y| \geq \min\{t_x, t_y\}$. Such a collection $\langle \mathcal{I}, t \rangle$ of intervals and tolerances is called a *tolerance representation* where $\mathcal{I} = \{I_x \mid x \in V\}$ and $t = \{t_x \mid x \in V\}$. If graph G has a tolerance representation with $t_v \leq |I_v|$ for all $v \in V$, then G is called a *bounded tolerance graph* and the representation is called a *bounded tolerance representation.*

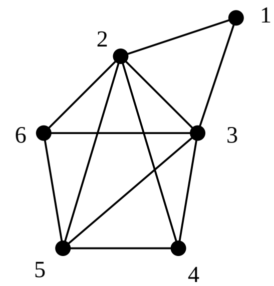

Figure 1.3: The tolerance graph for our motivating example, where I_4 and I_6 have a tolerance of 20 minutes and each of the others 5 minutes.

Consider once again our motivating example. If each of the tolerances were to be 5 minutes, then the tolerance graph would be the same as the interval graph since all of the non-empty intersections are longer than 5 minutes. However, if the tolerances of I_4 and I_6 were 20 minutes (or anything greater than 15 minutes) and each of the others 5 minutes, then the tolerance graph would have no edge between v_4 and v_6, as shown in Figure 1.3. In this case, the vertices of the tolerance graph can be colored using 4 colors, which provides a consistent assignment of meeting rooms.

We next look at some additional examples of tolerance graphs. For tolerance representations, we draw the interval assigned to each vertex and list its tolerance next to it, as in the representation of the tree T_2 in Figure 1.4. Notice that the vertex c_3 has infinite tolerance. In fact, any tolerance greater than $|I_{c_3}|$ would work equally well. In Chapter 3, we will see that every tolerance representation of T_2 must have some vertex whose tolerance is greater than its interval length.

For bounded tolerance representations, the tolerance assigned to vertex v is at most the length of the interval $I_v = [L(v), R(v)]$ assigned to v. In this case, we sometimes find it clearer to show the tolerances visually using shading. We shade in the interval from $L(v)$ to $L(v) + t_v$ above I_v and shade in the interval from $R(v) - t_v$ to $R(v)$ below I_v. Figure 1.5 shows a bounded tolerance representation of the graph $K_{1,3}$ in which tolerances are indicated by shading.

The exercises at the end of this chapter will help the reader to become familiar with the concepts presented. Our formal study of tolerance graphs begins in Chapter 2. The remainder of this chapter is devoted to definitions,

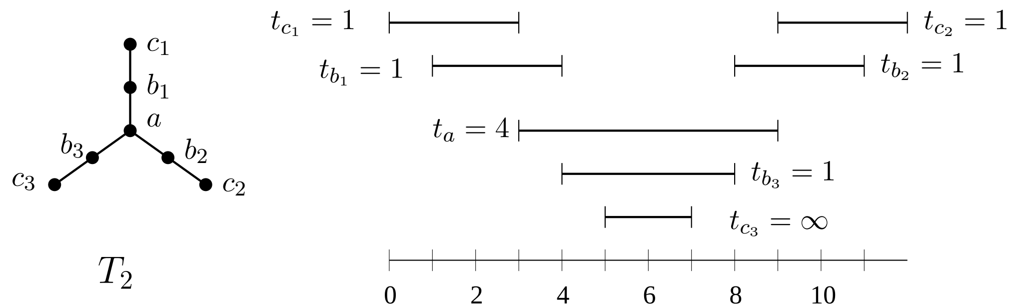

Figure 1.4: The graph T_2 and a tolerance representation of it.

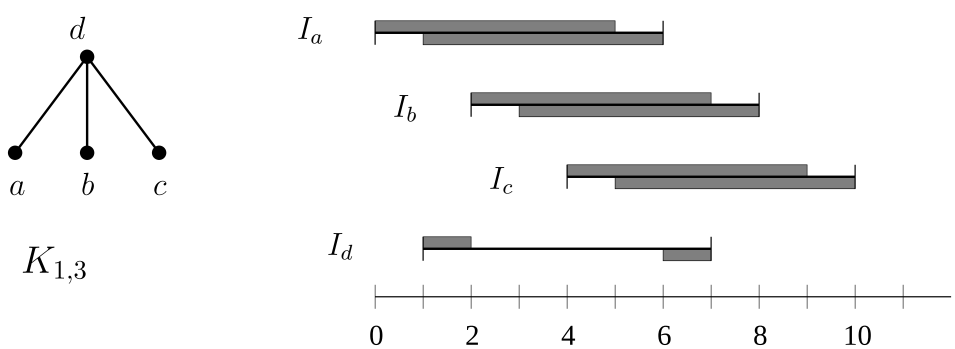

Figure 1.5: The graph $K_{1,3}$ and a bounded tolerance representation of it.

background and classical results.

1.4 Chordal Graphs, Comparability Graphs, and Properties of Interval Graphs

1.4.1 Chordal Graphs and Split Graphs

A graph G is a *chordal graph* if every cycle of length greater than or equal to 4 has a *chord*, that is, an edge connecting two vertices that are not consecutive on the cycle. For example, the graph in Figure 1.3 is chordal, and the edge (3,5) is a chord of the cycle [3,4,5,6,3]. The chordal graphs are a well known classical family of graphs, and they appear in many interesting applications including relational databases, matrix theory, statistics and biology. In the literature, chordal graphs are also called *triangulated graphs* [Ber73, Gol80] or *rigid circuit graphs* [Rob76]. The family of chordal graphs includes all interval graphs but does not include all tolerance graphs.

There are several interesting characterizations of chordal graphs which we will now review. We present their equivalence below in Theorem 1.1.

A vertex v is called *simplicial* if its *neighborhood* $\mathcal{N}(v) = \{w \in V(G) \,|\, \forall w \in E(G)\}$ is a clique, that is, every pair of neighbors of v are connected by an edge of the graph. Let $\sigma = [v_1, v_2, \dots, v_n]$ be an ordering of the vertices $V(G)$, and let $G_i = G_{\{v_i,\dots,v_n\}}$ denote the subgraph remaining after deleting $\{v_1, \dots, v_{i-1}\}$ from G. We define σ to be a *perfect elimination ordering (peo)* if v_i is a simplicial vertex in the graph G_i, for all i. For example, two possible perfect elimination orderings for the graph in Figure 1.3 are [4,6,5,1,3,2] and [1,4,3,5,2,6], but [3,4,5,6,1,2] is *not* a perfect elimination ordering for this graph.

A *maximum cardinality search (MCS)* of a graph G is done as follows: Initially all vertices are unnumbered and have counters set to zero. Choose an unnumbered vertex with largest counter, give it the next number, and add 1 to the counters of each of its neighbors. Continue doing this until all the vertices have been numbered. Suppose that the vertices were numbered in this way $[x_1, x_2, \dots, x_n]$, then we will call it a *maximum cardinality search ordering.* Such an MCS ordering for the graph in Figure 1.3 is [1,2,3,4,5,6].

