Characterizing Proximity Trees

Prosenjit Bose* William Lenhart† Giuseppe Liotta‡

Abstract

Complete characterizations are given for those trees that can be drawn as either the relative neighborhood graph, relatively closest graph, Gabriel graph or modified Gabriel graph of a set of points in the plane. The characterizations give rise to linear-time algorithms for determining whether a tree has such a drawing; if such a drawing exists one can be constructed in linear time in the real RAM model. The characterization of Gabriel graphs settles several conjectures of Matula and Sokal [17].

Keywords: Graph drawing, proximity graphs, Delaunay triangulation, Gabriel graph, relative neighborhood graph, relatively closest graph, minimum spanning tree, trees

1 Introduction

Many problems in pattern recognition and classification, geographic variation analysis, geographic information systems, computational geometry, computational morphology, and computer vision use the underlying *structure* (also referred to as the *skeleton* or *internal shape*) of a set of data points revealed by means of a *proximity graph* (see for example [24], [20], [8], [17], [25], [27], [1]). A proximity graph attempts to exhibit the relation between points in a point set. Two points are joined by an edge if they are deemed *close* by some proximity measure. It is the measure that determines the type of graph that results. Many different measures of proximity have been defined.

The *relatively closest graph* was introduced by Lankford [15]. A variant, known as the *relative neighborhood graph*, was introduced by Toussaint [23], who showed that the this graph contains the minimum spanning tree and is contained in the Delaunay triangulation. Combinatorial properties of relative neighborhood graphs were also discussed by Urquhart [26] and by Agarwal and Matoušek [1].

The *Gabriel graph*, introduced by Gabriel and Sokal [10], is a supergraph of the relative neighborhood graph. Matula and Sokal [17] established various properties of Gabriel graphs; moreover, they gave an optimal $O(n \log n)$ algorithm for computing the Gabriel graph of n points in the plane. A variation of the Gabriel graph, known as the *modified Gabriel graph*, was studied by Cimikowski [3].

*Research supported in part by NSERC and FCAR. Département de Mathématiques et Informatique, Université du Québec à Trois-Rivières, C.P. 500, Trois-Rivières, Québec, G9A 5H7, Canada. This research was conducted while the author was at the School of Computer Science of McGill University. jit@uqtr.uquebec.ca

†Department of Computer Science, Williams College, Williamstown, MA 01267. This work has been done when this author was visiting the School of Computer Science of McGill University. lenhart@cs.williams.edu

‡Dipartimento di Informatica e Sistemistica, Università di Roma 'La Sapienza', via Salaria 113, I-00198 Roma, Italia. This work has been done when this author was visiting the School of Computer Science of McGill University. liotta@infokit.ing.uniroma1.it

Other proximity graphs that have been studied are the *β-skeleton* [12], the *sphere of influence graph* [25], and the *γ-neighborhood graph* [27]. An extensive survey on the current research in proximity graphs can be found in Jaromczyk and Toussaint [14].

Much attention has been given over the past several years to developing algorithms for drawing graphs in the plane such that the resulting drawing has certain geometric properties. For example, Dillencourt [6] showed that any triangulation without chords or nonfacial triangles is drawable as a Delaunay triangulation. Di Battista and Vismara [5] characterized the allowable angles of the triangles of a Delaunay triangulation through a non-linear system of inequalities. Other interesting problems deal with trees. Monma and Suri [18] showed that no tree with vertex degree greater than six can be drawn as a minimum spanning tree and provided a drawing algorithm for trees with vertex degree at most five. Recently, Eades and Whitesides [7] have proved that it is $NP-$hard to decide whether a tree with vertices of degree six can be drawn as a minimum spanning tree. Trees that can be drawn as sphere of influence graphs were characterized by Jacobson et al [13]. As for other families of graphs, Lubiw and Sleumer [16] proved that maximal outerplanar graphs can be drawn both as relative neighborhood and Gabriel graphs. For a thorough overview on graph drawing problems and algorithms, the reader is referred to the excellent bibliography of Di Battista, Eades, Tamassia and Tollis [4].

In this paper, we study the problem of determining whether trees can be drawn as certain types of proximity graphs. A tree T can be drawn as a proximity graph when there exists a set of points in the plane such that the proximity graph of that set of points is isomorphic to T. It is known ([17], [26], [3]) that every tree with vertex degree less than or equal to three can be drawn as any of the types of proximity graphs mentioned in the abstract, and no tree with vertex degree greater than five can be drawn as any of those types of proximity graphs. Matula and Sokal [17] conjectured that a Gabriel tree cannot have a vertex of degree greater than four and cannot have two adjacent vertices of degree four. Sendov [22] claimed to have proved both conjectures, however a counter-example to the result used by Sendov in his proof was given by Garcia [11], leaving the conjectures still open. To date no complete graph-theoretic characterization of these classes of graphs has been given.

We give a complete characterization of trees that can be drawn as relative neighborhood graphs, relatively closest graphs, Gabriel graphs and modified Gabriel graphs. Our characterization of Gabriel trees settles in the affirmative the conjectures of Matula and Sokal.

For each type of proximity graph considered, we exhibit a set of trees, called *forbidden trees*, which we show cannot be drawn as that type of proximity graph. We then show that any non-forbidden tree can be drawn as that type of proximity graph. For each type of proximity graph, the test to determine whether a tree is forbidden can be made in time proportional to the size of the tree. Given a tree T that can be drawn as a particular type of proximity graph, we provide a drawing algorithm that computes a set of points such that the proximity graph of the set of points is isomorphic to T. The time complexity of our drawing algorithms, using the real RAM model, is linear in the size of the tree.

This paper is organized as follows: Section 2 gives necessary definitions. Section 3 completes the characterization of modified Gabriel trees. Section 4 presents the characterization of relatively closest trees and relative neighborhood trees as well as describing an algorithm for drawing such trees. Section 5 does the same for Gabriel trees. Closing remarks and open problems can be found in Section 6.

2 Preliminaries

We begin by defining some of the graph theoretic and geometric terminology used in this paper. For more details see [2] and [19].

A *graph* $G = (V, E)$ consists of a finite non empty set $V(G)$ of *vertices*, and a set $E(G)$ of unordered pairs of vertices known as *edges*. An edge $e \in E(G)$ consisting of vertices u and v is denoted by $e = uv$; u and v are called the *endpoints* of e and are said to be *adjacent* vertices or *neighbors*. The *degree* of a vertex $v \in V(G)$, denoted by $deg_G(v)$ (or just $deg(v)$ when no confusion will result), is the number of edges of $E(G)$ which have v as an endpoint. A *path* in a graph G is a finite non-null sequence $P = v_1 v_2 \ldots v_k$ where the vertices $v_1, v_2 \ldots v_k$ are distinct and $v_i v_{i+1}$ is an edge for each $i = 1 \ldots k-1$. The vertices v_1 and v_k are known as the *endpoints* of the path. A *cycle* is a path whose endpoints are the same. An *acyclic* graph is a graph that contains no cycles. A graph is *connected* if, for each pair of vertices $u, v \in V$, there is a path from u to v.

A *tree* is a connected acyclic graph. Let T be a tree, a vertex $v \in V(T)$ such that $deg_T(v) = 1$ is a *leaf* of T. Let T be a tree, $v \in V(T)$, and x is a neighbor of v; $T_x(v)$ denotes the connected component of $T - v$ which contains x. Similarly, $T_x[v]$ denotes $T_x(v)$ along with the vertex v and the edge xv. A *rooted tree* (T, r) is a tree T along with a distinguished vertex r (called the *root* of T). Vertices adjacent to r are called *children* of r and r is called the *parent* of those vertices. Each child x of r is in turn the root of $T_x(r)$. A *subgraph* of a graph $G = (V, E)$, is a graph $H = (V', E')$, where V' is a subset of V and E' is a subset of E such that all endpoints of edges in E' are elements of V'. A *spanning* subgraph of a graph G, is a subgraph of G whose vertex set is the same as the vertex set of G. If a spanning subgraph of a graph G is a tree, then it is known as a *spanning tree* of G. Two graphs are *isomorphic* if there is a bijection of the vertex sets which preserves adjacency.

We will be using a variety of geometric objects to describe our tree drawing algorithms. See also Figure 1. Let $\mathbb{R}^2$ denote the Euclidean plane. Given a point $p \in \mathbb{R}^2$ and a radius $\rho > 0$, $D[p, \rho]$ denotes the closed disk of radius ρ centered at p. We denote with $\overline{D}[p, \rho]$ the complement of $D[p, \rho]$ in $\mathbb{R}^2$. Any sector of $D[p, \rho]$ of angle $\alpha < 2\pi$ is denoted by $S[p, \alpha, d, \rho]$, where d is a vector with origin p, which bisects the angle α of the sector. A sector having angle α is called an *α-sector*. A sector of infinite radius is called a *wedge* and is denoted by $W[p, \alpha, d]$. The arc which forms the common boundary of $D[p, \rho]$ and $S[p, \alpha, d, \rho]$ is denoted by $A[p, \alpha, d, \rho]$. Given a ray r with source point p, we define $HP[r, p]$ to be the closed half-plane not containing r determined by the line through p orthogonal to r. Round brackets in the above notation refers to the open set as opposed to the closed set. For example, $D(p, \rho)$ denotes the open disk of radius ρ centered at p. Let $d(x, y)$ denote the Euclidean distance between x and y. Finally, given three points a, b, c, in general position, $\angle abc$ denotes the *strictly smaller* angle between segment $\overline{ba}$ and segment $\overline{bc}$, unless stated otherwise.

A *drawing* of a graph $G = (V, E)$ is a function which maps the vertices of G to points in the plane and edges of G to curves in the plane such that for each edge $e = uv$, the endpoints of the curve corresponding to e are the points in the plane corresponding to u and v. A drawing of G is called a *planar* drawing if no curve intersects itself or any other curve, except possibly at its endpoints. A graph is said to be planar if it admits a planar drawing. A *straight-line* drawing of a graph G is a drawing in which each edge corresponds to the line segment between its endpoints. All planar graphs admit straight-line planar drawings [9]. A set S of points in the plane, along with a set of line segments between points of S defines a graph in a natural way and at the same time a straight-line drawing of the graph. In this paper we will often refer to such a drawing as the graph itself.

Intuitively speaking, a *proximity graph* on a finite set $S \subset \mathbb{R}^2$ is obtained by connecting pairs of points of S with line segments if the points are considered to be *close* in some sense. Different definitions of closeness give rise to different proximity graphs. One technique for defining a proximity graph on a set of points is to select a geometric region defined by two points of S—for example the

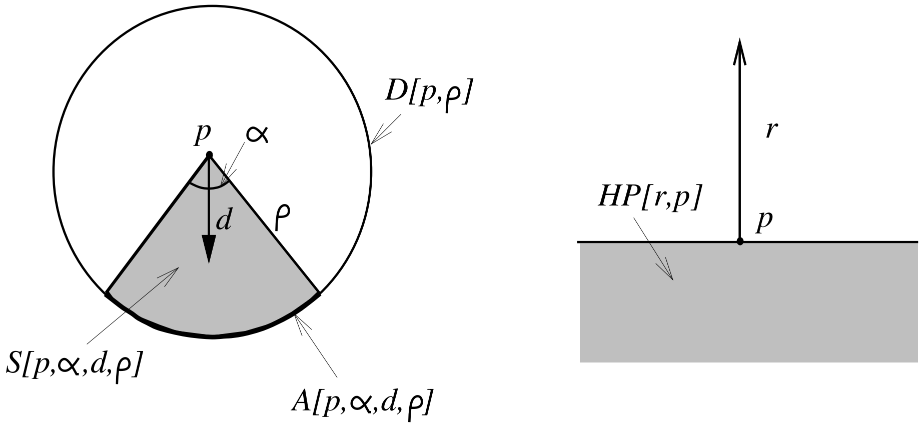


Figure 1: A closed disk, a sector, an arc, and a half-plane.

smallest disk containing the two points—and then specifying that a segment is drawn between the two points if and only if this region contains no other points from S. Such a region will be referred to as a *region of influence* of the two points. Five such definitions follow.

Given a set S of points in $\mathbb{R}^2$, the *relatively closest graph of* S, denoted by $RCG(S)$, has a segment between points u and v in S if the intersection of the closed disks of radius $d(u,v)$ centered at u and v is empty. This region of influence is referred to as the *closed lune* of u and v and is denoted by $L[u,v]$. Equivalently, $u, v \in S$ are adjacent if and only if

$$d(u,v) < \max[d(u,w), d(v,w)], \text{ for all } w \in S, w \neq u, v.$$

An example of such a graph is shown in Figure 2 (b). Observe that $RCG(S)$ need not be connected. The notation $RCG(S)$ is used to represent both the set of points and segments in the plane and the graph which these points and segments define. We say that a graph G is a *relatively closest graph* if G is isomorphic to $RCG(S)$ for some finite set $S \subset \mathbb{R}^2$. $RCG(S)$ is then called an *RCG drawing* of G or a *realization* of G as a relatively closest graph.

The *relative neighborhood graph of* S, denoted by $RNG(S)$, is defined similarly to $RCG(S)$, except that open disks are used instead of closed disks. That is, $RNG(S)$ is obtained by replacing $<$ with $\leq$ in the above inequality. The region of influence between u and v is referred to as the *open lune* of u and v, denoted by $L(u,v)$. An example of such a graph is depicted in Figure 2 (c). The $RNG(S)$ is always connected [23]. A *relative neighborhood graph* is a graph which is isomorphic to $RNG(S)$ for some $S \subset \mathbb{R}^2$.

The *Gabriel graph of* S, denoted by $GG(S)$, has as its region of influence the closed disk having segment $\overline{uv}$ as diameter. That is, two vertices $u, v \in S$ are adjacent if and only if

$$d^2(u,v) < d^2(u,w) + d^2(v,w), \text{ for all } w \in S, w \neq u, v.$$

This region is referred to as the *closed disk* of u and v, denoted as $D[u,v]$. See Figure 2 (d) for an example. A *Gabriel graph* is a graph which is isomorphic to $GG(S)$ for some $S \subset \mathbb{R}^2$.

The *modified Gabriel graph of* S, denoted by $MGG(S)$, is defined similarly to the Gabriel graph of S, with the difference that the open disk is used instead of the closed disk; that is, $\leq$ replaces $<$

in the previous inequality. The region of influence is referred to as the *open disk* of u and v, denoted by $D(u,v)$. An example of $MGG(S)$ can be found in Figure 2 (e). A *modified Gabriel graph* is a graph which is isomorphic to $MGG(S)$ for some $S \subset \mathbb{R}^2$.

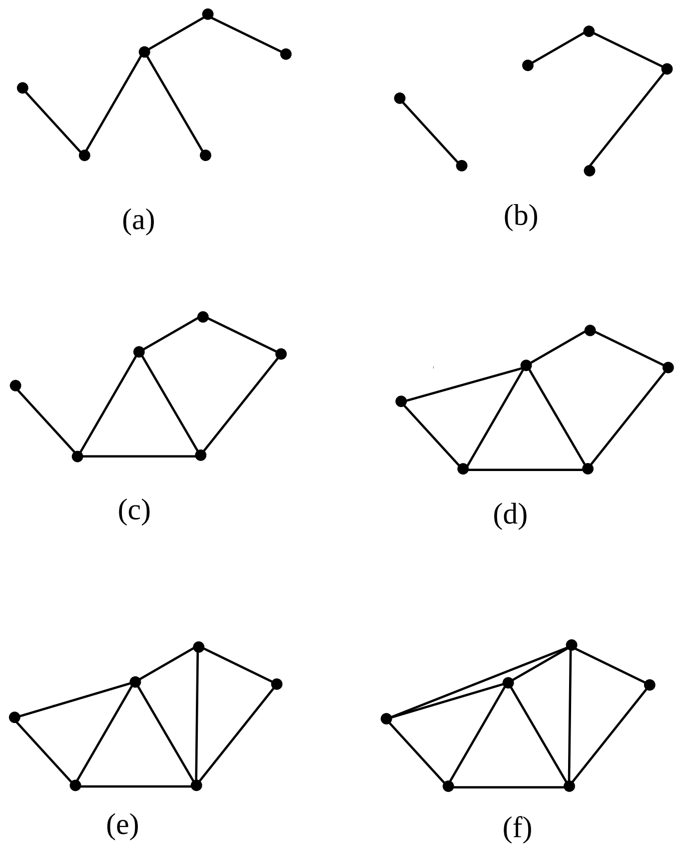


Figure 2: Six different proximity graphs on a set S of six points: (a) $MST(S)$, (b) $RCG(S)$, (c) $RNG(S)$, (d) $GG(S)$, (e) $MGG(S)$, (f) $DT(S)$.

