Angewandte Mathematik und Informatik
Universität zu Köln

Report No. 99.369

Level Planarity Testing in Linear Time

by

Michael Jünger, Sebastian Leipert and Petra Mutzel

1999

Partially supported by DFG-Grant Ju204/7-3, Forschungsschwerpunkt "Effiziente Algorithmen für diskrete Probleme und ihre Anwendungen"

Institut für Informatik
Universität zu Köln
Pohligstraße 1
50969 Köln

1991 Mathematics Subject Classification: 05C85, 68R10, 90C35
Keywords: Level Planar Graph, Level Planarity Testing, Hierarchy, PQ-tress

Level Planarity Testing in Linear Time

Michael Jünger[a] Sebastian Leipert[b]
Petra Mutzel[c]

[a] Institut für Informatik, Universität zu Köln, mjuenger@informatik.uni-koeln.de
[b] Institut für Informatik, Universität zu Köln, leipert@informatik.uni-koeln.de
[c] Max-Planck-Institut für Informatik, Saarbrücken, mutzel@mpi-sb.mpg.de

Abstract

A level graph $G = (V, E, \phi)$ is a directed acyclic graph with a mapping $\phi : V \to \{1, 2, \dots, k\}$, $k \geq 1$, that partitions the vertex set V as $V = V^1 \cup V^2 \cup \dots \cup V^k$, $V^j = \phi^{-1}(j)$, $V^i \cap V^j = \emptyset$ for $i \neq j$, such that $\phi(v) \geq \phi(u) + 1$ for each edge $(u, v) \in E$. The level planarity testing problem is to decide if G can be drawn in the plane such that for each level V^i, all $v \in V^i$ are drawn on the line $l_i = \{(x, k - i) \mid x \in \mathbb{R}\}$, the edges are drawn monotone with respect to the vertical direction, and no edges intersect except at their end vertices. If G has a single source, the test can be performed in $\mathcal{O}(|V|)$ time by an algorithm of Di Battista and Nardelli [1988] that uses the PQ-tree data structure introduced by Booth and Lueker [1976]. PQ-trees have also been proposed by Heath and Pemmaraju [1995, 1996] to test level planarity of level directed acyclic graphs with several sources and sinks. It has been shown in Jünger, Leipert, and Mutzel [1997] that this algorithm is not correct in the sense that it does not state correctly level planarity of every level planar graph. In this paper, we present a correct linear time level planarity testing algorithm that is based on two main new techniques that replace the incorrect crucial parts of the algorithm of Heath and Pemmaraju [1995, 1996].

1 Introduction

A fundamental issue in Automatic Graph Drawing is to display hierarchical network structures as they appear in software engineering, project management and database design. The network is transformed into a directed acyclic graph that has to be drawn with edges that are strictly monotone with respect to the vertical direction. Most applications imply a partition of the vertices into levels that have to be visualized by placing the vertices belonging to the same level on a horizontal line. These graphs are called level graphs. Using the PQ-tree data structure, Di Battista and Nardelli

[1988] have shown how to test if a level graph with a single source is level planar, i.e. can be drawn without edge crossings.

PQ-trees have also been proposed by Heath and Pemmaraju [1995, 1996] to test level planarity of level graphs with several sources and sinks. It has been shown in Jünger, Leipert, and Mutzel [1997] that this algorithm is not correct in the sense that it does not state correctly level planarity of every level planar graph. In this paper, we present a correct linear time level planarity testing algorithm that is based on two main new techniques that replace the incorrect crucial parts of the algorithm of Heath and Pemmaraju [1995, 1996].

This paper is organized as follows. After summarizing the necessary preliminaries in the next section, we give a short introduction to the PQ-tree data structure and the level planarity test presented by Heath and Pemmaraju [1995, 1996] in the third section. In Section 4 we summarize the incorrect crucial parts of this algorithm. A correct linear time level planarity test is presented in the fifth section. The proofs of correctness are presented in Section 6. We finish this paper with some remarks on the construction of a level planar embedding based on our algorithm.

2 Preliminaries

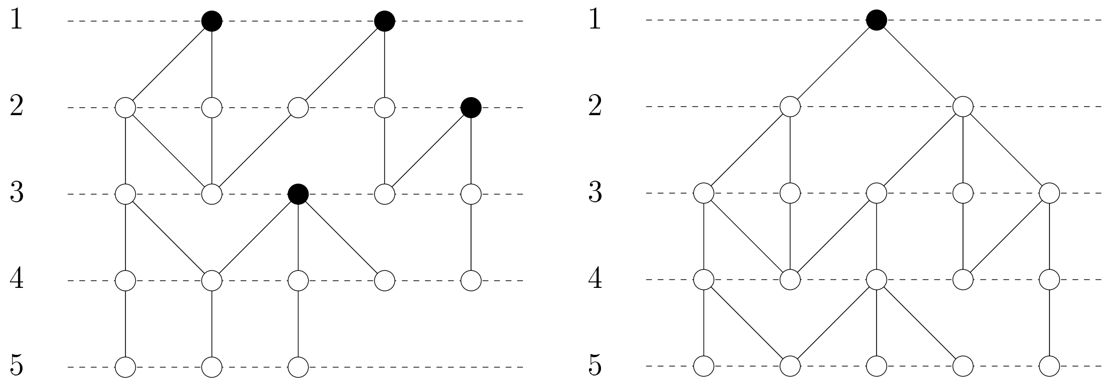
A level graph $G = (V, E, \phi)$ is a directed acyclic graph with a mapping $\phi : V \to \{1, 2, \ldots, k\}$, $k \geq 1$, that partitions the vertex set V as $V = V^1 \cup V^2 \cup \cdots \cup V^k$, $V^j = \phi^{-1}(j)$, $V^i \cap V^j = \emptyset$ for $i \neq j$, such that $\phi(v) \geq \phi(u) + 1$ for each edge $(u, v) \in E$. A vertex $v \in V^j$ is called a *level-j vertex* and V^j is called the *j-th level* of G. For a level graph $G = (V, E, \phi)$, we sometimes write $G = (V^1, V^2, \ldots, V^k; E)$.

A drawing of a level graph G in the plane is a *level drawing* if the vertices of every V^j, $1 \leq j \leq k$, are placed on a horizontal line $l_j = \{(x, k - j) \mid x \in \mathbb{R}\}$, and every edge $(u, v) \in E$, $u \in V^i$, $v \in V^j$, $1 \leq i < j \leq k$, is drawn as a monotonically decreasing curve between the lines l_i and l_j. A level drawing of G is called *level planar* if no two edges cross except at common endpoints. A level graph is *level planar* if it has a level planar drawing. A level graph G is obviously level planar if and only if all its components are level planar. We therefore may assume in the following without loss of generality that G is connected.

A level drawing of G determines for every V^j, $1 \leq j \leq k$, a total order $\leq_j$ of the vertices of V^j, given by the left to right order of the vertices on l_j. A *level embedding* consists of a permutation of the vertices of V^j for every $j \in \{1, 2, \ldots, k\}$ with respect to a level drawing. A level embedding with respect to a level planar drawing is called *level planar*.

A level graph $G = (V, E, \phi)$ is said to be *proper* if every edge $e \in E$ connects only vertices belonging to consecutive levels. Figure 1 shows two proper level graphs. If a level graph is not proper, it must have an edge $e = (v, w) \in E$ such that $v \in V^i$

and $w \in V^j$ with $1 \leq i < j - 1 \leq k - 1$. Such an edge is called a *long edge* and it is said to be *traversing* the levels l with $i < l < j$. Any nonproper level graph can be transformed into a proper level graph by replacing every long edge by a path having a dummy vertex for every traversed level. We therefore may assume that all level graphs are proper.



(a) A level graph.

(b) A hierarchy.

Figure 1: Examples of proper level graphs. Sources are drawn black.

A level graph G may have sinks and sources placed on various levels of the graph. A *hierarchy* is a level graph in which all sources belong to the first level V^1 of the graph. If G is a hierarchy with more than one vertex in V^1 it is always possible to add a new subset V^0 with exactly one vertex connected to every vertex of V^1. Such a transformation does not modify the planarity properties of the given hierarchy. As a consequence, it is sufficient to consider hierarchies with $|V^1| = 1$. Figure 1(b) shows a hierarchy.

3 The Approach by Heath and Pemmaraju

This section gives a brief introduction to the level planarity test of Heath and Pemmaraju [1995, 1996]. In order to test whether a level graph $G = (V, E, \phi)$ is level planar, it is sufficient to find an ordering $\leq_j$ of the vertices of every set V^j, $1 \leq j < k$, such that for every pair of edges $(u_1, v_1), (u_2, v_2) \in E$ with $u_1, u_2 \in V^j$ and $u_1 \leq_j u_2$, we have $v_1 \leq_{j+1} v_2$. Let G^j denote the subgraph of G induced by $V^1 \cup V^2 \cup \cdots \cup V^j$. Unlike G, G^j is not necessarily connected.

The basic idea of the level planarity testing algorithm presented by Heath and Pemmaraju [1995, 1996] is to perform a top-down sweep, processing the levels in the order $V^1, V^2, \ldots, V^k$ and computing for every level V^j, $1 \leq j \leq k$, a set of permutations of the vertices of V^j that appear in some level planar embedding of G^j. In case that the set of permutations of G^k is not empty, the graph $G = G^k$ is obviously level planar. A permutation induced by an ordering $\leq_j$ on the vertices of V^j is called a *witness* of G^j if the permutation appears in some level planar embedding of G^j.

