Upward Planar Drawings of Series-Parallel Digraphs with Maximum Degree Three
(Extended Abstract)

Md. Abul Hassan Samee and Md. Saidur Rahman

Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology (BUET).
{samee,saidurrahman}@cse.buet.ac.bd

Abstract. An upward planar drawing of a digraph G is a planar drawing of G where every edge is drawn as a simple curve monotone in the vertical direction. A digraph is upward planar if it has an embedding that admits an upward planar drawing. The problem of testing whether a digraph is upward planar is NP-complete. In this paper we give a linear-time algorithm to test the upward planarity of a series-parallel digraph G with maximum degree three and obtain an upward planar drawing of G if G admits one.

Key words: Graph Drawing, Upward Planar Drawing, Directed Acyclic Graph, Series-Parallel Graph, Algorithm, SPQ-tree.

1 Introduction

In an upward planar drawing of a digraph, every vertex is mapped to a point in the Euclidean plane and every edge is drawn as a simple curve monotone in the vertical direction without producing any crossing with other edges, as illustrated in Fig. 1(a). Upward planar drawings of digraphs find important applications in visualization of the hierarchical network structures which frequently arise in software engineering, project management and visual languages [BDMT98]. Unfortunately, not all digraphs have upward planar drawing. One can easily understand that if a digraph contains a cycle, then one of the edges on the cycle cannot be drawn monotonically in the upward direction (see the cycle induced by vertices b, c, h and g of digraph G' in Fig. 1(b)). A digraph is upward planar if it has an embedding which admits an upward planar drawing. Acyclicity is a necessary condition for a digraph to be upward planar. Throughout this paper, wherever we refer to a digraph, we mean an acyclic digraph. However, acyclicity is not a sufficient condition for upward planarity. For example, the acyclic digraph G'' in Fig. 1(c) is not upward planar; there are four possible upward planar

M. Kaykobad and Md. Saidur Rahman (Eds.): WALCOM 2007, pp. 28–45, 2007.

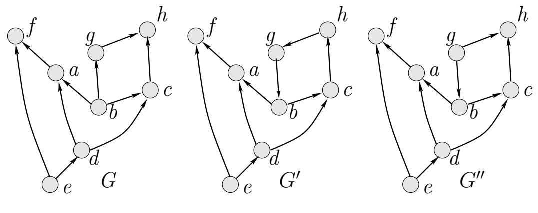

Fig. 1. (a) An upward planar digraph G, (b) a digraph G' which contains a cycle and therefore is not upward planar and (c) an acyclic digraph G'' which is not upward planar.

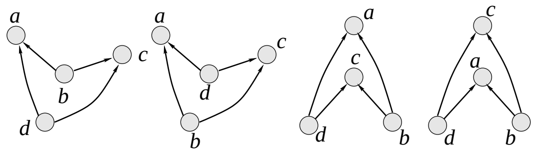

Fig. 2. The four possible embeddings of face F_1 of digraph G''.

embeddings of the undirected cycle induced by the vertices a, b, c and d (see Fig. 2), and starting with any of these four embeddings, the remaining edges cannot be added in any way to obtain an upward planar drawing of G''.

The problem of testing upward planarity of a digraph is one of the most challenging problems in the area of graph drawing and has been studied with extensive effort. Linear-time algorithms are known for testing whether a digraph admits a planar drawing [HT74,BL76]. Testing whether a digraph admits an upward drawing can also be solved in linear-time using the well-known topological sorting technique [CLRS01]. Nevertheless, combining these two properties makes the problem NP-hard [GT94]. The problem can be studied both in the fixed embedding setting and in the variable embedding setting. In the fixed embedding setting, the algorithm cannot alter the given embedding, and if that particular embedding is not upward planar, the output must be negative although some other embedding of the same graph could be upward planar. Fig. 3(a) shows an upward planar embedding of a digraph G whose embedding in Fig. 3(b) is not upward planar. Fig. 3(c) is another upward planar embedding of G. Bertolazzi *et al.* [BDLM94] have given an algorithm to test upward planarity in time $O(n^2)$ in the fixed embedding setting.

In the variable embedding setting, the algorithm can give a negative output only if there is no upward planar embedding of the input graph. Garg and Tamassia [GT01] proved that it is an NP-complete problem to determine whether a digraph has an upward planar drawing in the variable embedding setting. Nevertheless, the problem has been studied in the variable embedding setting for some restricted classes of digraphs [Pap95,HL96,BDMT98]. Garg and Tamassia

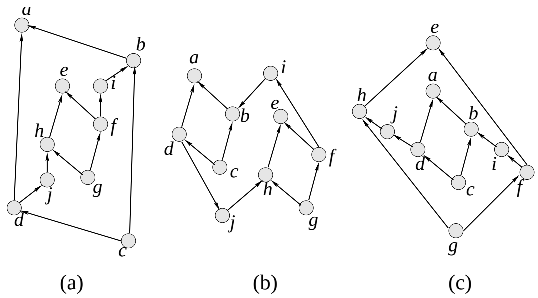

Fig. 3. (a) An upward planar digraph G and an upward planar embedding of G, (b) a non-upward planar embedding of G, and (c) another upward planar embedding of G.

[GT95] proved that a series-parallel digraph with single source and single sink is always upward planar. Unfortunately, a series-parallel digraph G with multiple sources and sinks may not be upward planar, and testing upward planarity for digraphs with multiple sources and sinks is more difficult. Recently, Didimo *et al.* [DGL06] provided an algorithm that tests upward planarity of series-parallel digraphs in time $O(n^4)$ in the variable embedding setting.

In this paper, we study upward planar drawings of series-parallel digraphs of $\Delta \leq 3$ with multiple sources and sinks in the variable embedding setting. For such a digraph G, we give a linear-time algorithm to construct an upward planar drawing of G if G admits one. The approach of our algorithm is different from the one presented in [DGL06] and the algorithm in [DGL06] requires time $O(n^4)$ even for a series-parallel digraph with the maximum degree three. The main idea of our algorithm is as follows. Our algorithm works in two phases, namely a testing phase and a construction phase. We begin with a decomposition tree called SPQ-tree $\mathcal{T}$ of G. In the testing phase, we traverse $\mathcal{T}$ bottom-up and test the feasibility of obtaining an upward planar drawing of G. If this phase fails, we declare that G is not an upward planar digraph. If the testing phase succeeds, we start the construction phase and using the information obtained in the testing phase, we obtain an upward planar embedding of G in a top-down traversal of $\mathcal{T}$.

The rest of the paper is organized as follows. Section 2 describes some definitions and presents preliminary results. In Section 3 we describe our primary findings on upward planarity of series-parallel digraphs with $\Delta \leq 3$. Section 4 presents our algorithm to test upward planarity and find an upward planar drawing of a biconnected series-parallel digraph with $\Delta \leq 3$. Finally, Section 5 is a conclusion.

2 Preliminaries

In this section we give some definitions and present preliminary results.