Before describing the final type of proximity graph discussed in this paper, the Delaunay triangulation, we present two definitions which will prove useful in our tree drawing algorithms. Let S be a finite set of points of $\mathbb{R}^2$ and let X and Y be any (not necessarily finite) subsets of points of $\mathbb{R}^2$. We say that S *separates* X and Y if for each $x \in X$ and each $y \in Y$, there is a point in S which lies in the region of influence determined by x and y. Thus if S' is a set of points that contains S, $x \in X \cap S'$ and $y \in Y \cap S'$, then the proximity graph of S' will not contain the edge from x to y. Our algorithms will proceed by representing some of the vertices of the tree to be drawn as a set $S \in \mathbb{R}^2$ and then finding regions separated by S in which we can draw the remaining portions of the tree.

The second definition concerns a certain type of graph drawing. Given some definition of proximity, we say that a rooted tree (T,r) has an α-*drawing* if there exists a set S such that T is isomorphic to the proximity graph of S and S is contained in some α-sector having as apex the point of S corresponding to r. Note that the radius of the sector is irrelevant; if (T,r) has some α-drawing, it can be drawn in any α-sector of any positive radius.

Given a set S of points in the plane and a planar straight-line graph on S, the graph partitions the plane into connected regions, called *faces*, exactly one of which, the *outer face*, is unbounded, and the rest of which, the *interior faces*, each are bounded by a closed curve consisting of edges of the graph. A *triangulation* of S is a planar straight-line graph on S such that every interior face has as its boundary a triangle. A *Delaunay triangulation* of a set S of points in the plane, denoted by $DT(S)$, is a triangulation of S such that for each interior face, the triangle which bounds that face has the property that the circle circumscribing the triangle contains no other points of the graph in its interior. A set S may admit more than one Delaunay triangulation, but only if S contains four or more co-circular points. A list of properties of the Delaunay triangulation can be found in [19]. An example of a Delaunay triangulation is given in Figure 2 (f).

Finally, we describe another graph, a *minimum spanning tree*, which is not defined in terms of a region of influence. Given a set S of points in the plane, consider a connected straight-line graph G on S, that is, a graph having as its edge set E a collection of line segments connecting pairs of vertices of S. Define the *weight* of G to be the sum of all of the edge lengths of G. Such a graph is called a *minimum spanning tree of* S, denoted by $MST(S)$, if its weight is no greater than the weight of any other connected straight-line graph on S. (It is easy to see that such a graph must be a tree.) In general, a set S may have many minimum spanning trees (for example, if S consists of the vertices of a regular polygon). An example of a minimum spanning tree is depicted in Figure 2 (a).

The following relationships among the different proximity graphs hold for any finite set S of points in the plane.

Lemma 2.1 *[3]* $RCG(S) \subseteq RNG(S) \subseteq GG(S) \subseteq DT(S)$

Lemma 2.2 *[23]* $MST(S) \subseteq RNG(S)$.

Lemma 2.3 *[3]* $GG(S) \subseteq MGG(S)$

$MST(S)$ need not be a subgraph of $RCG(S)$ since the latter is not even guaranteed to be connected; if, however, $RCG(S)$ is connected, then it does contain $MST(S)$. It is worth noting that in several papers $GG(S)$ is defined using open disks instead of closed disks (i.e. in these papers $GG(S)$ is what we have called $MGG(S)$). The definition we have chosen to adopt in the present paper is the one given by Gabriel and Sokal in the paper which first defines Gabriel graphs [10]. The distinction is worth noting since, for example, $GG(S)$ is guaranteed to be a subgraph of $DT(S)$ whereas $MGG(S)$ need not be (in fact, $MGG(S)$ may contain crossing segments). In fact, the characterization of GG trees and MGG trees will also exhibit this distinction.

3 Modified Gabriel Trees

In this section we characterize modified Gabriel trees. Matula and Sokal [17] showed that for any tree T of maximum degree at most three, a GG drawing of T can be computed in time proportional to the size of T in the real RAM model. It is easy to see that the drawings they compute are also MGG drawings. Thus our task in this section is to show that a modified Gabriel tree cannot have a vertex of degree greater than three. Let S be a finite set of points in the plane.

Lemma 3.1 *If $MGG(S)$ is a tree then the angle between any two adjacent edges is greater than $\pi/2$.*

Proof: Suppose v, x, y are vertices in S such that vx, vy are in $MGG(S)$. We will show that $d^2(x,y) > d^2(x,v) + d^2(y,v)$, which implies that $\angle xvy > \pi/2$. Let T_x and T_y be the subtrees, containing x and y respectively, obtained by removing v from $MGG(S)$ (see Figure 3).

Let $x' \in T_x$ and $y' \in T_y$ be two vertices such that

$$d(x', y') \leq d(\overline{x}, \overline{y}), \forall \overline{x} \in T_x \text{ and } \forall \overline{y} \in T_y.$$

D(x', y') does not contain any points of T_x or T_y by definition. It must, however, contain some point z in $MGG(S) - T_x - T_y$ since there is not an edge between x' and y' in $MGG(S)$. $MGG(S)$ is a tree, thus there must be a unique path from v to z, denoted by $P(v, z)$. This path contains no vertices of T_x or T_y, since z is in a connected subgraph of $MGG(S) - v$ different from both T_x and T_y. We can therefore conclude the following:

1. Since $MGG(S)$ is a minimum spanning tree, by Lemmas 2.1 and 2.2, and since $P(v, z) \cup P(v, x') \cup x'z$ forms a cycle where every edge is contained in $MGG(S)$ except for $x'z$, we have that $d(x', z) > d(x, v)$.
2. $d(y', z) > d(y, v)$ by the same argument as above.
3. $d(x, y) \geq d(x', y')$ by the relation between x' and y'.
4. $\angle x'zy' > \pi/2$ since z is contained in $D(x', y')$.

Therefore,

$$d^2(x,y) \geq d^2(x', y') > d^2(x', z) + d^2(y', z) > d^2(x, v) + d^2(y, v),$$

which concludes the proof.

Corollary 3.1 *A modified Gabriel tree cannot have a vertex of degree greater than three.*

As mentioned at the beginning of the section, the work of Matula and Sokal [17],implies that any tree of maximum degree at most three admits an MGG drawing, which can be efficiently computed. This, along with Corollary 3.1, completes the characterization of modified Gabriel trees, which we summarize in the following theorem.

Theorem 3.1 *A tree T is a modified Gabriel tree if and only if the maximum vertex degree of the tree is at most three. Furthermore, in the real RAM model, an MGG drawing of T can be computed in time proportional to the size of T.*

4 Relatively Closest and Relative Neighborhood Trees

In this section we characterize relatively closest and relative neighborhood trees. Cimikowski [3] has shown that no tree with a vertex of degree greater than five can be a relatively closest or relative neighborhood tree. We show here that any tree with maximum degree at most five is a relatively closest and a relative neighborhood tree. This immediately implies the result of Monma and Suri [18], that any tree having maximum degree at most five can be drawn in the plane as a minimum spanning tree of some set of points. We do this by exhibiting a drawing algorithm for such trees which produces a drawing similar to that given by Monma and Suri.

The following lemmas form the geometric foundations for the drawing algorithm.

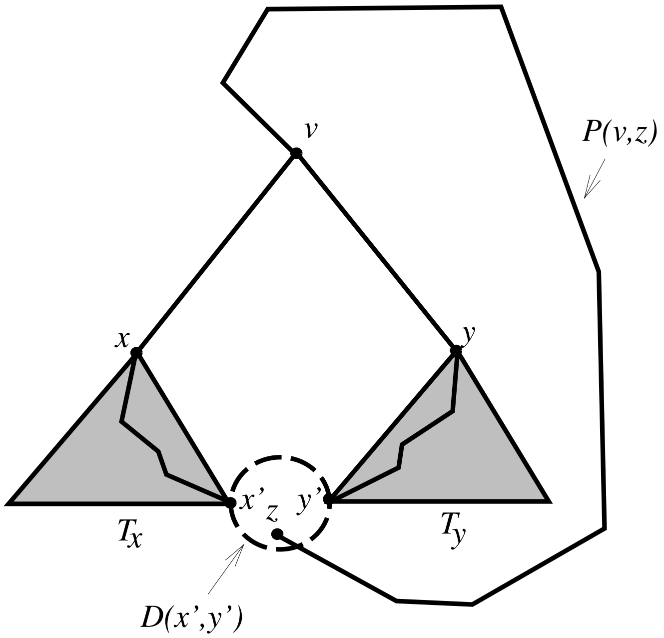


Figure 3: Illustration for Lemma 3.1.

Lemma 4.1 *Let $P = \{p_0, p_1, p_2\}$ be a set of points of $\mathbb{R}^2$ such that $\pi/3 < \angle p_1 p_0 p_2 < \pi$ and $d(p_0, p_1) = d(p_0, p_2)$. For any $\rho < (d(p_1, p_2) - d(p_0, p_1))/3$, $\{p_0\}$ separates $D[p_1, \rho]$ and $D[p_2, \rho]$ in both the relative neighborhood and relatively closest proximity graphs.*

Proof: Let ρ satisfy $0 < \rho < (d(p_1, p_2) - d(p_0, p_1))/3$. Let x be a point in $D[p_1, \rho]$ and let y be a point in $D[p_2, \rho]$. See Figure 4 (a). Note that $d(x, y) \geq d(p_1, p_2) - 2\rho$ and that $d(p_0, x) \leq d(p_0, p_1) + \rho$ by assumption. But, $d(p_1, p_2) - 2\rho > d(p_0, p_1) + \rho$. Therefore, $d(x, y) > d(p_0, x)$. By the same argument $d(x, y) > d(p_0, y)$. Thus, $L(x, y)$ contains p_0 (as does $L[x, y]$).

Lemma 4.2 *Given a disk $D[p_0, r]$ and a point p on the boundary of the disk. For any $0 < \rho < r$, there exists an arc $A = A[p, \alpha, d, \rho]$, with $\alpha > \pi$ and ray d with origin p_0 bisecting α, such that for every point $x \in A$, $d(x, p_0) > d(p, p_0)$.*

Proof: Given any $0 < \rho < r$, let a and b be the points of intersection between $D[p, \rho]$ and $D[p_0, r]$. See Figure 4 (b). Let β be the *strictly larger* angle between $\overline{pa}$ and $\overline{pb}$. Clearly, $\beta > \pi$. Let d be the bisector of β and let α be any angle $\pi < \alpha < \beta$. Consider the arc $A = A[p, \alpha, d, \rho]$; since A is entirely contained in $\overline{D}[p_0, r]$, for every point $x \in A$ we have that $d(x, p_0) > d(p, p_0)$.

We use the above tools to develop the following drawing algorithm.

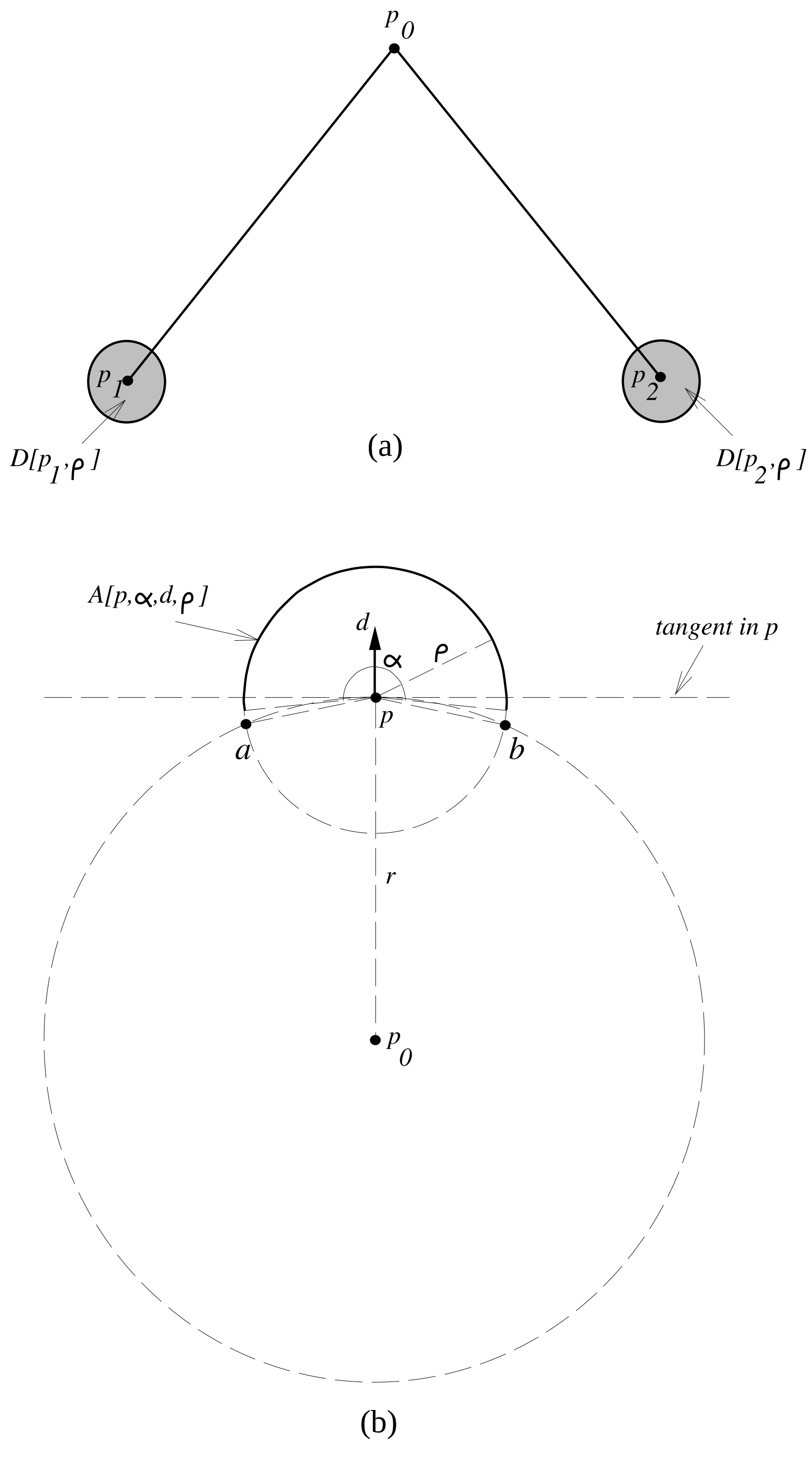


Figure 4: Illustrations for Lemma 4.1 and for Lemma 4.2. (a) The disks separated by $\{p_0\}$. (b) Arc $A[p, \alpha, d, \rho]$.

procedure $DrawRCG(T, u, p_0, r, p, \rho)$

Draw u at p.

If $deg_T(u) > 0$ then

Choose any $\rho', 0 < \rho' < \rho$.

Let $v_i, i = 1, \ldots, k$ be the neighbors of u.

