

# Upward Numbering Testing for Triconnected Graphs

M. Chandramouli, A. A. Diwan  
Dept. of Computer Science and Engineering,  
Indian Institute of Technology, Powai,  
Bombay 400 076,  
INDIA

Email: {mouli, aad}@cse.iitb.ernet.in

**Abstract.** In this paper, we look at the problem of upward planar drawings of planar graphs whose vertices have preassigned y-coordinates. We give a linear time algorithm for testing whether such an embedding is feasible for triconnected labelled graphs.

## 1 Introduction

For directed graphs, a notion similar to planarity (for undirected graphs) is that of an *upward planar* drawing, that is a planar drawing of the graph such that all the edges are directed upward (monotonic curves from a lower y-coordinate to a higher one). There has been a lot of work on upward planarity testing of various classes of graphs [8, 3]. Recently, Garg and Tamassia [6] have proved that the problem of upward planarity testing for arbitrary graphs is NP-complete.

In this paper, we consider the problem of *upward numbering testing* for planar graphs (see Section 3 for the definition). The main contribution of this paper is a linear time algorithm for *upward numbering* testing for triconnected labelled graphs. Battista and Nardelli [1] have given a linear time algorithm for recognising upward numberings for single source labelled digraphs. Lin [11], gave an algorithm for a subclass of digraphs having their sources and sinks on the outface, called proper s-t boundary hierarchical graphs. Recently Heath and Pemmaraju [7] have given a linear time algorithm for graphs with adjacent vertices having labels which differ by unity. This algorithm leads to a quadratic time algorithm for upward numbering testing for arbitrary graphs.

The rest of the paper is organised as follows. Section 2 looks at earlier work on upward planarity testing. Section 3 contains the algorithm for upward numbering testing. Section 4 gives an interesting connection between the upward drawings of a graph with distinct labels and an intersection graph realization problem.

## 2 Upward planar drawings of digraphs

In this section, we look at the prior work on upward planarity testing for digraphs. DiBattista and Tamassia [2] have shown that the problem of upward planarity testing is equivalent to the problem of augmenting a given digraph to obtain a planar s-t digraph. Kelly [9], and DiBattista and Tamassia [2], have proved the following theorem.

**Theorem 1.** *A digraph admits an upward planar drawing iff it is a subgraph of a planar s-t digraph.* □

DiBattista and Tamassia [2] have given a linear time algorithm for producing an upward polyline grid drawing of a s-t digraph.

### 3 Upward drawings of planar graphs with labels

We now consider the problem of upward drawings of graphs with the additional constraint that the vertices have labels attached to them, which denote the y-coordinate (level) at which they are to be placed.

#### 3.1 Preliminaries

We consider only planar graphs and henceforth refer to them only as graphs. A map  $f : V \rightarrow \mathbb{Z}^+$  (set of positive integers) is called a labelling of the graph.

**Definition 1** *A generalized s-t numbering of a graph  $G = (V, E)$  is a map  $f : V \rightarrow \{1, 2, \dots, N\}$ , such that  $|\{v : f(v) = 1\}|$  and  $|\{v : f(v) = N\}|$  are equal to one, and such that each vertex  $v \neq s, t$ , where  $f(s) = 1$  and  $f(t) = N$ , has two adjacent vertices  $u, w$  for which  $f(u) < f(v) < f(w)$  and the vertices  $s$  and  $t$  are adjacent.*

We call the value  $f(v)$  attached to vertex  $v$  the *label* associated with  $v$ . We denote a labelled graph as a 3-tuple  $G = (V, E, l)$  where  $l$  is the labelling function.

**Definition 2** *A labelling of the vertices of a planar graph such that no two adjacent vertices have equal labels is said to be an upward numbering if there exists a planar embedding of the graph such that all the vertices labelled  $i$  have y-coordinate  $i$  and all the edges are strictly monotonic curves.*

Without loss of generality, we can assume that the labels are positive integers between 1 and  $N$ . Let 1 and  $N$  be the smallest and largest labels in a labelling of a graph. We can make the following assumption about any labelling of a graph.