Let $G = (V, E)$ be a connected graph with vertex set V and edge set E. The *degree* of a vertex v, $deg(v)$ is the number of edges incident to v in G. We denote the maximum of the degree of the vertices of G by $\Delta(G)$. The *connectivity* $\kappa(G)$ of a graph G is the minimum number of vertices whose removal results in a disconnected graph or a single-vertex graph K_1. We say that G is *k-connected* if $\kappa(G) \geq k$. A planar drawing of G partitions the plane into topologically connected regions called *faces*. The unbounded face is called the *outer face*, the remaining faces are called *inner faces*. A *path* in G is an ordered list of distinct vertices $v_1, v_2, \ldots, v_q \in V$ such that $(v_{i-1}, v_i) \in E$ for all $i, 2 \leq i \leq q$. A path P is called a *u, v-path* if u and v are the first and last vertices in P respectively.

A graph $G = (V, E)$ is called a *series-parallel graph* (with source s and sink t) if either G consists of a pair of vertices connected by a single edge or there exist two series-parallel graphs $G_i = (V_i, E_i), i = 1, 2$, with source s_i and sink t_i such that $V = V_1 \cup V_2, E = E_1 \cup E_2$, and either $s = s_1$, $t_1 = s_2$ and $t = t_2$ or $s = s_1 = s_2$ and $t = t_1 = t_2$ [REN05]. A pair $\{u, v\}$ of vertices of a connected graph G is a *split pair* if there exist two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ satisfying the following two conditions:

1. $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{u, v\}$; and
2. $E = E_1 \cup E_2$, $E_1 \cap E_2 = \emptyset$, $|E_1| \geq 1$, $|E_2| \geq 1$.

Thus every pair of adjacent vertices is a split pair. A *split component* of a split pair $\{u, v\}$ is either an edge (u, v) or a maximal connected subgraph H of G such that $\{u, v\}$ is not a split pair of H.

Let G be a biconnected series-parallel graph. Let (s, t) be an edge of G. The SPQ-tree $\mathcal{T}$ of G with respect to a *reference edge* $e = (s, t)$ describes a recursive decomposition of G induced by its split pairs [GL99]. Tree $\mathcal{T}$ is a rooted ordered tree whose nodes are of three types: S, P and Q. Each node x of $\mathcal{T}$ corresponds to a subgraph of G, called its *pertinent graph* $G(x)$. Each node x of $\mathcal{T}$ has an associated biconnected multigraph, called the *skeleton* of x and denoted by *skeleton*(x). Tree $\mathcal{T}$ is recursively defined as follows.

• *Trivial Case*: In this case, G consists of exactly two parallel edges e and e' joining s and t. $\mathcal{T}$ consists of a single Q-node x, and the skeleton of x is G itself. The pertinent graph $G(x)$ consists of only the edge e'.

• *Parallel Case*: In this case, the split pair $\{s, t\}$ has three or more split components $G_0, G_1, \cdots, G_k, k \geq 2$, and G_0 consists of only a reference edge $e = (s, t)$. The root of $\mathcal{T}$ is a P-node x. The *skeleton*(x) consists of $k+1$ parallel edges $e_0, e_1, \cdots, e_k$ joining s and t, where $e_0 = e = (s, t)$ and $e_i, 1 \leq i \leq k$, corresponds to G_i. The pertinent graph $G(x) = G_1 \cup G_2 \cup \cdots \cup G_k$ is a union of $G_1, G_2, \cdots, G_k$. As an example, the skeleton of P-node P_2 in Fig. 4 consists of three parallel edges joining vertices a and c and Figure 4(d) depicts the pertinent graph of P_2.

• *Series Case*: In this case the split pair $\{s, t\}$ has exactly two split components, and one of them consists of the reference edge e. One may assume that the other split component has cut-vertices $c_1, c_2, \cdots, c_{k-1}$, $k \geq 2$, that partition the component into its blocks $G_1, G_2, \cdots, G_k$ in this order from s to t. Then the root of $\mathcal{T}$ is an S-node x. The skeleton of x is a cycle $e_0, e_1, \cdots, e_k$ where $e_0 = e$, $c_0 = s$, $c_k = t$, and e_i joins c_{i-1} and c_i, $1 \leq i \leq k$. The pertinent graph $G(x)$ of node x is a union of $G_1, G_2, \cdots, G_k$. For example, the skeleton of S-node S_3 in

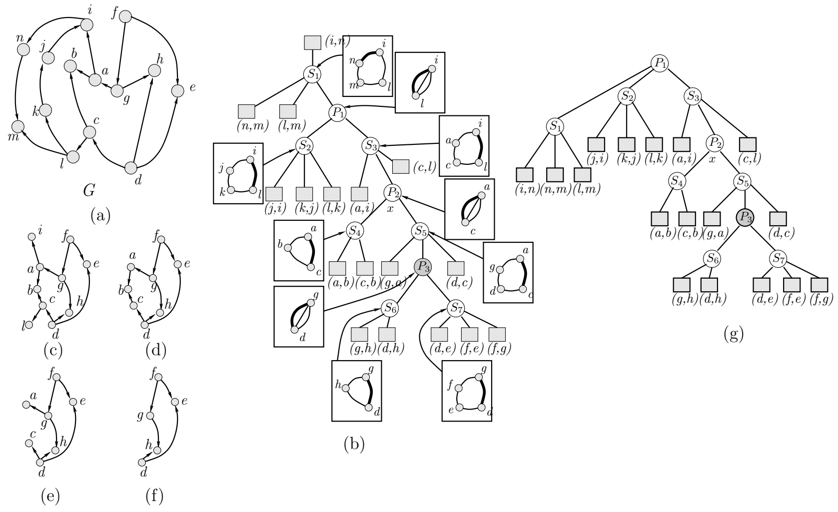

Fig. 4. (a)A biconnected series-parallel graph G with $\Delta = 3$, (b) SPQ-tree $\mathcal{T}$ of G with respect to reference edge (i, n), and skeletons of P- and S-nodes, (c) the pertinent graph $G(S_3)$ of S-node S_3, (d) the pertinent graph $G(P_2)$ of P-node P_2, (e) the pertinent graph $G(S_5)$ of S-node S_5, (f) the pertinent graph $G(P_3)$ of P-node P_3, (g) SPQ-tree $\mathcal{T}$ of G with P-node P_1 as the root

Fig. 4 is the cycle a, i, l, c, a and Figure 4(c) depicts the pertinent graph $G(S_3)$ of S_3.

In each of the cases mentioned above, we call the edge e the *reference edge* of node x. Except for the trivial case, node x of $\mathcal{T}$ has children $x_1, x_2, \cdots, x_k$ in this order; x_i is the root of the SPQ-tree of graph $G(x_i) \cup e_i$ with respect to the reference edge $e_i, 1 \leq i \leq k$. We call edge e_i *the reference edge of node* x_i, and call the endpoints of edge e_i the *poles* of node x_i. The tree obtained so far has a Q-node associated with each edge of G, except the reference edge e. We complete the SPQ-tree $\mathcal{T}$ by adding a Q-node, representing the reference edge e, and making it the parent of x so that it becomes the root of $\mathcal{T}$. An example of the SPQ-tree of a biconnected series-parallel graph in Fig. 4(a) is illustrated in Fig. 4(b), where the edge drawn by a thick line in each skeleton is the reference edge of the skeleton.

The SPQ-tree $\mathcal{T}$ defined above is the one used in [REN05] and is a special case of an "SPQR-tree"[DT96,GL99] where there is no R-node and the root of the tree is a Q-node corresponding to the reference edge e. One can easily modify $\mathcal{T}$ to an SPQ-tree $\mathcal{T}'$ with an arbitrary P-node as the root as illustrated in Fig. 4(g).