A PQ-tree is a data structure that represents the permutations of a finite set U in which the members of specified subsets occur consecutively. This data structure has been introduced by Booth and Lueker [1976] to solve the problem of testing for the consecutive ones property (see, e.g., Fulkerson and Gross [1965]). A PQ-tree is a rooted and ordered tree that contains three types of nodes: leaves, P-nodes, and Q-nodes. The leaves are in one to one correspondence with the elements of U. The P- and Q-nodes are internal nodes. In subsequent figures, P-nodes are drawn as circles while Q-nodes are drawn as rectangles.

The *frontier* of a PQ-tree T, denoted by frontier(T), is the sequence of all leaves of T read from left to right, and the frontier of a node X, denoted by frontier(X), is the sequence of its descendant leaves read from left to right. The frontier of a PQ-tree is a permutation of the set U. We use the notion frontier(T) and frontier(X) also to denote the set of elements in frontier(T) and frontier(X), respectively, its meaning being evident by context. An *equivalence transformation* specifies a legal reordering of the nodes within a PQ-tree. The only legal equivalence transformations are

(i) any permutation of the children of a P-node, and

(ii) the reverse permutation of the children of a Q-node.

Two PQ-trees T and T' are *equivalent* if and only if their underlying trees are equal and T can be transformed into T' by a sequence of equivalence transformations. The equivalence of two PQ-trees is denoted $T \equiv T'$. The set of *consistent permutations* of a PQ-tree is the set of all frontiers that can be obtained by a sequence of equivalence transformations and is denoted by

$$\mathrm{PERM}(T) \quad = \quad \{\mathrm{frontier}(T') \mid T' \equiv T\} \ .$$

Given a PQ-tree T over the set U and given a subset $S \subseteq U$, Booth and Lueker [1976] developed a pattern matching algorithm called *reduction* and denoted by REDUCE(T, S) that computes a PQ-tree T' representing all permutations of T in which the elements of S form a consecutive sequence. A tree with no nodes at all called *empty tree* is returned if the original tree could not be reduced for the specified set S.

If G^j is a hierarchy, the set of permutations of the vertices of V^j that appear in some level planar embedding of G^j can be represented by a PQ-tree T_j according to Di Battista and Nardelli [1988] by

- replacing every cut vertex by a P-node,
- replacing every biconnected component by a Q-node,
- replacing every vertex of V^j by a leaf, and
- rooting T at the node corresponding to the biconnected component or cut vertex that contains the source s.

In order to test whether the hierarchy G^{j+1} is level planar, Di Battista and Nardelli [1988] add for every edge (v_i, w), $w \in V^{j+1}$, $v_i \in V^j$, $i = 1, 2, \ldots, \mu$, $\mu \geq 1$ a virtual vertex w_{v_i} labeled w and virtual edges (v_i, w_{v_i}) to the graph G^j. The authors then try to compute for every vertex $w \in V^{j+1}$ a sequence of permutations of components around cut vertices and swappings of maximal biconnected components such that all virtual vertices labeled w form a consecutive sequence on the horizontal line l_{j+1}. If such a sequence can be found, it is obvious that the vertex w can be added to G^j without destroying level planarity. The process of computing the prescribed sequence can be efficiently done using the PQ-tree T_j, applying the function REDUCE to sets of leaves corresponding to virtual vertices with the same label, yielding a linear time algorithm. The algorithm of Di Battista and Nardelli [1988] makes use of the crucial property that in a hierarchy G each G^j is connected; a property that is not shared by general level graphs.

In case that G^j, $1 \leq j < k$, consists of more than one connected component, Heath and Pemmaraju suggest to use a PQ-tree for every component and formulate a set of rules of how to merge the PQ-trees T_1 and T_2 associated with the components F_1 and F_2, if F_1 and F_2 are both adjacent to some vertex $v \in V^{j+1}$.

During a *First Reduction Phase* Heath and Pemmaraju [1995, 1996] reduce the leaves of T_1 and T_2 corresponding to the vertex v, called the *pertinent* leaves. After successfully performing the reduction, the consecutive sequence of pertinent leaves is replaced by a single pertinent representative in both T_1 and T_2. Going up one of the trees T_i, $i \in \{1, 2\}$, from its pertinent representative, an appropriate position is searched, allowing the tree T_j, $j \neq i$, to be placed into T_i. After successfully performing this step the resulting tree T' has two pertinent leaves corresponding to the vertex v, which again are reduced and replaced by a single representative. If any of the steps fails, Heath and Pemmaraju state that the graph G is not level planar.

Merging two PQ-trees T_1 and T_2 corresponds to merging the two components F_1 and F_2 and is accomplished using certain information that is stored at the nodes of the PQ-trees. For any subset S of the set of vertices in V^j, $1 \leq j \leq m$, that belongs to a component F, define $\mathrm{ML}(S)$ to be the greatest $d \leq j$ such that $V^d, V^{d+1}, \ldots, V^j$ induces a subgraph in which all vertices of S occur in the same connected component. The level $\mathrm{ML}(S)$ is said to be the *meet level* of S. For a Q-node Y in the corresponding PQ-tree T_F with ordered children $Y_1, Y_2, \ldots, Y_t$ integers denoted by $\mathrm{ML}(Y_i, Y_{i+1})$, $1 \leq i < t$, are maintained satisfying $\mathrm{ML}(Y_i, Y_{i+1}) =$

ML(frontier(Y_i)∪frontier(Y_{i+1})). For a P-node X a single integer denoted by ML(X) that satisfies ML(X) = ML(frontier(X)) is maintained. Furthermore, define LL(F), the *low indexed level*, to be the smallest d such that F contains a vertex in V^d and maintain this integer at the root of the corresponding PQ-tree. The *height* of a component F in the subgraph G^j is j − LL(F). The LL-value merely describes the size of the component.

Figure 2 shows an example of a graph G^5 with two components. The LL-value of the left component 2 is 1 and the LL-value of the right component is 2. Figure 3 shows the PQ-trees corresponding to the graph G^5 together with the ML-values that are stored at the nodes. The maintenance of the ML-values during the pattern matching algorithm REDUCE is straightforward.

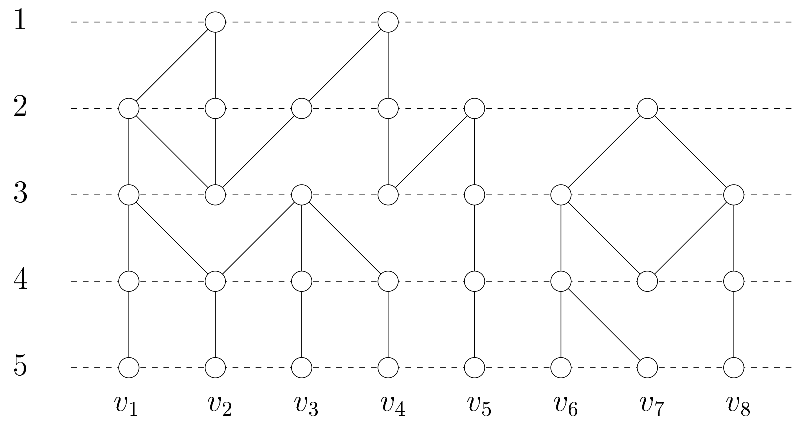


Figure 2: A G^5 with two components.

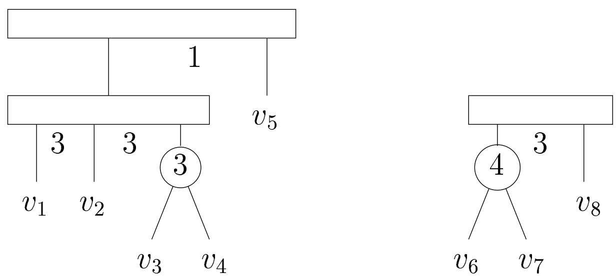


Figure 3: PQ-trees corresponding to G^5 shown in Fig. 2.

Using these LL- and ML-values, Heath and Pemmaraju [1995, 1996] describe a set of rules how to connect two PQ-trees and claim that the pertinent leaves of the new tree T' are reducible if and only if the corresponding component F' is level planar.

Proposition 3.1 (Heath and Pemmaraju [1995, 1996]). *Suppose that X is the least common ancestor of a pair of leaves v and w in a PQ-tree. If X is a*

P-node, then

$$\text{ML}(\{v, w\}) = \text{ML}(X) \ .$$

Proposition 3.2 (Heath and Pemmaraju [1995, 1996]). *Suppose that X is the least common ancestor of a pair of leaves v and w in a PQ-tree. Suppose further that X is a Q-node with ordered children $X_1, X_2, \ldots, X_t$ such that $v \in \text{frontier}(X_p)$ and $w \in \text{frontier}(X_q)$, where $1 \leq p < q \leq t$. Then*

$$\text{ML}(\{v, w\}) = \min\{\text{ML}(X_i, X_{i+1}) \mid p \leq i < q\} \ .$$

The next proposition of Heath and Pemmaraju [1996] formally states the fact that as we follow a path in a PQ-tree from a leaf to the root, the encountered ML-values are non increasing.

