Upright Drawings of Planar Graphs on Three Layers

Md. Jawaherul Alam, Mashfiqui Rabbi, and Md. Saidur Rahman*

No Institute Given

Abstract. An upright drawing of a planar graph G on k layers is a planar straight-line drawing of G, where each vertex of G is placed on a set of k horizontal lines, called layers and no two adjacent vertices are placed on the same layer. In this paper, we give a linear-time algorithm to check whether or not a given planar graph G admits an upright drawing on three layers and if so, to obtain such a drawing of G. To the best of our knowledge, this is the first algorithm for upright drawings of planar graphs on k layers for $k > 2$.

1 Introduction

An *upright drawing of a graph* G is a planar straight-line drawing of G where the vertices of G are placed on a set of horizontal lines, called layers such that no two adjacent vertices are placed on the same layer. For example, Fig 1(b) and (c) illustrates two planar straight-line drawings Γ_1 and Γ_2 of the graph G in Fig 1(a). Among these two drawings, Γ_1 is not an upright drawing of G since the two adjacent vertices a and b are placed on the same layer in Γ_1. However, Γ_2 is an upright drawing of G that occupies three layers. On the other hand, the graph H of Fig. 1(d) does not admit any upright drawing on three layers although it admits an upright drawing Γ_H on four layers. One can infer from this simple example that the problem of determining whether a given graph admits an upright drawing on k layers for a given value of k is quite challenging. However, the problem is quite trivial for where the number of layers is less than three. In this paper, we give an algorithm to determine whether a graph G admits an upright drawing on three layers.

An upright drawing of a planar graph is a variant of the well studied graph drawing convention, named "layered drawings" [War77,Sud05]. A *layered drawing* of a planar graph G is a planar straight-line drawing of G such that the vertices are drawn on a set of layers. Thus an upright drawing of G is a layered drawing of G with the additional constraint that no two adjacent vertices of G are placed on the same layer. Layered drawings have important applications in VLSI layouts [Len90], DNA-mapping [WG86], information visualization [BETT99,KW01] etc. In some application areas, it is often desirable to obtain

* Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka-1000, Bangladesh. E-mail: jawaherul@yahoo.com, mashfiqui.rabbi@gmail.com, saidurrahman@cse.buet.ac.bd

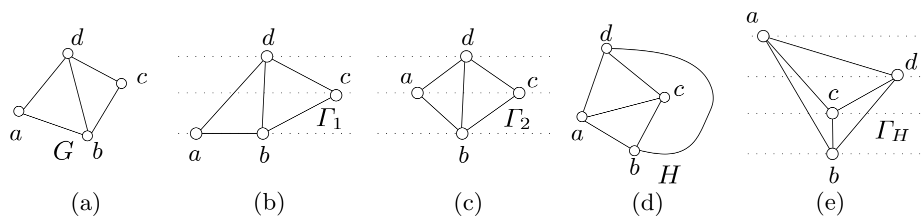


Fig. 1. (a) The graph G, (b) a drawing Γ of G on two layers.

upright drawings of a planar graphs on a desired number of layers. For example in the "standard cell" technology employed during the VLSI layout design process, the VLSI modules are placed on some constant number of previously fixed rows so that they can be lined up in rows on the integrated circuit. The placement of these modules thus gives a layered drawing of the graph obtained from the VLSI circuit where each vertex represents a module in the circuit and each edge represents an interconnection between two modules. Since the modules in a standard cell are designed so that the input and output lines are emitted from the top and the bottom of each module, this drawing is upright. Therefore a VLSI circuit can be placed on a VLSI chip with k rows using the standard cell technology if and only if the corresponding graph admits an upright drawing on k layers.

There has not been much work on upright drawings of planar graphs. Suderman gave a linear-time algorithm to obtain an upright drawing of a tree with pathwidth h on $\lceil 3h/2 \rceil$ layers [Sud04] but for upright drawings of a planar graph there is no such previous algorithm. Nevertheless, there is some significant results on algorithms to check whether a planar graph admits a layered drawing on two and three layers [Bie98,FLW03,CSW04] etc. There is also a previously known algorithm to check whether a planar graph admits an upright drawing on three layers with the additional constraint that adjacent vertices are placed only on adjacent layers in the drawing [Sud05]. However, for a general case of upright drawings of planar graphs, there is no necessary and sufficient condition for a graph to admit an upright drawing on three layers or on any fixed number of layers greater than two. In this paper, we give a linear-time algorithm that checks whether a given planar graph G admits an upright drawing on three layers and obtains an upright drawing of G on three layers if it admits one.