Let G be a planar digraph. G is a *series-parallel digraph* if the underlying undirected graph of G is a series-parallel graph. The SPQ-tree of a series-parallel digraph G is exactly the same as the one of the underlying undirected series-parallel graph of G. In the remainder of this paper, we consider an SPQ-tree

$\mathcal{T}$ of a series-parallel digraph G with a P-node as the root. If $\Delta(G) = 2$, then the underlying undirected graph of G is a cycle and $|E| - |V| = 0$. It has been shown in [HL05] that all acyclic digraphs with $|E| - |V| < 2$ are upward planar. Hence, for $\Delta(G) = 2$, G is always upward planar. One may thus assume that $\Delta(G) \geq 3$, and that the root P-node of $\mathcal{T}$ has three or more children. Then the *pertinent digraph* $G(x)$ of each node x is the subgraph of G induced by the edges corresponding to all descendant Q-node of x. Based on the assumption that $\Delta(G) = 3$, the following facts hold [REN05].

Fact 1 *Let (s, t) be the reference edge of an S-node x of $\mathcal{T}$, and let $x_1, x_2, \cdots, x_k$ be the children of x in this order from s to t. Then*

(i) each child x_i of x is either a P-node or a Q-node;
(ii) both x_1 and x_k are Q-nodes; and
(iii) x_{i-1} and x_{i+1} must be Q-nodes if x_i is a P-node where $2 \leq i \leq k - 1$.

Fact 2 *Each non-root P-node of $\mathcal{T}$ has exactly two children. A child of a non-root P-node can be an S- or a Q-node.*

A P-node in an SPQ-tree $\mathcal{T}$ is *primitive* if it does not have any descendant P-node in $\mathcal{T}$. Let x be a primitive P-node in $\mathcal{T}$. Let x_l and x_r be the left and right child of x in $\mathcal{T}$ respectively. Then the underlying undirected graph of $G(x) = G(x_l) \cup G(x_r)$ is a cycle and hence the digraph $G(x)$ is upward planar [HL05]. Therefore, the pertinent digraph $G(x)$ of every primitive P-node x in $\mathcal{T}$ is always upward planar. We define that the *height* of a primitive P-node is zero. The *height* of any other P-node is $(i + 1)$ if the maximum of the heights of its descendant P-nodes is i. The P-node P_3 in Fig. 4(b) is a primitive P-node. The heights of the other two P-nodes P_2 and P_1 in Fig. 4(b) are 1 and 2 respectively.

Let G be a planar digraph. A drawing Γ of G is an *upward planar drawing* if it has no edge-crossing and all the edges of G are drawn as simple curves monotonically increasing in the vertical direction. G is an *upward planar digraph* if G admits an upward planar drawing. One can observe that the following lemma holds for an upward planar digraph G.

Lemma 1. *A digraph G is upward planar if and only if every subgraph H of G is upward planar.*

Let G be a digraph with a fixed planar embedding. A vertex v of G is *bimodal* if the circular list of edges incident to v can be partitioned into two (possibly empty) lists, one consisting of incoming edges and the other consisting of outgoing edges. If all vertices of G are bimodal then G is called bimodal. Acyclicity and bimodality are necessary conditions for the upward planarity of an embedded planar digraph [BDLM94]. However, they are not sufficient conditions.

Let f be a face of an embedded planar bimodal digraph G and suppose that the boundary of f is visited clockwise if f is an inner face, and counterclockwise if f is the outer face. Let $\alpha = (e_1, v, e_2)$ be a triplet such that v is a vertex of the boundary of f and e_1, e_2 are two incident edges of v that are consecutive on the boundary of f. Triplet α is called an *angle of f*. We call an angle α a *switch* *angle of* f if either the direction of e_1 is opposite to the direction of e_2 on the boundary of f or e_1 and e_2 coincide. Note that if e_1 and e_2 coincide then G is not biconnected. If e_1 and e_2 are both incoming in v, then α is a *sink-switch of* f and if they are both outgoing, then α is a *source-switch of* f. A source or a sink of G is called a *switch vertex* of G and a vertex that is not a switch vertex is called an *ordinary vertex* of G. In the remainder of this paper we refer to a switch angle of a face f by calling it simply a *switch of* f.

Let G be an embedded planar digraph. Let Γ be an upward planar drawing of G and let α be an angle of a face f of G. We assign a label F to the angle α in face f if α is not a switch of f. Otherwise α is a switch of f, and we label α in f with a letter L if α has a value greater than π in Γ and with a letter S if the value of α in Γ is less than π. We assign labels to all angles of G as mentioned above and obtain a labeled embedded digraph. We call this labeled embedded digraph an *upward planar representation* of G and denote it by U_G. The drawing Γ is said to be an upward planar drawing that *preserves* U_G [DGL06]. Figure 5(b) illustrates U_G of the graph G in Fig. 5(a) for the upward planar drawing Γ in Fig. 5(c). It is mentionable that any arbitrary labeling of the switches of G may not have a corresponding upward planar drawing and hence may not be regarded as an upward planar representation of G. The conditions which must be met in order to obtain U_G are described below.

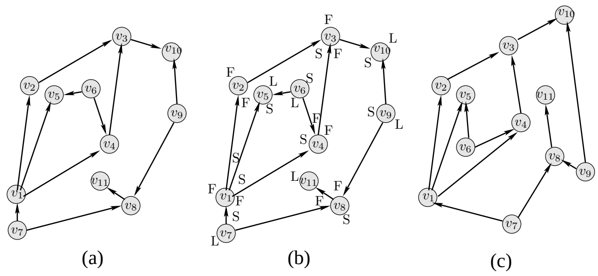

Fig. 5. Illustration of upward planar representation and upward planar drawing.

Let G be a digraph with a fixed planar embedding. Let Φ be an assignment which assigns a label of either L, S, or F to each angle of every face of G. For vertex v of G, we denote by $L(v)$, $S(v)$, and $F(v)$ the number of angles at v that Φ labels with L, S, and F respectively. For a face f of G, we denote by $L(f)$, $S(f)$, and $F(f)$ the number of angles of f that are labeled by Φ with L, S, and F respectively. We call the assignment Φ an *upward consistent assignment* if the following two conditions hold for Φ:

(i) for each switch vertex v of G, $L(v) = 1$, $S(v) = deg(v) - 1$, $F(v) = 0$, and for each ordinary vertex v of G, $L(v) = 0$, $S(v) = deg(v) - 2$, $F(v) = 2$; and
(ii) for each inner face f, $S(f) - L(f) = 2$ and for the outer face f, $S(f) - L(f) = -2$.