- We can assume that there is a unique vertex labelled 1 and also a unique vertex labelled  $N$ . We can also assume that these two vertices are adjacent.
- We can assume that in any face that there are no three consecutive vertices with increasing labels. Note that as we are considering triconnected graphs, the faces are uniquely defined.

For all labelled graphs, we denote the unique vertex with the smallest label by  $s$  and the unique vertex with the largest label by  $t$ . Following the terminology of digraphs, we call a vertex a *source* (*sink*) if it has no neighbouring vertices with labels *smaller* (*larger*) than it. In [2] it has been shown that every planar *st-digraph* admits an *upward drawing*. We prove a similar result for labelled digraphs. The following theorem follows easily from the algorithms given in [12] to construct polyline representations (planar embeddings such that the edges are polygonal segments) of a planar graph from its s-t numbering.

**Theorem 2.** *A labelling of a planar graph  $G$  is an upward numbering if and only if the graph can be augmented to a planar graph through the addition of edges such that the given labelling is a generalised  $s$ - $t$  numbering of the augmented graph, with no two adjacent vertices having the same label.*  $\square$

The following lemma establishes a property of graphs with a generalised  $s$ - $t$  numbering.

**Lemma 3.** *Given a graph  $G = (V, E, l)$ . Let  $u$  and  $v$  be two vertices of  $G$ . Then  $l$  is a generalised  $s$ - $t$  numbering only if there exists a path between  $u$  and  $v$  such that all the labels are less than  $\max(l(u), l(v))$  and there exists a path such that all the labels are greater than  $\min(l(u), l(v))$ .*  $\square$

### 3.2 Resolving sources and sinks

Our strategy for testing for *upward numbering* is the following. We try to add edges to a given embedding of a graph to *resolve* all sources and sinks other than  $s$  and  $t$ , that is to add outgoing edges to sinks and incoming edges to sources to get a planar graph, with a generalised  $s$ - $t$  numbering. We say that such an embedding is a *feasible embedding*.

As we are considering triconnected graphs we have to examine only a polynomial number of embeddings as the embedding is decided by the choice of outerface. Moreover we want the edge  $(s, t)$  to lie on the outerface. As an edge can be shared by exactly two faces we have to consider only two embeddings. If the graph were to have an upward planar drawing then there are at least two embeddings which are feasible namely the embedding having the  $s$ - $t$  edge as the leftmost edge and the one with it as the rightmost edge. Hence we have to test exactly one of the embeddings with the edge from the vertex labelled  $s$  to the vertex labelled  $t$  on the outermost face.

### 3.3 Reducing the face size

When we are testing a given embedding of a triconnected graph, we can perform certain transformations to make the faces smaller in size. The following lemmas describe ways of adding edges to faces in a given embedding of a labelled triconnected graph to decrease the face size.

Let  $G = (V, E)$  be a graph and let  $e$  be an edge which does not belong to  $E$ . Then by  $G + e$  we denote the graph  $G$  along with the additional edge  $e$ . We represent the fact that  $l$  is an upward numbering of  $G$  iff it is an upward numbering of  $H$  by  $(G, l) \equiv (H, l)$ .

We omit the proof of the following lemmas, which show that testing for upward numbering of a given labelled graph is equivalent to testing on the same graph with certain additional edges. The first lemma shows that if in any face, there exist vertices  $u$  and  $v$  such that all the vertices between them on one of the face paths joining them have labels lying strictly between those of  $u$  and  $v$ , then

the edge  $(u, v)$  can be added to the face. The second lemma shows that an edge, joining a vertex with the largest label in a face and a vertex with the smallest label in a face, can be added to the face.