Proposition 3.3 (Heath and Pemmaraju [1995, 1996]). *Suppose that node X is the parent of an internal node Y in a PQ-tree. Define x as follows:*

$$x = \begin{cases} \text{ML}(X) & \text{if } X \text{ is a } P\text{-node;} \\ \max\{\text{ML}(Y, Z) \mid Z \text{ is a child of } X \text{ adjacent to } Y\} & \text{if } X \text{ is a } Q\text{-node.} \end{cases}$$

Define y as follows:

$$y = \begin{cases} \text{ML}(Y) & \text{if } Y \text{ is a } P\text{-node;} \\ \min\{\text{ML}(Y_i, Y_{i+1}) \mid 1 \leq i < t\} & \text{if } Y \text{ is a } Q\text{-node with ordered} \\ & \text{children } Y_1, Y_2, \ldots, Y_t. \end{cases}$$

Then $x \leq y$ holds.

A detailed description of the merge operations of Heath and Pemmaraju [1995, 1996] is now given. Let $G = (V, E, \phi)$ be a k-level graph and F_1 and F_2 be two components of G^j, $1 \leq j < k$, both being adjacent to the same vertex $v \in V^{j+1}$. Let T_1 and T_2 be the PQ-trees of F_1 and F_2, both representing all level planar embeddings of their corresponding components after the application of the first merge phase for the level $j+1$. Identifying the vertices labeled v of the components F_1 and F_2 constructs a new component F. For this new component F a new PQ-tree T is needed that represents all level planar embeddings of F. Heath and Pemmaraju now formulate a set of rules of how to construct the PQ-tree T using the two existing ones T_1 and T_2.

Without loss of generality, we may assume that $\text{LL}(T_1) \leq \text{LL}(T_2)$. Thus component F_2 is the smaller component and an embedding of F_1 has to be found such that F_2 can be nested within the embedding of F_1. This corresponds to adding the root of T_2 as a child to a node of the PQ-tree T_1 constructing a new PQ-tree T'. In order to find an appropriate location to insert T_2 into T_1, we start with the leaf labeled v in T_1 and proceed upwards in T_1 until a node X' and its parent X are encountered satisfying one of the following five conditions.

Merge Condition A The node X is a P-node with $\mathrm{ML}(X) < \mathrm{LL}(T_2)$. Attach T_2 as child of X in T_1.

Merge Condition B The node X is a Q-node with ordered children $X_1, X_2, \ldots, X_t$, $X' = X_1$, and $\mathrm{ML}(X_1, X_2) < \mathrm{LL}(T_2)$. Replace X' in T_1 by a Q-node Y having two children, X' and the root of T_2. The case where $X' = X_t$ and $\mathrm{ML}(X_{t-1}, X_t) < \mathrm{LL}(T_2)$ is symmetric.

Although the ML-value of X' and T_2 is 0 (since they are not connected), we set for technical purposes $\mathrm{ML}(X', T_2) := \mathrm{ML}(X', X_2)$. This is legal, since the two leaves labeled v in the frontier of X' and T_2 are reduced after applying the merge operation and replaced by a single representative which removes the value $\mathrm{ML}(X', T_2)$ from the tree. We will see later, how we benefit from this value.

Merge Condition C The node X is a Q-node with ordered children $X_1, X_2, \ldots, X_t$, $X' = X_i$, $1 < i < t$, and $\mathrm{ML}(X_{i-1}, X_i) < \mathrm{LL}(T_2)$ and $\mathrm{ML}(X_i, X_{i+1}) < \mathrm{LL}(T_2)$. Replace X' by a Q-node Y with two children, X' and the root of T_2, and set (for technical purposes again) $\mathrm{ML}(X', T_2) := \max\{\mathrm{ML}(X_{i-1}, X'), \mathrm{ML}(X', X_{i+1})\}$.

Merge Condition D The node X is a Q-node with ordered children $X_1, X_2, \ldots, X_t$, $X' = X_i$, $1 < i < t$, and

$$\mathrm{ML}(X_{i-1}, X_i) < \mathrm{LL}(T_2) \leq \mathrm{ML}(X_i, X_{i+1}) \ .$$

Attach the root of T_2 as child of X between X_{i-1} and X', and set (for technical purposes) $\mathrm{ML}(X_{i-1}, T_2) := \mathrm{ML}(X_{i-1}, X')$ and $\mathrm{ML}(T_2, X') := \mathrm{ML}(X_{i-1}, X')$.

In case that

$$\mathrm{ML}(X_i, X_{i+1}) < \mathrm{LL}(T_2) \leq \mathrm{ML}(X_{i-1}, X_i) \ ,$$

attach the root of T_2 as child of X between X' and X_{i+1} and set the ML-values correspondingly.

Merge Condition E The node X' is the root of T_1. Reconstruct T_1 by inserting a Q-node Y as new root of T_1 with two children X' and the root of T_2 and set $\mathrm{ML}(X', T_2) := 0$.

The merge operations for the conditions B and C both take care of the fact that the subgraph corresponding to T_2 can be embedded on either side of the subgraph corresponding to X' with respect to the subgraph X. By construction, Heath and Pemmaraju [1996] make the following observations on the merge rules A, B, . . . , E.

Observation 3.4 (Heath and Pemmaraju [1996]). *Let* $\pi_1 \in \text{PERM}(T_1)$ *be a permutation of the leaves of* T_1*, such that* frontier(X') *is adjacent to a leaf labeled* w*, and* $\text{ML}(\text{frontier}(X') \cup \{w\}) < \text{LL}(T_2)$*. For any* $\pi_2 \in \text{PERM}(T_2)$*, there exists a permutation* $\pi \in \text{PERM}(T')$ *with* T' *being the new PQ-tree that is consistent with* π_1 *and* π_2 *and in which the* frontier(T_2) *occurs immediately after* frontier(X') *and immediately before* w*.*

Observation 3.5 (Heath and Pemmaraju [1996]). *Let* $\pi_1 \in \text{PERM}(T_1)$ *be a permutation such that the leaves of* frontier(X') *appear at the end of* π_1*. Let* $\pi_2 \in \text{PERM}(T_2)$ *be an arbitrary permutation of the leaves of* T_2*. Then there exists a permutation* $\pi \in \text{PERM}(T')$ *that is consistent with* π_1 *and* π_2 *and in which the leaves in* frontier(T_2) *occur immediately after* frontier(X')*.*

4 On the Incorrectness on the Approach of Heath and Pemmaraju

In Jünger, Leipert, and Mutzel [1997] we have shown that the order of merging the components is important for testing a level graph. Within the second merge phase, components are merged in an arbitrary order. We show that choosing an arbitrary order may result in PQ-trees that are not reducible with respect to a vertex v (although the graph is level planar). Consider four different components F_1, F_2, F_3, F_4 and their corresponding PQ-trees T_1, T_2, T_3, T_4 each having a pertinent leaf corresponding to some level-$(j+1)$ vertex v. For simplicity we assume that for every component the leaf labeled v appears in all permutations at one end of the PQ-tree. Assume further that the smallest common ancestor of a leaf labeled v and any other leaf adjacent to it is a Q-node. Figure 4 shows such a component F_i and its corresponding PQ-tree T_i. The number $l_i = \text{ML}(\{w^i_{p_i}, v\})$ is the ML-value between the pertinent leaf labeled v and the frontier of its left neighbor.

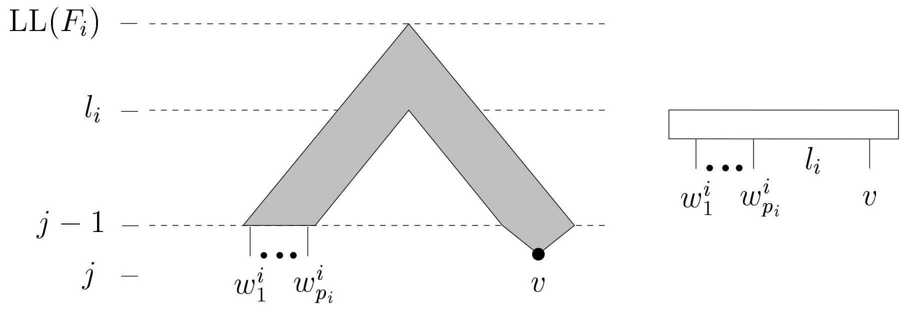


Figure 4: Component F_i and its corresponding PQ-tree T_i.

Assuming that the condition

$$\text{LL}(F_1) \le l_1 < \text{LL}(F_2) \le l_2 < \text{LL}(F_3) \le l_3 < \text{LL}(F_4) \le l_4$$

on the ML- and LL-values of the components holds, it is possible to merge all four components into one component such that the pertinent leaves form a consecutive sequence. Figure 5 shows the four components, indicating how the components can be merged constructing a level planar embedding.

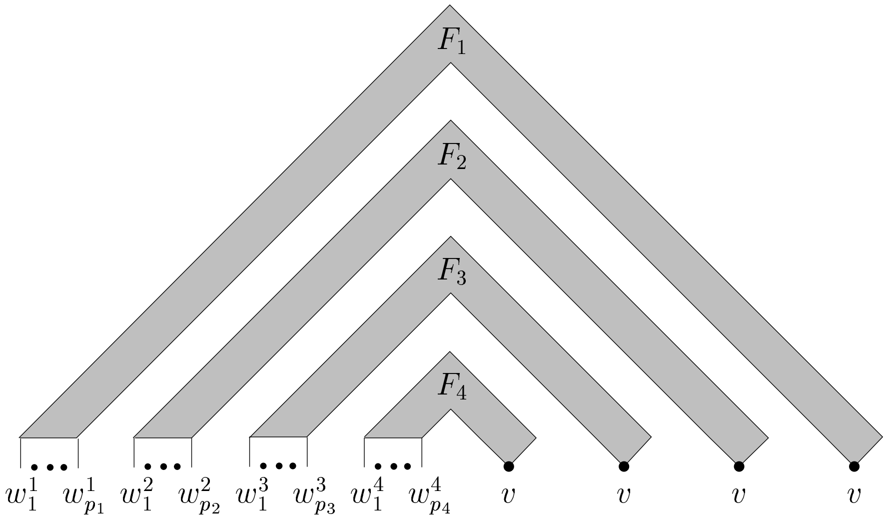


Figure 5: Possible level planar arrangement of the components F_1, F_2, F_3, F_4.