The rest of this section is organized as follows. In Section 2, we give some preliminary definitions that has been used throughout the paper. In Section 3, we present our algorithm to detect whether a tree T admits an upright drawing on three layers and to obtain an upright drawing of T on three layers if one exists. Finally, Section 6 is a conclusion.

2 Preliminaries

In this section, we give some definitions and present an outline of our algorithm.

Let $G = (V, E)$ be a simple graph with vertex set V and edge set E. We also denote the set of vertices of G by $V(G)$ and the set of edges of G by $E(G)$. Let (u, v) denote an edge of G joining two vertices u and v of G. Two vertices u and v of G are *neighbors* of each other if G has an edge (u, v). The edge (u, v) is said to be incident to the vertices u and v and the vertices u and v are said to be adjacent to each other. The *degree* of a vertex v in G is the number of incident edges of v in G. A graph G is a subgraph of another graph H if $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$. Let G be a subgraph of another graph H. Then $H - G$ denote the subgraph of H obtained from H by deleting all the vertices in G and the edges incident to these vertices.

A *path* P in a graph G is a sequence of vertices $v_0, v_1, \ldots, v_n$ such that G contains an edge $e_i = (v_{i-1}, v_i)$ for each $i, (1 \leq i \leq n)$ and all the vertices v_i, $(0 \leq i \leq n)$ are distinct. Such a path P is also called a v_0, v_n-path of G. The vertices v_0 and v_n of the path P are called the *end-vertices* of P. A graph G is *connected* if there exists a u, v-path in G for every pair of vertices $u, v \in V$. Otherwise, the graph G is disconnected. A component of a graph G is a connected subgraph of G which is not a subgraph of any other connected subgraph of G. A *cycle* in a graph G is a path in G whose end-vertices are the same. A *tree* is a connected graph which does not contain any cycle. A vertex u of a tree T having degree one in T is called a *leaf* of T. A vertex u of T having degree greater than one in T is called an *internal vertex* of T.

A k-th caterpillar is defined as follows.

(i) For $k = 1$, a k-th caterpillar is a single vertex tree.
(ii) For $k > 1$, a k-th caterpillar is a tree that contains a path S, called the spine, such that each component of $T - S$ is a $(k-1)$-th caterpillar.

The caterpillarity of a tree is k if it is a k-th caterpillar but not a $k+1$-th caterpillar.

A *k-layer drawing* of a tree T is a drawing of T where each vertex of T is placed on one of k horizontal lines called layers and each edge of T is drawn as a straight-line segment without any edge crossings. An *upright drawing of T on k layers* is a k-layer drawing Γ of T such that no two adjacent vertices in T are placed on the same layer in Γ. A *proper drawing of T on k layers* is a k-layer drawing Γ of T such that all the adjacent vertices in T are placed on consecutive layers in Γ. Note that a proper drawing of a tree T is also an upright drawing of T but the reverse is not true.

3 Upright Drawings of Trees

In this section, we give a necessary and sufficient condition for a tree to admit an upright drawing on three layers. Based on this characterization, we also give a linear time algorithm to check whether a given tree admits an upright drawing on three layers and to obtain such a drawing if it does.

It is well known that a tree T admits an upright drawing on two layers if and only if T is a caterpillar [omK97]. The following lemma extends this result for upright drawing on three layers.

Lemma 1. *A tree T admits an upright drawing on three layers if and only if T is an extended caterpillar.*

Proof. We first assume that T has an upright drawing Γ on three layers. Let u and v be the leftmost and the rightmost vertices of T in Γ. Let S denote the unique path between u and v in T. This path divides the entire area of Γ into two regions, both of which have the maximum height at most two. Hence each component of $T - S$ admits an upright drawing on two layers and hence it is a caterpillar [omK97]. Therefore by definition, T is an extended caterpillar and S is the corresponding spine.