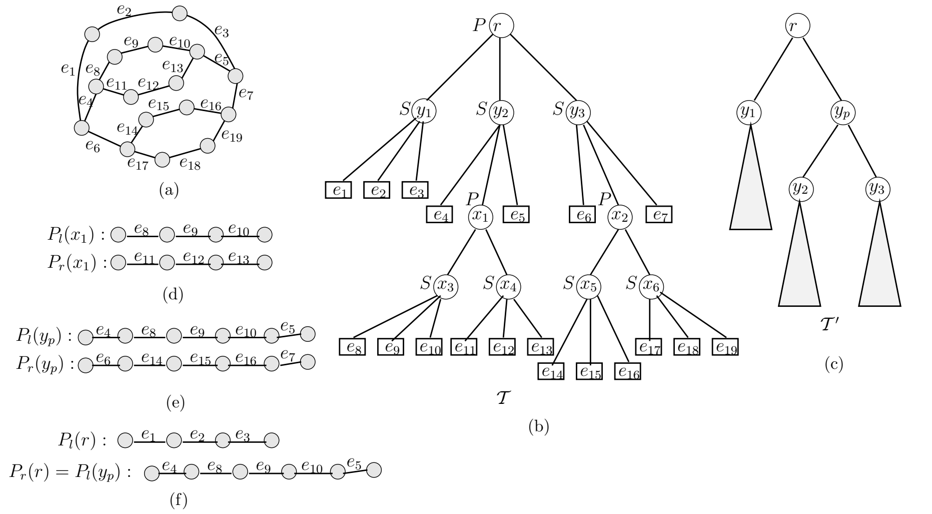

Fig. 6. (a) A biconnected series-parallel graph G, (b) an SPQ-tree $\mathcal{T}$ of G, (c) $\mathcal{T}$ with a dummy P-node y_p, (d)–(f) pole-paths of node x_1, y_p and r respectively.

One can intuitively understand the necessity of conditions (i) and (ii) for an upward planar drawing. Condition (i) must hold due to the necessity of bimodality, while condition (ii) must hold due to basic geometric consideration for upward planarity. Conditions (i) and (ii) are also sufficient for upward planarity of G as stated in [DGL06]. Hence the following lemma holds.

Lemma 2. *Let G be an acyclic planar bimodal embedded digraph such that each angle of G is assigned a label L, S or F under some assignment Φ. Then the labellings of the angles in G define an upward planar representation U_G of G if and only if Φ is upward consistent.*

Given an upward planar representation U_G of G, it is always possible to construct an upward planar straight-line drawing of G in linear-time [DGL06].

3 Feasible Labellings

In this section, we introduce the notion of "feasible set" of values for labelling the angles of G to obtain an upward planar embedding of G.

Let $\mathcal{T}$ be a given SPQ-tree of G with a P-node r as its root. Then r has three children and every other P-node of $\mathcal{T}$ has exactly two children. Let y_1, y_2 and y_3 be the three children of r. We insert a dummy P-node y_p as a child of r and make y_2 and y_3 the two children of y_p. Let $\mathcal{T}'$ be the resulting tree (see Fig. 6(b) and (c)). Every P-node of $\mathcal{T}'$ has exactly two children and the poles of r and y_p in $\mathcal{T}'$ are the same now. For any two P-nodes z_1 and z_2 in $\mathcal{T}'$, we say that z_1 is the *parent P-node* of z_2 (equivalently, z_2 is a *child P-node* of z_1) if

z_2 is a descendant of z_1 and there is no other P-node between z_1 and z_2 on the z_1, z_2-path in $\mathcal{T}'$. By this definition, y_p is a child P-node of r.

Let x be a P-node with poles u, v. Let x_l and x_r be the left and right child of x respectively. We define two edge-disjoint u, v-paths $P_l(x)$ and $P_r(x)$ as follows:

(i) if x is primitive, then each of $G(x_l)$ and $G(x_r)$ is a path. In this case, $P_l(x) = G(x_l)$, $P_r(x) = G(x_r)$ (see Fig. 6(d)).
(ii) if x is not primitive and is not the root of $\mathcal{T}$, then let y denote a child P-node of x in the left subtree of x and y' denote a child P-node of x in the right subtree of x. In this case, $P_l(x)$ will consist of all the child Q-nodes of x_l and $P_l(y)$, for each child P-node y in the left subtree. Similarly, $P_r(x)$ will consist of all the child Q-nodes of x_r and $P_l(y')$, for each child P-node y' in the right subtree (see Fig. 6(e)).
(iii) if x is the root of $\mathcal{T}$, then x_l is either an S- or a Q-node and x_r is the dummy P-node. In this case, $P_l(x)$ is defined as in case (ii) above and $P_r(x) = P_l(x_r)$. (see Fig. 6(f)).

We call each of $P_l(x)$ and $P_r(x)$ a *pole path* of x. In the remainder of this paper, we use $C(x)$ to denote the cycle $P_l(x) \cup P_r(x)$ and $F(x)$ to denote the face bounded by $C(x)$. Let x be a non-primitive P-node in $\mathcal{T}$ and y denote a child P-node of x. We call a switch of $F(x)$ a *free switch of* $F(x)$ if the switch is neither on $P_l(y)$ nor at the poles of y for any child P-node y of x.

We now introduce the notion of *feasible labeling of* $P_l(x)$ and the *feasible set* of a P-node x in $\mathcal{T}$. Let $U_{G(x)}$ be an upward planar representation of $G(x)$. Then an upward planar representation of $C(x)$ can be obtained from $U_{G(x)}$. This can be done by simply deleting all those vertices and edges of $G(x)$ which are not in $C(x)$. Let $U_{C(x)}$ denote this upward planar representation of $C(x)$ and Φ be the corresponding upward consistent assignment. Let $L(x)$ and $S(x)$ denote the number of L- and S- labels assigned by Φ to those switches of $F(x)$ which are on the path $P_l(x)$. If $L(x) = p$ and $S(x) = p + q$, then $S(x) - L(x) = q$. We say that q is a *feasible value of* $S - L$ *for labeling the switches of* $F(x)$ *on path* $P_l(x)$. Let *Feasible*(x) denote the set of all feasible values of $S - L$ for labeling the switches on path $P_l(x)$. We regard *Feasible*(x) as the *feasible set* of the P-node x. An assignment of labels to the switches on $P_l(x)$ is a *feasible labeling of* $P_l(x)$ if $S(x) - L(x) = q$ for some $q \in$ *Feasible*(x). The following fact follows the definition of feasible labeling.

Fact 3 *For a given feasible value $q \in$ Feasible(x), there is a corresponding upward planar representation of $G(x)$.*

Let x be a non-primitive P-node in $\mathcal{T}$ and y be a child P-node of x in $\mathcal{T}$. Since $G(y)$ is a subgraph of $G(x)$, given $U_{G(x)}$ we can obtain an upward planar representation $U_{G(y)}$ of $G(y)$ and hence the following fact holds.

Fact 4 *For a given feasible value $q \in$ Feasible(x), there is a corresponding feasible value $q' \in$ Feasible(y) for each child P-node y of x.*

We now have the following lemma.

Lemma 3. *$G(x)$ has an upward planar representation if and only if $P_l(x)$ can be given a feasible labeling.*

Proof. Necessity. The necessity of the condition follows from the definition of feasible labeling.