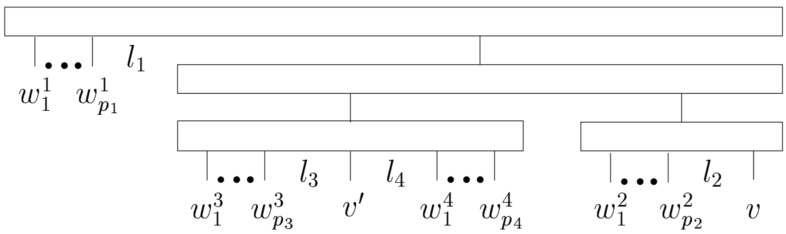


Figure 6: PQ-tree T''' whose pertinent leaves labeled v are not reducible. The vertex v' denotes the single representative of the the leaves labeled v corresponding to the components F_1, F_3 and F_4.

Consider the following merge operations on the components F_1, F_2, F_3, F_4 and their corresponding PQ-trees:

1. merge F_1 and F_4 into a component F',
2. merge F' and F_3 into a component F'',
3. merge F'' and F_2 into a component F'''.

The resulting PQ-tree T''' corresponding to F''' is shown in Figure 6. The pertinent leaves labeled v do not form a consecutive sequence in any permissible permutation of the PQ-tree T'''. Hence the algorithm presented by Heath and Pemmaraju [1995, 1996] states nonlevel planarity for certain level planar graphs.

Furthermore, components of G^j that have just one level-j vertex are not treated properly. In fact, they may be inserted at wrong positions in other PQ-trees. This is due to the fact that during the first merge phase the algorithm reduces for every PQ-tree all leaves with the same label and replaces them by a single representative. Clearly, this replacement corresponds to the construction of new interior faces in the corresponding subgraph. However, PQ-trees are not designed to carry information about interior faces, hence the information about the "space" within these interior faces gets lost. It is easy to see that situations may occur where components being adjacent to just one level-j vertex have to be embedded within one of these interior faces. The approach of Heath and Pemmaraju [1995, 1996] does not detect this fact, which is another reason that it may incorrectly state the non level planarity of a level planar graph.

Heath and Pemmaraju [1995, 1996] claim that their algorithm can be implemented using only $O(|V|)$ time. This is true for the merge and reduce operations. However, considering two PQ-trees T_1, T_2 both having a leaf labeled v and a leaf labeled w, Heath and Pemmaraju [1996] suggest to merge the trees T_1 and T_2 at the leaves labeled v constructing a new PQ-tree T and then reduce T with respect to the leaves labeled v as well as with respect to the leaves labeled w. It is not clear how the update operations that are necessary for detecting both pairs of leaves can be done in $O(|V|)$ time, Heath and Pemmaraju [1995, 1996] do not discuss this matter.

We will combine two new strategies to eliminate the problems we encountered in the algorithm of Heath and Pemmaraju [1995, 1996].

5 A Linear Time Level Planarity Test

In this section we discuss how to construct a correct algorithm LEVEL-PLANARITY-TEST that tests a level graph $G = (V^1, V^2, \ldots, V^k; E)$ for level planarity. Before we describe our algorithm, called LEVEL-PLANARITY-TEST, let us recall some old and introduce some new terminology. Since G^j is not necessarily connected, let m_j denote the number of components of G^j and let F_i^j, $i = 1, 2, \ldots, m_j$, denote the components of G^j. Figure 7 shows a G^4 with $m_4 = 2$ components F_1^4 and F_2^4. The set of vertices in F_i^j is denoted by $V(F_i^j)$.

Let H_i^j be the graph arising from F_i^j as follows: For each edge $e = (u, v)$, where u is a vertex in F_i^j and $v \in V^{j+1}$ we introduce a *virtual vertex* with label v and a *virtual edge* that connects u and this virtual vertex. Thus there may be several virtual vertices with the same label, adjacent to different components of G^j and each with exactly one entering edge. The form H_i^j is called the *extended form* of F_i^j and the set of virtual vertices of H_i^j is denoted by frontier(H_i^j). Figure 8 shows possible extended forms H_1^4 and H_2^4 of the example in Fig. 7. The virtual vertices on level 5 are denoted by their labels. The frontier of H_1^4 consists of one virtual vertex labeled u, two vertices labeled v, and two vertices labeled w.

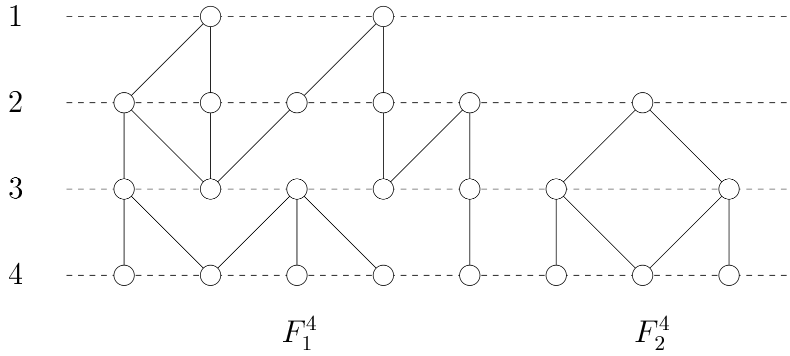


Figure 7: A G^4 with $m_2 = 2$ components F_1^4 and F_2^4.

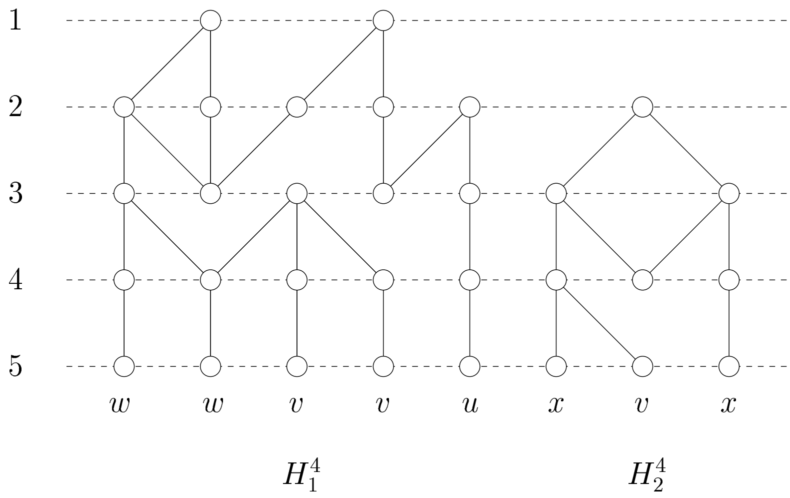


Figure 8: Two possible extended forms H_1^4 and H_2^4 of Fig. 7.

The set of virtual vertices of H_i^j that are labeled $v \in V^{j+1}$ is denoted by S_i^v. Figure 9 shows the sets S_1^w, S_1^v, and S_1^u, the set of virtual vertices labeled w,v and u of H_1^4, respectively. The set S_1^x is empty.

The graph that is created from an extended form H_i^j by identifying all virtual vertices with the same label to a single vertex is called *reduced extended form* and denoted by R_i^j. Figure 10 shows the reduced extended forms R_1^4 and R_2^4 of H_1^4 and H_2^4. In R_1^4 the vertices labeled w have been identified and the vertices labeled v have been

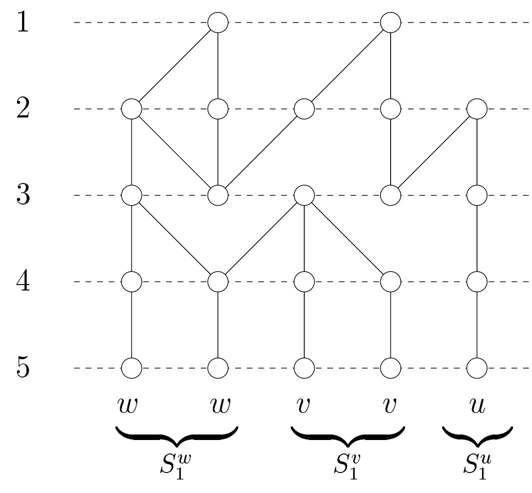


Figure 9: Sets S_1^w, S_1^v, and S_1^u of H_1^4.

identified. In order to identify the two vertices labeled x in R_2^4, it was necessary to permute the left most vertex labeled x and v. Both forms R_1^4 and R_2^4 now have exactly one vertex labeled v.