Theorem 1.1. *The following conditions are equivalent:*

(i) *G is a chordal graph.*

(ii) *G has a perfect elimination ordering.*

(iii) *The reversal $[x_n, \ldots, x_2, x_1]$ of any MCS ordering of G is a perfect elimination ordering.*

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

The equivalence (i)$\Leftrightarrow$(ii) is due to Dirac; (i)$\Leftrightarrow$(iii) to Tarjan; (i)$\Leftrightarrow$(iv) independently to Buneman, Gavril and Walters; see [BLS99, Gol80, Gol84, MM99] for a proof of this theorem and for additional references.

Both conditions (ii) and (iii) suggest algorithms for recognizing chordal graphs. Using (ii), one would repeatedly look for and eliminate a simplicial vertex, breaking ties arbitrarily, until either all vertices are eliminated (success) or no simplicial vertex can be found (failure). This greedy method is correct since once a vertex becomes simplicial, it remains simplicial in any induced subgraph. Using (iii), one would carry out a maximum cardinality search while testing its reversal to verify that it is a perfect elimination ordering (success) or is not a peo (failure). The latter method gives a more efficient algorithm, having complexity $O(n + e)$ for a graph with n vertices and e edges, see [BBH02, Gol80, Gol84, TY84].

There are also efficient, polynomial time algorithms for finding a minimum coloring, maximum clique, maximum stable set, or a minimum clique cover of a chordal graph. In general, these graph problems are NP-complete, which means that chordal graphs are indeed a very special family of graphs.

We conclude this section by defining and characterizing the class of split graphs. A graph $G = (V, E)$ is called a *split graph* if its vertex set can be partitioned $V = X \cup Y$ into a stable set X and a clique Y. The graph in Figure 1.3 is a split graph with partition $X = \{1, 4\}$ and $Y = \{2, 3, 5, 6\}$.

The *complement* $\overline{G}$ of G is the graph where $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{xy \mid xy \notin E(G), x \neq y\}$. Since a stable set in G is a clique in the complement $\overline{G}$, and vice versa, G is a split graph if and only if $\overline{G}$ is a split graph. Földes and Hammer [FH77] have given the following characterization of split graphs.

Theorem 1.2. *The following conditions are equivalent:*

(i) *G is a split graph.*

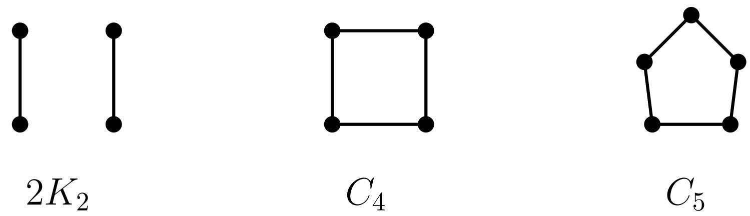

Figure 1.6: The forbidden subgraphs characterizing split graphs.

(ii) *G and $\overline{G}$ are chordal graphs.*

(iii) *G contains none of the graphs $2K_2$, C_4, C_5 as an induced subgraph, (see Figure 1.6).*

For a proof of this theorem and for further reading on chordal graphs and split graphs, see [BLS99], [Gol80] and [MM99]. We will see split graphs again in Chapter 11.

1.4.2 Comparability Graphs and Transitive Orientations

A *transitive orientation* F of graph $G = (V, E)$ is an assignment of a direction, or orientation, to each edge in E such that if $xy \in F$ and $yz \in F$ then $xz \in F$. A graph is called a *comparability graph* if it has a transitive orientation. For example, the even length chordless cycles $C_{2k} (k \geq 2)$ are comparability graphs, but the odd length chordless cycles C_5, C_7, etc. are not comparability graphs. Comparability graphs are also known as *transitively orientable (TRO)* graphs. Additional examples of comparability graphs and their transitive orientations can be found in Figure 1.7. Figure 1.8 shows several graphs which have no transitive orientation. Gallai [Gal67] gave a list of forbidden subgraphs that characterizes the class of comparability graphs, (see also [Duc84]). The name "comparability" graph comes from the observation that relation F is a strict partial ordering of V whose comparability relation is precisely E. We will discuss more about ordered sets in Section 1.5.

Comparability graphs can be recognized, and a transitive orientation can be produced, using the following well known greedy method: (a) Choose an orientation of an arbitrarily chosen edge. (b) Propagate all other orientations forced by this and all subsequently oriented edges (usually called the

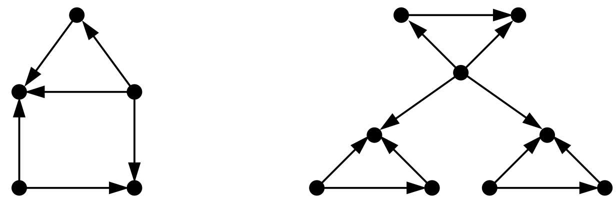

Figure 1.7: Some transitive orientations.

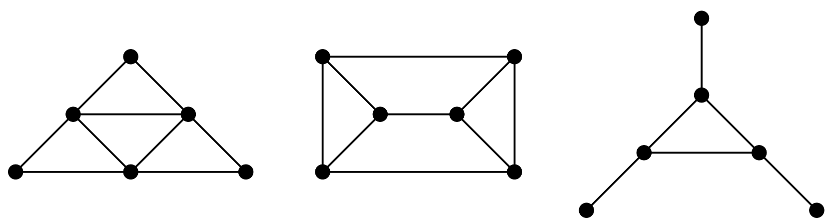

Figure 1.8: Some graphs which are *not* transitively orientable.

implication class). If at some point an edge is forced in both opposite directions, exit with failure. (c) When no other orientations are forced, add the oriented edges to F and remove them from E. If the graph still has some edges, repeat this sequence of steps. When this algorithm finishes, F will be a transitive orientation. The reader unfamiliar with this topic is referred to [Gol80, Gol84]. This method can be implemented to run in $O(n \cdot e)$ time for a graph with n vertices and e edges, or by a more careful counting $O(\sum_{v \in V} d_v^2)$, where d_v is the degree of v. (The *degree* of a vertex v is the number of edges that have v as an endpoint, that is, $d_v = |\mathcal{N}(v)|$.)

Asymptotically faster algorithms for recognizing comparability graphs, which use a technique called modular decomposition, have been given in [MS99]. In [MS99], the authors show how to find an orientation F of an arbitrary graph G such that F is a TRO of G if and only if G is a comparability graph. This is very good if there is other information guaranteeing that G is a comparability graph. However, this alone does not recognize comparability graphs, since the algorithm simply produces an orientation which is *not* transitive when G is *not* a comparability graph. Hence, to complete it to a recognition algorithm, one must test F to determine if it is transitive. The complexity of their method uses $O(n+e)$ time to produce F and $O(n^{\alpha})$

to test whether F is transitive, where $O(n^\alpha)$ is the complexity to perform transitive closure or $n \times n$ matrix multiplication (currently $n^{2.376}$).

The complements of comparability graphs, called *cocomparability graphs*, are of particular interest in this book since, as we will see in the next chapter, all bounded tolerance graphs are cocomparability graphs. Cocomparability graphs also have a characterization as the intersection graphs of function diagrams [GRU83] which we present in Section 1.6.

1.4.3 Interval Graphs

We defined interval graphs in Section 1.2 as being the intersection graphs of intervals on a line. Interval graphs have several important characterizations which we will review here. One of these is the equivalence of interval graphs and the graphs that are both chordal and cocomparability. A second relates to the notion of an asteroidal triple of vertices which we now define.