Choose an angle $\alpha > \pi$ and a vector d with origin p such that for every point $x \in A[p, \alpha, d, \rho']$, $d(x, p_0) > d(p, p_0)$.

Choose points $p_i, i = 1, \ldots k$ such that

1. $p_1, \ldots p_k$ appear in clockwise order along arc $A[p, \alpha, d, \rho']$
2. $\angle p_i p p_{i+1} > \pi/3$ for $0 < i < k$

Choose $\rho_1 > 0$ such that

1. $D[p_i, \rho_1]$ is contained in $D[p, \rho] \cap \overline{D}[p_0, r]$
2. $D[p_i, \rho_1]$ and $D[p_{i+1}, \rho_1]$ are separated by $\{p\}$ for $0 < i < k$

For $i \leftarrow 1, \ldots, k$

Draw segment $\overline{pp_i}$.

$DrawRCG(T_{v_i}(u), v_i, p, \rho', p_i, \rho_1)$

end procedure.

Figure 5: Procedure *DrawRCG*.

Lemma 4.3 *Any tree T with maximum vertex degree at most five is a relatively closest (relative neighborhood) tree. Furthermore, in the real RAM model, an RCG (or RNG) drawing of T can be computed in time proportional to the size of T.*

Proof: Let T be a tree with maximum vertex degree at most five. We present a procedure which constructs a set of points $S \in \mathbb{R}^2$ such that both $RCG(S)$ and $RNG(S)$ are isomorphic to T. By guaranteeing that for each pair $u, v \in S$, uv is an edge of T iff both $L(u, v)$ and $L[u, v]$ are empty, the set S will represent T as both a relatively closest and relative neighborhood tree.

Let u be a vertex of T whose degree is at most four, and consider the rooted tree (T, u). Procedure *DrawRCG* (see Figure 5) receives as input (T, u), a point $p_0 \in \mathbb{R}^2$, a radius $r > 0$ (these are used to form the disk $D[p_0, r]$), a point p on the boundary of $D[p_0, r]$ and a radius ρ ($0 < \rho < r$). The output is an RCG and RNG drawing of (T, u). The rooted tree (T, u) will be drawn in the region $D[p, \rho] \cap \overline{D}[p_0, r]$ (see Figure 6(a)).

Notice that the procedure is recursively called on rooted trees where the degree of the root is at most four. Namely, the first call of the procedure has as input a rooted tree with a root of degree at most four (we can guarantee this on the first call by picking any node of degree at most four as the root of the tree), an arbitrary point $p_0 \in \mathbb{R}^2$, an arbitrary radius $r > 0$, any point p on the boundary of $D[p_0, r]$ and any $\rho, 0 < \rho < r$. Since the maximum vertex degree of the tree is at most five, each recursive call of *DrawRCG* will be made with a rooted subtree whose root has degree at most four. Thus, in the proof of correctness to follow, we will assume that the tree to be drawn always has a root of degree at most four.

We prove the correctness of the procedure by induction on the size of the tree. If the input tree has exactly one vertex, procedure *DrawRCG* produces a correct drawing. Suppose that the procedure works correctly on all trees (of maximum degree at most five) having fewer than n vertices

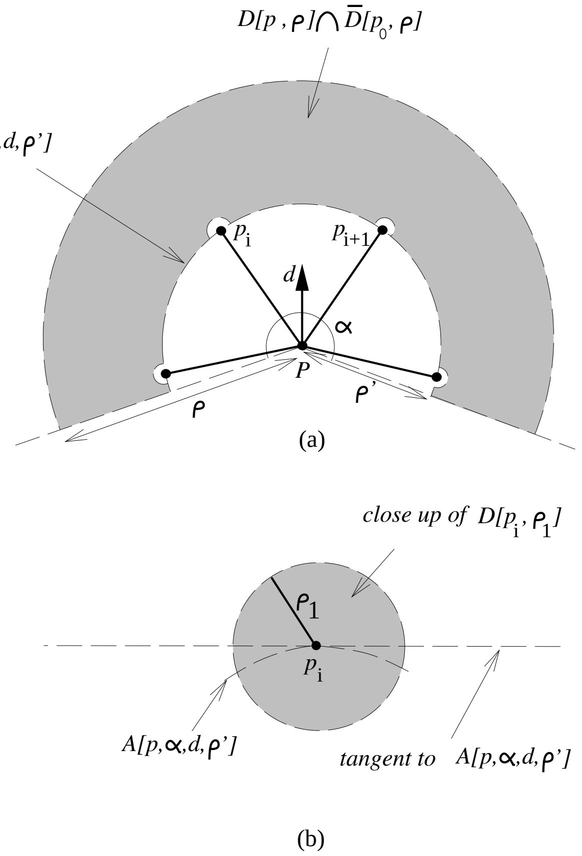


Figure 6: Illustration for procedure *DrawRCG*.

where the root of the tree has degree at most 4. Now, consider a tree (T, u) with n vertices, and root u with degree at most 4; thus, u has k children, where $k \leq 4$

Procedure *DrawRCG* first draws the root u at point p. Then, it defines a new set of points p_i on arc $A[p, \alpha, d, \rho']$ in the region $D[p, \rho] \cap \overline{D}[p_0, r]$ such that $\angle p_i p p_{i+1} > \pi/3$ $(0 < i < k)$ (see Figure 6(a)). The existence of this arc is guaranteed by Lemma 4.2. Thus, $\{p\}$ separates each pair $\{p_i\}$ and $\{p_j\}$ $(i \neq j)$, and $\{p\}$ also separates each $\{p_i\}$ from $\{p_0\}$.

Next, for each $i \leq k$, a disk $D[p_i, \rho_1]$ is computed, where the subtree rooted at p_i will be placed (see Figure 6(b)). The disks $D[p_i, \rho_1]$ are chosen to be properly contained in $D[p, \rho] \cap \overline{D}[p_0, r]$, and such that $D[p_i, \rho_1]$ and $D[p_j, \rho_1]$ are separated by $\{p\}$ for all $i, j, 0 < i < j \leq k$. The existence of such disks is guaranteed by Lemma 4.1. Finally, since each subtree has a root with degree at most four, by the inductive hypothesis, they are correctly drawn. Thus, by induction, the algorithm produces a correct drawing.

In terms of time complexity, since *DrawRCG* performs a constant amount of work per call and is called once for each edge in the tree, the drawing of the tree can be computed in time proportional to the size of the tree.

An example of a drawing produced by *DrawRCG* is given in Figure 7. The first call of *DrawRCG* used vertex u as root; ρ was chosen to be slightly less than r.

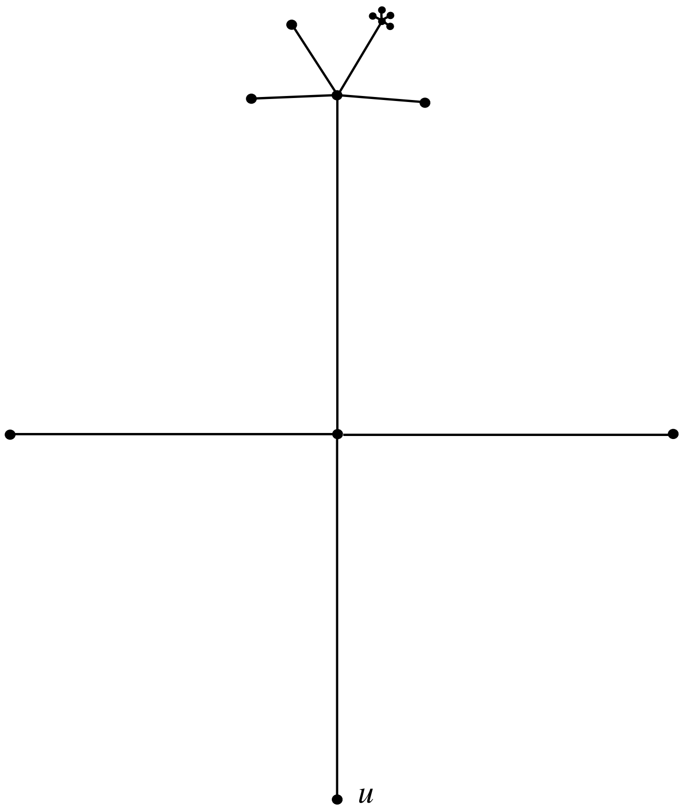


Figure 7: A drawing produced by procedure *DrawRCG*.

Thus, we have a complete characterization of relatively closest and relative neighborhood trees, which we summarize below.

Theorem 4.1 *A tree T is a relatively closest or relative neighborhood tree if and only if the maximum vertex degree of the tree is at most five. Furthermore, in the real RAM model, an RCG or RNG drawing of T can be computed in time proportional to the size of T.*

5 Gabriel Trees

In this section, we characterize Gabriel trees. Matula and Sokal [17] conjectured that a Gabriel tree cannot have a vertex of degree greater than four and cannot have two adjacent vertices of degree four. Sendov [22] claimed to have proved both conjectures, however a counter-example to the result used by Sendov in his proof was shown in [11]. One consequence of our characterization of Gabriel trees is the resolution of both of these conjectures in the affirmative. The section is organized as follows. In Subsection 5.1 we discuss the relationship between the Gabriel tree and the Delaunay triangulation of a set of points. An infinite class of trees that cannot be drawn as Gabriel trees is presented in Subsection 5.2. Finally, a drawing algorithm for non forbidden trees is shown in Subsection 5.3.

5.1 Gabriel Trees and Delaunay Triangulations

Let S be a finite set of points in the plane and let $DT(S)$ be a Delaunay triangulation of S. We investigate some properties that relate $DT(S)$ to $GG(S)$. We first mention a useful property of any triangulation of S.

Observation 5.1 If va and vb are two edges in a triangulation of a set S, then there exists a path from a to b which does not contain v and such that every vertex on the path is adjacent to v.

For any three points x, y and z of S, let $D[x, y, z]$ be the smallest closed disk containing x, y and z, and $D(x, y, z)$ the corresponding open disk.

Lemma 5.1 *Let ab be an edge of $DT(S)$.*

1. *If ab is contained in two triangles $\triangle(abc)$ and $\triangle(abd)$ of $DT(S)$ then $ab \in GG(S)$ iff $\angle acb$ and $\angle adb$ are both less than $\pi/2$.*
2. *If ab is contained in only one triangle $\triangle(abc)$ of $DT(S)$ then $ab \in GG(S)$ iff $\angle acb < \pi/2$.*

Proof: If $\angle acb$ and $\angle adb$ are both less than $\pi/2$ then $D[a, b] \subseteq D(a, b, c) \cup D(a, b, d) \cup \{a, b\}$. Since $D(a, b, c)$ and $D(a, b, d)$ contain no points of S, $D[a, b]$ contains no points of S other than a and b and therefore ab is in $GG(S)$ (see Figure 8). For the other direction, we observe that if either $\angle acb \geq \pi/2$ or $\angle adb \geq \pi/2$ then $D[a, b]$ contains either c or d and so $ab \notin GG(S)$. A similar argument holds in the case that ab is contained in only one triangle of $DT(S)$.

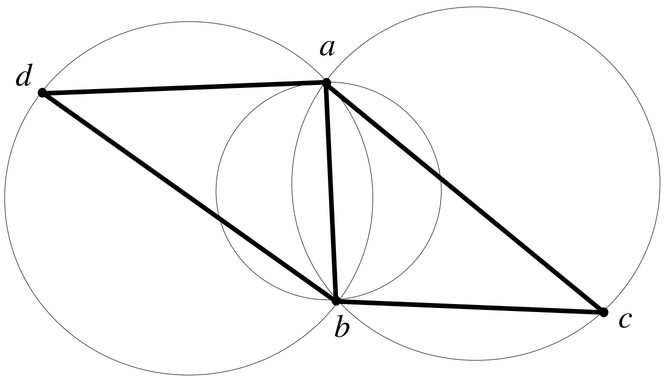


Figure 8: Figure for the proof of Lemma 5.1.

Theorem 5.1 *Let $GG(S)$ be a tree. Then, for all cycles C in $DT(S)$, there exists an edge $ab \in C$ not in $GG(S)$ such that for some $p \in S$, $\triangle(apb)$ is a face of $DT(S)$, $\triangle(apb)$ lies inside C, and $\angle apb \geq \pi/2$.*

Proof: Consider any two cycles C and C' in $DT(S)$. We will say that C is *contained* in C' if the region bounded by C' contains the region bounded by C. This gives a partial order on the cycles of $DT(S)$. Since $GG(S)$ is a tree, any cycle C must contain an edge that is not in $GG(S)$. Consider then the set of all cycles which do not satisfy the conclusion of the Theorem. If this set is empty, then we are done; otherwise this set contains a cycle C which is maximal with respect to containment, i.e. there does not exist another cycle in the set that contains C. Consider an edge ab on C which is not in $GG(S)$ (see Figure 9 (a)). Since $ab \notin GG(S)$, by Lemma 5.1 it must lie on a triangle $\triangle(apb)$ of $DT(S)$ such that $\angle apb \geq \pi/2$. Since ab does not satisfy the conclusion of the Theorem, $\triangle(apb)$ must lie outside C. Let C' be the smallest (in terms of containment) cycle of $DT(S)$ which contains both C and $\triangle(apb)$ (see Figure 9 (b)). Then the edges of C' consist of edges of C and at least one of ap and bp. Thus C' violates the conclusion of the Theorem, since the edges ap and pb do not satisfy the conclusion of the Theorem ($\triangle(apb)$ is a triangle of $DT(S)$ that lies inside C', where the angle with apex a and the one with apex b are both less than $\pi/2$). This, however, contradicts the maximality of C.

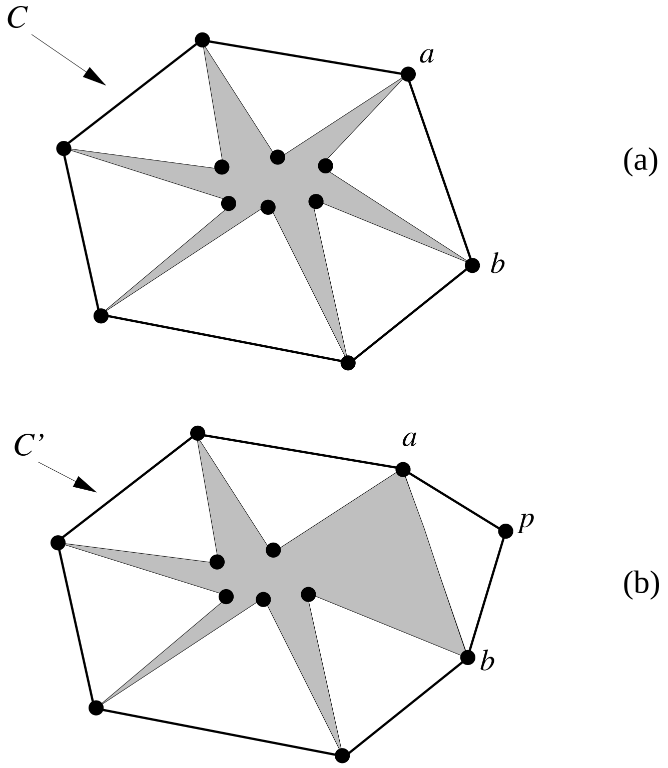


Figure 9: Figure for the proof of Theorem 5.1: (a) cycle C and its interior; (b) cycle C' and its interior.

5.2 Forbidden Trees and Wide Trees

We begin by establishing some properties of GG drawings of trees. Let S be a set of points in the plane. The first property can be verified using the same argument as in the proof of Lemma 3.1.

Lemma 5.2 *If $GG(S)$ is a tree then the angle between any two adjacent edges is greater than or equal to $\pi/2$.*

Corollary 5.1 *If $GG(S)$ is a tree it cannot have a vertex of degree greater than four.*

An immediate consequence of Theorem 5.1 is the following.