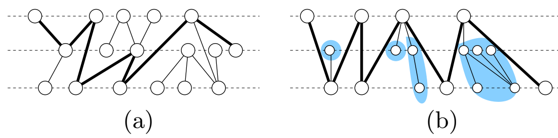


Fig. 2. (a) The spine of a tree that admits an upright drawing on three layers, (b) An upright drawing of an extended caterpillar on three layers.

We now assume that T is an extended caterpillar and we give an algorithm to obtain an upright drawing of T on three layers. Let us denote these three layers as l_1, l_2 and l_3 from top to bottom. Let $S = v_0, v_1, \ldots, v_f$ be the spine of T. We place the vertices of S on l_1 and l_3 layers such that consecutive vertices on S are placed on different layers and the x-coordinate of v_i is greater than the x-coordinate of v_{i-1} for $1 \leq i \leq f$. (See Fig. 2(b).) Since each component C of $T - S$ is a caterpillar, C admits an upright drawing Γ_C on two layers due to [omK97]. Let v_S be the vertex of S that is adjacent to some vertex v_C of C in T. Then we can place the drawing Γ_C of C (possibly after mirroring) on the two layers other than the layer on which v_S is placed and add the edge (v_S, v_C) using straight-line segment without any edge crossings. We thus obtain an upright drawing of T on three layers as illustrated in Fig. 2(b). $\mathcal{Q.E.D.}$

The proof of the sufficiency of the above lemma gives a linear-time algorithm to obtain an upright drawing of an extended caterpillar on three layers if the spine (or at least two end vertices of the spine) is given. If the spine is not specified, one can detect whether a given tree T is an extended caterpillar using this algorithm by considering each pair of leaves of T the end-vertices of an spine. However, this naïve approach takes $O(n^2)$ time. In the rest of this section, we give an outline of an algorithm that detects whether a given tree T is an extended caterpillar and in the positive case, find the spine of T in $O(n)$ time. Before presenting that, we need to define the notion of "compressing a vertex in a graph".

A vertex v of degree two in a graph G is said to be *compressable* if both of its neighbors have degree greater than two. Let v be a compressable vertex in a graph G and let v_l and v_r be the two neighbors of v in G. Then *to compress the* *vertex* v in G is the operation of deleting the vertex v (and its incident edges) from G and adding the edge (v_l, v_r) to G.

We now have the following lemma.

Lemma 2. *Let T be a tree and let T_c be the tree obtained compressing all the compressable vertex of the tree T. Let T^* be the tree obtained from T_c by deleting all the leaves of T_c. Then T is an extended caterpillar if and only if there is a path S^* in T^* such that each component of $T^* - S^*$ is a path.*

Proof. We first assume that T is an extended caterpillar and S is a spine of T as illustrated in Fig. 3(a) where the bold edges represents the spine. Let S^* be the subgraph of T^* obtained by compressing all the compressable vertices of S in T as illustrated in Fig. 3(c) by bold edges. We now show that for each component

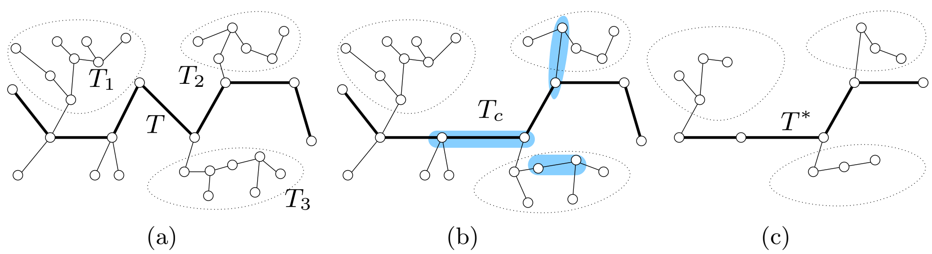


Fig. 3. (a) An extended caterpillar T, (b) The tree T_c obtained by compressing T, (c) The tree T^* obtained from T_c by deleting all the leaves of T_c.