Three vertices $v_1, v_2, v_3 \in V(G)$ form an *asteroidal triple*(AT) of G if, for all permutations i, j, k of {1,2,3}, there is a path from v_i to v_j which avoids using any vertex in the *closed neighborhood* $\mathcal{N}[v_k] = \{v_k \cup \mathcal{N}(v_k)$. An easy way to verify this for v_k is to delete $\mathcal{N}[v_k]$ and test whether v_i and v_j remain in the same connected component of $G - \mathcal{N}[v_k]$. It also follows from the definition that the three vertices of an asteroidal triple are pairwise nonadjacent. For example, $\{c_1, c_2, c_3\}$ is an asteroidal triple in the tree T_2 in Figure 1.4.

A graph is called *asteroidal triple free* (*AT-free*) if it contains no asteroidal triple. Golumbic, Monma and Trotter [GMT84] showed that every cocomparability graph is AT-free, which we prove in Theorem 1.13. More recently, Corneil, Olariu and Stewart [COS97] have given other mathematical and algorithmic properties characterizing AT-free graphs. The connection with interval graphs is given in the following theorem. Additional characterizations of interval graphs can be found in [Gol80] and [BLS99].

Theorem 1.3. *The following conditions are equivalent:*

(i) *G is an interval graph.*

(ii) *G is chordal and a cocomparability graph.*

(iii) *G is chordal and AT-free.*

The equivalence (i)$\Leftrightarrow$(ii) is due to Gilmore & Hoffman [GH64], and (i)$\Leftrightarrow$(iii) is due to Lekkerkerker & Boland [LB62] who also gave a list of forbidden subgraphs which characterize interval graphs.

Efficient algorithms which run in $O(n + e)$ time are known for recognizing an interval graph with n vertices and e edges, as well as for solving the coloring, clique and stable set problems on interval graphs [BL76]. As an illustration, let's consider the case where we are given an interval representation $\mathcal{I} = \{I_v \mid v \in V\}$ for $G = (V, E)$, and we want to color the intervals using a minimum number of colors so that intersecting intervals are assigned different colors. This is equivalent to coloring the vertices of G so that adjacent vertices get different colors. The following procedure handles the coloring.

Algorithm for Coloring a Set of Intervals

> Sort the intervals according to their left endpoints. Sweep across the representation from left to right, assigning colors in a first fit manner, that is, when a new interval is encountered, always assign the lowest numbered available color, and when an interval is finished, its color becomes available again.

It is an easy exercise, or a good exam question, to show that this "greedy" coloring algorithm is optimal. In particular, during the left to right sweep, just at the point where the highest numbered color k is used, we will find a clique of size k.

We will see this greedy coloring algorithm again in Chapter 4 being applied to representations of probe graphs. We will also show how to color a tolerance representation in Chapter 9.

1.4.4 Unit Interval Graphs and Proper Interval Graphs

An interval graph G that has a representation in which each interval has the same (unit) length is called a *unit interval graph*. Similarly, if G has a representation in which no interval properly contains another interval, G is called a *proper interval graph*. Clearly, a unit representation is also proper. It is easy to verify that the bipartite graph $K_{1,3}$ does not have a proper interval representation. The following classical result of Roberts [Rob69] tells us that the unit interval graphs are equivalent to the proper interval graphs, and they are further equivalent to the $K_{1,3}$-free interval graphs.

Theorem 1.4. *The following conditions are equivalent:*

(i) *G is a unit interval graph.*

(ii) *G is a proper interval graph.*

(iii) *G is an interval graph and is $K_{1,3}$-free.*

We conclude this section with a very useful lemma, which will allow us to assume certain canonical properties of an interval representation, for example, distinct endpoints. We write $I_x \ll I_y$ to mean that the interval I_x is completely to the left of interval I_y.

Lemma 1.5. *A set of intervals $\mathcal{I} = \{I_v \mid v \in V\}$ can be transformed into another set $\mathcal{I}' = \{I'_v \mid v \in V\}$ in which all interval endpoints are distinct, and this transformation preserves the following relationships:*

(i) $I_x \ll I_y \iff I'_x \ll I'_y$

(ii) $I_x \subset I_y \iff I'_x \subset I'_y$

(iii) $|I_x| = |I'_x|$

In particular, (ii) *shows that the transformation preserves the "proper" property and* (iii) *implies that it preserves the "unit" property.*

Proof. Let $\mathcal{I} = \{I_v \mid v \in V\}$ be a set of intervals where $I_v = [L(v), R(v)]$ for all $v \in V$. If there is a repeated endpoint, let $S = \{L(v), R(v) \mid v \in V\}$ be the set of endpoints in the representation, let ϵ be the smallest positive difference between elements of S, and let s be the smallest repeated endpoint in S. If there exist $x \in V$ with $R(x) = s$, pick the one whose interval I_x is the longest and replace I_x by $I'_x = [L(x) + \epsilon/2, R(x) + \epsilon/2]$. Otherwise, pick $x \in V$ with $L(x) = s$ and $|I_x|$ as large as possible and replace I_x by $I'_x = [L(x) - \epsilon/2, R(x) - \epsilon/2]$. It is not hard to see that this new collection satisfies (i), (ii) and (iii) and there is one fewer pair of elements sharing an endpoint. If necessary, recompute ϵ and repeat until all endpoints are distinct. $\square$

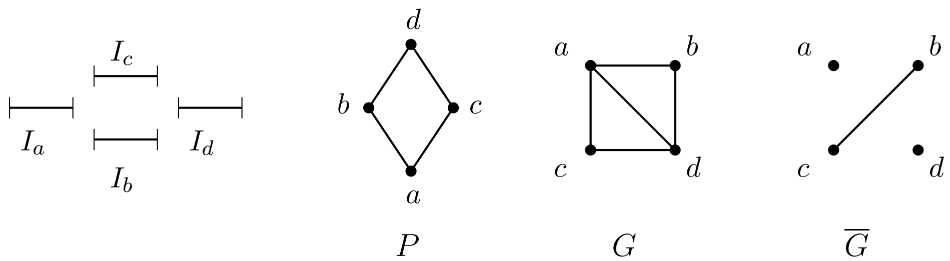

Figure 1.9: An interval representation of ordered set P, its comparability graph G and its incomparability graph $\overline{G}$.

1.5 Ordered Sets

An *ordered set* $P = (X, \prec)$ consists of a ground set X and a binary relation $\prec$ on X which is irreflexive, transitive and therefore asymmetric. Two elements $x, y \in X$ are *comparable* in P if $x \prec y$ or $y \prec x$; otherwise x and y are *incomparable* which we denote $x \parallel y$. We say that y *covers* x if $x \prec y$ and there is no z with $x \prec z \prec y$. Ordered sets (also known as *orders* or *posets*) are often depicted by their *Hasse diagrams* in which edges implied by transitivity are not drawn. For example, Figure 1.9 shows the Hasse diagram of the order P whose only comparabilities are $a \prec b$, $a \prec c$, $b \prec d$, $c \prec d$ and $a \prec d$.

A *linear order* (or *chain*) is one with no incomparabilities and an *antichain* is an order with no comparabilities. The *dual* of the ordered set $P = (X, \prec)$ is the order $P^d = (X, \prec^d)$ with $x \prec y \iff y \prec^d x$.