**Lemma 4.** *Given a labelled graph  $G = (V, E, l)$ . Let  $f$  be a face such that there are consecutive vertices  $x_0, x_1, x_2, \dots, x_k$  such that  $l(x_0) = i$  and  $l(x_k) = j$  and  $i < l(x_r) < j$ ,  $1 \leq r < k$ . Let  $H = (V, E \cup \{(x_0, x_k)\})$ . Then  $(G, l) \equiv (H, l)$ .*  $\square$

**Lemma 5.** *Given a labelled graph  $G = (V, E, l)$ . Let  $f$  be a face and let  $x_i$  and  $x_j$  be vertices with  $l(x_i) = 1$  and  $l(x_j) = N$ , where 1 and  $N$  are the smallest and largest labels on the face. Let  $H$  be the graph obtained by adding the edge  $(x_i, x_j)$  to the face  $f$ . Then  $(G, l) \equiv (H, l)$ .*  $\square$

From the previous lemmas, we note the following facts.

1. If there are two vertices labelled  $n$  on the face, they can both be joined to a vertex labelled 1.
2. If there are two vertices labelled 1 on the face, they can both be joined to a vertex labelled  $n$ .

We can add edges to the faces using Lemma 4 and 5 till no more edges can be added. We now look at the labels on the faces which are remaining. We can show by simple arguments that the face labels must be one of the four cases depicted in Figure 1.

**Lemma 6.** *In each of the four basic face configurations shown in Figure 1, the only possible resolution of sources and sinks is one in which, either all the sources or all the sinks are resolved in any face.*

**Proof: Case 1:** There is a unique vertex with the smallest label. In this case all the sources must be resolved in the face. Otherwise let  $v$  be a vertex (source) not resolved in the face. Clearly  $x_1, x_2, x_3$  are not resolved in the face. Hence there exist paths from  $x_1$  to  $x_3$  and from  $x_2$  to  $v$ , with labellings as in Lemma 3. But these two paths cannot exist at the same time without violating planarity. Hence it follows that all the sources in the face must be resolved in the face and all of them are resolved by connecting them to the source with the smallest label.

**Case 2:** There is a unique vertex with the largest label. As in Case 1, all the sinks must be resolved in the face by connecting them to the sink with the largest label.

**Case 3:** Consider the arc  $S$  between  $x_1$  and  $x_4$ . If there are no vertices labelled 2 or  $n - 1$  on this arc then it follows that we can add the edge  $(x_1, x_4)$  to the face  $f$ . But we have assumed that all such edges have already been added to the face. Hence there must be a vertex labelled 2 or  $n - 1$  in  $S$ . Also, one of the vertices  $x_1$  or  $x_4$  must be resolved in the face, otherwise by Lemma 3 we have

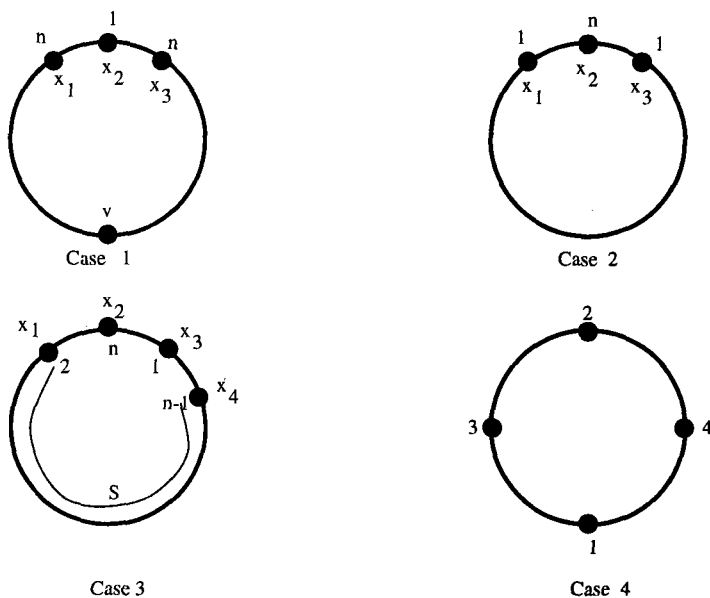


Fig. 1. Basic face configurations

two vertex disjoint paths between  $x_1$  and  $x_3$ , and  $x_2$  and  $x_4$  which lie totally outside the face. These paths must cross, violating the planarity of the resolved graph.