Corollary 5.2 *If $GG(S)$ is a tree and if ab and bc are two edges of $GG(S)$ such that $\angle abc = \pi/2$, then $\triangle(abc)$ is in $DT(S)$.*

Proof: By Observation 5.1, $DT(S)$ contains a path from a to c not including b such that every edge on that path is adjacent to b. That path, together with edges ab and bc form a cycle. By Theorem 5.1, some edge xy on the path must be such that $\angle xby \geq \pi/2$. But $\angle abc = \pi/2$, thus xy is the edge ac and so $\triangle(abc)$ is in $DT(S)$.

Lemma 5.3 *If $GG(S)$ is a tree then it cannot contain as a subtree a path $P = abcd$ such that $\angle abc = \pi/2$ and $\angle acd < \pi/2$.*

Proof: We proceed by contradiction. Suppose such a path P existed in $GG(S)$. See Figure 10(a).

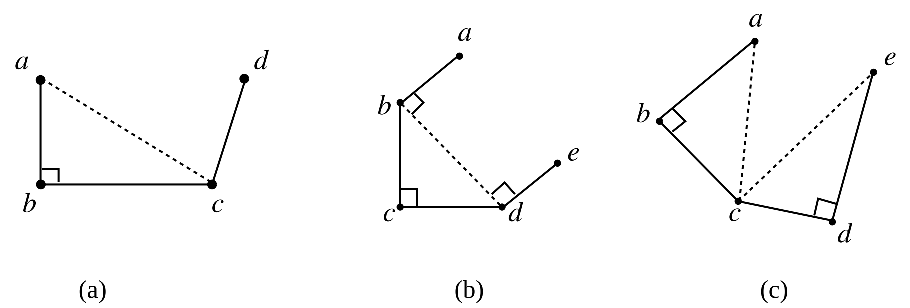


Figure 10: The forbidden paths of (a) Lemma 5.3, (b) Lemma 5.4 and (c) Lemma 5.5.

By Corollary 5.2, the edge ac is in $DT(S)$. By Observation 5.1, $DT(S)$ contains a path from a to d not containing c such that each vertex on the path is adjacent to c, call this path $P(a,d)$. Let $C = P(a,d) \cup P$. By Theorem 5.1, C contains an edge xy not in $GG(S)$ such that $\angle xcy \geq \pi/2$. This, however, is impossible since $\angle acd < \pi/2$.

Corollary 5.3 *A Gabriel tree cannot contain two adjacent vertices of degree four.*

Proof: The proof follows from Lemmas 5.2 and 5.3.

Observe that corollaries 5.1 and 5.3 together settle the conjectures of Matula and Sokal.

Lemma 5.4 *If $GG(S)$ is a tree then it cannot contain as a subtree a path $P = abcde$ such that $\angle bcd = \pi/2$, $\angle abd = \pi/2$ and $\angle bde = \pi/2$.*

Proof: We proceed by contradiction. Suppose $P \in GG(S)$. The argument is similar to the proof of Lemma 5.3. See Figure 10(b). By Corollary 5.2, $bd \in DT(S)$. Consider the subpath $P' = abcd$ of P and the path of $DT(S)$, from a to d, guaranteed by Observation 5.1. Now, since $\angle abd = \pi/2$, we can conclude that $ad \in DT(S)$ (otherwise, using the same reasoning of Lemma 5.3, P' and the path from a to d would contradict the conclusions of Theorem 5.1).

Let $P(d,e)$ be the path of $DT(S)$ from d to e, and let $C = P(a,d) \cup P$. By Theorem 5.1, C contains an edge xy not in $GG(S)$ such that $\angle xdy \geq \pi/2$. This, however, is impossible since $\angle ade < \pi/2$.

Lemma 5.5 *If $GG(S)$ is a tree then it cannot contain as a subtree a path $P = abcde$ such that $\angle abc = \pi/2$, $\angle cde = \pi/2$ and $\angle ace < \pi/2$.*

Proof: By Corollary 5.2, ac and ce are both in $DT(S)$. See Figure 10(c). Thus by the same arguments used in the proofs of Lemmas 5.3 and 5.4, $\angle ace$ cannot be less than $\pi/2$.

In the following lemmas and corollaries, all subscripts are considered modulo 4.

Lemma 5.6 *Assume that $GG(S)$ is a tree. Let v be a vertex of degree four in $GG(S)$ and let $v_i, 0 \leq i \leq 3$ be the neighbors of v in clockwise order. If some v_i has degree three then $\angle v_{i-1}v_iv_{i+1} \leq \pi/2$.*

Proof: Assume that some v_i has degree three. Let u, w and v be the neighbors of v_i in clockwise order. See Figure 11. By Lemma 5.3, $\angle v_{i-1}v_iu \geq \pi/2$ as is $\angle wv_iv_{i+1}$. By Lemma 5.2, $\angle uv_iw \geq \pi/2$ also. Thus, $\angle v_{i-1}v_iv_{i+1} \leq \pi/2$.

We are now in a position to describe the infinite class of trees that cannot be drawn as a Gabriel graph (referred to as *forbidden trees*). We first describe the smallest member of this family.

Corollary 5.4 *If $GG(S)$ is a tree it cannot contain a vertex of degree four adjacent to four vertices of degree three.*

Proof: Suppose otherwise. Let v have degree four and let v_0, v_1, v_2 and v_3 be the four neighbors of v in clockwise order. By Lemma 5.6, for each $i, 0 \leq i \leq 3$, $\angle v_{i-1}v_iv_{i+1} \leq \pi/2$. Thus each of these four angles must be equal to $\pi/2$, since v_0, v_1, v_2, v_3 form the vertices of a convex quadrilateral. But now $GG(S)$ contains a path (in fact, several) of the type forbidden by Lemma 5.4.

Corollary 5.5 *If $GG(S)$ is a tree and contains a subtree T' consisting of a vertex v of degree four adjacent to three vertices of degree three and a fourth vertex v_0 of degree one then (T', v_0) does not have an α-drawing for any $\alpha \leq \pi/2$.*

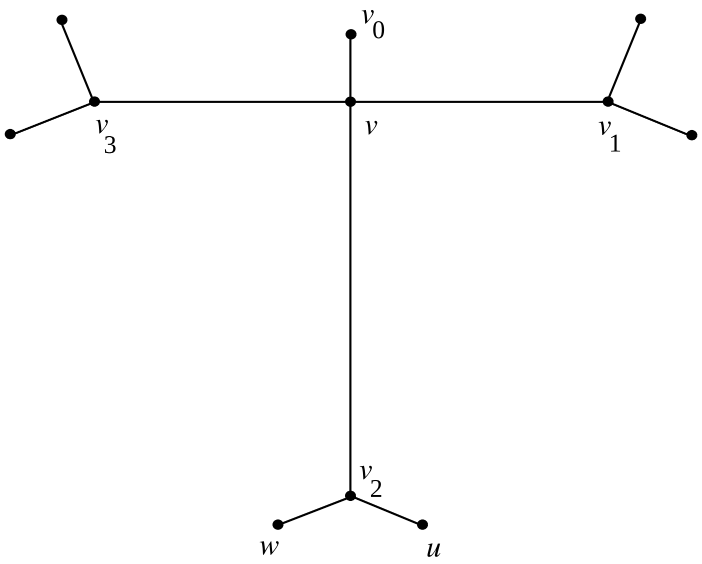


Figure 11: Illustration for Lemma 5.6 and for Corollary 5.5. In the figure, $\angle v_3 v_0 v_1 > \pi/2$.

Proof: Let v_0, v_1, v_2 and v_3 be the neighbors of v in clockwise order, where v_1, v_2 and v_3 all have degree three (see Figure 11). Then, by Lemma 5.6, for $i = 1, 2, 3$, $\angle v_{i-1} v_i v_{i+1} \leq \pi/2$. In fact, to avoid the configuration of Lemma 5.4, at least one of these three angles must be strictly less than $\pi/2$. Thus $\angle v_3 v_0 v_1 > \pi/2$, since v_0, v_1, v_2, v_3 form the vertices of a convex quadrilateral. This implies that (T', v_0) cannot be α-drawn in any α-sector having apex v_0 and $\alpha \leq \pi/2$.

It turns out that rooted trees which can only be drawn in α-sectors where $\alpha > \pi/2$ play an important role in characterizing those trees which are not Gabriel trees.

Definition 5.1 A rooted tree (T, r) is *wide* if

1. $deg(r) = 2$ or
2. $deg(r) = 1$, r has neighbor v of degree four and each rooted tree $(T_x(v), x)$, $x \neq r$, is wide.

Figures 11 and 12 show examples of wide rooted trees. By Lemma 5.2, if the root r of a wide tree has degree $deg(r) = 2$, then no α-drawing of (T, r) can have $\alpha < \pi/2$. In fact, this also holds for wide rooted trees in which the root has degree one, as the next lemma shows.

Lemma 5.7 *Let (T, v_0) be a wide rooted tree with $deg(v_0) = 1$. Then (T, v_0) has no α-drawing for any $\alpha \leq \pi/2$.*

Proof: We will actually show the following: Given a realization of (T, v_0) as a Gabriel tree, letting v denote the neighbor of v_0 and letting v_0, v_1, v_2 and v_3 denote the neighbors of v in clockwise order then $\angle v_3 v_0 v_1 > \pi/2$. From this the lemma follows.

We proceed by induction on the number of vertices in (T, v_0). The base case is the eleven vertex tree with vertex v_0 adjacent to vertex v of degree four and such that each neighbor of v other than v_0 has two neighbors of degree one (see Figure 11). The truth of the lemma for this graph follows from Corollary 5.5. Suppose now the statement is true for any tree on fewer than n vertices. We prove it for trees having n vertices. If v_1, v_2, and v_3 have degree three, then again the statement is true by Corollary 5.5. Suppose a vertex adjacent to v, say v_2 has degree two. See Figure 12. Since $(T_{v_2}(v), v_2)$ is wide, the other neighbor of v_2 must have degree 4. Call this neighbor w and let w_1 and w_2 be the neighbors of w which come before and after v_2 in clockwise order around w. From Lemma 5.5, we conclude that $\angle w_1 v_2 v_1 \geq \pi/2$ and $\angle w_2 v_2 v_3 \geq \pi/2$. By induction, $\angle w_2 v_2 w_1 > \pi/2$. Thus $\angle v_3 v_2 v_1 < \pi/2$. Also, angles $\angle v_2 v_1 v_0$ and $\angle v_0 v_3 v_2$ are each at most $\pi/2$ by Lemma 5.6. Therefore, $\angle v_1 v_0 v_3 > \pi/2$, since v_0, v_1, v_2, v_3 form the vertices of a convex quadrilateral. A similar argument holds if v_1 and/or v_3 has degree two.

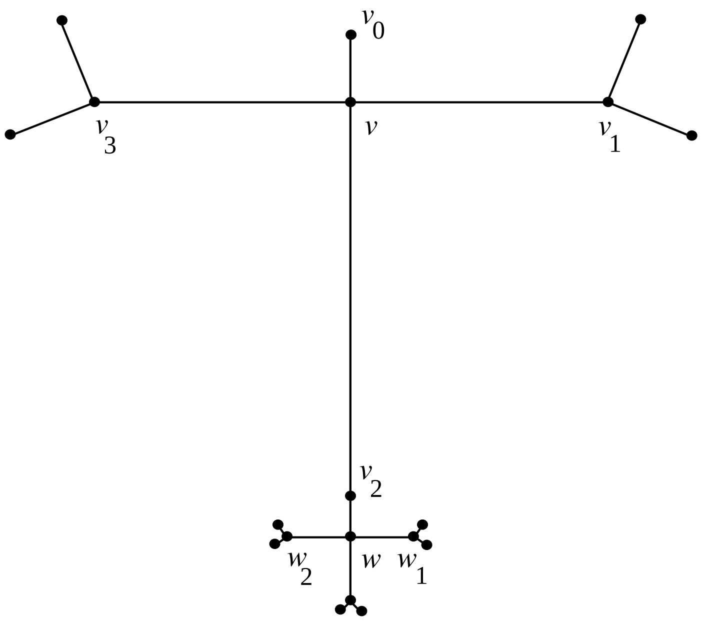


Figure 12: Illustration for Lemma 5.7. In the figure, $\angle v_3 v_0 v_1 > \pi/2$.

We are now in a position to describe the class of forbidden trees.

Definition 5.2 A tree T is *forbidden* if it contains any of the following

1. A vertex of degree at least five
2. Two adjacent vertices of degree four
3. A vertex v of degree four such that for each neighbor x of v, the subtree $(T_x(v), x)$ is wide.

See Figure 13 for an example of a forbidden tree.

Lemma 5.8 *No forbidden tree T is a Gabriel graph.*

Proof: If T has a vertex of degree greater than four, then by Corollary 5.1 it cannot be realized as a Gabriel tree. Similarly, by Corollary 5.3, T cannot be realized as a Gabriel tree if it has two adjacent vertices of degree four. Suppose now that T contains a vertex v of degree four with neighbors v_0, v_1, v_2, and v_3, such that the subtrees $(T_{v_i}(v), v_i), (i = 0 \ldots 3)$ rooted at v_i are wide. Suppose T has been realized as a Gabriel tree. As was shown in the proof of Lemma 5.7, for each $i = 0 \ldots 3$, $\angle v_{i-1} v_i v_{i+1} > \pi/2$. But this is impossible since $v_0, \ldots, v_3$ form the vertices of a convex quadrilateral.

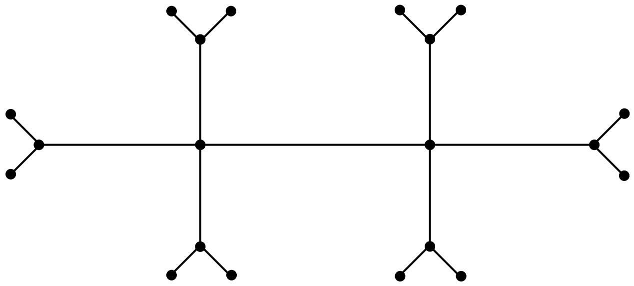


Figure 13: A forbidden tree.

5.3 How to Draw a Gabriel Tree

To complete our characterization of Gabriel trees, we need to establish the converse of Lemma 5.8. To this end, we describe a drawing algorithm which, given a non-forbidden tree T, produces a set of points S such that $GG(S)$ is isomorphic to T. We begin with some geometric preliminaries that are fundamental to the drawing algorithm.

Lemma 5.9 *Let $P = \{p, p_1, p_2\}$ be a set of points of $\mathbb{R}^2$ be such that $\angle p_1 p p_2 > \pi/2$, let $S = S[p, \beta, d, \rho]$ $(\pi/2 < \beta < \pi)$ be a sector such that p_1 and p_2 are in the interior of S, and let d_1 and d_2 be the vectors in directions $p_1 - p$ and $p_2 - p$ respectively. There exist angles $\alpha_1, \alpha_2 > 0$ such that*

1. *$S_i = S[p, \alpha_i, d_i, \rho]$ is in the interior of S, for $i = 1, 2$.*
2. *P separates S_1 and S_2.*

Proof: See Figure 14. Since both p_1 and p_2 are points in the interior of S, there always exists an angle α_1 and an angle α_2 such that both $S[p, \alpha_1, d_1, \rho]$ denoted by S_1 and $S[p, \alpha_2, d_2, \rho]$ denote by S_2 are in the interior of S. Let α_1 and α_2 be chosen such that $\angle p_1 p p_2 - (\frac{\alpha_1 + \alpha_2}{2}) > \pi/2$, then for any point $x \in S_1$ and any point $y \in S_2$, $\angle xpy > \pi/2$.