Two graphs are naturally associated with the order $P = (V, \prec)$. The *comparability graph* $G = (V, E)$ of P has edge set $E = \{xy \mid x \prec y \text{ or } y \prec x\}$ and the *incomparability graph* $\overline{G} = (V, \overline{E})$ has edge set $\overline{E} = \{xy \mid x \parallel y\}$. Figure 1.9 shows an order P and its comparability graph G and its incomparability graph $\overline{G}$. Note that the incomparability graph of any order is always a cocomparability graph and conversely, any cocomparability graph is the incomparability graph of an order.

1.5.1 Interval Orders

An ordered set $P = (V, \prec)$ is an *interval order* if each element $v \in V$ can be assigned a real interval I_v so that $x \prec y \iff I_x$ is completely to the left

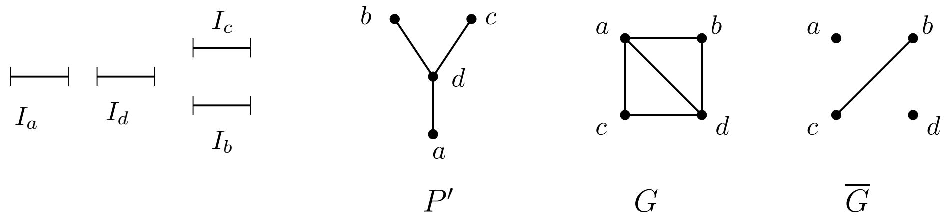

Figure 1.10: A different interval order P' with the same comparability graph and incomparability graph as P in Figure 1.9.

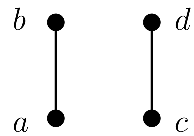

Figure 1.11: The order $\mathbf{2}+\mathbf{2}$.

of I_y. The set of intervals $\{I_v \mid v \in V\}$ is an *interval order representation* of P. The same set of intervals also provides an interval graph representation of the incomparability graph $\overline{G}$ of P since $I_x \cap I_y \neq \emptyset \iff x \parallel y$ in P, as illustrated in Figure 1.9. Note, however, that different interval orders may give rise to the same incomparability graph. For example, the set of intervals in Figure 1.10 gives an interval representation of the order P' and the incomparability graph $\overline{G}$.

The name "interval order" first appears in [Fis70], and [Fis85] gives a modern treatment of the subject. However, its origins go back to Norbert Weiner [Wei14] whose definition of a interval order (which he called a *relation of complete sequence*) was not then known to Fishburn, see [FM92].

Interval orders have a well-known forbidden suborder characterization which we give below. The order $\mathbf{2}+\mathbf{2}$ consists of four elements a, b, c, d whose only comparabilities are $a \prec b$ and $c \prec d$ (see Figure 1.11). More generally, the order $\mathbf{r}+\mathbf{s}$ consists of two chains: one with r elements, the other with s elements, and everything in the first chain is incomparable to everything in the second chain.

Theorem 1.6. *[Fis70] An ordered set is an interval order if and only if it has no suborder isomorphic to* $\mathbf{2}+\mathbf{2}$.

1.5.2 Dimension and Interval Dimension

The intersection of orders $P_1 = (X, \prec_1), P_2 = (X, \prec_2), \ldots, P_k = (X, \prec_k)$ with the same ground set is the order $P = (X, \prec)$ where $x \prec y \iff x \prec_i y$ for $i = 1, 2, \ldots, k$.

A *linear extension* of $P = (X, \prec)$ is a linear order $L = (X, \prec_L)$ so that $x \prec_L y$ whenever $x \prec y$. Thus a linear extension of P has all the comparabilities of P plus additional comparabilities to make L linear. One can show that any order is the intersection of all its linear extensions (Exercise 1.12). This makes the following notion of dimension well-defined.

A *linear realizer* of an order P is a set of linear orders whose intersection is P. The *dimension* of P (denoted $dim(P)$) is the size of a smallest linear realizer of P. It is easy to see that $dim(P) = 2$ and $dim(P') = 2$ for the examples in Figures 1.9 and 1.10. Every transitive orientation of a comparability graph is an order, so it has a well defined dimension. An important result, which we will prove in Section 7.3 is that every transitive orientation of a comparability graph G has the same dimension (i.e., dimension is a comparability invariant.) Thus, we can denote this common value as $dim(G)$. For a comprehensive treatment of dimension theory of ordered sets, see [Tro92].

Similarly, an *interval realizer* of an order P is a set of interval orders whose intersection is P, and the *interval dimension* of P (denoted $idim(P)$) is the size of a smallest interval realizer of P. For example, the order B in Figure 1.12 has $idim(B) = 2$ and an interval realizer of it is shown. The order B can not have interval dimension 1 since it contains suborders isomorphic to $\mathbf{2} + \mathbf{2}$.

The interval dimension is also known to be a comparability invariant [HKM91]. Since linear orders are interval orders, interval dimension is well-defined and $idim(P) \leq dim(P)$ for all P. In Chapters 5 and 10, we will be interested in the class of orders P with $idim(P) \leq 2$.

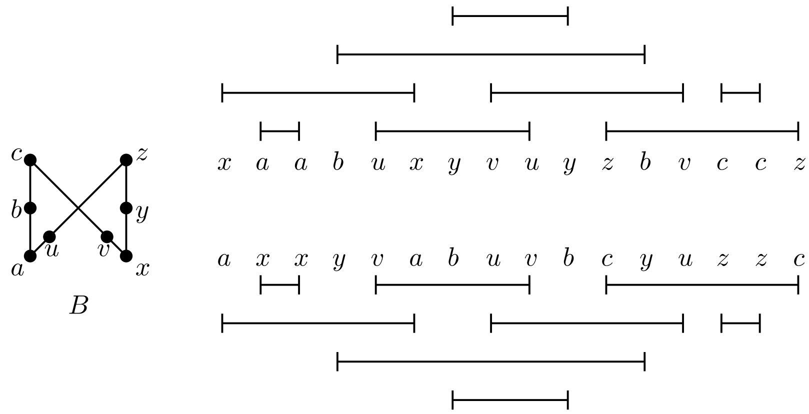

Figure 1.12: The order B with $idim(B) = 2$ and an interval realizer of it.

1.6 The Hierarchy of Permutation, Parallelogram, Trapezoid, Function and AT-free Graphs

In this section, we survey a hierarchy of well-known graph classes arising from intersection diagrams.

A graph $G = (V, E)$ is a *permutation graph* if there is a permutation π of $V = \{1, 2, 3, \ldots, n\}$ so that for vertices i, j we have $ij \in E$ if and only if the order of i and j are reversed in π. For example, the path P_4, with edge set $\{(1, 2), (1, 4), (3, 4)\}$ is a permutation graph using $\pi = [2, 4, 1, 3]$ (see Figure 1.13). If graph G is a permutation graph using π, then its complement $\overline{G}$ is also a permutation graph using the reversal of π. We record this fact as a remark.

Remark 1.7. A graph G is a permutation graph if and only if its complement $\overline{G}$ is a permutation graph.