C of $T - S$, the component in $T^* - S^*$ obtained from C is a path. since C is a caterpillar, deleting all the leaves of C yields a path P [Wes05]. Again, all the leaves of C are also leaves of T and T_c except possibly one leaf v_C of C, which is adjacent to a vertex v_S of S in T. For example, all the leaves of the component T_1 of $T - S$ in Fig. 3(a) are also leaves of T but for each of the components T_2 and T_3 of $T - S$, the black-colored leaf is not a leaf of T. We may assume that the degree of v_S is at least three, since otherwise v_S is and end-vertex of S and we can extend the spine S of T upto v_C. Let v'_C be the neighbor of v_C on C. If the degree of v'_C is also greater than two in C as for the component T_2 of $T - S$ in Fig. 3(a), then v_C is compressable in T and deleted in T_c. Hence the subgraph of T^* corresponding to C is the path P. On the other hand, if the degree of v'_C is two as for the component T_3 of $T - S$ in Fig. 3(a), then v_C is not deleted in T^* but the component of $T^* - S^*$ corresponding C is still a path with only the vertex v_C added to P. Thus each component of $T^* - S^*$ is a path as illustrated in Fig. 3(c).

We now assume that there is a path S^* in T^* such that each component of $T^* - S^*$ is a path and prove that T is an extended caterpillar. Let u and v be the two end-vertices of S^*. Let S be the unique path in T between u and v. Note that S^* and S contains the same set of vertices in the same order except possibly some vertices of S are compressed or deleted in T^*. If each component of $T - S$ is a caterpillar, then T is an extended caterpillar where S is the spine. We thus assume that there is a component C in $T - S$, which is not a caterpillar. Then C has the "Y-graph" as illustrated in Fig. 4(a) as its subgraph [Wes05]. Then, without loss of generality, we assume that this Y-graph is connected to S in any of the three different ways as illustrated in Fig. 4(b)–(d). One can easily see that

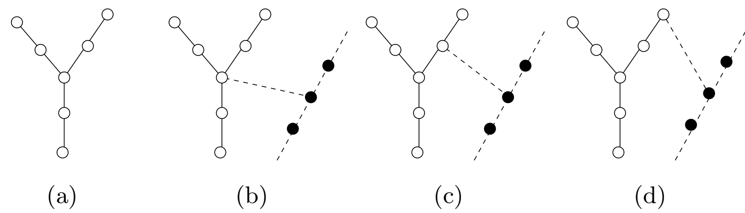


Fig. 4. (a) A Y-graph, (b)–(c) three ways of connection between a Y-graph and S.

in each of these cases, the component of $T^* - S^*$ correspoding to C is not a path, which is a contradiction. Therefore each component of $T - S$ is a caterpillar and hence T is an extended caterpillar, where S is the spine of T. $\mathcal{Q.E.D.}$

One can check in linear time whether a tree T has a path S such that each component of $T - S$ is a path by the algorithm presented in [ASRR08]. In the positive case, these algorithms also find such a path S for the given tree T that satisfies the above condition. The algorithms presented in Lemmas 1 and 3 then obtain an upright drawing of a given tree T on three layers if one exists. Therefore summarizing all the lemmas in this section, we obtain the following theorem.

Theorem 1. *One can determine in linear time whether a given tree T admits an upright drawing on three layers and obtain such a drawing of T if one exists.*

4 Upright Drawings of Biconnected Graphs

In this section, we first give a necessary and sufficient condition for a biconnected plane graph to admit an upright drawing on three layers. We also give a linear-time algorithm to obtain an upright drawing of a biconnected plane graph on three layers if it admits one. Finally, we show that if a biconnected planar graph G admits an upright drawing on three layers, then one can find in linear time an embedding Γ of G that admits an upright drawing on three layers.

We first have the following lemma that establishes a necessary condition for a biconnected plane graph to admit an upright drawing on three layers.

Lemma 3. *Let a biconnected plane graph G admit an upright drawing Γ on three layers. Then the simple weak dual graph G^* of G is a path.*

Proof. We first prove that G^* does not contain any cycle. Assume for a contradiction that G^* contains a cycle C. Then the faces of G corresponding to those vertices of G^*, that belong to the cycle C, induces at least one internal vertex of G with degree three or more as illustrated in Fig. 5(a). Therefore, to prove that G^* contains no cycle, it is sufficient to prove that G does not have any internal vertex with degree three or more.