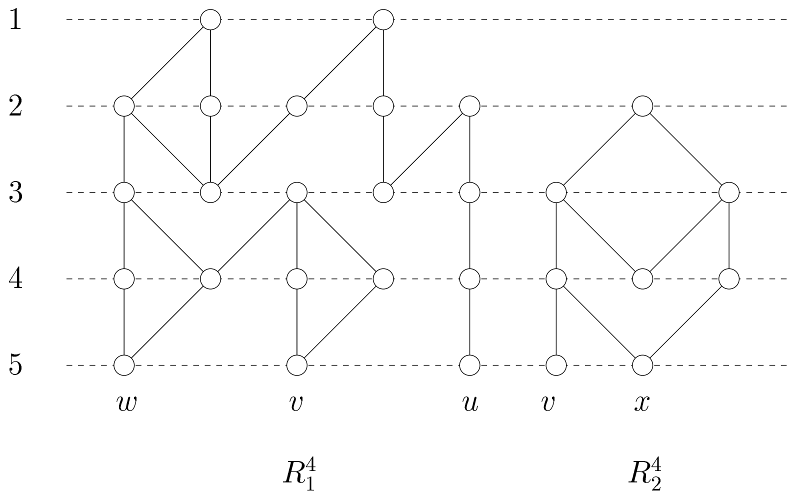


Figure 10: Reduced extended forms R_1^4 and R_2^4 of H_1^4 and H_2^4.

The set of virtual vertices of R_i^j is denoted by frontier(R_i^j). If S_i^v of H_i^j is not empty, we denote the vertex with label v of R_i^j (i.e., the vertex that arose from identifying all virtual vertices of S_i^v) by v_i and update $S_i^v = \{v_i\}$. The graph arising from the identification of two virtual vertices v_i and v_l (labeled v) of two reduced extended forms R_i^j and R_l^j is denoted $R_i^j \cup_v R_l^j$. We call $R_i^j \cup_v R_l^j$ a *merged reduced form.* The vertex arising from the identification of v_i and v_l is denoted by $v_{\{i,l\}}$ (and labeled by

v of course). If $\mathrm{LL}(R_i^j) \leq \mathrm{LL}(R_l^j)$ we say R_l^j is *v-merged* into R_i^j. The form that is created by v-merging R_l^j into R_i^j and identifying all virtual vertices with the same label $w \neq v$ is again a reduced extended form and denoted by R_i^j (thus renaming $R_i^j \cup_v R_l^j$ with the name of the "higher" form). Figure 11 shows the resulting merged reduced extended form $R_1^4 \cup_v R_2^4$ after R_2^4 (the smaller form) has been v-merged into R_1^4 (the higher form). Since R_1^4 is the higher form, $R_1^4 \cup_v R_2^4$ is renamed into R_1^4.

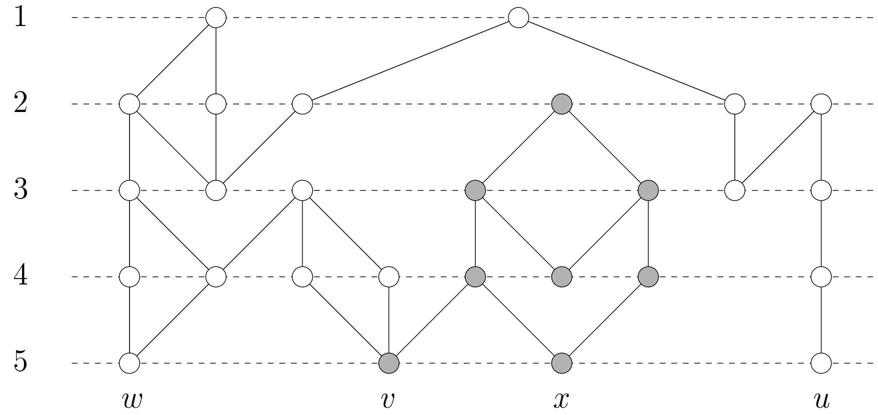


Figure 11: A Merged reduced extended form $R_1^4 \cup_v R_2^4$ after R_2^4 has been v-merged into R_1^4. The former vertices of R_2^4 are drawn shaded.

As will be shown later in this paper, we omit scanning for leaves with the same label after we have v-merged several reduced extended forms. This is done in order to achieve linear running time. However, this strategy results in improper reduced extended forms, possibly having several virtual vertices with the same label. These forms are called *partially reduced extended forms.*

If some reduced extended form has been v-merged into R_i^j, the form R_i^j is called *v-connected*, otherwise R_i^j is called *v-unconnected.* Thus, R_1^4 shown in Fig. 10 is v-unconnected, R_1^4 shown in Fig. 11 is v-connected.

A reduced extended form R_i^j that is v-unconnected for all $v \in V^{j+1}$ is called *primary*. A reduced extended form R_i^j that is v-connected for at least one $v \in V^{j+1}$ is called *secondary*. Again, R_1^4 shown in Fig. 10 is primary, R_1^4 shown in Fig. 11 is secondary.

Let R_i^j be a reduced extended form such that $S_i^v \neq \emptyset$ for some $v \in V^{j+1}$ and $S_i^w = \emptyset$ for all $w \in V^{j+1} - \{v\}$, then R_i^j is called *v-singular.* Figure 12 shows a v-singular form.

Let $\mathcal{T}(G^j)$ be the set of level planar embeddings of all components of G^j. We will show that in case that G^j is level planar, the set of permutations of level-j vertices in level planar embeddings of each component F_i^j of G^j can be described by a *PQ*-tree $T(F_i^j)$. Clearly this is true, if every F_i^j is a hierarchy (Di Battista and Nardelli [1988]). Hence it remains to be shown that it is also possible to maintain a *PQ*-

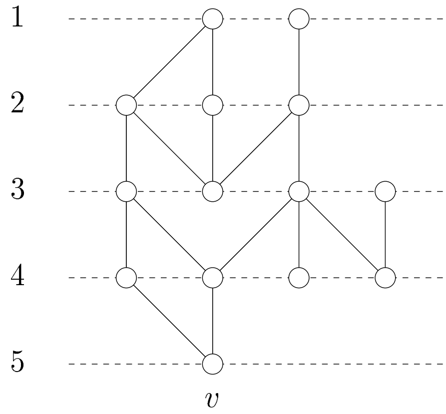


Figure 12: A v-singular form.

tree for every component F_i^j that is not a hierarchy. As has been shown in Booth and Lueker [1976], it is straightforward to construct from $T(F_i^j)$ a PQ-tree $T(H_i^j)$ associated with H_i^j. Thus the leaves of $T(H_i^j)$ correspond to the virtual vertices of H_i^j and we label the leaves of $T(H_i^j)$ as their counterparts in H_i^j. By construction, $\mathcal{T}(G^j)$ is a set of PQ-trees. Considering a function CHECK-LEVEL that computes for every level j, $j = 2, 3, \ldots, k$ the set $\mathcal{T}(G^j)$ of level planar embeddings of the components G^j, the algorithm LEVEL-PLANARITY-TEST can be formulated as follows.

Bool **LEVEL-PLANARITY-TEST**($G = (V^1, V^2, \ldots, V^k; E)$)
begin
  Initialize $\mathcal{T}(G^1)$;
  for $j := 1$ to $k - 1$ do
    $\mathcal{T}(G^{j+1})$ = CHECK-LEVEL($\mathcal{T}(G^j), V^{j+1}$);
    if $\mathcal{T}(G^{j+1}) = \emptyset$ then
      return "false";
  return "true";
end.

We introduce two new strategies that lead to a correct algorithm as well as new techniques for obtaining linear running time. One strategy is to sort all PQ-trees with a leaf labeled v in their frontier according to their LL-values and merge them according to this ordering. We show that the new PQ-tree constructed by the application of this ordering represents all possible level planar embeddings of the corresponding new component. Our second strategy for a correct treatment of v-singular components consists of keeping at every single representative the size of the largest interior face that has been constructed by identifying the corresponding virtual vertices. When merging a PQ-tree of a v-singular component into another PQ-tree with lower LL-value, this information is checked first. When merging two non singular components, this information has to be updated when introducing a new single representative. Here we have to take into account that merging two components results into something that we call a *cavity*. Considering the intersection $\mathcal{C}$ of the half space $\{x \in \mathbb{R}^2 \mid x_2 \geq k - j - 1\}$ and the outer face of a level planar drawing of the current extended forms, a v-cavity $\mathcal{C}_v$ is defined to be a region of $\mathcal{C}$ such that v is adjacent to the region. Obviously v can be adjacent to several such regions. Moreover, these regions are not unique, since they depend on the current embedding. This is no drawback, since we only need to maintain a lower bound on the size of the largest v-cavity which can be easily implemented using the PQ-trees and the LL- and ML-values of Heath and Pemmaraju [1995, 1996]. Figure 13 shows such a v-cavity. The arrow on the right side of the figure determines the height of the cavity. A v-singular form can only be level planar embedded within this cavity, if it is smaller than the height of the cavity. We define $\mathrm{LL}(C_v)$ to be the *low indexed level* of a v-cavity $\mathcal{C}_v$ as $\mathrm{ML}(\{w \in V^{j+1} \mid w \text{ is on the boundary of } \mathcal{C}_v\})$. The height of a v-cavity C_v is $j + 1 - \mathrm{LL}(C_v)$. The following lemma reveals how to obtain a lower bound on the height of the largest v-cavity in every level planar embedding of two forms that have been v-merged. If two PQ-trees have been merged at their leaves labeled v, the LL-value of a v-cavity is obviously smaller or equal to the ML-value stored at the root of the pertinent subtree. Notice that we here make use of the ML-values that have been "technically" set during the merge operations.