Alternatively, a permutation graph can be viewed as the intersection graph of line segments in a matching diagram as follows. Write the elements $\{1, 2, 3, \ldots, n\}$ in order on a horizontal line L_1, and underneath write them in the order of π on another horizontal line L_2. For $k = 1, 2, 3, \ldots, n$, connect the two occurrences of k with a straight line segment S_k. Then

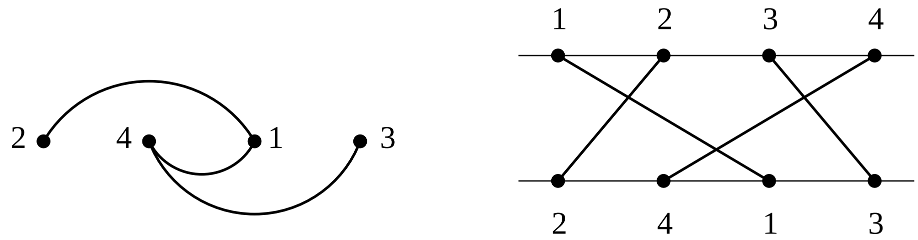

Figure 1.13: The path P_4 as a permutation graph.

$ij \in E(G) \iff S_i \cap S_j \neq \emptyset$ (see Figure 1.13). We call such a representation a *permutation diagram*.

Permutation graphs are characterized by the following theorem. The equivalence (i) $\Leftrightarrow$ (ii) is due to [PLE71] and (ii) $\Leftrightarrow$ (iii) is due to [DM41]. For a proof of this and a more comprehensive treatment of permutation graphs, see [Gol80].

Theorem 1.8. *The following are equivalent.*

(i) *G is a permutation graph.*

(ii) *G is both a comparability graph and a cocomparability graph.*

(iii) $\dim(G) = 2$.

We now successively generalize permutation diagrams and permutation graphs to other geometric forms. Figure 1.14 shows the hierarchy of these classes together with a sample diagram for each.

Let L_1 and L_2 be two horizontal lines with L_1 above L_2. A *parallelogram diagram* consists of L_1, L_2 and a set of n parallelograms $\{P_i \mid i = 1, \ldots, n\}$ where each P_i has parallel sides along L_1 and L_2. A *trapezoid diagram* consists of L_1, L_2 and a set of n trapezoids $\{T_i \mid i = 1, \ldots, n\}$ where the parallel sides of each T_i lie on L_1 and L_2. We allow degenerate trapezoids (and parallelograms), that is, the sides along L_1 and/or L_2 may be points, in which case the resulting trapezoid may be a triangle or a straight line segment. Thus, a permutation diagram is also a parallelogram diagram, which in turn is also a trapezoid diagram.

A continuous curve f connecting a point on L_1 with a point on L_2 is called a *function line* if, whenever two points (x, y) and (x', y') on f have

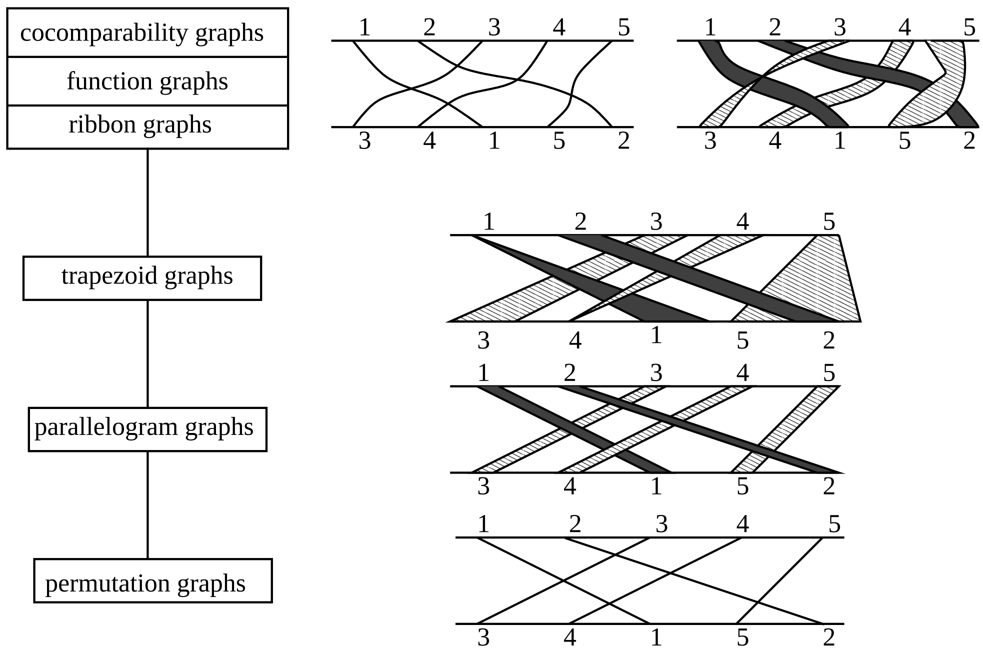

Figure 1.14: A hierarchy of graph classes and their associated intersection diagrams.

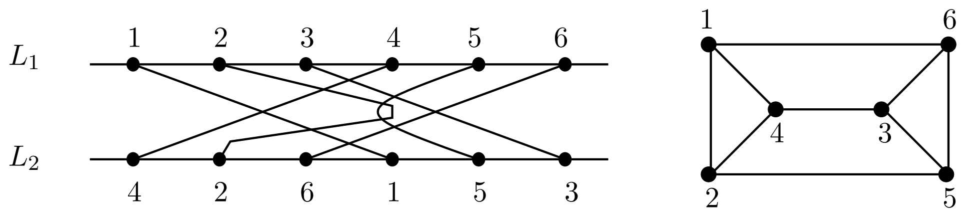

Figure 1.15: A function diagram and its intersection graph (which is isomorphic to $\overline{C_6}$).

the same horizontal value $y = y'$, the points must be equal, i.e., $x = x'$. A *function diagram* consists of L_1, L_2 and a set of n function lines connecting points on L_1 and L_2. The function diagram in Figure 1.15 has six function lines. Finally, we define a *ribbon* to be the area bounded by two function lines, and a *ribbon diagram* to consist of L_1, L_2 and a set of n ribbons. We note that a trapezoid is a ribbon whose bounding function lines are straight.

Definition 1.9. If R_i and R_j are ribbons (trapezoids, parallelograms), we write $R_i \ll R_j$ if R_i and R_j do not intersect and R_i is completely to the left of R_j; or formally, for every horizontal line L, cutting through the diagram, all points on the interval $R_i \cap L$ are to the left of all points on the interval $R_j \cap L$.

We now define the classes of *parallelogram* graphs, *trapezoid* graphs, *function* graphs and *ribbon* graphs to be the family of intersection graphs of their respectively named diagrams.

Remark 1.10. Clearly, these graph families satisfy the containments:
permutation $\subseteq$ parallelogram $\subseteq$ trapezoid $\subseteq$ ribbon.

In Figure 2.8 we will see these classes again as part of a larger hierarchy in which separating examples are given. The next result justifies the placement of “cocomparability,” “function” and “ribbon” graphs in the same box of Figure 1.14 by proving these classes are equivalent.