If one of the vertices labelled 2 is resolved in the face it follows that all the vertices labelled 2 are resolved in the face. Let  $v$  be a vertex labelled 2, which is not resolved in the face. Otherwise, by Lemma 3, there are paths between  $x_2$  and  $x_4$ , and  $x_3$  and  $v$  which do not share a vertex and which lie completely outside the face in any resolution of the face. Hence these paths must cross, implying that there is no planar resolution. Similarly, we can show that if one of the vertices labelled  $n-1$  is resolved in the face, then all the vertices labelled  $n-1$  must be resolved in the face.

Subcase A: There is a vertex labelled 2 in  $S$ .

$x_1$  is resolved in the face: Now assuming that all the vertices labelled 2 are resolved in the face by connecting them to  $x_3$  as shown in Figure 2(a). In face  $f_1$ , there is no vertex labelled 2 in the arc  $T$ . Hence the vertices  $x_5$  and  $x_6$  must have labels which are the largest in the face. If some other vertex has a larger label in arc  $T$ , then we can add an edge connecting this vertex and the vertex labelled 2. But we have assumed that all such edges have already been added. We can add the edges  $(x_3, x_5)$  and  $(x_3, x_6)$  to the face. Hence from Case 1 we get that all the sources must be resolved in the face. The same argument can be applied to each of the faces. For the last face, that is the face containing  $x_4$ , either  $x_7$  is adjacent to  $x_4$  or there is a vertex labelled 2 or  $n-1$  on the path between them. If there is a vertex labelled  $n-1$ , then the edge between  $x_7$  and

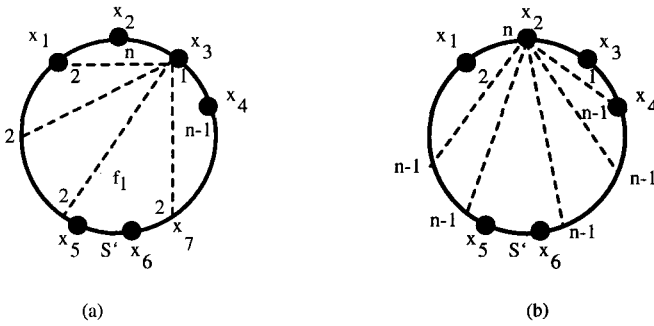


Fig. 2. Resolving all sources or all sinks

the vertex can be added. As we have assumed that all such edges have already been added, there must be a vertex labelled 2 adjacent to  $x_4$ . This face is now exactly like the other faces. Hence all the sources must be resolved in  $f$ .

Similarly if a single vertex labelled  $n - 1$  is resolved in the face, we can show that all the sinks are resolved in the face as shown in Figure 2(b).

**Subcase B:** In this case there is a vertex labelled  $n - 1$  in arc  $S$ . The arguments are exactly similar to the above case and it follows that either all the sources or all the sinks are resolved in the face.

**Case 4:** This case is trivial. □

### 3.4 Algorithm for upward numbering testing

The algorithm for upward numbering testing is shown in Figure 3. We now prove the correctness of the algorithm. We show that at each step of the algorithm there is either a forced vertex, that is a vertex which can be resolved in exactly one face or a vertex which cannot be resolved in any of the faces. If neither of these two conditions hold, then for each unresolved source or sink there are exactly two faces in which they can be resolved. Both these faces are of size four. In this case, we show that the choice of face for resolution is not critical.