Lemma 5.10 *Let $P = \{p, p_0, p_1, p_2, p_3\}$ be a set of points of $\mathbb{R}^2$ be such that*

1. *$\angle p_i p p_{i+1} = \pi/2$, for $i = 0, \ldots, 3$*

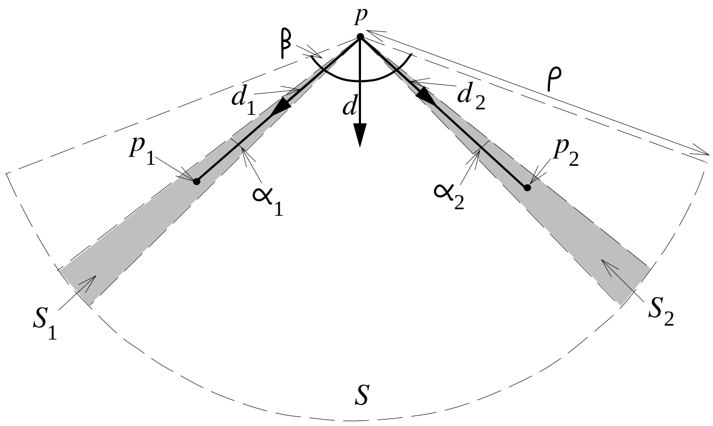


Figure 14: Figure for proof of Lemma 5.9.

2. $\angle p_3 p_0 p_1 > \pi/2$ *and*

3. $\angle p_{i-1} p_i p_{i+1} < \pi/2$, *for* $i = 1, \ldots, 3$

There exist angles β_i, *vectors* d_i *and radii* $\rho_i > 0$, *for* $i = 0, \ldots, 3$, *such that*

1. $\beta_0 > 0$
2. $\beta_i > \pi/2$, *for* $i = 1, \ldots 3$
3. P *separates each pair* $S_i = S[p_i, \beta_i, d_i, \rho_i]$ *and* $S_j = S[p_j, \beta_j, d_j, \rho_j]$, $i \neq j$.

Proof: In what follows, all subscripts are taken modulo 4. See Figure 15.

Let Q denote the convex quadrilateral having vertices $p_0, \ldots, p_3$. Let l_i and r_i be two vectors emanating from p_i external to Q such that

1. the clockwise angle β_i between l_i and r_i is less than π
2. the clockwise angle between vector l_i and $\overline{p_i p_{i-1}}$ is greater than $\pi/2$
3. the clockwise angle between vector r_i and $\overline{p_i p_{i+1}}$ is greater than $\pi/2$.

Let d_i be the bisector of angle β_i and refer to $W(p_i, \beta_i, d_i)$ as W_i. Given the bounds on the angles $\angle p_{i-1} p_i p_{i+1}$ ($i = 0, \ldots, 3$), we can guarantee that $\beta_1, \beta_2, \beta_3$ are greater than $\pi/2$ and that β_0 is greater than 0. Now, let F_i be the region $W_i \cap HP[r_{i-i}, p_{i-1}] \cap HP[l_{i+1}, p_{i+1}]$.

Given a point $x \in F_i$ and a point $y \in F_{i+1}$, we see that $\angle x p_i y > \pi/2$ since $y \in HP[r_i, p_i]$. Similarly, $\angle y p_{i+1} x > \pi/2$ since $x \in HP[l_{i+1}, p_{i+1}]$. Similar arguments hold for any non-consecutive pairs of regions F_j and F_k ($j \neq k$). Thus any sector with apex p_i and angle β_i contained in F_i can serve as a sector S_i.

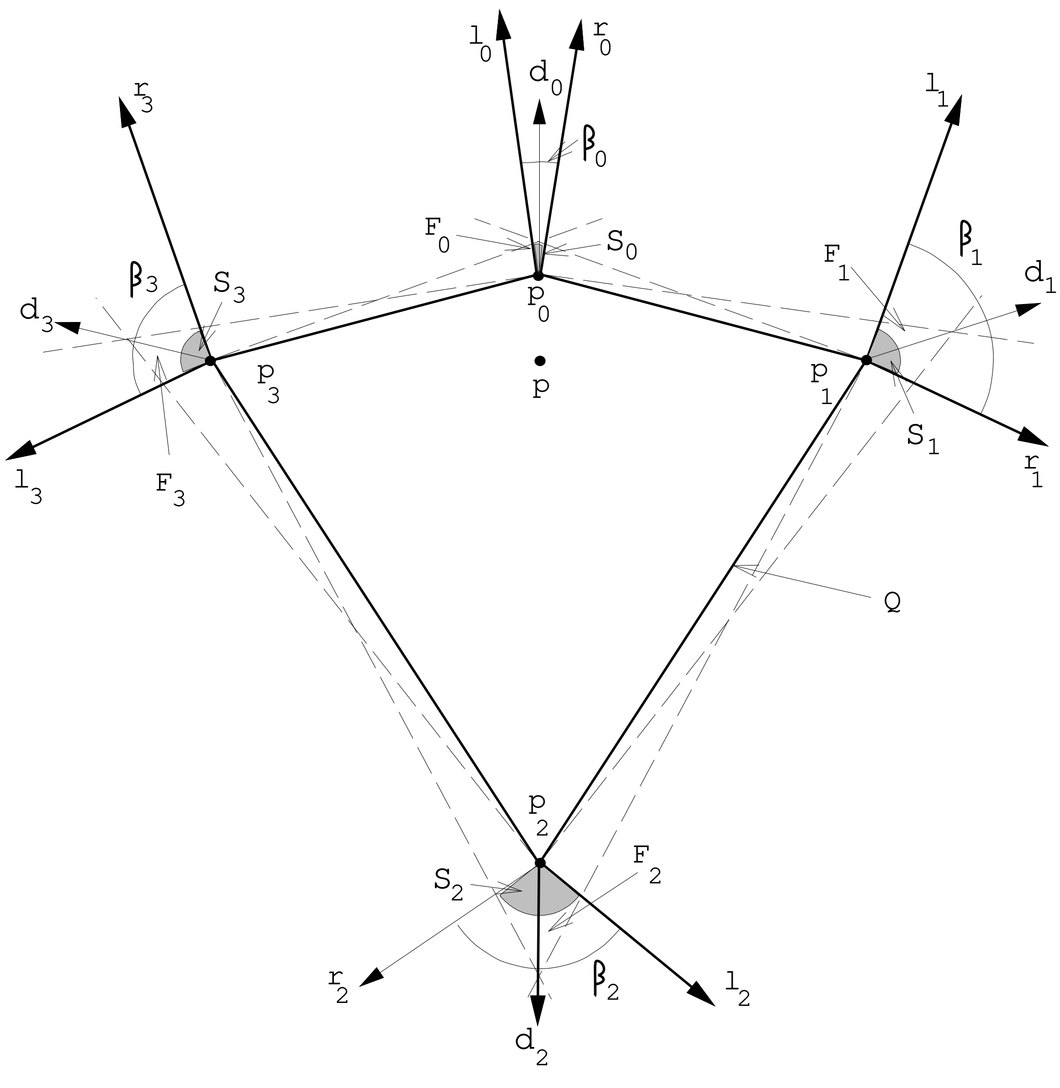


Figure 15: Illustration for Lemma 5.10.

To complete the characterization of Gabriel trees, we describe a drawing algorithm which, given a non-forbidden tree T and a vertex u of T of degree at most two, produces a set of points S such that $GG(S)$ is an α-drawing of (T, u). The algorithm works essentially by drawing u and each neighbor v of u, then recursively drawing the subtrees $(T_v(u), v)$ rooted at each neighbor of u. In order to be able to correctly place the neighbors v of u, the algorithm needs to know whether $(T_v(u), v)$ is wide before attempting to draw it. Thus, it will be necessary to provide the algorithm with information about the wideness of the rooted subtrees of T. We therefore first present a procedure *LabelGG* (see Figure 16), which will determine whether T is forbidden, and if T is not forbidden, will compute the information necessary for the drawing algorithm.

Lemma 5.11 *Given a tree T_0 and a leaf u_0 of T_0, the procedure LabelGG will determine whether T_0 is forbidden. Moreover, if T_0 is not forbidden, the procedure will give each vertex u of degree less than four a label $l(u)$ such that*

$$l(u) = \begin{cases} \text{“wide”} & \text{if the subtree of } (T_0, u_0) \text{ with root } u \text{ is wide} \\ \text{“not wide”} & \text{otherwise.} \end{cases}$$

Proof: We prove that *LabelGG* has the desired performance by verifying that the following invariant holds for the procedure:

> If $LabelGG(T, u)$ executes a Stop instruction then T_0 is a forbidden tree otherwise each vertex of degree less than four in the subtree T of (T_0, u_0) having root u has a label satisfying the conclusions of the Lemma.

A Stop instruction has the effect of ending all computation.

We begin by observing that all calls $LabelGG(T, u)$ have the property that $deg_T(u) \leq 2$: The first call, $LabelGG(T_0, u_0)$ has this property since u_0 is a leaf and no subsequent call is made without ensuring that this inequality holds. Note also that when a vertex of degree greater than four is found, the procedure executes a Stop instruction; the same occurs when a vertex of degree four having a neighbor of degree four is found. Thus we may restrict our attention during the rest of the proof to those trees of maximum degree at most four having no pair of adjacent degree four vertices.

We now establish by induction the truth of the invariant. By the preceding remarks, we may assume that $deg_T(u) \leq 2$. If u is a leaf of T_0, then $LabelGG(T, u)$ labels u with “not wide”. Suppose that the invariant holds on all calls $LabelGG(T, u)$ where T has fewer than $n > 1$ vertices. Assume now that *LabelGG* is invoked with a subtree (T, u) of (T_0, u_0) having n vertices. There are two cases to consider.

Case 1: $deg_T(u) = 1$.

Let v be the neighbor of u. If $deg_T(v) \leq 3$, then *LabelGG* is invoked on $(T_v(u), v)$ (i.e. the subtree of (T, u) having root v). Since v has degree at most two in this subtree, by induction either a Stop instruction is executed, in which case T_0 is forbidden or each vertex of the subtree receives a label satisfying the conclusions of the Lemma. In the latter case, u correctly receives the label “not wide”.

If $deg_T(v) = 4$, then for each neighbor v_1, v_2, v_3 of v, other than u, *LabelGG* is invoked on $(T_{v_i}(v), v_i)$, as long as the degree of v_i in T is less than four. Again by induction, either a Stop instruction is executed and T_0 is forbidden or each v_i is labelled appropriately. Now the labels of the v_i are used to assign the correct label to u, since (T, u) is wide if and only if each of these subtrees has label “wide”.

procedure $LabelGG(T, u)$
Case: $deg_T(u) = 0$.
$l(u) \leftarrow$ "not wide".
Case: $deg(u) = 1$.
Let v denote the neighbor of u.
Case: $deg(v) \leq 3$
$LabelGG(T_v(u), v)$.
$l(u) \leftarrow$ "not wide".
Case: $deg_T(v) = 4$.
Let $v_1, \ldots, v_3$ denote the neighbors of v other than u.
For $i = 1, \ldots, 3$:
If $deg(v_i) \geq 4$ then Stop
$Label(T_{v_i}(v), v_i)$
If any $l(v_i) =$ "not wide" then $l(u) \leftarrow$ "not wide"
Else $l(u) \leftarrow$ "wide"
Case: $deg_T(v) > 4$.
Stop.
Case: $deg_T(u) = 2$
Let v_0 and v_1 denote the neighbors of u.
$LabelGG(T_{v_0}[u], u)$.
If $l(u) =$ "wide" then Stop
$LabelGG(T_{v_1}[u], u)$.
If $l(u) =$ "wide" then Stop
$l(u) \leftarrow$ "wide".
end procedure.

Figure 16: Procedure *LabelGG*.

Finally, if $deg_T(v) > 4$, a Stop instruction is executed, since T_0 is forbidden.

Case 2: $deg_T(u) = 2$

Let v_0 and v_1 denote the neighbors of u. If $(T_{v_0}[u], u)$ is wide, then v_0 has degree four and so T_0 is forbidden, since u has degree three in T_0. The same holds for $(T_{v_1}[u], u)$. Therefore, if u is labelled "wide" in the invocation of *LabelGG* on either of these two trees, a Stop instruction is executed. Otherwise u is labelled "wide" since it has degree two.

We are now in a position to present a procedure which produces a *GG* drawing of any non-forbidden tree T.

Lemma 5.12 *Let T_0 be a non-forbidden tree and let $u_0 \in V(T_0)$ be a leaf of T_0. Then*

1. *If (T_0, u_0) is wide, then it can be β-drawn for any $\beta > \pi/2$.*
2. *If (T_0, u_0) is not wide then it can be α-drawn for any $\alpha > 0$.*

Proof: Let T_0 be a tree and let u_0 be a leaf of T_0. Since T_0 is not forbidden, it has maximum vertex degree at most four. We also assume that the vertices of T_0 of degree less than four have been labelled by procedure *LabelGG*. Given a point $q \in \mathbb{R}^2$, angles α and β, $\pi > \beta > \pi/2 > \alpha > 0$, a vector d and a radius $\rho > 0$, procedure *DrawGG* (see Figure 17), will either

- produce an α-drawing of (T, u) in $S[q, \alpha, d, \rho]$ if $l(u) =$ "not wide" or
- produce a β-drawing of (T, u) in $S[q, \beta, d, \rho]$ if $l(u) =$ "wide".

The proof of correctness is by induction on the size of T. In what follows we adopt the notation of Figure 17. Observe first that *DrawGG*, like *LabelGG*, is only invoked on subtrees (T, u) of (T_0, u_0) where $deg_T(u)$ is at most two and that *DrawGG* is invoked on exactly the same subtrees (T, u) on which *LabelGG* was invoked. If T has a single vertex, *DrawGG* produces the correct type of drawing. Suppose that the procedure works correctly on all trees (of maximum degree at most four and not forbidden) having fewer than n vertices. Assume now that *DrawGG* is invoked on a subtree (T, u) of (T_0, u_0) having n vertices. There are two cases to consider.

Case 1: $deg_T(u) = 1$.

Let v be the neighbor of u. This case is split into two sub-cases:

Case 1a: $l(u) =$ "not wide".

If $deg_T(v) < 4$, then a sector $S[q', \beta, d, \rho']$ is found which lies in the interior of $S[q, \alpha, d, \rho]$. By induction, $(T_v(u), v)$ will be drawn inside this smaller sector, guaranteeing an α-drawing of (T, u) in $S[q, \alpha, d, \rho]$. See Figure 18 (a).

If $deg_T(v) = 4$, then v has neighbors $v_0, \ldots, v_3$, where $v_2 = u$ and $l(v_0) =$ "not wide". Now points $P = \{p, p_0, \ldots, p_3\}$ disposed as in Lemma 5.10, where $p_2 = q$, are selected in the interior of $S[q, \beta, d, \rho]$. This allows sectors $S_i = S[p_i, \beta_i, d_i, \rho_i']$ to be found such that each S_i is in the interior of $S[q, \beta, d, \rho]$ and such that P separates each pair of sectors S_i and S_j. See Figure 18 (b). Each subtree $(T_{v_i}(v), v_i), i = 0, 1, 3$ is then, recursively, drawn correctly in the sector S_i.

Case 1b: $l(u) =$ "wide".

Again, $deg_T(v) = 4$, then v has neighbors $v_0, \ldots, v_3$, where now $v_0 = u$. Points $P = \{p, p_0, \ldots, p_3\}$, where $p_0 = q$, are selected in the interior of $S[q, \beta, d, \rho]$ so as to satisfy the hypotheses of Lemma 5.10. This allows sectors $S_i = S[p_i, \beta_i, d_i, \rho_i']$ to be found such that each S_i is in the interior of $S[q, \beta, d, \rho]$ and such that P separates each pair of sectors S_i and S_j. See Figure 19 (a). Each subtree $(T_{v_i}(v), v_i), i = 1, 2, 3$ is then, recursively, drawn correctly in the sector S_i.