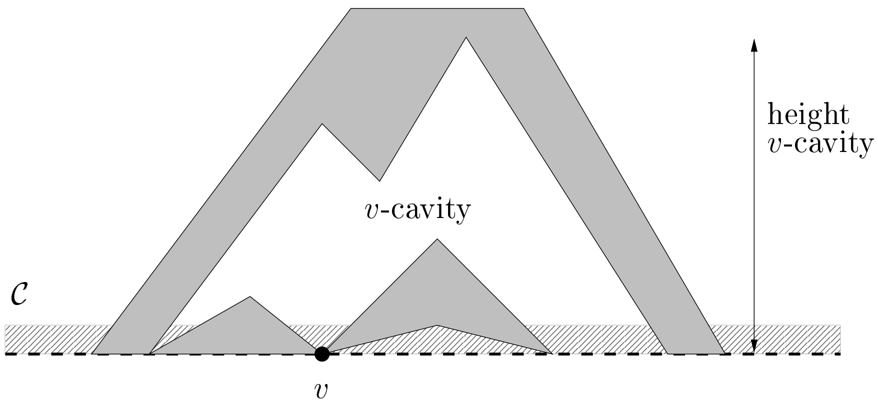


Figure 13: A v-cavity.

Lemma 5.1. *Let R_1^j and R_2^j be two nonsingular partially reduced extended forms of G^j of a level planar graph G with $S_1^v \neq \emptyset$ and $S_2^v \neq \emptyset$. Let T_1 and T_2 be their corresponding PQ-trees with $\mathrm{LL}(T_1) \leq \mathrm{LL}(T_2)$, representing all level planar embeddings of R_1^j, and R_2^j, respectively. Let T be the PQ-tree constructed by v-merging T_2 into T_1. Let X be the root of the pertinent subtree in T. If X is a P-node let $h = \mathrm{ML}(X)$, and if X is a Q-node let Y and Z be the (only) pertinent children of X and let $h = \mathrm{ML}(Y, Z)$. Let $\mathcal{C}_v$ be the largest v-cavity in an arbitrary level planar embedding of $R_1^j \cup_v R_2^j$. Then the following holds.*

$$\mathrm{LL}(C_v) \leq h.$$

Proof. By construction of the merge operation we have that $h < \mathrm{LL}(T_2)$ holds. Since R_2^j can be embedded level planar in R_1^j, there must exist in every level planar embedding at least one v-cavity $\mathcal{C}_v$ such that $\mathrm{LL}(C_v) \leq h$. □

As we have mentioned, merging two PQ-trees at leaves labeled v may result in a PQ-tree T with several leaves labeled $w \neq v$. Linearity of the algorithm is achieved by not applying the strategy of reducing the leaves labeled w, since it is not clear if the detection of these leaves reveals linear running time. We reduce these leaves labeled w only when considering their PQ-tree T for a merge operation at w. Thus we first reduce all leaves labeled w in every tree and then merge these trees at w. We show that the modified algorithm works correctly.

When merging PQ-trees at leaves labeled v, update operations have to be applied to the leaves of the new tree, since the leaves must know the PQ-tree that they belong to. To avoid the usage of *Fast-Union-Find-Set* operations which sum up to $\mathrm{O}(|V|\alpha(|V|, |V|))$ operations, we apply the following strategy. Leaves are updated only when they are involved in a reduce or merge operation. In order to update the leaves, we traverse all nodes from the considered leaf to its root. Let U be a set of PQ-trees with leaves labeled v in their frontier. We show that if this strategy is applied for all leaves except for the leaves in the PQ-tree with the lowest LL-value in U, the number of operations is proportional to the number of operations needed to reduce all these leaves. We do not need to know the PQ-tree with the lowest LL-value in U. It is easy to see that this tree is implicitly defined. Hence we can avoid for every merge operation the traversal of the tree corresponding to the highest component. Thus the total number of operations needed to perform the updates is proportional to the number of operations needed in all calls of the function REDUCE, where Booth and Lueker [1976] proved that the total number of operations in all calls of REDUCE for planar graphs is in $\mathcal{O}(|V|)$.

The procedure CHECK-LEVEL is divided into two phases. The *First Reduction Phase* constructs the PQ-trees corresponding to the reduced extended forms of G^j. Every PQ-tree $T(F_i^j)$ that represents all level planar embeddings of some component F_i^j is transformed into a PQ-tree $T(H_i^j)$ representing all level planar embeddings of the extended form H_i^j. We continue to reduce in every PQ-tree $T(H_i^j)$ all leaves with the same label, thereby constructing a new PQ-tree, representing all level planar embeddings of H_i^j, where leaves with the same label occupy consecutive positions. If one of the reductions fails, G cannot be level planar. Leaves with the same label v are replaced by a single representative v_i. Such a single representative v_i gets the same label v, storing either a value $\mathrm{PML}(v_i) = \mathrm{ML}(R)$ if the root R of the pertinent subtree is a P-node or a value $\mathrm{QML}(v_i) = \min\{\mathrm{ML}(x, y) \mid x, y$ consecutive children of R, x pertinent or y pertinent$\}$, if the root R is a Q-node. The default value of $\mathrm{QML}(v_i)$ and $\mathrm{PML}(v_i)$ is set to $k+1$. These values store the height of the largest new interior face that is constructed by merging the vertices labeled v and are needed to handle singular components correctly.

PQ-trees of different components are merged in the *Second Reduction Phase* using a function INSERT, if the components are adjacent to the same vertex v on level $j+1$. Given the set of leaves labeled v, we first determine their corresponding PQ-trees. If some leaves labeled v are in the frontier of the same PQ-tree, we reduce them and replace them by a single representative. The PQ-trees are then merged pairwise in the order of their sizes. We show that using this ordering a PQ-tree $T(F)$ is constructed that represents all possible level planar embeddings of the merged components. If there is more than one v-singular reduced extended form, $v \in V^{j+1}$, we only need to merge the largest one of these forms. If it is possible to embed this form level planar, all other v-singular forms obviously can be embedded level planar as well. Even though v may not be the only common vertex in the merged components, we do not reduce leaves with label $w \neq v$ in the PQ-tree in order to obtain a linear time algorithm. If one of the reduce or merge operations fails while applied in this phase, the graph G is not level planar. The PML- and QML-values are updated by using a function UPDATE. The function REPLACE removes all leaves with a common label v after these leaves have been reduced (and therefore are consecutive in all permissible permutations) and replaces them by a single representative (see also Booth and Lueker [1976]). Finally we add for every source of V^{j+1} its corresponding PQ-tree. Thus the set of PQ-trees constructed by the function CHECK-LEVEL represents all level planar embeddings of every component of G^{j+1}. The following code fragment contains operations that perform on the graph G. They are kept in the code for documentation purposes. Any implementation would of course rely only on the manipulation of the PQ-trees.

Once the Second Reduction Phase is complete, the function CHECK-LEVEL finishes with some final updates.

$\mathcal{T}(G^{j+1})$ **CHECK-LEVEL(**$\mathcal{T}(G^j), V^{j+1}$**)**
begin
  First Reduction Phase
  for every component F_i^j in G^j and its corresponding PQ-tree $T(F_i^j)$ do
    construct H_i^j;
    construct $T(H_i^j)$ (from the PQ-tree obtained in the previous iteration);
  for $v = 1$ to $|V^{j+1}|$ do
    for every extended form H_i^j do
      if $S_i^v \neq \emptyset$ then
        if REDUCE($T(H_i^j), S_i^v$) $= \emptyset$ then return $\emptyset$;
        else
          let v_i be a single representative of S_i^v;
          UPDATE(S_i^v,v_i);
          REPLACE(S_i^v,v_i);
    for every extended form H_i^j do
      $T(R_i^j) := T(H_i^j)$;

Second Reduction Phase

```
for $v = 1$ to $|V^{j+1}|$ do
   for every leaf labeled $v$ do
      find the corresponding $PQ$-tree;
   for every found $PQ$-tree $T(R_i^j)$ do
      if $S_i^v \geq 2$ then
         if REDUCE($T(R_i^j)$,$S_i^v$) $= \emptyset$ then return $\emptyset$;
         else
            let $\tilde{v}$ be a single representative of $S_i^v$;
            UPDATE($S_i^v$,$\tilde{v}$);
            REPLACE($S_i^v$,$\tilde{v}$);
   reorder indices such that $S_1^v, S_2^v, \ldots, S_p^v \neq \emptyset$, and $S_{p+1}^v, S_{p+2}^v, \ldots, S_{m_j}^v = \emptyset$;
   let $q$ be the number of $v$-singular reduced extended forms;
   eliminate all $v$-singular $R_i^j$ except for the one with the lowest LL-value;
   renumber the remaining $R_i^j$ from 1 to $p - q + 1$;
   $p := p - q + 1$;
   sort the $R_i^j$, such that LL($R_1^j$) $\leq$ LL($R_2^j$) $\leq$ LL($R_3^j$) $\leq \cdots \leq$ LL($R_p^j$);
(*) for $i = 2$ to $p$ do
      $T(R_1^j) :=$ INSERT($T(R_1^j), T(R_i^j), v$);
      $R_1^j := R_1^j \cup_v R_i^j$;
      if REDUCE($T(R_1^j), S_1^v$) $= \emptyset$ then return $\emptyset$;
      else
         let $\tilde{v}$ be a single representative of $S_1^v$;
         UPDATE($S_1^v$,$\tilde{v}$);
         REPLACE($S_1^v$,$\tilde{v}$);
```

Final Updates

```
update the pointers of the leaves to their $PQ$-trees;
add for every source a corresponding $PQ$-tree to $\mathcal{T}(G^j)$;
return $\mathcal{T}(G^{j+1})$;
end.
```

We now describe the function INSERT for merging the PQ-trees corresponding to two components. All five rules presented by Heath and Pemmaraju can be adapted, but contrary to their algorithm, we have to deal with the fact that a PQ-tree may correspond to a singular component. The merge operation is encapsulated within the function MERGE. Merging two PQ-trees is handled by the method INSERT. Let LL(T_{large}) and LL(T_{small}) be two PQ-trees such that $S_{large}^v \neq \emptyset$ and $S_{small}^v \neq \emptyset$, and LL($T_{large}$) $\leq$ LL(T_{small}). Assume further that S_{large}^v and S_{small}^v have been reduced and replaced by a single representative v_{large} and v_{small}, respectively, and that $S_{large}^v = \{v_{large}\}$ and $S_{small}^v = \{v_{small}\}$, respectively. INSERT returns a new PQ-tree T_{merge}. The method does not reduce the pertinent sequence, nor does it replace pertinent leaves by a single leaf. Observe that in case frontier(T_{small}) $= S_{small}^v$, we do not really add T_{small} to T_{large} if the component corresponding to T_{small} can be embedded in an interior face or a v-cavity of the component corresponding to T_{large}.