Sufficiency. Let us assume that $P_l(x)$ is given a feasible labeling according to a feasible value $q \in \mathit{Feasible}(x)$. Then it follows from Fact 3 and Fact 4 that there exists an assignment Φ of S- and L- labels to the switches of $F(x)$ such that *a*) $P_l(x)$ is labeled according to the feasible value q, *b*) for each child P-node y of x, $P_l(y)$ receives a feasible labeling, *c*) for each y, switches of $F(x)$ at the poles of y (if any) are assigned labels in such a way that $P_r(y)$ can be embedded inside the face where $P_l(y)$ is given a feasible labeling, and *d*) $S(F(x)) - L(F(x)) = 2$ from the definition of upward consistent assignment. Since Φ exists, one can find it by trying each value from the feasible set $\mathit{Feasible}(y)$ of each child P-node y. Then this same process may be applied recursively on $P_l(y)$ for each child P-node y of x and finally $U_{G(x)}$ can be computed.

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

We immediately get the following corollary from Lemma 3.

Corollary 1. *Let x be a P-node in $\mathcal{T}$. Then the pertinent digraph $G(x)$ of x has no upward planar representation if and only if the set Feasible(x) is empty.*

Let u be a pole of y such that there is a switch angle u_x of $F(x)$ at vertex u. As mentioned in condition (c) in the proof of Lemma 3, we need to label u_x in such a way that $P_r(y)$ can be embedded inside the face in which $P_l(y)$ is given a feasible labeling. At the same time, we must also ensure that the labeling of the switches at vertex u satisfies condition (i) of an upward consistent assignment. Since $\Delta(G) = 3$ and G is biconnected, the degree of every vertex in G is either two or three. Any label assigned to a switch at a vertex of degree two of G always satisfies condition (i) of upward consistent assignment. We therefore need to concentrate on labeling the switches at the vertices of degree three of G. Since G is a series-parallel digraph, only the poles of the P-nodes of $\mathcal{T}$ can have degree three. In regard to labeling the switches at the poles of the P-nodes of $\mathcal{T}$, we have the following lemma.

Lemma 4. *Let x and y be two P-nodes in $\mathcal{T}$ such that x is the parent of y. Let u be a pole of y such that there is a switch angle u_x of $F(x)$ at vertex u. Then u_x must be labeled with an L-label when $P_r(y)$ is embedded inside the face $F(x)$ and with an S-label when $P_r(y)$ is embedded in the exterior of the face $F(x)$ if the following (a) or (b) hold.*

(a) u is an ordinary vertex of G and,
(b) u is a switch vertex of G and either $P_r(y)$ is embedded inside the face $F(x)$ with a large angle at the switch of $F(y)$ at pole u or $P_r(y)$ is embedded in the exterior of the face $F(x)$ with a small angle at the switch of $F(y)$ at pole u.

We have omitted the proof of Lemma 4 in this extended abstract. In each of the two cases of Lemma 4, we define u as an *L-pole of node y*. The following lemma is a direct consequence of Lemma 4.

Lemma 5. *Let x and y be two P-nodes in $\mathcal{T}$ such that x is the parent of y in $\mathcal{T}$. If both the poles of y are L-poles, then in order to obtain an upward planar drawing, both the switches of $F(x)$ at the poles of y must be assigned the same label.*

The labels assigned to the switches (if any) of $F(y)$ at the poles of y play important roles in upward planarity testing as described in the following facts.

Fact 5 *Let a switch vertex u of G be a pole of node y, and let u_y be the switch angle of $F(y)$ at pole u. Then u is an L-pole of node y if (a) u_y is labeled with L and $F(y)$ is embedded as an inner face, and (b) u_y is labeled with S and $F(y)$ is embedded as the outer face.*

Fact 6 *Let an ordinary vertex u of G be a pole of node y. Let u contain a switch angle of $F(y)$. Then (a) $F(y)$ must be embedded as the outer face if u_y is labeled with L, and (b) $F(y)$ must be embedded as an inner face if u_y is labeled with S.*

We omit the proofs of Fact 5 and Fact 6 here since the facts are intuitive consequences of the definition of L-poles.

Let x be a P-node of $\mathcal{T}$ and y be a child P-node of x. We now define a *legitimate labeling for node y* which will be used extensively throughout the remainder of this paper. A labeling of the switches of $F(x)$ on $P_l(y)$ and at the poles of y is called a *legitimate labeling for node y* if the following (a) and (b) hold: (a) $P_l(y)$ is given a feasible labeling and (b) if a pole u of y is an L-pole then the labeling of u_x satisfies Lemma 4, otherwise both S and L labels are considered for labeling u_x. In the remainder of this paper, we use *Legitimate*(y) to denote the set of values of $S - L$ inside $F(x)$ corresponding to a legitimate labeling for node y. We define a labeling of the switches on $P_r(x)$ to be a *legitimate labeling of $P_r(x)$* if it performs a legitimate labeling for each child P-node y of x in the right subtree of x and it considers both S and L labels for the free switches on $P_r(x)$. We similarly define a *legitimate labeling of $P_l(x)$*. In the remainder of this paper, we use q_r and q_l to denote the value of $S - L$ inside $F(x)$ for a legitimate labeling of $P_r(x)$ and $P_l(x)$, respectively, and we use q_{pole} to denote the value of $S - L$ inside $F(x)$ for labeling the switches (if any) of $F(x)$ at the two poles of x. We say that $2 - (q_{pole} + q_r)$ is a *possible feasible value* of x. If we can find a legitimate labeling of $P_l(x)$ for which $q_l = 2 - (q_{pole} + q_r)$, then q_l will be a feasible value of node x and will be included in the feasible set *Feasible*(x) of x.

As stated earlier, our objective is to compute *Feasible*(x) for each P-node x in $\mathcal{T}$. Given *Feasible*(y) for each child P-node y of x, we can always compute *Feasible*(x). For this purpose, one should find each possible assignment of labels to the switches of $F(x)$ that ensures the conditions (b)–(d) mentioned in the proof of Lemma 3. Any algorithm following a brute-force approach to accomplish this would yield exponential time complexity. The approach for constructing $U_{G(x)}$ outlined in the proof of Lemma 3 would also yield exponential time complexity. In our algorithm which we describe in the next section, we show that we can compute *Feasible*(x) in linear-time by using the concepts introduced thus far. Furthermore, if $G(x)$ is upward planar, then we can also obtain $U_{G(x)}$ in linear-time.

4 An Upward Planar Drawing Algorithm

In this section we give a linear-time algorithm to test the upward planarity of a biconnected series-parallel digraph G with $\Delta(G) = 3$. If G is upward planar, then we also construct an upward planar representation of G in linear-time. An outline of our algorithm is given below.

Our algorithm consists of two phases, namely, the testing phase and the construction phase. In the testing phase, we traverse the P-nodes of $\mathcal{T}$ in a bottom-up fashion and at each P-node x, we test the upward planarity of $G(x)$. For this purpose, we compute the feasible set *Feasible*(x) of node x. If x is primitive, then computing *Feasible*(x) is quite straight forward. On the other hand, if x is non-primitive, then we can compute *Feasible*(x) from the feasible sets of the child P-nodes of x. If we succeed in this bottom-up traversal to find *Feasible*(r), where r is the root of $\mathcal{T}$, then we declare G as an upward planar digraph and start our second phase in which we construct an upward planar representation of G. On the other hand, if we find that *Feasible*$(x)= \emptyset$ for any P-node x in $\mathcal{T}$, then from Corollary 1 and Lemma 1, we declare that G is not upward planar. One can easily understand that if we consider only the combinatorial embeddings of the skeleton of each P-node of $\mathcal{T}$, then our decision regarding the upward planarity of G that we make in a *single* traversal of $\mathcal{T}$ may be incorrect. Therefore, we ensure that our method considers every planar embedding of the skeleton of each P-node of $\mathcal{T}$; nevertheless, our algorithm achieves linear-time as we will show in this section.