Since the vertices of G that are on the top and bottom layers in Γ are on the outer face of G, all the internal vertices of G are on the middle layer. Let us assume that there is an internal vertex v of G with degree three or more. Then v is placed on the middle layer in Γ. Since Γ is an upright drawing, none of the neighbors of v in G are placed on the middle layer in Γ. Hence, either the top or the bottom layer contains at least two neighbors of v in G. Without loss of generality, let us assume that v has two neighbors u and w on the top layer in Γ. Since v is an internal vertex, there must be a path between u and w along the top layer as illustrated in Fig. 5(b). However, this is not possible since no two adjacent vertices are placed on the same layer in Γ.

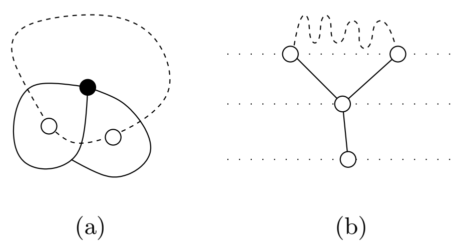


Fig. 5. (a) A cycle in G^* induces an internal vertex with degree three or more in G, (b) G does not contain an internal vertex of degree three or more.

We thus assume that there is no cycle in G^*. Since Any cycle requires at least three layers for any upright drawing, every face of G occupies all three layers in Γ. Therefore each face shares edges with at most two other faces of G, one to its left and one to its right in Γ. Hence, every vertex of G^* has degree at most two and G^* is a path. $\mathcal{Q.E.D.}$

We call a biconnected plane graph *a dual-path biconnected graph* if its simple weak dual graph is a path. The above lemma implies that if a biconnected plane graph G admits an upright drawing on three layers, then G is a dual-path biconnected graph. However, this condition is not sufficient. Before we give a necessary and sufficient condition for a biconnected plane graph to admit an upright drawing on three layers, we need some definitions. Let G be a dual-path biconnected plane graph with at least two faces and let G^* be its simple weak dual path. Let F_l and F_r be the faces corresponding to the leftmost and the rightmost vertices of G^* respectively. Let F'_l and F'_r be the faces of G, immediately right to the face F_l and immediately left to the face F_r respectively.

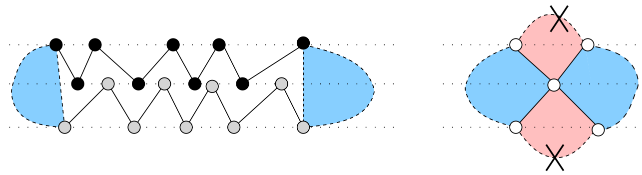
If we delete those vertices of F_l that are not on the face F_l' and those vertices of F_r that are not on the face F_r', then the outer cycle of G is divided into two paths. Let us denote these two paths by P_t and P_b and call them *top path* and *bottom path* respectively. Let the two end vertices of P_t be u and u' and the two end vertices of P_b be v and v' where u and v are on the face F_l and u' and v' are on the face F_r in G. We call u the *left-top*, v the *left-bottom*, u' the *right-top* and v' the *right bottom vertex* of G. We denote by $odd(u)$ ($even(u)$) the set of vertices that are at odd (even) distance from u along P_t. We also denote by $odd(v)$ ($even(v)$) the set of vertices that are at odd (even) distance from v along P_b. Similarly we define the notations $odd(u')$, $even(u')$, $odd(v')$ and $even(v')$. We now have the following theorem that gives a necessary and sufficient condition for a biconnected plane graph to admit an upright drawing on three layers.

Theorem 2. *Let G be a dual path biconnected graph with at least two faces. and let u and v be the left-top and left-bottom vertices of G. Let us define three sets of vertices; $S_{tb} = odd(u) \cup odd(v) \cup V_{in}$, $S_{tm} = odd(u) \cup even(v) \cup V_{in}$ and $S_{mb} = even(u) \cup odd(v) \cup V_{in}$, where V_{in} is the set of internal vertices of G. Then G admits an upright drawing on three layers if and only if at least one of the three sets of vertices S_{tb}, S_{tm} and S_{mb} is independent in G and contains no vertices of degree greater than three in G.*

Proof. Let F_l and F_r be the faces corresponding to the leftmost and the rightmost faces of G respectively and let P_t and P_b be the top path and bottom path of G respectively. Let us also assume that u' and v' are respectively the right-top and the right-bottom vertices of G. Then u and u' are the two end vertices of P_t and v and v' are the two end vertices of P_b, where u and v are on the face F_l and u' and v' are on the face F_r.