Case 2: $deg_T(u) = 2$

Let v_0 and v_1 denote the neighbors of u. Since T_0 is not forbidden, both v_0 and v_1 are labelled "not wide". Now, points $P = \{p, p_1, p_2\}$, where $p = q$ are selected from the interior of $S[q, \beta, d, \rho]$ so as to satisfy the hypotheses of Lemma 5.9. This allows sectors $S_i = S[p, \alpha_i, d_i, \rho]$ to be found such that both S_1 and S_2 are in the interior of $S[q, \beta, d, \rho]$ and such that $\{p\}$ separates S_1 and S_2. See Figure 19 (b). Each subtree $(T_{v_i}[u], u), i = 1, 2$ is then, recursively, drawn correctly in the sector S_i.

For an example of a tree produced by procedure *DrawGG* see Figure 12.

It is easy to see that the drawing procedure of the preceding lemma takes linear time using the real RAM model of computation and that the labelling procedure takes linear time in the comparison-based model. Thus Lemmas 5.8 and 5.12 together form the proof of the following theorem.

procedure $DrawGG(T, u, \alpha, \beta, q, d, \rho)$

Draw u at q.

Case: $deg_T(u) = 1$.

Let v denote the neighbor of u.

Case: $l(u) =$ "not wide".

If $degT(v) < 4$ then

Choose $\rho' < \rho$ so that for $q' = q + (\rho - \rho')d$, $S[q', \beta, d, \rho']$ lies in the interior of $S[q, \alpha, d, \rho]$.

Draw a segment from q to q'.

$DrawGG(T_v(u), v, \alpha, \beta, q', d, \rho')$

Else

Let $v_0, \ldots, v_3$ denote the neighbors of v, where $v_2 = u$ and $l(v_0) =$ "not wide".

Choose $p, p_0, \ldots, p_3$ in the interior $S = S[p, \beta, d, \rho]$, so that $p_2 = q$ and satisfying the hypotheses of Lemma 5.10.

Let β_i, d_i, ρ_i be as described in Lemma 5.10.

Choose $\rho_i' < \rho_i$ so that sector $S_i = S[p_i, \beta_i, d_i, \rho_i']$ is contained in the interior of sector S.

Draw v at p and draw segments from p to each p_i.

For $i = 0, 1, 3$: $DrawGG(T_{v_i}(v), v_i, \alpha, \beta_i, p_i, d_i, \rho_i')$.

Case: $l(u) =$ "wide".

Let $v_0, \ldots, v_3$ denote the neighbors of v, where $v_0 = u$.

Choose $p, p_0, \ldots, p_3$ in the interior $S = S[p, \beta, d, \rho]$, so that $p_0 = q$ and satisfying the hypotheses of Lemma 5.10.

Let β_i, d_i, ρ_i be as described in Lemma 5.10.

Choose $\rho_i' < \rho_i$ so that sector $S_i = S[p_i, \beta_i, d_i, \rho_i']$ is contained in the interior of sector S.

Draw v at p and draw segments from p to each p_i.

For $i = 1, 2, 3$: $DrawGG(T_{v_i}(v), v_i, \alpha, \beta_i, p_i, d_i, \rho_i')$.

Case: $deg_T(u) = 2$.

Let v_1 and v_2 be the neighbors of u.

Let $p = q$ and choose p_1, p_2 in the interior of $S[p, \beta, d, \rho]$ to satisfy the hypotheses of Lemma 5.9.

Let α_i, d_i be as described in Lemma 5.9.

Draw segments from p to p_1 and from p to p_2.

For $i = 1, 2$,

$DrawGG(T_{v_i}[u], u, \alpha_i, \beta, p_i, d_i, \rho)$.

end procedure.

Figure 17: Procedure $DrawGG$.

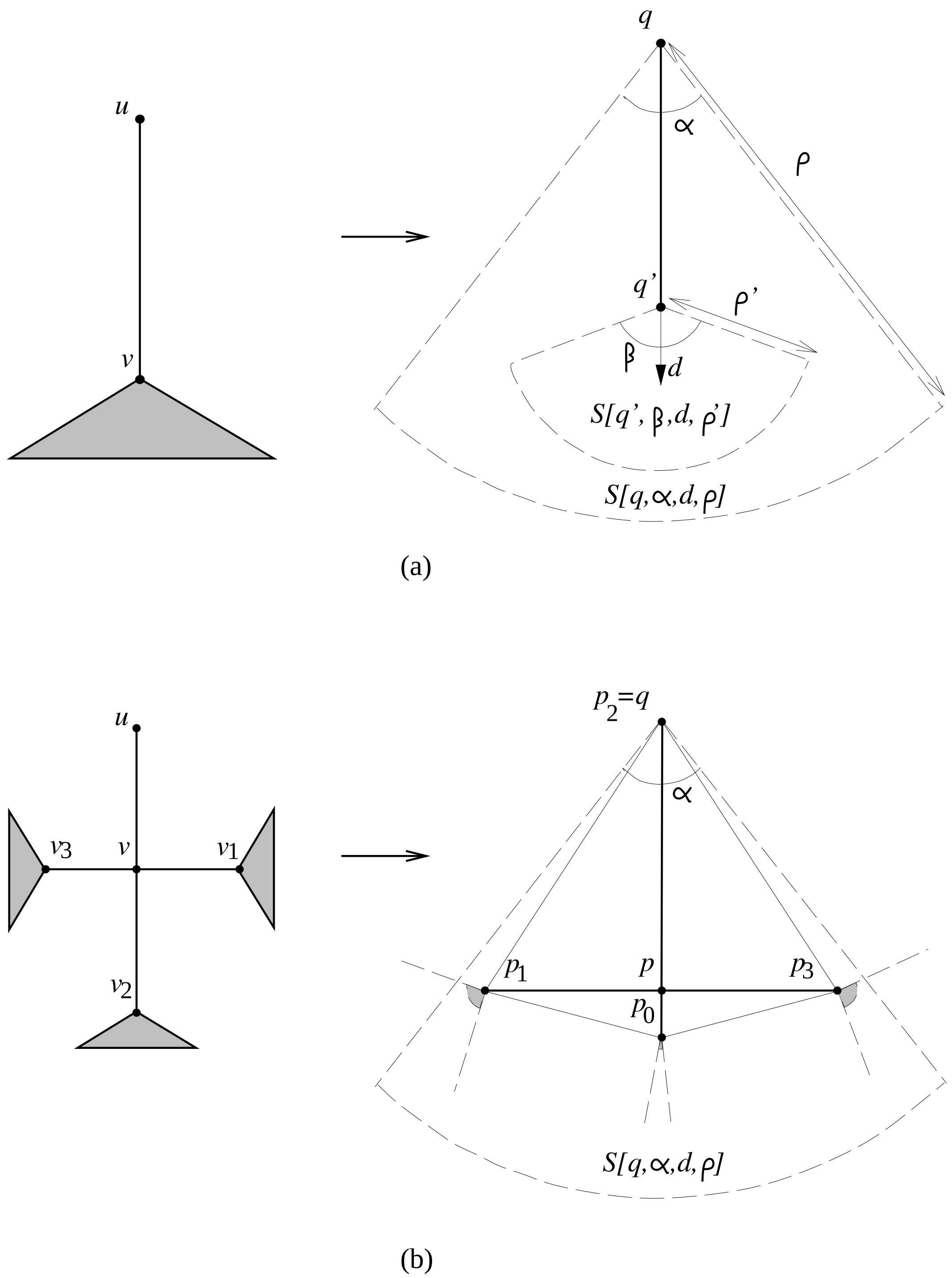


Figure 18: Illustration for Lemma 5.12; (a) $l(u)$ = "not wide" and $deg_T(v) < 4$; (b) $l(u)$ = "not wide" and $deg_T(v) = 4$.

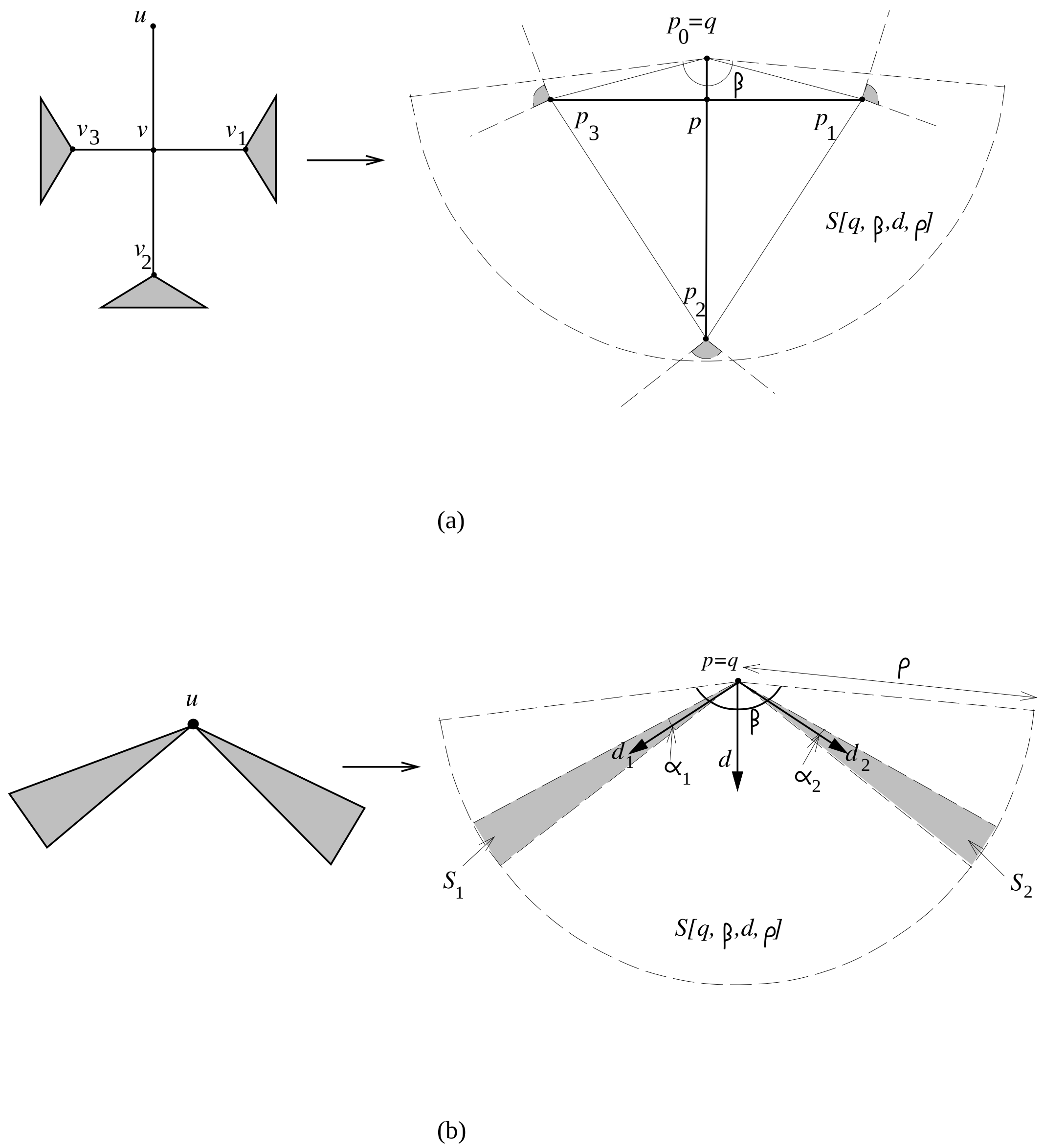


Figure 19: Illustration for Lemma 5.12; (a) $l(u) =$ " wide" and $deg_T(v) = 4$; (b) $deg_T(u) = 4$.

Theorem 5.2 *A tree is a Gabriel tree if and only if it is not forbidden, and the recognition problem for Gabriel trees can be solved in time proportional to the size of the tree. Furthermore, in the real RAM model, a GG drawing of a non-forbidden tree can be computed in time proportional to the size of the tree.*

6 Conclusion

In this paper we have provided complete characterizations of trees that can be drawn as relatively closest graphs, relative neighborhood graphs, Gabriel graphs or modified Gabriel graphs. The characterizations give rise to linear time recognition and drawing algorithms for each type of proximity graph. The problems of completely characterizing which graphs admit RNG, RCG, MGG or GG drawings remain open. Even characterizing those planar graphs which admit MGG drawings appears to be difficult.

The model of computation used is the real RAM. Although the decision algorithms do not need such a strong model of computation, the drawing algorithms do, since the coordinates for the vertices are real numbers. Suppose, for example, we compute an RNG drawing of a path P of n vertices by means of the algorithm presented in Section 4. Depending on the sizes of the radii of the sectors in which the subtrees are drawn, the ratio between the shortest edge and the longest edge in the drawing may be $O(k^n)$ for some $k > 1$. This leads to a drawing of exponential area. It is easy, however, to produce an RNG drawing of P where the edge lengths are all equal. This leads to the following question: Given an RNG tree, is it possible to produce an RNG drawing of the tree which requires area at most polynomial in the size of the tree? The same question is also of interest for RCG, GG and MGG trees.

Acknowledgments We thank G.Di Battista, G.Toussaint, and S.Whitesides for their comments and suggestions. We are particularly grateful to G.Blanco and J.Garcia for the interesting discussions that produced the ideas underlying Theorem 5.1; also, they showed us a counter example to Sendov's theorem. Finally, we thank David Thompson and Al Van Houtte for providing a stimulating research environment.

References

[1] P. K. Agarwal and J. Matoušek. Relative Neighbourhood Graphs in Three Dimensions. *Computational Geometry: Theory and application,* **2**, 1992, pp. 1-14.

[2] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications.* Elsevier Science, New York, New York, 1976.

[3] R. J. Cimikowski. Properties of Some Euclidean Proximity Graphs. *Pattern Recognition Letters,* **13**, 1992, pp. 417-423.

[4] G. Di Battista, P. Eades, R. Tamassia and I.G. Tollis. Algorithms for Automatic Graph Drawing: An Annotated Bibliography. To appear in *Computational Geometry: Theory and Applications.*

[5] G. Di Battista and L. Vismara. Angles of Planar Triangular Graphs. *Proc. ACM Symposium on Theory of Computing,* 1993, pp. 431-437.

[6] M. B. Dillencourt. Graph-Theoretical Conditions for Inscribability and Delaunay Realizability. University of Califoria Irvine, Tech. Rep. 92-90.

[7] P. Eades and S. Whitesides. Personal communication. 1993.

[8] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the Shape of a Set of Points in the Plane. *IEEE Transaction on information Theory,* **29**, 1983, pp. 551-559.

[9] I. Fary. On Straight Lines Representation of Planar Graphs. *Acta Sci. Math. Szeged,* **11**, 1948, ppl 229-233.

[10] K. R. Gabriel and R. R. Sokal. A New Statistical Approach to Geographical Analysis. *Systematic Zoology,* **18**, 1969, pp. 54-64.

[11] J. Garcia. A Counter-Example to a Theorem of Sendov. Manuscript in preparation.

[12] D. G. Kirkpatrick and J. D. Radke. A Framework for Computational Morphology. *Computational Geometry, G. T. Toussaint,* Elsevier, Amsterdam, 1985, pp. 217-248.

[13] M. S. Jacobson, M. J. Lipman, and F. R. McMorris. Trees that are Sphere-of-Influence Graphs. University of Louisville, Tech. Rep. 0191/2.

[14] J. W. Jaromczyk and G. T. Toussaint. Relative Neighborhood Graphs and Their Relatives. *Proceedings of the IEEE,* **80, 9**, 1992, pp. 1502-1517.