We omit the proof of the following lemma.

**Lemma 7.** *Let  $G = (V, E, l)$  be a labelled triconnected graph. Let  $f$  be a face such that two of the sources on the face are connected to each other by a path lying outside the face, such that the vertices have strictly decreasing labels. Then there exists a sink, which has an unique choice of face for resolution or there exists a sink, which cannot be resolved in any face.* □

We use the above property, while proving the next lemma.

**Lemma 8.** *At every step of the algorithm, there is a forced vertex, or an unresolvable vertex or a vertex for which the choice of face for resolution is not critical.*

**Algorithm Upward-Numbering-Testing**

1. Find an embedding of the graph such that  $s$  and  $t$  lie on the outface.
2. Add edges to the faces using lemmas 4 and 5 till no more edges can be added to any of the faces.
3. while there exists an unresolved source or sink (other than  $s$  or  $t$ ), do
4.     if there is a vertex which cannot be resolved in any of the faces  
        stop. { The current numbering is not an upward numbering }.
5.     endif
6.     if there exists a forced vertex  
        resolve the vertex in the unique face.  
        resolve all the sources or sinks in the same face.
7.     endif
8.     if there is no forced vertex  
        pick any arbitrary vertex.  
        resolve the vertex in any of the faces in which it can be resolved.  
        resolve all the sources or sinks in the same face.
9.     endif
10. endwhile

Fig. 3. Upward numbering testing

**Proof:** We assume that there are no vertices which are forced or have no choice of face for resolution. We now show that there must exist a vertex for which the choice of face for resolution is not critical. Let  $s$  be a vertex such that all vertices, with labels smaller than  $l(s)$  have been resolved. As there are no forced vertices or unresolvable vertices, it follows that  $s$  can be resolved in at least two faces.

**Case 1:**  $s$  can be resolved in faces  $f_1$  and  $f_2$ , which do not share an edge, as shown in the top of Figure 4. Let  $s_1$  and  $s_2$  be the sources with the smallest labels in  $f_1$  and  $f_2$  respectively. By our assumption, it follows that  $s_1$  and  $s_2$  have been resolved. Hence there exists a path  $P$  from  $s_1$  and  $s_2$ , which have labels less than or equal to  $\max(s_1, s_2)$ . By Lemma 7, we can always assume that the paths do not use any vertex in  $f_1$  or  $f_2$ . Consider the largest sink  $v$  in the region  $A$  marked in Figure 4. If it lies in the interior of the region, then clearly it cannot be resolved in any face which is a contradiction. If it lies on the boundary of  $A$  it must lie either on  $P_1$  or on  $P_2$ . In either case  $v$  is a forced vertex, which contradicts the fact that there are no forced vertices.

**Case 2:**  $s$  can be resolved in two faces  $f_1$  and  $f_2$ , which share an edge, as shown in the lower half of Figure 4. Let  $s_1$  and  $s_2$  be the smallest sources in  $f_1$  and  $f_2$  respectively. There are two subcases in this case and they are shown in the lower half of Figure 4.

**Subcase A:** We again consider the largest sink in region  $A$  and argue as in the previous case.

**Subcase B:** We consider the largest sink in the region  $A$ . If it lies on  $P_1$  or  $P_2$  (other than  $v$ ), we have a forced vertex. We are left now with the case when the