T_{merge} **INSERT**$(T_{large}, T_{small}, v)$
begin
  if frontier$(T_{small}) \neq S^v_{small}$ then
    T_{large} := MERGE(T_{large},T_{small},v);
  else if PML$(v_{large}) \neq k+1$ then
    if PML$(v_{large}) <$ LL(T_{small}) then
      do nothing;
    else
      T_{large} := MERGE(T_{large},T_{small},v);
  else if QML$(v_{large}) \neq k+1$ then
    if QML$(v_{large}) <$ LL(T_{small}) then
      do nothing;
    else
      T_{large} := MERGE(T_{large},T_{small},v);
  return the new PQ-tree T_{large};
end.

The method UPDATE is a straightforward implementation of finding a lower bound on the height of a cavity that could possibly embed singular components.

void **UPDATE**(S_i^v,v')
begin
  $\text{PML}_{\min} := \min\{\text{PML}(\tilde{v}) \mid \tilde{v} \in S_i^v\}$;
  $\text{QML}_{\min} := \min\{\text{QML}(\tilde{v}) \mid \tilde{v} \in S_i^v\}$;
  let X be the root of the pertinent subtree.
  if X is a P-node then
    $\text{PML}_X := \text{ML}(X)$;
  else
    $$\text{QML}_X := \min\left\{\text{ML}(Y,Z) \;\middle|\; \begin{array}{l} Y, Z \text{ consecutive children of } X, \\ Y \text{ and } Z \text{ pertinent} \end{array}\right\};$$
  if $\min\{\text{PML}_{\min}, \text{PML}_X\} < \min\{\text{QML}_{min}, \text{QML}_X\}$ then
    $\text{PML}(v') := \min\{\text{PML}_{\min}, \text{PML}_X\}$;
    $\text{QML}(v') := k+1$;
  else
    $\text{QML}(v') := \min\{\text{QML}_{\min}, \text{QML}_X\}$;
    $\text{PML}(v') := k+1$;
end.

6 Proof of Correctness

In this section we prove the correctness of the level planarity test. The strategy is to apply an inductive argument. Since a subgraph of G that is induced by a source and its outgoing edges is a trivial hierarchy, we have according to Di Battista and Nardelli [1988] that for such a subgraph there exists a PQ-tree that represents the set of level planar embeddings. We need to show that throughout every iteration the PQ-trees are correctly maintained and the set of permissible permutations always represents exactly the set of level planar embeddings of the corresponding form.

In Lemma 6.1, the first reduction phase is proven to be correct. Proving the correctness of the second reduction phase is more involved. We show in Lemmas 6.3 and 6.4 that merging a set of PQ-trees at their leaves labeled v is performed correctly if the functions INSERT and REDUCE are applied as described in the Section 5. Since some of these PQ-trees may have several leaves labeled v in their frontier due to earlier w-merge operations on vertices labeled $w \neq v$, we show Lemma 6.2 that the reduction of the leaves labeled v is performed correctly.

We start with a lemma on the correctness of the first reduction phase. Let us assume that we are given a k-level planar graph G, an extended form H_i^j, $1 \leq j < k$, of G, and a PQ-tree T_i that represents all level planar embeddings of H_i^j. To prove the correctness of the first phase we show that there exists a PQ-tree $\tilde{T}_i$ equivalent to T_i, such that all leaves with a common label appear consecutively in the frontier of $\tilde{T}_i$. If such a PQ-tree exists, we are obviously able to reduce for every $v \in V^{j+1}$ the PQ-tree T_i with respect to the leaves labeled v and to replace these leaves by a single representative. It is easy to see that this new PQ-tree represents level planar embeddings of the reduced extended form R_i^j and it remains to show that the PQ-tree represents exactly all level planar embeddings of R_i^j.

Lemma 6.1. *Let $G = (V, E, \phi)$ be a level planar graph with $k > 1$ levels. Let F_i^j, $i \in \{1, 2, \ldots, m_j\}$, be an arbitrary component of G^j, $1 \leq j < k$, and let H_i^j be its extended form and R_i^j be its reduced extended form. If T_i is a PQ-tree representing all level-planar embeddings of H_i^j, the PQ-tree T_i' constructed from T_i by reducing every set S_i^v and replacing it by a single representative v_i witnesses all level planar embeddings of R_i^j.*

Proof. Although the results of the lemma should be clear by the previous discussions, we give the proof in full detail. We first show that there exists a PQ-tree $\tilde{T}_i$ that is equivalent to T_i, such that for all $v \in V^{j+1}$, the leaves corresponding to S_i^v occupy consecutive positions in the frontier of $\tilde{T}_i$.

Consider an arbitrary level planar embedding $\mathcal{E}(R_i^j)$ of the reduced extended form R_i^j and let π be the witness of $\mathcal{E}(R_i^j)$. The level-j neighbors $w \in V(F_i^j)^j$ of $v \in V^{j+1}$ in F_i^j form a consecutive sequence on level j in $\mathcal{E}(R_i^j)$ (except for possible sinks). Every edge $e = (w, v)$, $w \in V(F_i^j)^j$, $v \in V^{j+1}$ corresponds to a virtual vertex of S_i^v.

Therefore, we get a level planar embedding of $\mathcal{E}(H_i^j)$ of H_i^j by replacing every edge e by a virtual edge with an incident virtual vertex labeled v in $\mathcal{E}(R_i^j)$. See Fig. 14 for an illustration. Let π' be a witness to $\mathcal{E}(H_i^j)$. By construction, all virtual vertices labeled v form a consecutive sequence in π' for every $v \in V^{j+1}$. Since $\mathcal{E}(H_i^j)$ is a level planar embedding of H_i^j, its witness π' must be in PERM(T_i). Thus there exists a PQ-tree $\tilde{T}_i$ equivalent to T_i with frontier$(\tilde{T}_i) = \pi'$.

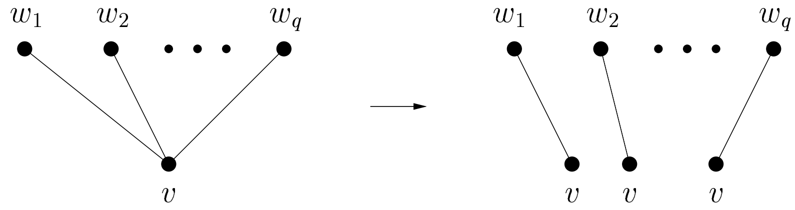


Figure 14: Illustration of the proof of Lemma 6.1. For every edge (w_i, v) $w_i \in V(F_i^j)^j$, $i = 1, 2, \ldots, q$, a virtual edge with a virtual vertex labeled v is introduced.

The existence of the PQ-tree $\tilde{T}_i$ that is equivalent to T_i guarantees that the reduction of T_i with respect to S_i^v for every $v \in V^{j+1}$ is successful. These reductions construct a PQ-tree $\tilde{T}_i'$ with PERM$(\tilde{T}_i') \subseteq$ PERM(T_i). Furthermore we have $\pi' \in$ PERM$(\tilde{T}_i')$ and we may assume that $\pi' =$ frontier$(\tilde{T}_i')$.

Replacing in $\tilde{T}_i'$ all leaves with a common label by a single representative, we obtain a PQ-tree T_i', where we have by construction for the witness π of $\mathcal{E}(R_i^j)$ that $\pi \in$ PERM(T_i'). Thus T_i' represents all level planar embeddings of R_i^j. □

Consider a set of partially reduced extended forms that contain vertices labeled v in their frontier. Before the PQ-trees corresponding to these forms can be merged, we have to ensure that there exists at most one leaf labeled v in the frontier of each PQ-tree. However, some of these PQ-trees may have more than one leaf labeled v in their frontier, due to previous merge operations on leaves labeled $w \neq v$. Thus before showing the correctness of the merge operation, we first show that reducing in a PQ-tree all leaves with the same label v constructs a PQ-tree that represents all level planar embeddings of the corresponding partially reduced extended form with the virtual vertices labeled v identified.