[15] P. M. Lankford. Regionalization: Theory and Alternative Algorithms. *Geographical Analysis,* **1**, 1969, pp. 196-212.

[16] A. Lubiw, and N. Sleumer, All Maximal Outerplanar Graphs are Relative Neighborhood Graphs. *Proceedings of the Fifth Canadian Conference on Computational Geometry, Waterloo,* 1993, pp. 198-203.

[17] D. W. Matula and R. R. Sokal. Properties of Gabriel Graphs Relevant to Geographic Variation Research and the Clustering of Points in the Plane. *Geographical Analysis,* **12, 3**, 1980, pp. 205-222.

[18] C. Monma and S. Suri. Transitions in Geometric Minimum Spanning Trees. *Proc. ACM Symposium on Computational Geometry,*, 1991, pp. 239-249.

[19] F. P. Preparata and M. I. Shamos, *Computational Geometry – an Introduction.* Springer-Verlag, New York, 1985.

[20] J. D. Radke. On the shape of a set of points. *Computational Morphology,* ed. G. T. Toussaint, Elsevier, Amsterdam, 1988, pp. 105-136.

[21] M. I. Shamos and D. Hoey. Closest Point Problem. *Proc. of the 8th Annual IEEE Symposium on the Foundations of Computer Science,* 1975, pp. 151-162.

[22] B. Sendov. Planar Neighbourhood Graphs without Cycles. *Comptes Rendus de l'Academie Bulgare des Sciences,* **44, 4**, 1991, pp. 23-25.

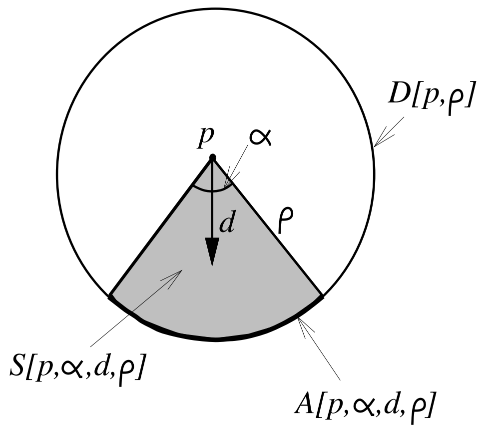
[23] G. T. Toussaint. The Relative Neighborhood Graph of a Finite Planar Set. *Pattern Recognition,* **12**, 1980, pp. 261-268.

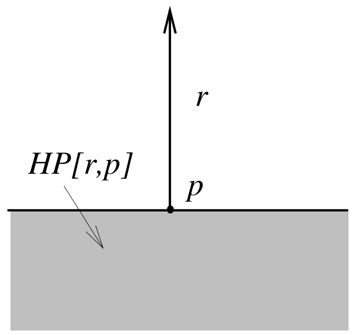
[24] G. T. Toussaint. Pattern Recognition and Geometrical Complexity *Proc. Fifth Int. Conf. on Pattern Recognition (Miami Beach),* 1980, pp. 1340-1347, .

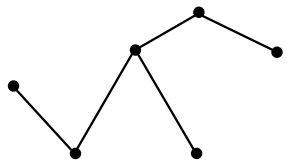
[25] G. T. Toussaint. A Graph-Theoretical Primal Sketch. *Computational Morphology,* ed. G. T. Toussaint, Elsevier, Amsterdam, 1988, pp. 229-260.

[26] R. B. Urquhart. Some Properties of the Planar Euclidean Relative Neighbourhood Graph. *Pattern Recognition Letters,* **1, 5**, 1983, pp. 317-322.

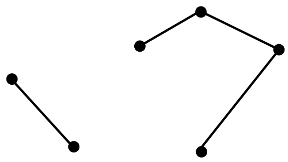
[27] R. C. Vetkamp. The γ-Neighbourhood Graph. *Computational Geometry: Theory and Applications,* **1**, 1992, pp. 227-246.


D[p,ρ]
p
∝
d
ρ
S[p,∝,d,ρ]
A[p,∝,d,ρ]

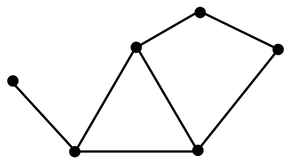

r
HP[r,p]
p



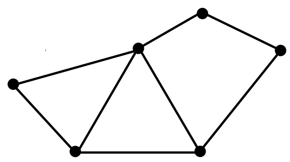
(a)



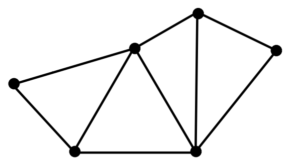
(b)



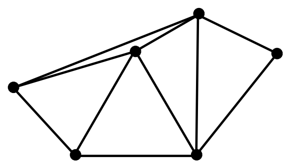
(c)



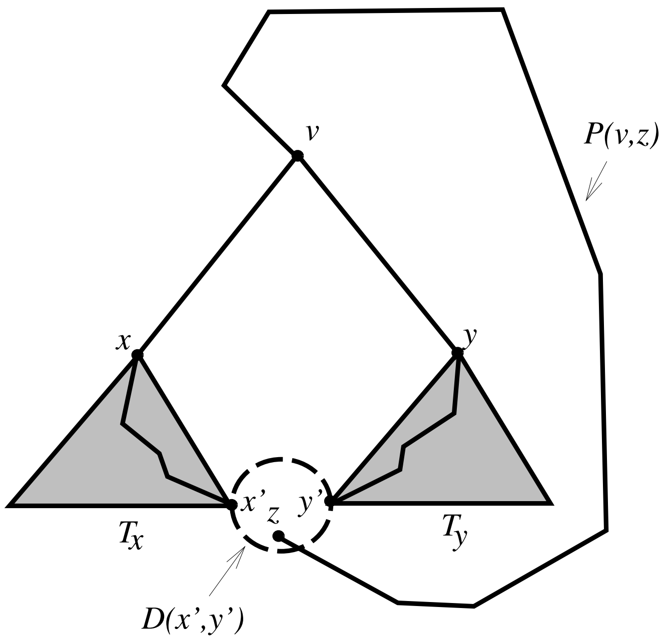
(d)

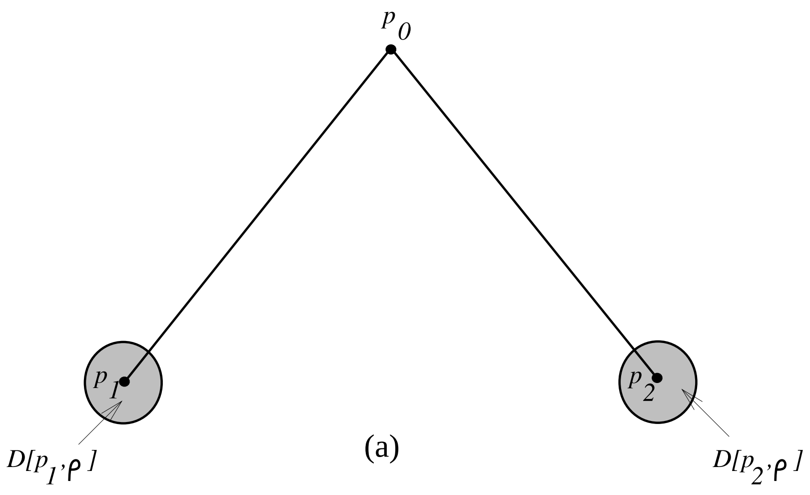


(e)



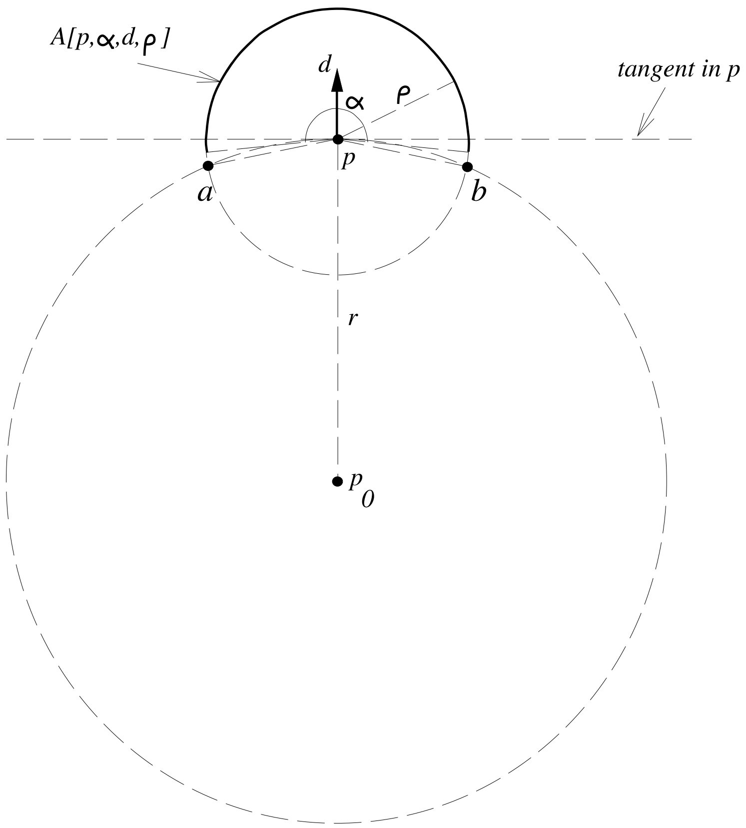
(f)


v
P(v,z)
x
y
x'
y'
z
T_x
T_y
D(x',y')



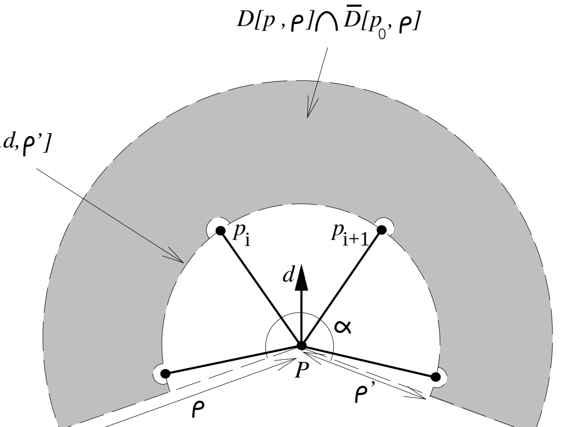
p_0
p_1
p_2
$D[p_1, \rho]$
$D[p_2, \rho]$

(a)

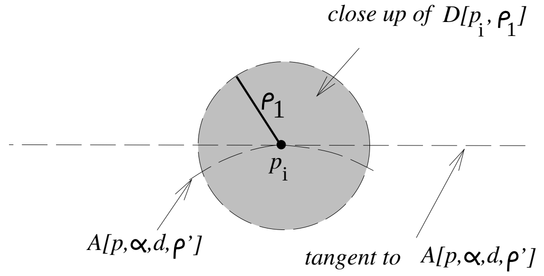


$A[p, \alpha, d, \rho]$
d
α
ρ
tangent in p
p
a
b
r
p_0

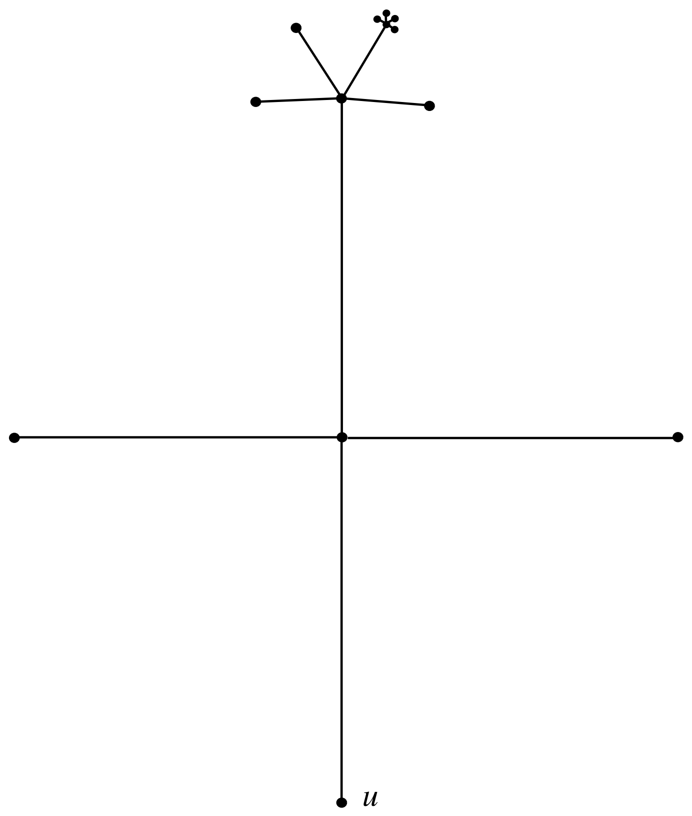
(b)

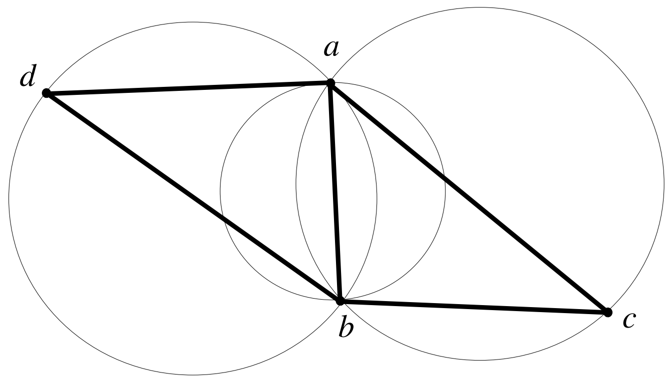

D[p , ρ] ∩ D̄[p0, ρ]
A[p,∝,d,ρ']
pi
pi+1
d
∝
P
ρ
ρ'

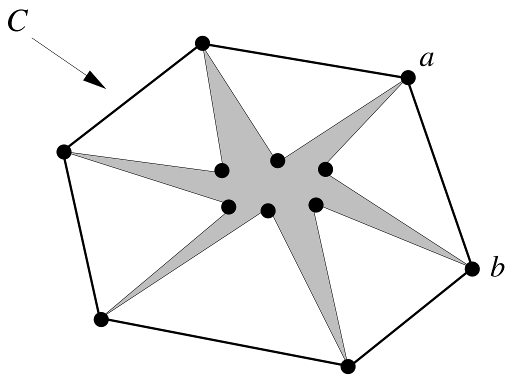
(a)


close up of D[pi, ρ1]
ρ1
pi
A[p,∝,d,ρ']
tangent to A[p,∝,d,ρ']

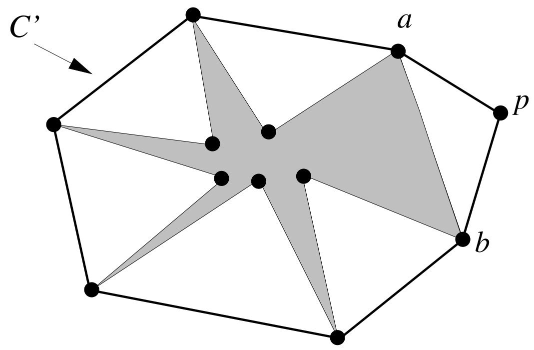
(b)


u

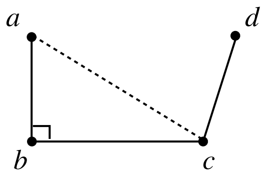

a
d
b
c



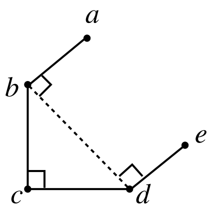
(a)



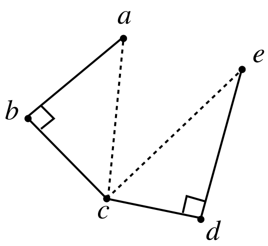
(b)