Consider the following special type of function diagram in which the curves are piecewise linear. Let $L_1, L_2, \ldots, L_{k+1}$ be horizontal lines each labeled from left to right by a permutation of the numbers $1, 2, \ldots, n$. For each i $(1 \leq i \leq n)$ the curve f_i consists of the union of the k straight line segments which join i on L_t with i on L_{t+1} $(1 \leq t \leq k)$. When $k = 1$, this is

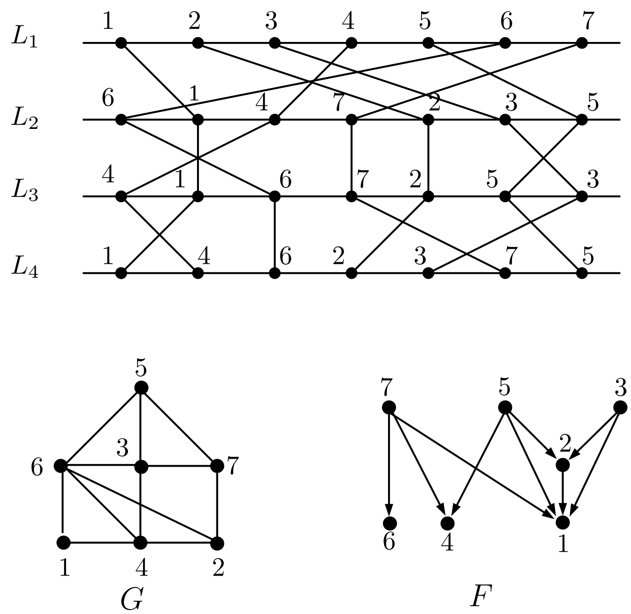

Figure 1.16: A concatenation of three permutation diagrams, its intersection graph G and a transitive orientation F of the complement $\overline{G}$.

just a permutation diagram; when $k \geq 2$, it is called the *concatenation* of k permutation diagrams (see Figure 1.16).

In the following theorem, the equivalences (i) $\Leftrightarrow$ (iii) $\Leftrightarrow$ (iv) are due to Golumbic, Rotem and Uruttia [GRU83] and their equivalence with (ii) was observed in Golumbic and Lewenstein [GL00].

Theorem 1.11. *The following are equivalent.*

(i) *G is a function graph.*

(ii) *G is a ribbon graph.*

(iii) *G is a cocomparability graph.*

(iv) *G is the intersection graph of a concatenation of permutation diagrams.*

Proof. (iv) $\Longrightarrow$ (i) $\Longrightarrow$ (ii). This is immediate since a concatenation of permutation diagrams is a function diagram, and a function diagram is a ribbon diagram where each pair of bounding curves is equal.

(ii) $\Longrightarrow$ (iii). Let G be the intersection graph of the ribbon diagram whose set of ribbons is $R_1, R_2, \ldots, R_n$. Since i and j are adjacent in the complement $\overline{G}$ if and only if R_i and R_j do *not* intersect, we may define an orientation F of $\overline{G}$ as follows

$$ij \in F \Longleftrightarrow R_i \ll R_j$$

where $R_i \ll R_j$ is defined in Definition 1.9. The orientation F is obviously transitive, so $\overline{G}$ is a comparability graph.

(iii) $\Longrightarrow$ (iv). Let $\overline{G}$ be the comparability graph of an order $P = (X, \prec)$, and let $\mathcal{L} = \{L_1, L_2, \ldots, L_{k+1}\}$ be a realizer of P. We may assume, without loss of generality, that $X = \{1, 2, \ldots, n\}$. We will build a concatenation of permutation diagrams whose intersection graph will be G.

For each linear order L_i $(1 \leq i \leq k+1)$, draw a horizontal line and label n points on the line with the elements of X from left to right according to the order L_i. We also use L_i to denote this line and its labeled points. We stack these $k+1$ horizontal lines as shown in the example in Figure 1.16. Let the curve f_i consist of the union of the k straight line segments which join i on L_t with i on L_{t+1} $(1 \leq t \leq k)$. We will show that this concatenation of permutation diagrams represents G. If $ij \in E(G)$, then i and j are not comparable in P, so there are linear orders $L_r, L_s \in \mathcal{L}$ such that $i \prec_r j$ and $j \prec_s i$. Therefore, f_i and f_j intersect somewhere within the area between the horizontal lines L_r and L_s. Otherwise, if $ij \in E(\overline{G})$, then either $i \prec_t j$ for all $L_t \in \mathcal{L}$ and f_i lies completely to the left of f_j, or $j \prec_t i$ for all $L_t \in \mathcal{L}$ and f_i lies completely to the right of f_j. In either case, f_i and f_j do not intersect, which completes the proof of the theorem. $\square$

By choosing $\mathcal{L}$ to be a minimum realizer above, Golumbic, Rotem and Uruttia [GRU83] proved the following result which we state as a remark.

Remark 1.12. If ℓ is the minimum value for which G is the intersection graph of a concatenation of ℓ permutation diagrams, then $dim(\overline{G}) = \ell + 1$.

We conclude this section by adding asteroidal triple free (AT-free) graphs to our hierarchy of Figure 1.14, showing that

$$\text{cocomparability} \subseteq \text{AT-free}.$$

Indeed, this inclusion is proper because the chordless cycle C_5 is AT-free but is not a cocomparability graph.

Theorem 1.13 ([GRU83]). *All cocomparability graphs are AT-free.*

Proof. If $G = (V, E)$ is a cocomparability graph, then G is the intersection graph of a function diagram D, by Theorem 1.11. Suppose, for a contradiction, that G has an asteroidal triple $\{a, b, c\}$, and consider their associated function lines f_a, f_b, f_c in the diagram D. Since a, b, c are pairwise nonadjacent, the curves f_a, f_b, f_c do not intersect one another. Therefore, one of them, say f_b, lies totally between the other two.

Now consider what happens if we remove f_b and all curves which intersect it. We will obtain a function diagram for $G_{V-\mathcal{N}[b]}$ in which a and c are separated into distinct connected components. This contradicts the assumption that $\{a, b, c\}$ is an asteriodal triple, and proves the theorem. $\square$

1.7 Other Families of Graphs

1.7.1 Weakly Chordal Graphs

Weakly chordal graphs, as the name suggests, are a generalization of chordal graphs. They have gained interest in the recent literature, and will play an important role in our study of tolerance graphs in the next chapter.

Hayward [Hay85] introduced the class of *weakly chordal* graphs (also called *weakly triangulated*) as those with no induced subgraph isomorphic to C_n or to $\overline{C_n}$ for $n \geq 5$. Since $\overline{C_5} = C_5$, and $\overline{C_n}$ contains induced copies of C_4 for $n \geq 6$, the class of weakly chordal graphs contains the class of chordal graphs.

We will call vertices x and y a *two-pair* if every chordless path between x and y has exactly two edges. The weakly chordal graphs have been characterized using two-pairs as follows.