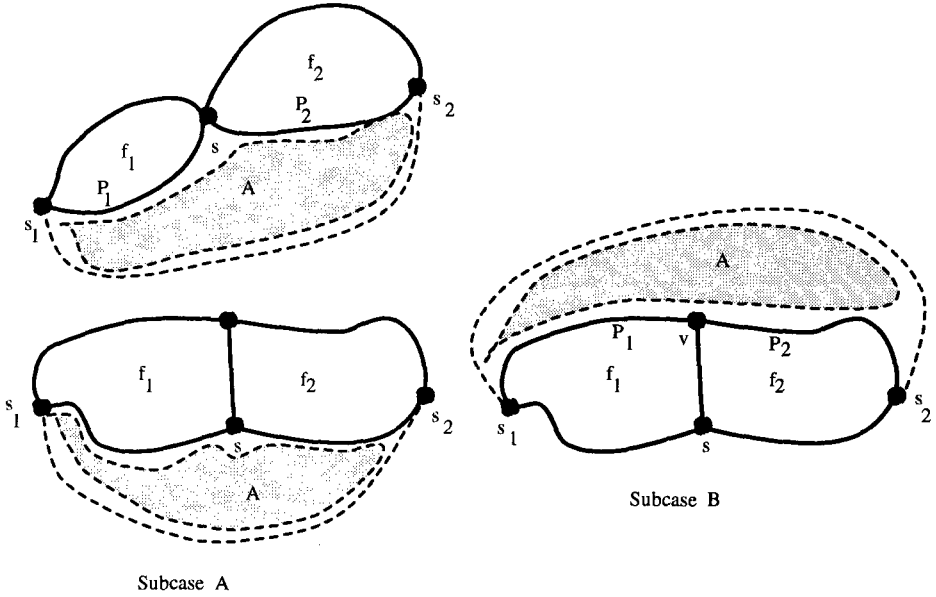


Fig. 4. Faces share a vertex or edge

largest sink is  $v$ . If  $v$  itself is a forced vertex, we are done. We are now left with the case when  $v$  can be resolved in both  $f_1$  and  $f_2$ . If both the face cycles are of size four, then  $s$  and  $v$  can be resolved in either

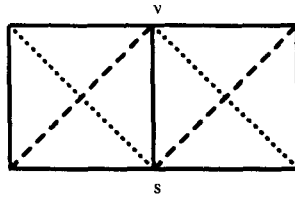


Fig. 5. Face cycles of size four

of the ways shown in Figure 5. Note that only these vertices can be resolved in these faces. Also the manner in which they are resolved within the faces does not affect the choices for other faces.

The other case is that at least one of the face sizes must be larger than four. Let  $f_2$  be the face whose size is larger than four. All the vertices on  $P_1$  have labels between  $s_1$  and  $v$ . Similarly, all the vertices on  $P_2$  have labels between  $s_2$  and  $v$ . Hence we can assume that these vertices are adjacent, as no such edge can be added to any of the faces. From the previous discussion on the types of labellings on the smallest faces, we get that  $l(v)$  is the second largest label in each of the



faces  $f_1$  and  $f_2$ , where  $n_1$  is the largest label in  $f_1$  and  $n_2$  in  $f_2$ . Similarly, it follows that  $l(s) = 2$ . Also there is a vertex  $y$  which is adjacent to  $s_2$  such that  $l(y) = n_2$ .

As the edge  $(s, y)$  cannot be added to  $f_2$ , there must exist a vertex  $w$ , with  $l(w) = 2$ , which is adjacent to  $y$ .

$w$  already resolved: If the vertex  $w$  has already been resolved, then there exists a