In the construction phase, we perform a top-down traversal of the P-nodes of $\mathcal{T}$. We start the construction phase with a feasible labeling of $P_l(r)$ where r is the root of $\mathcal{T}$. Then in a top-down traversal of $\mathcal{T}$, at each P-node x we assign labels to the switches of $F(x)$ such that the assignment satisfies conditions (a)–(d) given in the proof of Lemma 3. This procedure is carried on the basis of information gathered in the testing phase. At the end of this traversal we obtain the final upward planar representation U_G of G.

We now start with the description of our procedure to determine the feasible set of a primitive P-node. Let x be a P-node in $\mathcal{T}$. In the remainder of this paper, we use the symbols n_r, n_l and n_x to denote the number of switches of $F(x)$ on the path $P_r(x)$, on the path $P_l(x)$ and at the two poles of x, respectively. We also adopt the notation $[low \, .. \, high]$ to denote the set of integers in which the numbers are listed in ascending order and the first number is low, the last number is $high$ and if not mentioned explicitly, the periodicity of the numbers is 2. Let x be a primitive P-node and q be a possible feasible value of x. Then q will be a feasible value of x and included in *Feasible*(x) if $|q| \leq n_l$. Let $I_0 = [-n_r + 2 \, .. \, n_r + 2]$, $I_- = [-n_r + (2 - n_x) \, .. \, n_r + (2 - n_x)]$ and $I_+ = [-n_r + (2 + n_x) \, .. \, n_r + (2 + n_x)]$. Then we have the following lemmas regarding the possible feasible values of a primitive P-node x in $\mathcal{T}$ whose proofs are omitted in this extended abstract.

Lemma 6. *Let x be a primitive P-node of $\mathcal{T}$. Let I_{inner} and I_{outer} be the set of possible feasible values of node x for embedding $F(x)$ as an inner face and the outer face, respectively. Then the following (a) and (b) hold.*

(a) $I_{inner} = I_{outer} = I_0 = [-n_r + 2 \,..\, n_r + 2]$*, if* $n_x = 0$*; and*
(b) $I_{inner} = I_- = [-n_r + (2 - n_x) \,..\, n_r + (2 - n_x)]$ *and* $I_{outer} = I_+ = [-n_r + (2 + n_x) \,..\, n_r + (2 + n_x)]$*, if* $n_x > 0$ *and each of these* n_x *poles of* x *is an ordinary vertex of* G*.*

Lemma 7. *Let* x *be a primitive P-node of* $\mathcal{T}$*. Let* I_{inner} *and* I_{outer} *be the set of possible feasible values of node* x *for embedding* $F(x)$ *as an inner face and the outer face, respectively. If* $n_x = 2$ *and one of the two poles of* x *is a switch vertex of* G *and the other is an ordinary vertex of* G*, then the following (a) and (b) hold.*

(a) $I_{inner} = I_- \cup I_0 = [-n_r \,..\, n_r + 2]$ *and* $I_{outer} = I_+ \cup I_0 = [-n_r + 2 \,..\, n_r + 4]$*; and*
(b) For every $q \in I_0$*, the pole of* x *which is also a switch vertex of* G *is an L-pole.*

Lemma 8. *Let* x *be a primitive P-node of* $\mathcal{T}$*. Let* I_{inner} *and* I_{outer} *be the set of possible feasible values of node* x *for embedding* $F(x)$ *as an inner face and the outer face, respectively. If* $n_x > 0$ *and each of these* n_x *poles of* x *is a switch vertex of* G*, then the following (a)–(d) hold.*

(a) For $n_x = 1$*,* $I_{inner} = I_{outer} = I_- \cup I_+$ *and for* $n_x = 2$*,* $I_{inner} = I_{outer} = I_- \cup I_+ \cup I_0$*.*
(b) For every $q \in I_-$*, each of the* n_x *poles of* x *is an L-pole if* $F(x)$ *is embedded as the outer face.*
(c) For every $q \in I_+$*, each of the* n_x *poles of* x *is an L-pole if* $F(x)$ *is embedded as an inner face.*
(d) For every $q \in I_0$*, one of the poles of* x *is an L-pole if* $n_x = 2$*.*

Having computed the feasible set for a primitive P-node, we now proceed towards the computation of feasible set of any P-node in $\mathcal{T}$. To compute a feasible value for a non-primitive P-node x in $\mathcal{T}$, we must ensure that for every child P-node y of x, we are performing a legitimate labeling. Therefore, we first demonstrate how we can find the set *Legitimate*(y) for child P-node y of x. Let $q \in$ *Feasible*(y). By giving a feasible labeling to $P_l(y)$ satisfying $q \in$ *Feasible*(y), we can obtain three possible embeddings of $F(y)$ as shown in Fig. 7(b)–(d). In Fig. 7(b) and (c) $F(y)$ is embedded as an inner face. Hence, in these two cases, exactly one of $P_r(y)$ and $P_l(y)$, but not both, appears at the outer face. On the other hand, in Fig. 7(d), $F(y)$ is embedded as the outer face, and hence both $P_l(y)$ and $P_r(y)$ have been drawn drawn at the outer face. From a more theoretical point of view, the three figures in Fig. 7(b)–(d) actually correspond to three possible planar embeddings of the skeleton of node y. We can discard the other three possible planar embeddings of the skeleton of node y because they are just the mirror reflections of the three embeddings shown. In order to obtain an embedding as shown in Fig. 7(b) we have to label the switches on $P_l(y)$ in such a way that the labels of these switches yield $S - L = -q$ inside face $F(x)$. Similarly, in order to obtain the embeddings shown in Fig. 7(c) and (d)

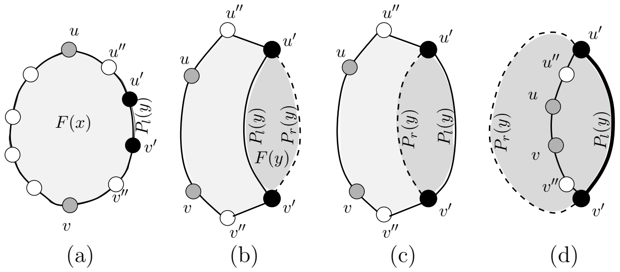

Fig. 7. Three possibilities to consider from P-node x to embed the facial cycle $F(y)$.

the labels of the switches on $P_l(y)$ should yield $S - L = q$ inside face $F(x)$. We can determine the legitimate values of $S - L$ inside $F(x)$ for the three scenarios shown in Fig. 7 by considering the following three possible cases: (i) *Both the poles of y are L-poles:* for the three embeddings in Fig. 7(b)–(d), we should have $2 - q$, $-2 + q$, $2 + q$ respectively as the value of $S - L$ inside $F(x)$. (ii) *Exactly one of the poles of y is an L-pole:* we should have $1-q$, $-1+q$, $1+q$ respectively for the embeddings in Fig. 7(b)–(d). (iii) *None of the poles of y is an L-pole:* we should have $-q$, $+q$, $+q$ respectively for the embeddings in Fig. 7(b)–(d). For the cases (ii) and (iii), if a pole of y contains a free switch of $F(x)$, then the value of $S - L$ inside $F(x)$ for labeling that switch would be ± 1, since it can be assigned either of the two possible labels.