Theorem 1.14. *The following are equivalent.*

(i) *G is a weakly chordal graph.*

(ii) *Every induced subgraph of G is either a clique or has a two-pair.*

(iii) *If edges are repeatedly added between two-pairs in G, the result is eventually a clique.*

The implication (ii)$\Longrightarrow$(i) follows from the observation that nonadjacent vertices in C_k or $\overline{C_k}$ (for $k \geq 5$) are not a two-pair (a good exercise). The implication (i)$\Longrightarrow$(ii) is due to Hayward, Hoàng Maffray [HHM90], and (i)$\Leftrightarrow$(iii)

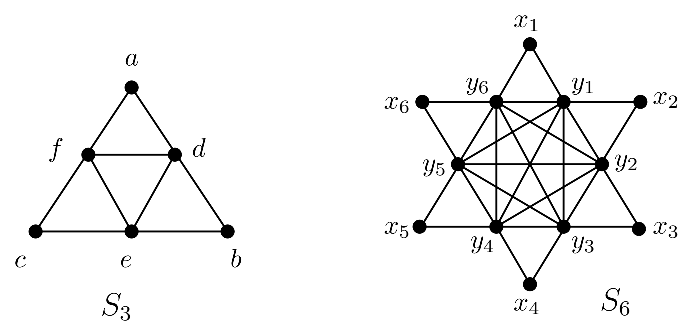

Figure 1.17: The suns S_3 and S_6.

is due to Spinrad and Sritharan [SS95]. The latter equivalence also leads to an $O(n^4)$ recognition algorithm for weakly chordal graphs.

1.7.2 Strongly Chordal Graphs

The strongly chordal graphs have been studied only recently and specialize chordal graphs in several ways. We will encounter these graphs in Chapters 11 and 12. We next define strongly chordal graphs and give additional definitions which will be used in Theorem 1.16 to characterize strongly chordal graphs using chords of a cycle, forbidden subgraphs and elimination orderings.

Let $C = [u_1, u_2, \ldots, u_{2k}, u_1]$ be a cycle of even length $2k \geq 6$. A chord $v_i v_j \in E(G)$ is called an *odd chord* if one of i and j is even and the other is odd, that is, it divides C into two even length cycles.

A graph G is defined to be a *strongly chordal* if it is chordal and every cycle of even length greater than or equal to 6 has an odd chord. The graph in Figure 1.3 is strongly chordal, however, the graph S_3 in Figure 1.17 is not strongly chordal since the even cycle $[a, d, b, e, c, f, a]$ has no odd chord.

A vertex x is called a *simple* vertex if the following condition holds for closed neighborhoods: for every pair of neighbors y and z of x, either $\mathcal{N}[y] \subseteq \mathcal{N}[z]$ or $\mathcal{N}[z] \subseteq \mathcal{N}[y]$. An ordering of the vertices $[v_1, v_2, \ldots, v_n]$ is called a *simple elimination ordering* for G if v_i is a simple vertex in the graph G_i, for all i, where, as before, $G_i = G_{\{v_i, \ldots, v_n\}}$ denotes the subgraph of G remaining after deleting $\{v_1, \ldots, v_{i-1}\}$. Note that the graph S_3 in Figure 1.17 has no simple vertex, so it does not have a simple elimination ordering.

A *strong elimination ordering* is defined to be an ordering of the vertices $[v_1, v_2, \ldots, v_n]$ where, for all $i < j < k < \ell$, if $v_i v_k, v_i v_\ell, v_j v_k \in E(G)$ then $v_j v_\ell \in E(G)$.

Remark 1.15. It is an easy exercise to verify that simple elimination orderings and strong elimination orderings are special cases of perfect elimination orderings (Exercise 1.10).

The graph S_3 is one of a family of forbidden subgraphs characterizing strongly chordal graphs. They are known in the literature both as suns and as trampolines. The *k-sun* S_k $(k \geq 3)$ consists of $2k$ vertices, a stable set $X = \{x_1, x_2, \ldots, x_k\}$ and a clique $Y = \{y_1, y_2, \ldots, y_k\}$, and edges $E_1 \cup E_2$ where $E_1 = \{x_1y_1, y_1x_2, x_2y_2, y_2x_3, \ldots, x_ky_k, y_kx_1\}$ forms the *outer cycle* and $E_2 = \{y_iy_j \mid i \neq j\}$ forms the inner clique. Figure 1.17 shows the graphs S_3 and S_6 and motivates the name sun. The suns are split graphs, so they are chordal by Theorem 1.2, but they are not strongly chordal since the outer cycle has no odd chord.

The next theorem due to Farber [Far83] summarizes the characterizations of strongly chordal graphs.

Theorem 1.16. *The following are equivalent.*

(i) *G is a strongly chordal graph.*

(ii) *G has a simple elimination ordering.*

(iii) *G is chordal and sun-free.*

(iv) *G has a strong elimination ordering.*

For further reading on strongly chordal graphs, and additional characterizations, see [MM99] and [BLS99].

1.7.3 Threshold Graphs

A graph $G = (V, E)$ is called a *threshold graph* if there exist positive weights a_i $(i \in V)$ and a threshold $t > 0$ such that

$$S \subseteq V \text{ is a stable set } \Leftrightarrow \sum_{s \in S} a_s \leq t.$$

We will see threshold graphs again in Chapters 4, 11 and 12. The class of threshold graphs was introduced by Chvátal and Hammer [CH77] who proved the next characterization theorem. A vertex which is adjacent to every other vertex is called *universal*; a vertex which is adjacent to no other vertex is called *isolated*.

Let $0 < \delta_1 < \delta_2 < \cdots < \delta_m < |V|$ be the vertex degrees of the nonisolated vertices of G, where the δ_i are distinct and there may be many vertices of degree δ_i; further, let $\delta_0 = 0$, even if there are no isolated vertices. The *degree partition* of V is given by $V = D_0 \cup D_1 \cup \cdots \cup D_m$, where D_i is the set of all vertices of degree δ_i. Only D_0 is possibly empty.

Theorem 1.17. *The following are equivalent.*

(i) *G is a threshold graph.*

(ii) *$\overline{G}$ is a threshold graph.*

(iii) *There exist positive weights w_i $(i \in V)$ and a threshold $\theta > 0$ such that $xy \in E \Leftrightarrow w_x + w_y > \theta$.*

(iv) *Repeatedly removing either a universal or an isolated vertex from G results eventually in the empty set.*

(v) *G does not contain any of P_4, C_4 or $2K_2$ as an induced subgraph.*

(vi) *For all distinct vertices $x \in D_i$ and $y \in D_j$, we have $xy \in E \iff i + j > m$.*

Additional equivalent conditions and proofs can be found in [MP95]. Theorem 1.17 immediately implies the following.

Theorem 1.18. *Threshold graphs are chordal, co-chordal, comparability and cocomparability graphs; hence, they are also interval, split and permutation graphs.*

Proof. Let G be a threshold graph. The chordality of G follows from (v) and co-chordality then follows from the equivalence of (i) and (ii). To show G is a comparability graph, we fix an ordering $\prec$ on $V(G)$ using (iii) where $x \prec y$ whenever $w_x < w_y$. Now orient $E(G)$ according to $\prec$. This orientation will be transitive using (iii). Thus G is a comparability graph, and it is also a cocomparability graph using the equivalence of (i) and (ii). The remaining conclusions follow from Theorems 1.3, 1.2 and 1.8. $\square$

1.8 Other Reading and General References

In this book, it would be impossible to present all of the topics in graph theory that would be of interest to a researcher studying tolerance graphs. For further reading and reference we offer a modest list of important works that should be consulted.

- M.C. Golumbic, "*Algorithmic Graph Theory and Perfect Graphs",* Academic Press [1980] provides an introduction to classes of perfect graphs such as comparability graphs, chordal graphs and interval graphs. In addition to the mathematical foundations, there is an emphasis on applications as well as algorithms and complexity.

Four books have appeared recently which cover advanced research in this area. They are the following, and are a must for any graph theory library.