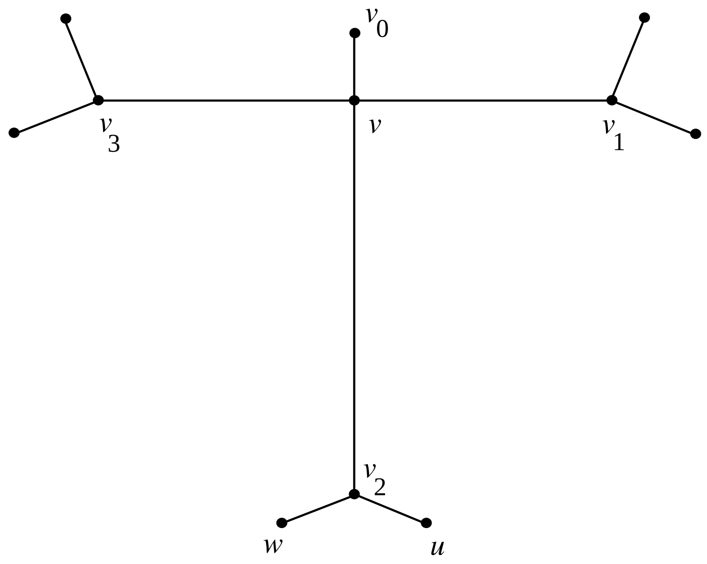
(a)



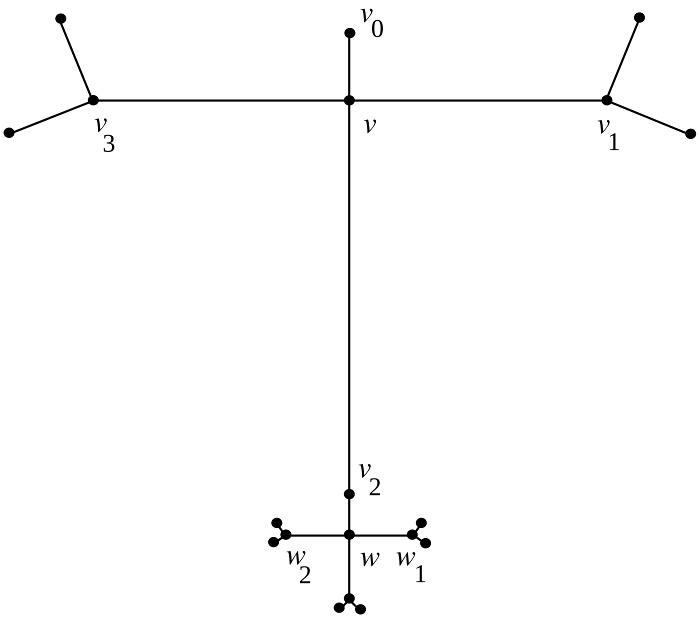
(b)

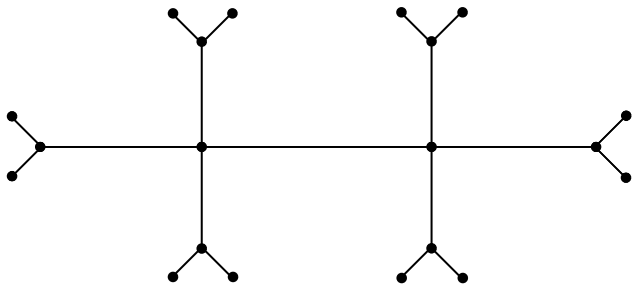


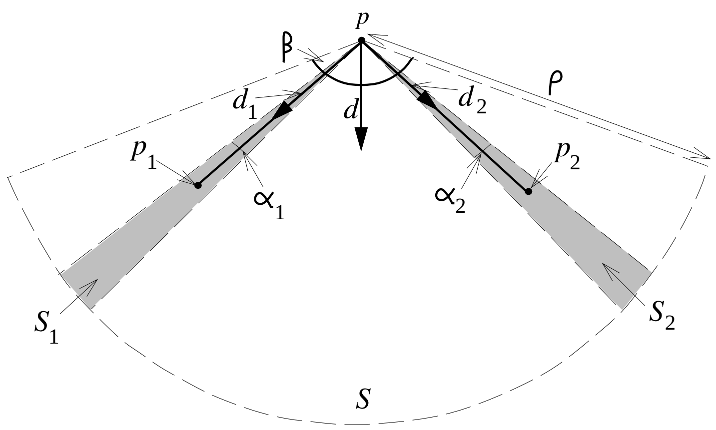
(c)



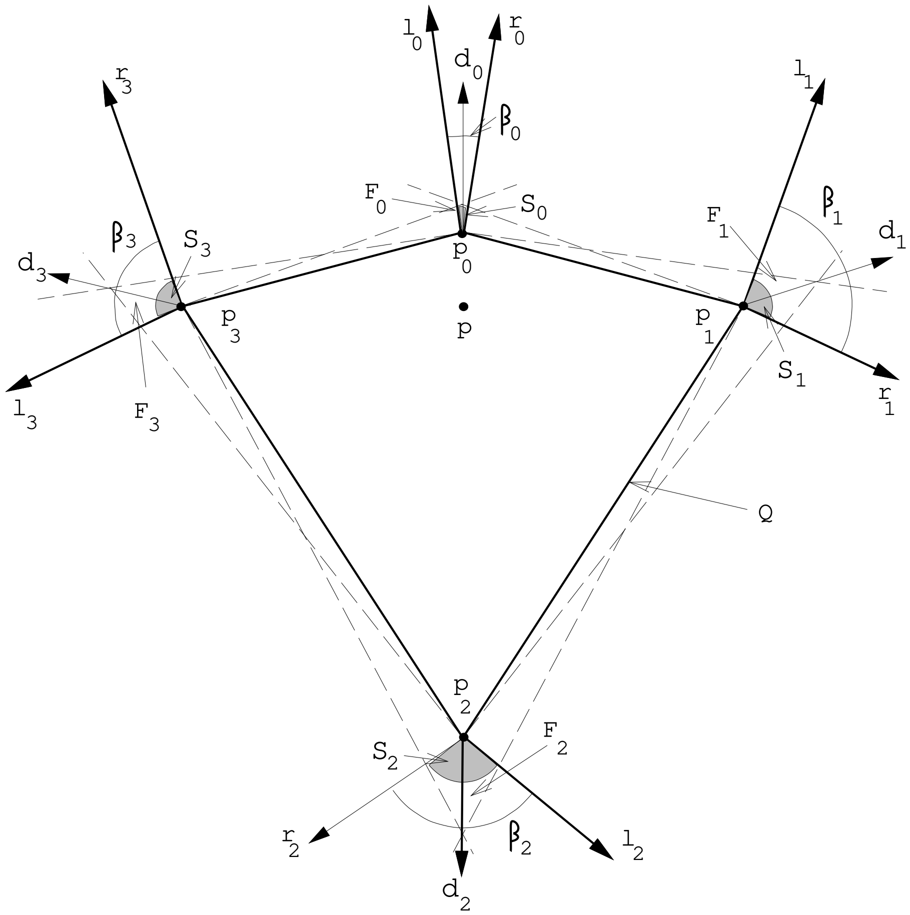
v_0
v_3
v
v_1
v_2
w
u


v_0
v_3
v
v_1
v_2
w_2
w
w_1

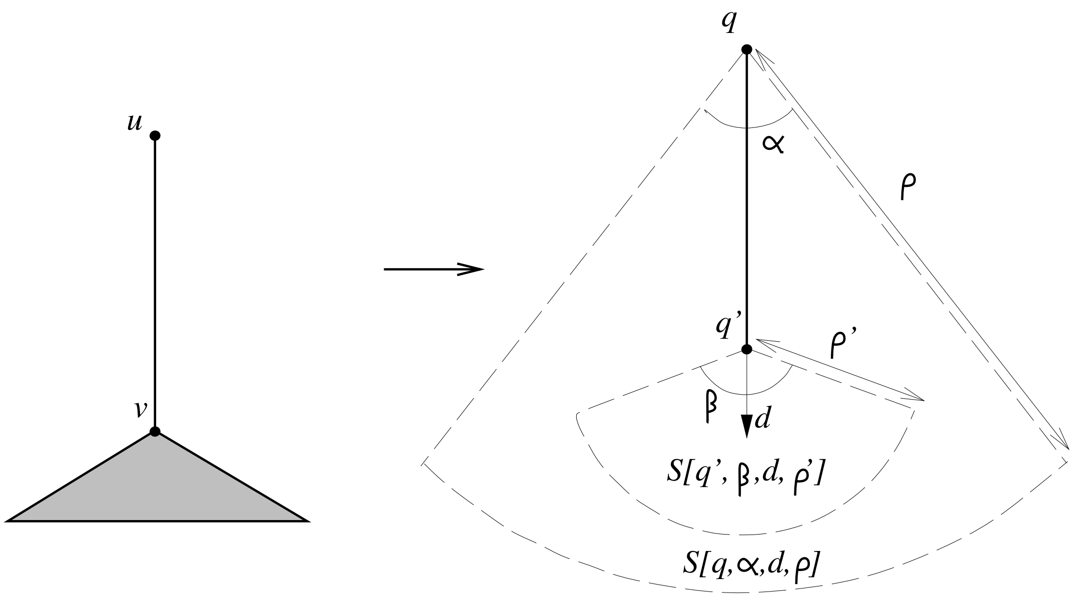




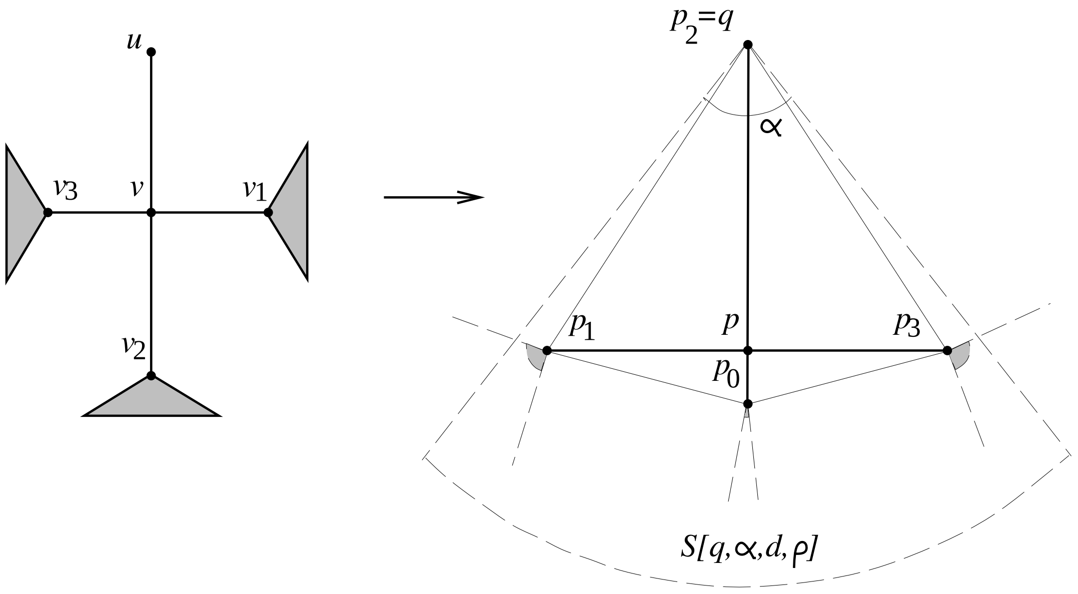
p
β
ρ
d1
d
d2
p1
p2
α1
α2
S1
S2
S



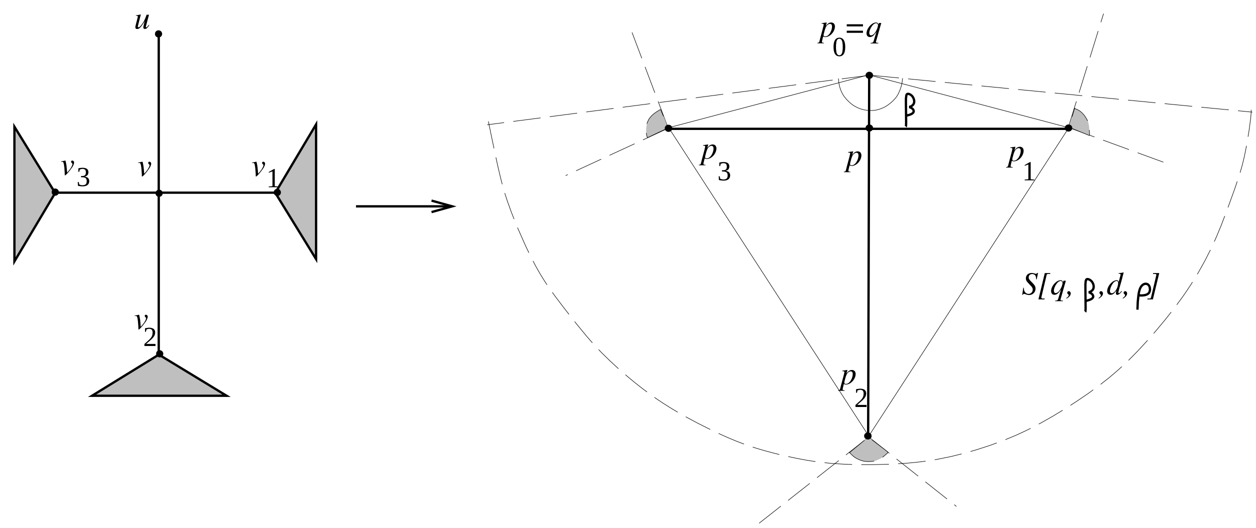
l0
r0
d0
β0
F0
S0
p0
p
r3
β3
S3
d3
p3
l3
F3
l1
β1
F1
d1
p1
S1
r1
Q
p2
S2
F2
r2
β2
l2
d2



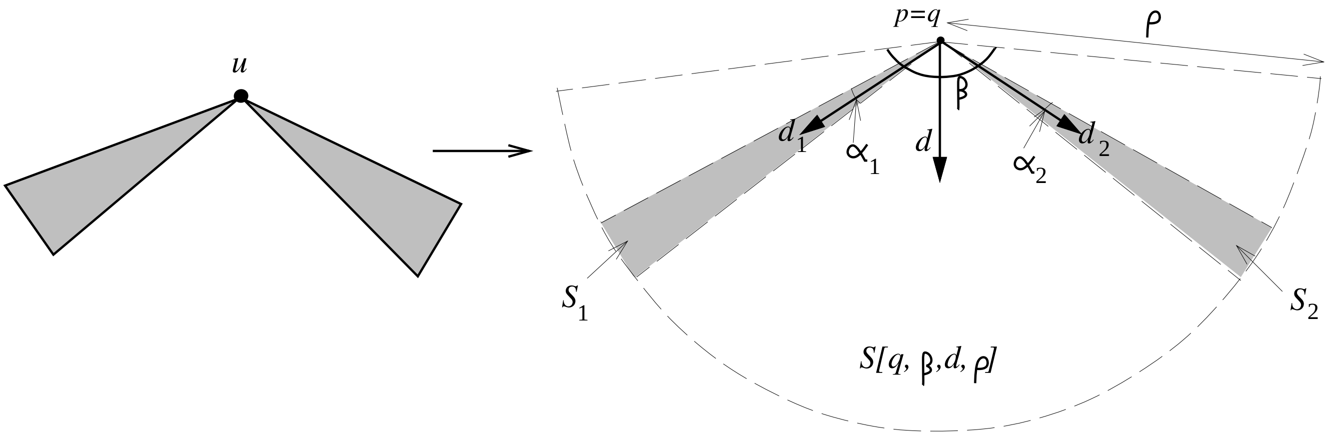
(a)



(b)



(a)



(b)