We now show how we can compute *Legitimate*(y) when y is a primitive P-node. In the following we first consider only the legitimate values resulting from embedding $F(y)$ as an inner face. Let q_m denote the maximum of these legitimate values. In Lemma 10 we show that, if $F(y)$ is embedded as the outer face, then at most two new legitimate values can be obtained, namely, q_m+2 and $q_m + 4$. We have seen in Lemma 6, 7 and 8 that the set of feasible values which can be satisfied for embedding $F(y)$ as an inner face is of the form: $[lo .. hi]$. We now have the following lemma regarding the legitimate values for a primitive P-node y when $F(y)$ is embedded as an inner face.

Lemma 9. *Let y be a primitive P-node in $\mathcal{T}$ and x be the parent P-node of y in $\mathcal{T}$. Let n_{Lpole} denote the number of L-poles of node y. Let Feasible(y)= $[lo .. hi]$ and k denote the number of switches of $F(x)$ at those poles of y which are not L-poles. Then for embedding $F(y)$ as an inner face the following (a) and (b) hold.*

(a) If $lo = hi = 0$ and $n_{Lpole} = 2$ then Legitimate(y)= $\{-2, +2\}$; and
(b) Otherwise, Legitimate(y)= $[-(max + k) .. (max + k)]$, where max is the maximum of $|n_{Lpole} - lo|$ and $|n_{Lpole} - hi|$.

We now have the following lemma regarding the legitimate values of a primitive P-node x when $F(x)$ is embedded as the outer face.

Lemma 10. *Let x be a primitive P-node in $\mathcal{T}$. Let q_m be the maximum of all the legitimate values obtained by embedding $F(x)$ as an inner face. Then at most two new legitimate values, namely, $q_m + 2$ and $q_m + 4$ can be obtained by embedding $F(x)$ as the outer face.*

In the following lemma, we address the issue of computing *Feasible*(x) for a non-primitive P-node x in $\mathcal{T}$.

Lemma 11. *Let x be a non-primitive P-node in $\mathcal{T}$. Let $y_1, \ldots, y_l$ be the child P-nodes of x in $\mathcal{T}$. Then Feasible(x) can be computed from Feasible$(y_1), \ldots,$ Feasible(y_l).*

Proof. Let $y_1, y_2, \ldots, y_{left}$ be the child P-nodes in the left subtree of x and $y'_1, y'_2, \ldots, y'_{right}$ be the child P-nodes in the right subtree of x. Let h denote the height of the P-node x in $\mathcal{T}$. We prove the claim by induction on h.

We first assume that $h = 1$. Then every child P-node of x is primitive. Let y be a child P-node of x. According to Lemma 6, 7 and 8, y has two types of feasible values, namely, the feasible values satisfying which $F(y)$ can be embedded as an inner face and the feasible values satisfying which $F(y)$ can be embedded as the outer face. Among all the child P-nodes y of x, for at most one y, we can embed $F(y)$ as the outer face since planarity would be violated otherwise. Hence, for each y, we first consider those values from *Feasible*(y) satisfying which we can embed $F(y)$ as an inner face, later we handle the feasible values satisfying which we can embed $F(y)$ as the outer face.

As we have shown in Lemma 9, if y_i is a child P-node of x in the left subtree of x, then the set of legitimate values for embedding $F(y)$ as an inner face is $[-p_i \;..\; p_i]$ with a periodicity of either 2 or 4 for some integer $p_i\,(1 \leq i \leq left)$. Similarly, if y'_i is a child P-node of x in the right subtree of x, then the set of legitimate values for embedding $F(y)$ as an inner face is $[-p'_i \;..\; p'_i]$ with a periodicity of either 2 or 4 for some integer $p'_i\,(1 \leq i \leq right)$. Let p' denote the number of free switches of $F(x)$ on path $P_r(x)$. Since a labeling of each of these free switches can yield $S - L = \pm 1$ inside $F(x)$, we would have $[-p' \;..\; p']$ as the set of values for labeling these switches. Therefore, the set of legitimate values for a labeling of $P_r(x)$ will be of the form $[-(p' + \sum p'_i) \;..\; (p' + \sum p'_i)]$ with a periodicity of either 2 or 4. Taking $k = p' + \sum p'_i$, we can obtain $[-k \;..\; k]$ as the set of legitimate values for a labeling of $P_r(x)$. From this set of values, we can obtain the set of possible feasible values of x (i.e., we can obtain I_{inner} and I_{outer}) exactly in the same way as we have described in Lemma 6, 7 and 8. It now remains to determine which of these possible feasible values will be the feasible values of x. For this purpose, we first determine the legitimate a labeling of $P_l(x)$ exactly in the same way as we determined the legitimate values for a labeling of $P_r(x)$. Every possible feasible value of x which is also a legitimate value for a labeling of $P_l(x)$, will be a feasible value of x and hence, will be included in *Feasible*(x). It is mentionable that, in this computation, we do not need to check every value from the former set with every value of the latter set. Rather, we can obtain the whole information in time $O(1)$ from the periodicity and the first and last values of these two sets. Having computed *Feasible*(x), we can compute *Legitimate*(x) for embedding $F(x)$ as an inner face as illustrated in Lemma 9 for a primitive P-node and we can also compute the possible changes in these legitimate values if $F(x)$ is embedded as the outer face as illustrated in Lemma 10 for a primitive P-node.

We now consider those feasible values of each child P-node y of x satisfying which $F(y)$ can be embedded as the outer face. We said previously that any such embedding can increase the legitimate values for y by at most $+2$ and $+4$. Hence, regardless of the choice of y, any such embedding can cause a change of $i \in \{\pm 2, \pm 4\}$ in the legitimate values for x. Let *External*(x) be the set of possible changes in the legitimate values for x if $F(x)$ or $F(z)$ is embedded as the outer face where z is a descendant P-node of x. Then *External*$(x) \subseteq \{i + j : i \in \{\pm 2, \pm 4\}$ *and* $j \in \{2, 4\}\} = [-2 .. +8]$. Along with *Legitimate*(x), we pass this set of possible changes to the parent P-node of x in $\mathcal{T}$.

We next assume that $h > 1$ and that the children of x have handed the following two quantities to x. (i) *Legitimate*(y) for every child P-node y of x and (ii) the possible changes in the legitimate values for node y that can be obtained by embedding either $F(y)$ as the outer face, or $F(z)$ as the outer face where z is a descendant P-node of y. In a manner exactly similar to the case for $h = 1$, we can compute the feasible values of x first by considering only those feasible values of each child P-node y of x which can be satisfied while embedding $F(y)$ as an inner face. From this we determine the set *Legitimate*(x). Next we determine the possible changes in the legitimate values for node x that are obtainable by embedding either $F(x)$ as the outer face, or $F(z)$ as the outer face where z is a descendant P-node of x. By doing so, we have again equipped ourselves with *Feasible*(x) and also with the above two quantities (a) and (b) which we would pass to the parent of x if x is not the root of $\mathcal{T}$, or if x is the root of $\mathcal{T}$, then we can start our second phase of constructing U_G by using a feasible value from *Feasible*(x). $\mathcal{Q.E.D.}$

We call the algorithm outlined in the proofs of Lemma 6, Lemma 7, Lemma 8 and Lemma 11 Algorithm **UP-Tester**. We now have the following theorem.