- A. Brandstädt, V.B. Le and J.P. Spinrad, "*Graph Classes: A Survey*", SIAM, Philadelphia, [1999] is an extensive and invaluable compendium of the current status of complexity and mathematical results on hundreds on families of graphs. It is comprehensive with respect to definitions and theorems, and citing over 1100 references.

- T.A. McKee and F.R. McMorris, "*Topics in Intersection Graph Theory*", SIAM, Philadelphia, [1999] is a focused monograph on structural properties, presenting definitions, major theorems with proofs and many applications.

- N.V.R. Mahadev and U.N. Peled, "*Threshold Graphs and Related Topics*", North-Holland, [1995] is a thorough and extensive treatment of all research done in the past years on threshold graphs, threshold dimension and orders, and a dozen new concepts which have emerged.

- W.T. Trotter, "*Combinatorics and Partially Ordered Sets*", Johns Hopkins University Press, Baltimore, [1992] is a valuable book which covers new directions of investigation and research on ordered sets with an emphasis on dimension theory.

Other important classical books are Roberts [Rob76] and Fishburn [Fis85]. All these references illustrate the many uses of the intersection graph model, which has become a necessary and important tool for solving real-world problems, and the rich mathematical structures motivated by them.

Temporal Reasoning. One of the "traditional" applications of interval graphs is reasoning about time intervals, which started with the original questions of Hajós in 1957 and Benzer in 1959 (see [Gol80] page 171). Temporal reasoning is an essential part of many applications in artificial intelligence (AI). Given a set of explicit relationships between certain events, we would like to be able to infer additional relationships which are implicit in those given. For example, the transitivity of "before" and "contains" may allow us to derive information regarding the sequence of events. Seriation problems ask for a mapping of temporal events onto the time line such that all the given relations are satisfied, that is, a consistent scenario. Similarly, there are problems of scheduling, planning, and story understanding in which one is interested in constructing a time line where each particular event or phenomenon or task corresponds to an interval representing its duration.

Allen [All83] introduced a model for temporal reasoning using the thirteen primitive interval relations obtained by considering all possible orderings of their four endpoints. Several authors working in AI have studied and adapted Allen's model further, and have incorporated such models into reasoning systems. The paper by Golumbic and Shamir [GS93] has provided a bridge linking some of these temporal reasoning notions from the AI community with those of the combinatorics community and extending results in both disciplines. We also refer the reader to Golumbic [Gol98] which is a survey paper[1] in the same spirit as this book. It describes a number of directions of current work on reasoning about time, many of which employ graph algorithms.

1.9 Exercises

Exercise 1.1. Let $\mathcal{I} = \{I_i\}$ for $i = 0, ..., 6$ where $I_i = [i, 8 + 6i - i^2]$.

(a) What is the interval graph represented by $\mathcal{I}$?

(b) If $t_i = 2i + 1$, what is the tolerance graph represented by $\langle \mathcal{I}, \{t_i\} \rangle$?

(c) If $t'_i = 7 - i$, what is the tolerance graph represented by $\langle \mathcal{I}, \{t'_i\} \rangle$?

(d) What is the size of the largest clique in each of these graphs?

(e) What is the size of the largest stable set in each of these graphs?

[1] This survey paper also includes some of that author's newest illustrative stories, "Will Allan get to Judy's in time?" and "Goldie and the Four Bears".

Exercise 1.2. Find a tolerance representation for the chordless 4-cycle C_4 in Figure 1.6.

Exercise 1.3. Find a maximum cardinality search (MCS) ordering for each of the graphs in Figure 1.8. Check whether the reversal of these MCS orderings are perfect elimination orderings. Explain your findings in terms of Theorem 1.1.

Exercise 1.4. Prove Theorem 1.2.

Exercise 1.5. (a) Give a transitive orientation (TRO) for the graph in Figure 1.3.

(b) Give an argument for why each of the graphs in Figure 1.8 does not have a transitive orientation.

Exercise 1.6. Let $G = (V, E)$ be a graph, and let $\overline{G} = (V, \overline{E})$ be its complement. Prove the following:

If F_1 is a TRO of G and F_2 is a TRO of $\overline{G}$, then $F_1 \cup F_2$ is transitive, i.e., a TRO of the complete graph.

Exercise 1.7. At the Center for Disease Research each new researcher (i.e., doctoral student) visits the Germ Exposure Room once during the first day of the semester, and is exposed to all the bacteria of everyone who is there at the time. How can we assign the researchers to a minimum number of offices in such a way that no one will be exposed to a new person? Give a graph theoretic solution.

Exercise 1.8. Let $G_{20} = (V, E)$ be a graph with vertices $\{v_1, v_2, ..., v_{20}\}$ and edges $(v_i, v_j) \in E \iff i + j \geq 18$.

(a) What is the size of the largest clique of G_{20}?

(b) Prove that G_{20} is an interval graph.

(c) Find a perfect elimination ordering for the vertices of G_{20}.

Exercise 1.9. What graph is represented by the intersection diagrams in Figure 1.14? Show that this graph is not a threshold graph.

Exercise 1.10. Show that all simple elimination orderings and all strong elimination orderings are perfect elimination orderings.

Exercise 1.11. LALE Airline has published the following schedule and has exactly four B737 and two B757 aircraft available.

Flight	Departs TelAviv	Arrives TelAviv	Aircraft
TelAviv-Athens-TelAviv #1	7:00	12:30	B757
TelAviv-Athens-TelAviv #2	11:30	17:00	B737
TelAviv-Athens-TelAviv #3	13:00	18:30	any
TelAviv-Athens-TelAviv #4	16:00	21:30	any
TelAviv-Rome-TelAviv #5	9:00	19:30	B757
TelAviv-Cairo-TelAviv #6	10:30	15:00	B737
TelAviv-Istanbul-TelAviv #7	19:00	23:50	any
TelAviv-Amman-TelAviv #8	16:30	19:30	B737
TelAviv-Milan-TelAviv #9	15:00	23:50	B757

(a) Assume that minimum "ground time" between flights is 75 minutes. Can LALE meet its schedule above? Explain why.

(b) What is the minimum number of B757 aircraft required if LALE adds the three additional flights below? Explain your answer in terms of interval graphs.

Additional Flights	Departs TelAviv	Arrives TelAviv	Aircraft
TelAviv-Bucharest-TelAviv #1	6:30	13:30	B757
TelAviv-Athens-TelAviv #2	14:30	20:00	B757
TelAviv-Eilat-TelAviv #3	21:00	23:30	B757

Exercise 1.12. Show that any order is the intersection of all its linear extensions.

Exercise 1.13. Give a transitive orientation F for the chordless 6-cycle C_6, and draw the associated Hasse diagram for this order. Prove that this order has dimension 3 and interval dimension 3.

Exercise 1.14. The graph G in Figure 1.16 is a cocomparability graph since its complement $\overline{G}$ has a transitive orientation. Does G have a transitive orientation? Is G a permutation graph? Why?

Exercise 1.15. Let G be a chordal graph and $n = |V(G)|$. Show that the number of maximal cliques in G is at most n. (Hint: Let $[v_1, v_2, ..., v_n]$ be a perfect elimination ordering, and consider the sets $\{v_i\} \cup [\mathcal{N}(v_i) \cap V(G_i)]$).