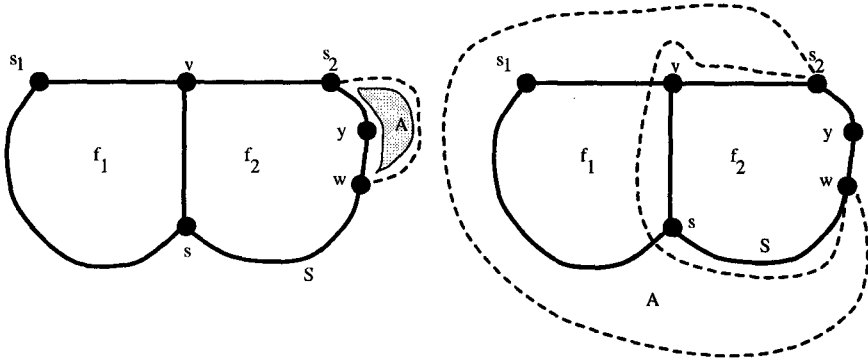


Fig. 6. Vertex  $w$  already resolved

path between  $w$  and  $s_2$ . The path can lie in either of the ways depicted in Figure 6. Consider the largest sink in the enclosed region in the first configuration in Figure 6. If it is in the interior it cannot be resolved. If it lies on the boundary it must be the vertex  $y$  which again cannot be resolved. Consider the second configuration shown in Figure 6. In this case consider the largest sink in the region bounded by the dotted line. If it lies in the interior then it cannot be resolved. If it lies on the boundary it must be  $v$ , but  $v$  cannot be the largest sink as it can be resolved in  $f_1$ .

$w$  is not already resolved: If there exists another face in which  $w$  can be resolved and they do not share an edge, we can argue as previously.

If the two faces share an edge, let  $s_3$  be the smallest source vertex in  $f_3$  (the other face in which  $w$  can be resolved). As  $s_2$  and  $s_3$  have already been resolved, there exists a path  $P$  between  $s_2$  and  $s_3$ . The path  $P$  can lie in either of the ways shown in Figure 7.

**Figure 7(a):** Look at the largest sink in region  $A$ . If it lies on the boundary, then it must lie between  $y$  and  $s_3$ . Hence there is a unique choice of face for resolution. If it lies in the interior it cannot be resolved.

**Figure 7(b):** Consider the largest sink in the region shown. As  $l(v) = n - 1$  it cannot lie between  $s$  and  $w$ . If it lies between  $w$  and  $s_3$  there is a unique face for resolution.

Another possible configuration, is when  $f_3$  and  $f_2$  share the edge  $(w, z)$ . In this case, it can be argued using exactly similar arguments that there is either a forced vertex or an unresolvable vertex. This completes the proof of the lemma.

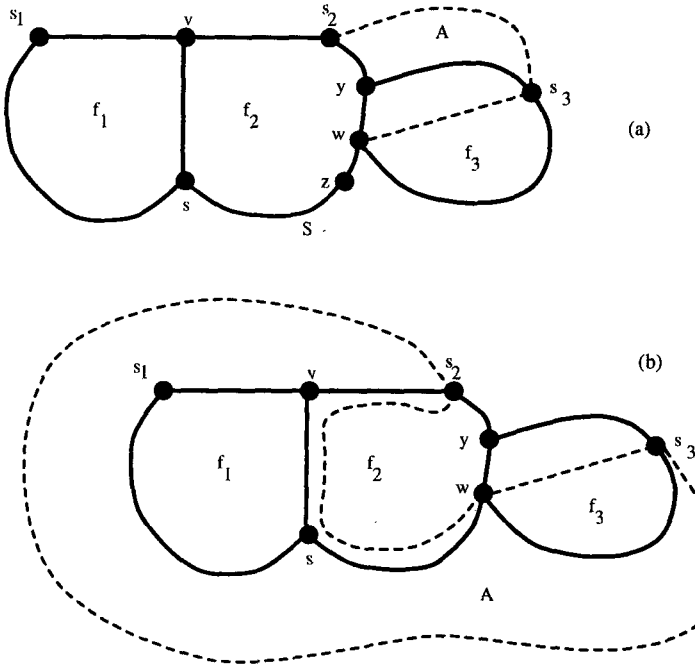


Fig. 7. Vertex  $w$  not already resolved

□

**Theorem 9.** *Algorithm Upward-Numbering-Testing produces a graph with a generalised  $s$ - $t$  numbering, of which  $G$  is a subgraph, if  $l$  is an upward numbering in linear time.*