Theorem 1. *Let G be a series-parallel digraph with $\Delta(G) \leq 3$. Then the upward planarity of G can be tested in linear-time.*

Proof. Let $\mathcal{T}$ be the SPQ-tree of G. From Lemma 6, Lemma 7, Lemma 8 and Lemma 11, we can find *Feasible*(x) for any P-node x in $\mathcal{T}$ by applying Algorithm **UP-Tester**. If **UP-Tester** finds *Feasible*$(x) = \emptyset$, then from Corollary 1, $G(x)$ is not upward planar and hence, from Lemma 1 G itself is not upward planar. We can show that the operations required by Algorithm **UP-Tester** to compute the feasible sets of all the P-nodes of $\mathcal{T}$ can be performed in time linear to the number of P-nodes in $\mathcal{T}$. The details of these computations are omitted in this extended abstract. Since the number of P-nodes of $\mathcal{T}$ is linear in the number of vertices of G, the upward planarity of G can be tested in time $O(n)$. $\mathcal{Q.E.D.}$

Finally, we give the following theorem regarding the construction of an upward planar representation of G, U_G.

Theorem 2. *Let r be the root of the SPQ-tree $\mathcal{T}$ of G. If G is upward planar, then starting with a feasible labeling of $P_l(r)$, an upward planar representation of G can be constructed in linear-time.*

Proof. Our proof is constructive. We show here how we can perform a feasible labeling of $P_l(y)$ for each child P-node y of x given that $P_l(x)$ has been labeled with a feasible value. Let q be the feasible value satisfied for labeling $P_l(x)$. If q requires that for some descendant P-node z of x, $F(z)$ should be embedded as the outer face, then at first we determine that P-node. Depending on the satisfied feasible value q, we know what labels should be assigned to the switches (if any) at the poles of x. Hence we label the switches (if any) at the poles of x. Let $S - L = q_p$ inside $F(x)$ for the labels assigned to the switches (if any) at the poles of x. As shown in the proof of Lemma 11, we can easily compute the sets *Legitimate*(y) that altogether yielded the set *Feasible*(x). Let the set of all possible values for labeling the switches on $P_l(x)$ be $[l \, .. \, h]$. Then let $i = h - q$. We iterate through each of the computed legitimate sets. Let *Legitimate*$(y) = [l_y \, .. \, h_y]$. If $i > (h_y - l_y)$ then we satisfy the feasible value corresponding to l_y for node y and reduce i by $h_y - l_y$. Then we proceed with the next node. When we find $i \leq (h_y - l_y)$ at a node y, we satisfy the feasible value corresponding to $(h_y - i)$ for that P-node. For each of the remaining P-nodes in the left subtree of x, we satisfy the feasible value corresponding to h_y. We then perform the same operations in order to satisfy the value $2 - q - q_p$ for the switches on $P_r(x)$. Clearly, the whole tree can be traversed in linear-time while performing these operations at each x while the completion of the traversal indicates that we have obtained an upward planar embedding of G in which the switches have been labeled according to an upward consistent assignment. This completes the proof of the claim. $\mathcal{Q.E.D.}$

5 Conclusion

In this paper, we gave a simple linear-time algorithm to test upward planarity and in the positive case, obtain an upward planar drawing of a series-parallel digraph with the maximum degree three. Since our attention was confined to series-parallel digraphs with the maximum degree three, it looks difficult to extend this algorithm in a straight forward way for more general classes of digraphs and also for series-parallel digraphs with higher degrees. It is left as a future work to find other characterizations of upward planarity of series-parallel digraphs and devise efficient algorithms for upward planarity testing and upward planar drawings of series-parallel digraphs with higher degrees.

References

[BCDTT94] P. Bertolazzi, R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis, *How to draw a series-parallel digraph*, International Journal of Computational Geometry and Applications, 4 (4), pp. 385–402, 1994.

[BDLM94] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino, *Upward drawings of triconnected digraphs*, Algorithmica, 6 (12), pp. 476–497, 1994.

[BDMT98] P. Bertolazzi, G. Di Battista, C. Mannino and R. Tamassia *Optimal upward planarity testing of single-source digraphs*, SIAM Journal on Computing, 27, pp. 132–169, 1998.

[BL76] K. S. Booth and G. S. Lueker, *Testing the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, J. Comp. Sys. Sci., 13, pp. 335–379, 1976.

[CDTT94] R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis, *Dynamic graph drawing:trees, series-parallel digraphs, and planar st-digraphs*, SIAM Journal on Computing, 24 (5), pp. 970–1001, 1995.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, McGraw-Hill Book Company, Second Edition, 2001.

[DGL06] W. Didimo, F. Giordano, G. Liotta, *Upward spirality and upward planarity testing*, Proc. of GD'05, Lecture Notes in Computer Science, 3843, pp. 117–128. Springer-Verlag, 2006.

[DT96] G. Di Battista and R. Tamassia, *On-line planarity testing*, SIAM J. Comput., 25(5), pp. 956–997, 1996.

[GL99] A. Garg and G. Liotta, *Almost bend-optimal planar orthogonal drawings of biconnected degree-3 planar graphs in quadratic time*, Proc. of GD'99, Lect. Notes in Computer Science, 1731, pp.38–48, 1999.

[GT94] A. Garg and R. Tamassia, *On the computational complexity of upward and rectilinear planarity testing* Proc. Graph Drawing 1994, Lecture Notes in Computer Science, 894, pp. 286–297. Springer-Verlag, 1995.

[GT95] A. Garg and R. Tamassia, *Upward planarity testing*, Order, 12, pp. 109–133, 1995.

[GT01] A. Garg and R. Tamassia, *On the computational complexity of upward and rectilinear planarity testing*, SIAM Journal on Computing, 31(2), pp. 601–625, 2001.

[HL96] M. D. Hutton and A. Lubiw, *Upward planarity testing of single-source acyclic digraphs*, SIAM Journal on Computing, 25(2), pp. 291–311, 1996.

[HL05] P. Healy and K. Lynch, *Fixed-parameter tractable algorithms for testing upward planarity*, Proceedings of SOFSEM'05, Lecture Notes in Computer Science, 3381, pp. 199–208, 2005.

[HT74] J. Hopcroft and R. E. Tarjan, *Efficient planarity testing*, Journal of the ACM, 21(4), pp. 549–568, 1974.

[Pap95] A. Papakostas, *Upward planarity testing of outerplanar dags*, Proc. of GD'95, Lecture Notes in Computer Science, 894, pp. 298–306, 1995.

[REN05] M. S. Rahman, N. Egi, and T. Nishizeki, *No-bend orthogonal drawings of series-parallel graphs*, Proc. of GD'05, Lecture Notes in Computer Science, 3843, pp. 409–420. Springer-Verlag, 2006.

[VTL82] J. Valdes, R. E. Tarjan and E. L. Lawler, *The Recognition of Series-Parallel Digraphs*, SIAM J. Comput., 11 (2), pp. 298–313, 1982.