We first assume that G admits an upright drawing Γ on three layers. Since the drawing of each face requires all three layers of Γ, the faces of G are placed in the order of the corresponding vertices along the simple weak dual graph of G and the two faces F_l and F_r are drawn at the leftmost and rightmost position in Γ. Then the two paths P_t and P_b must be drawn between the drawings of F_l and F_r as illustrated in Fig. 6(a). One of these two paths (say P_t) must be drawn using the top and the middle layers only and the other (say P_b) using the middle and the bottom layers only since otherwise, there will be some edge crossings. However, since u and v are on the common boundary of F_l and the face immediately to the right of F_l, either u and v are adjacent to each other or both of them are adjacent to some internal vertex. In both the cases, both u and v are not placed on the middle layer since no two adjacent vertices are placed on the same layer in Γ. Therefore, there are three possible cases regarding the placement of the two vertices u and v on these three layers;

(i) u on the top layer and v on the bottom layer
(ii) u on the top layer and v on the middle layer
(iii) u on the middle layer and v on the bottom layer

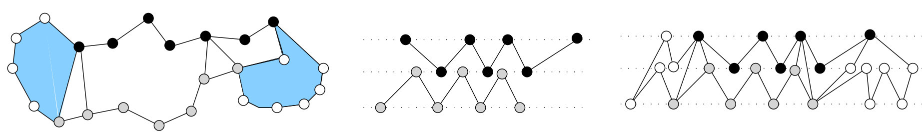


(a) (b)

Fig. 6. (a) (P_t) and P_b are drawn bertween the drawing of F_l and F_r, (b) no vertex placed on the middle layer have degree greater than three.

Let us first assume that u and v are placed on the top and bottom layers respectively in Γ. Then all the vertices of the set $S_{tb} = odd(u) \cup odd(v) \cup V_{in}$ must be placed on the middle layers in Γ. By a similar reasoning, it can be showed that the set of vertices S_{tm} (S_{mb}) are placed on the middle layer in Γ if u is placed on the top (middle) layer and v is placed on the middle (bottom) layer in Γ. Again the vertices of G placed on the same layer of an upright drawing gives an independent set in G. Furthermore Since G is biconnected, no vertex w of G with degree greater than three is placed on the middle layer in Γ since otherwise, w would be a cut vertex in G as illustrated in Fig. 6(b). Therefore, at least one of the three sets S_{tb}, S_{tm} and S_{mb} is independent in G and contains no vertices with degree greater than three in G.

We now assume that at least one of the three sets (say S_{tb}) is independent in G and contains no vertices of degree greater than three in G as illustrated in Fig.7(a). Under this assumption we will constructively obtain an upright drawing



(a) (b) (c)

Fig. 7. (a) A dual-path biconnected graph G where S_{tb} is independent and have no vertices with degree greater than three, (b) drawing of P_t and P_b, (c) drawing of G.

of G on three layers. We place the vertices of P_t on the top and middle layer in the increasing order of x-coordinate such that all the vertices of the set $even(u)$ are placed on the top layer and all the vertices of the set $odd(u)$ are placed on the middle layer. Similarly, we place the vertices of P_b on the bottom and middle layer in the increasing order of x-coordinate such that all the vertices of the set $even(u)$ are placed on the bottom layer and all the vertices of the set $odd(u)$ are placed on the middle layer. While placing the vertices of the P_b path on the bottom and middle layer, we take special care so that if a vertex v_t of P_t and a vertex v_b of P_b are adjacent to each other or have an internal vertex as their common neighbor, then these two vertices are placed in such positions that we can add an edge between them without creating any edge crossings. (See Fig. 7(b).) This is always possible because no vertices of $odd(u)$ and $odd(v)$ have degree greater than three. We note that all the internal vertices of G have degree two in G and have exactly one neighbor from P_t and exactly one neighbor from P_b they all belong to the set S_{tb}. We place all these internal vertex of G in such a position on the middle layer that we can add an edge between these vertices and their neighbors on the two paths P_t and P_b. We finally place the vertices of the two faces F_l and F_r that have not been yet placed and add all the edges of G to complete the drawing. (See Fig. 7(c).) $\mathcal{Q.E.D.}$

The above lemma gives a necessary and sufficient condition for a biconnected plane graph to admit an upright drawing on three layers. We now address the problem for a biconnected planar graph. Let G be a biconnected planar graph and we want to check whether G admit an upright drawings on three layers. Since G may have an exponential number of embeddings, a naive approach of checking all these embeddings for an existence of an upright drawing on three layers would take an exponential amount of time. But the following lemma implies that we can find a suitable embedding Γ in linear time so that only checking the embedding Γ of G for an existence of an upright drawings on three layers would suffice for the graph G itself.