**Proof:** The correctness follows from the previous lemmas. We just give a brief sketch of the time complexity arguments. The first phase in which edges are to be added to the various faces can be carried out in time linear in face size for each face. We maintain a list of sources and sinks from which we can add edges and as we traverse the face, we add edges depending on the label associated with the new source or sink. The second phase in which the sources and sinks are to be resolved can be carried out by maintaining a graph of sources and sinks and faces in which they can be resolved. Hence the algorithm produces a graph with a generalised  $s$ - $t$  numbering in linear time. □

#### 4 Grid intersection graphs and upward drawings with labels

A bipartite graph is a *grid intersection graph*, if it can be realized as the intersection graph of horizontal and vertical line segments in the plane. Such graphs

have been studied in literature [10, 4]. A bipartite graph can be represented by the adjacency matrix, whose rows correspond to the vertices in one of the colour classes and whose columns correspond to members of the other class. The following problem was raised in [10, 4]: If the vertices on the left hand side class of the bipartite graph are linearly ordered, and are to be represented as horizontal segments placed at the same level (y-coordinate) as their order, can the bipartite graph be realized as an intersection graph.

In the case that all the vertices on the right side are vertices of degree two, then such a realization can be converted to an upward planar drawing of the graph (corresponding to the bipartite graph) with labels (as defined by the linear order). The upward drawing corresponding to the following matrix is shown in Figure 8.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Each horizontal segment can be contracted to a single point producing the above

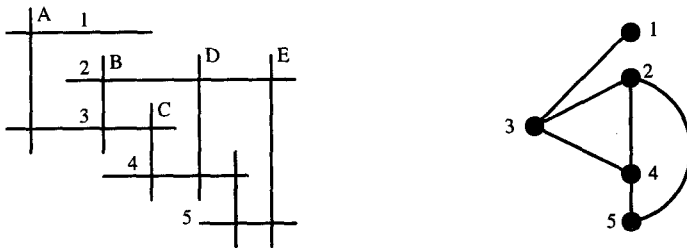


Fig. 8. An upward drawing of an IHV graph with labels

drawing. The process is clearly reversible, that is, such a drawing can be converted to a realization of the corresponding bipartite graph using horizontal and vertical segments alone (see [12] for details). Hence testing for the realization is easy for bipartite graphs arising from triconnected graphs.

## 5 Discussion

We have given an algorithm to test for *upward numbering* for triconnected graphs. The problem of testing *upward numbering* for arbitrary graphs has some interesting connections with the problem of characterising grid intersection graphs [10, 5].

## References

1. G. Di Battista and E. Nardelli. *An Algorithm for Planarity Testing of Hierarchical Graphs*, volume 246 of *Lecture Notes in Computer Science*, pages 277–289. Springer-Verlag, 1987.
2. G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoretical Computer Science*, 61:175–198, 1988.
3. P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, to appear.
4. M. Chandramouli. *Upward Planar Graph Drawings*. PhD thesis, IIT Bombay, 1994.
5. M. Chandramouli and A. A. Diwan. Intersection graphs of horizontal and vertical line segments in the plane, 1992. Unpublished manuscript.
6. A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. In *Graph Drawing 94, DIMACS*, 1994.
7. L. S. Heath and S. Pemmaraju. Recognizing leveled-planar dags in linear time. In *Proceedings of Graph Drawing '95*, 1995.
8. M. D. Hutton and A. Lubiw. Upward planar drawing of single source acyclic digraphs. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 203–211, 1991.
9. D. R. Kelly. Fundamentals of planar ordered sets. *Discrete Mathematics*, 63:197–216, 1987.
10. J. Kratochvíl. A special planar satisfiability problem and some consequences of its np-completeness. *Discrete Appl. Math.* (to appear).
11. X. Lin. *Analysis of Algorithms for Drawing Graphs*. PhD thesis, Department of Computer Science, University of Queensland, 1992.
12. R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Disc. and Comp. Geometry*, 1(4):321–341, 1986.