Lemma 4. *Let G be a biconnected planar graph that admits an upright drawing on three layers. Then One can find in linear time an embedding Γ of G such that the plane graph corresponding to Γ admits an upright drawings on three layers.*

Proof. We first describe our algorithm to find a desired embedding Γ of G in linear time. Then we will prove that the embedding Γ given by our algorithm admits an upright drawing on three layers.

If a biconnected plane graph admits an upright drawing on three layers, then all its internal vertices have degree at most two as pointed out in Lemma 4. Therefore we first find an embedding Γ_1 of G where all the internal vertices in Γ_1 have degree two in G. We obtain a graph G_1 from G by adding a vertex w and adding edges between w and all the vertices with degree greater than two in G. Since the graph G admits an upright drawing, it has an embedding where all the internal vertices have degree two, i.e. all the vertices of degree greater than two are in the outer face. Therefore, the new graph G_1 is planar. We now find such an embedding Γ' of G_1 that has the vertex w in the outer face. Then deleting the vertex w (and all its incident edges) from Γ' yields an embedding Γ_1 of G such that all the internal vertices in Γ_1 have degree two in G.

$\mathcal{Q.E.D.}$

5 Upright Drawings of General Graphs

In this section we give a necessary and sufficient condition for a general planar graph to admit an upright drawing on three layers. We assume that the graph is connected since otherwise, the condition can be tested for each component of the graph. Beofre presenting the necessary and sufficient condition, we need some definitions.

Let G be a connected planar graph. A *biconnected component* of G is a maximal biconnected subgraph of G. We also denote each component of the graph obtained by deleting the vertices of all the biconnected components from G by a *tree part*. A tree part T of G is said to be connected to a vertex u of a biconnected component of G if a vertex v of T is adjacent to v. A tree part T of G is *trivial* if T consists of a single vertex. We also call a tree part a *2-layer tree component* if it is not trivial and is a caterpillar. Again, we call a tree part a *3-layer tree component* if it is an extended caterpillar but is not a caterpillar. A 2-layer or a 3-layer tree component T is said to be compatible with a biconnected component B if an end-verex of a spine of T is adjacent to a vertex of B; otherwise T is said to be *non-compatible* with B.

Let a biconnected component B of G admits an upright drawing Γ on three layers. A vertex v of B is called a left (right) boundary vertex of Γ if there is no vertex or edge to the left (right) of v in Γ. The left (right) boundary of Γ consists of all the left (right) boundary vertices of Γ. A boundary vertex of Γ is either a left or a right boundary vertex. A boundary vertex of Γ is called a top, bottom or middle layer boundary vertex if it is on the top, bottom or middle layer respectively. Two boundary vertices of Γ are said to share a common boundary if both of them are either left boundary vertices or right boundary vertices. A biconnected component B_1 is said to be connected to a compatible biconnected component B_2 if B_1 and B_2 admits upright drawings Γ_{B_1} and Γ_{B_2} respectivle on three layers such that one of the following conditions holds.

- Γ_{B_1} and Γ_{B_2} has a common middle layer boundary vertex v, and all the tree parts connected to a boundary vertex on the same boundary as v in Γ_{B_1} and Γ_{B_2} are trivial.
- Γ_{B_1} and Γ_{B_2} has a common boundary vertex v, and v is a middle layer boundary vertex in neither Γ_{B_1} nor Γ_{B_2}.
- Γ_{B_1} has a boundary vertex v_1 adjacent to a boundary vertex v_2 of Γ_{B_2}, and both v_1 and v_2 are not middle layer boundary vertices of Γ_{B_1} and Γ_{B_2} respectively
- There is a tree part T of G such that T is an extended caterpillar where the end vertices of a spine of T are adjacent to a boundary vertex of Γ_{B_1} and a boundary vertex of Γ_{B_2} respectively.

We now have the following theorem that gives a necessary and sufficient condition for a connected graph to admit an upright drawing on three layers.

Theorem 3. *Let G be a connected planar graph. G admits an upright drawing on three layers if and only if each of its biconnected components B admits an upright drawing Γ_B on three layers such that for each biconnected component B, no tree part is connected to an internal vertex of B, all the tree parts connected to a vertex of B other than the boundary vertices are trivial and for each of the left and right boundaries of Γ_B, one of the following conditions (1)–(4) holds.*

(1) There is no middle layer boundary vertex that is connected to a tree part; one of the top and bottom layer boundary vertex is connected to at most one compatible 3-layer tree component or one compatible biconnected component and zero or more caterpillars and the other boundary vertex is connected to at most one compatible 2-layer tree component and zero or more trivial tree parts.
(2) There is a middle layer boundary vertex which is connected to exactly one compatible 3-layer tree component or exactly one non-compatible 2-layer tree component or exactly one compatible biconnected component and at most one compatible 2-layer tree component and zero or more trivial tree parts are connected to the vertices on the same boundary of Γ_B.
(3) There is a middle layer boundary vertex which is connected to at most one compatible 2-layer tree component and zero or more trivial tree parts and at most one of the top and bottom boundary vertex is connected to at most one compatible 3-layer tree component or one compatible biconnected component and zero or more caterpillars and all the tree parts connected to the other boundary vertex are trivial.
(4) There is a middle layer boundary vertex which is connected to exactly two compatible caterpillar and the other tree parts connected to the the vertices on the same boundary are trivial.

6 Conclusion

In this paper, we have given a necessary and sufficient condition to check whether a given tree T admits an upright drawing on three layers. Based on this characterization, we have also given a recognition and drawing algorithm for upright drawings of trees on three layers. We have also given a necessary condition for a tree to admit an upright drawing on k layers for a given value of $k > 3$. It remains as our future work to obtain a necessary and sufficient condition for a tree to obtain an upright drawing on k layers for any given value of k.

References

[ASRR08] M. J. Alam, M. A. H. Samee, M. M. Rabbi, and M. S. Rahman. Upward drawing of trees on the minimum number of layers (extended abstract). In *the Proceedings of the Workshop on Algortihms and Computation (WALCOM) 2008*, volume 4921 of *Lecture Notes in Computer Science*, pages 88–99. Springer-Verlag, 2008.

[BETT99] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, Upper Saddle River, New Jersey, 1999.

[Bie98] T. C. Biedl. Drawing planar partitions i: Ll-drawings and lh-drawings. In *the Proceedings of the fourteenth annual symposium on Computational geometry (SCG) 1998*, pages 287–296. ACM, 1998.

[CSW04] S. Cornelsen, T. Schank, and D. Wagner. Drawing graphs on two and three lines. *Journal of Graph Algorithms and Applications*, 8(2):161–177, 2004.

[FLW03] S. Felsner, G. Liotta, and S. K. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. *Journal of Graph Algorithms and Applications*, 7(4):363–398, 2003.

[KW01] M. Kaufmann and D. Wagner. *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Computer Science*. Springer-Verlag, London, UK, 2001.

[Len90] T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layouts*. Wiley, New York, NY, USA, 1990.

[omK97] U. Fößmeier and M. Kaufmann. Nice drawings for planar bipartite graphs. In *the Proceedings of the 3rd Italian Conference on Algorithms and Complexity (CIAC) 1997)*, pages 122–134. Springer-Verlag, 1997.

[Sud04] M. Suderman. Pathwidth and layered drawing of trees. *International Journal of Computational Geometry and Applications*, 14(3):203–225, 2004.

[Sud05] M. Suderman. Proper and planar drawings of graphs on three layers. In *the Proceedings of Graph Drawing 2005*, volume 3843 of *Lecture Notes in Computer Science*, pages 434–445. Springer-Verlag, 2005.

[War77] J. N. Warfield. Crossing theory and hierarchy mapping. *IEEE Transactions on Systems, Man, and Cybernetics*, 7:502–523, 1977.

[Wes05] D. B. West. *Introduction to Graph Theory*. 2005.

[WG86] M. S. Waterman and J. R. Griggs. Interval graphs and maps of dna. *Bulletin of Mathematical Biology*, 48:189–195, 1986.