Lemma 6.2. *Let $G = (V, E, \phi)$, be a level graph with $k > 1$ levels. Let $v \in V^{j+1}$, $1 \leq j < k$. Let R_i^j be a level planar partially reduced extended form with $S_i^v \neq \emptyset$ and with $|S_i^w| \leq 1$ for all $w = 1, 2, \ldots, v-1$. Let $T(R_i^j)$ be the corresponding PQ-tree, representing all level planar embeddings of R_i^j.*

Let F be the subgraph constructed from R_i^j by identifying all virtual vertices labeled v to a single vertex v. Let $T(F)$ be the PQ-tree constructed as described in the second merge phase of CHECK-LEVEL, by reducing in $T(R_i^j)$ all leaves labeled v. Then

PERM($T(F)$) *is exactly the set of permutations of level-*$(j+1)$ *vertices that appear in level planar embeddings of* F.

Proof. We first show that $T(F)$ represents level planar embeddings of F. Applying the function REDUCE to the PQ-tree $T(R_i^j)$ with respect to the leaves labeled v, creates either a PQ-tree $\tilde{T}(R_i^j)$ such that the vertices labeled v occupy consecutive positions or an empty PQ-tree. If $\tilde{T}(R_i^j)$ is not the empty PQ-tree, we have that $\text{PERM}(\tilde{T}(R_i^j)) \subseteq \text{PERM}(T(R_i^j))$ represents all level planar embeddings of R_i^j such that all leaves labeled v form a consecutive subsequence. Replacing these leaves by a single representative constructs a PQ-tree $T(F)$ that represents level planar embeddings of F.

We now prove that for any $\mathcal{E}_F^j$ a PQ-tree equivalent to $T(F)$ exists that represents exactly the permutation of the level-$(j+1)$ vertices of $\mathcal{E}_F^j$. The idea is to transform the level planar embedding $\mathcal{E}_F^j$ into a level planar embedding of R_i^j giving us valid information on the PQ-tree. The transformation replaces in $\mathcal{E}_F^j$ the vertex v by a sequence of virtual vertices. We associate every virtual vertex of v with a primary reduced extended form R_l^j, $l \neq j$, that has been w-merged, $w \in \{1, 2, \ldots, v-1\}$, into R_i^j in an earlier iteration of the second reduction phase if this reduced extended form was adjacent to w. However, in order to perform this transformation, we first need to show that the incoming edges of w that are associated with R_l^j, $l \neq j$, appear consecutively around w in $\mathcal{E}_F^j$.

Let q be the number of level-$(j+1)$ vertices of F. Let $\pi = [w_1, w_2, \ldots, w_q]$ be a witness of the embedding $\mathcal{E}_F^j$. F is either primary or it has been constructed by merging reduced extended forms at vertices $w \in \{1, 2, \ldots, v-1\}$. Hence the set of incoming edges of v can be partitioned into $p \geq 1$ sets of incoming edges belonging to the primary reduced extended forms that have been w-merged to create R_i^j. Let $R_1^j, R_2^j, \ldots, R_p^j$ be these primary reduced extended forms.

We first show that for every $l = 1, 2, \ldots, p$, the incoming edges of v that belong to a reduced extended form R_l^j, $l \in \{1, 2, \ldots, p\}$, appear consecutively in the clockwise order of incoming edges of v in the embedding $\mathcal{E}_F^j$. To show this, we assume first that only primary reduced extended forms are w-merged into R_i^j.

Let R_ν^j and R_μ^j, $\nu, \mu \in \{1, 2, \ldots, p\}$, $\nu \neq \mu$, be two primary reduced extended forms such that R_ν^j and R_μ^j have been w_ν-merged and w_μ-merged into R_i^j, respectively, where $w_\nu, w_\mu \in \{1, 2, \ldots, v-1\}$. Let $y_1, y_2, \ldots, y_c$, $c \geq 1$ be the set of level-j neighbors of v in R_ν^j and let $z_1, z_2, \ldots, z_d$, $d \geq 2$, be the set of level-j neighbors of v in R_μ^j. Assume that in the clockwise order of neighbors of v in the level planar embedding of $\mathcal{E}_F^j$ a vertex $y_l \in \{y_1, y_2, \ldots, y_c\}$ appears between the vertices z_a and z_b with $z_a, z_b \in \{z_1, z_2, \ldots, z_d\}$, $z_a \neq z_b$. Since R_μ^j is connected and v-unconnected (since it is primary), there exists a path P_μ in R_μ^j connecting the vertices z_a and z_b neither using v nor w_ν. Since R_ν^j is connected, there exists a path P_ν in R_ν^j connecting the vertices y_l and w_ν not using v. Both paths cross each other but have no vertex in common, a contradiction to the level planar embedding $\mathcal{E}_F^j$. See Fig. 15 for illustration.

In case that primary reduced forms are w-merged, $w < v$, before they are merged into R_i^j (and thus secondary reduced extended forms are w-merged into R_i^j), similar reasoning holds.

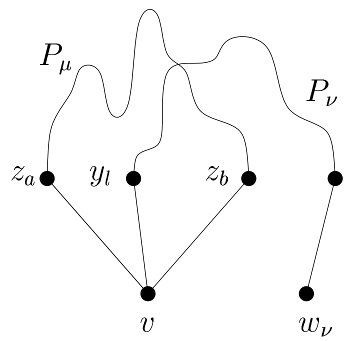


Figure 15: Illustration of the proof of Lemma 6.2. If in $\mathcal{E}_F^j$ the sequence of incoming edges of v belonging to R_μ^j is separated by some edge (y_l, v) with $y_l \notin V(R_\mu^j)$, two vertex disjoint paths exist that cross each other.

We construct from $\mathcal{E}_F^j$ an embedding $\mathcal{E}'$. Introduce for every reduced extended form R_ν^j, $\nu \in \{1, 2, \ldots, m_j\}$, that has been w-merged into F a vertex v_ν if $S_\nu^v \neq \emptyset$. Replace v by a sequence of virtual vertices v_ν, such that v_ν is adjacent to the same vertices as v in R_ν^j. Label each vertex v_ν with v. See for an illustration the example shown in Fig. 16.

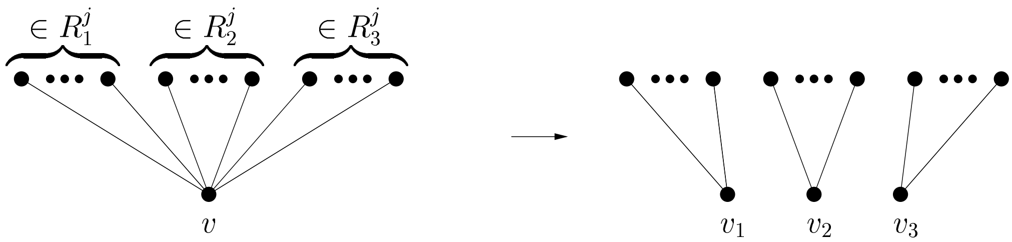


Figure 16: Illustration of the proof of Lemma 6.2. For the forms R_1^j, R_2^j, and R_3^j we introduce virtual vertices v_1, v_2, and v_3. We replace v by the sequence of virtual vertices, not reordering the incoming edges of v.

Since the incoming edges of v that correspond to a reduced extended form R_ν^j appear consecutively around v, the so constructed embedding $\mathcal{E}'$ is obviously level planar. Furthermore the graph corresponding to $\mathcal{E}'$ is identical to R_i^j, and we have by assumption that the witness π' of $\mathcal{E}'$ is in $\mathrm{PERM}(T(R_i^j))$. Since the witness π arises from π' by identifying all (consecutive) leaves labeled v, we have by construction that $\pi \in \mathrm{PERM}(T(F))$. $\square$

The next lemma shows a more technical result that is needed for proving the correctness of the second reduction phase. Let R_i^j and R_l^j be two partially reduced extended forms and let T_i and T_l be their corresponding PQ-trees where $\mathrm{LL}(T_i) \leq \mathrm{LL}(T_l)$. When merging the PQ-trees T_i and T_l at leaves labeled v, we insert the PQ-tree T_l into the PQ-tree T_i. After applying any of the merge operations of Heath and Pemmaraju [1996], the tree T_l is completely contained as a subtree of T_i. While the frontier of T_i has changed (by inserting T_l as a subtree) the frontier of T_l has not changed at all. Hence, all leaves in frontier(T_l), including the leaf labeled v, form a consecutive sequence in the new PQ-tree T_i.

This implies that if we want to use these merge operations for PQ-trees, the level-(j+1) vertices of R_l^j must form a consecutive sequence on level $j+1$ in every level planar embedding of $R_i^j \cup_v R_l^j$. However, this is not the case in general. Consider the example shown in Fig. 17 showing four partially reduced extended forms $R_1^j, R_2^j, R_3^j, R_4^j$ that have been v-merged. The forms are constructed similarly to the components F_1, F_2, F_3, F_4 that are shown in the counterexample of Fig. 5 on page 10. If we first v-merge R_4^j into R_1^j and then v-merge R_3^j into R_1^j and then v-merge R_2^j into R_1^j we know already from Section 4 that the PQ-tree constructed by this sequence of merge operations is not reducible (see Fig. 6 on Page 10). In fact, there exist level planar embeddings of $R_1^j \cup_v R_2^j \cup_v R_3^j \cup_v R_4^j$ such that the virtual vertices of R_2^j do not form a consecutive sequence on level $j+1$. Such an embedding is shown in Fig. 17 where the virtual vertices $w_1^2, w_2^2, \ldots, w_{q_2}^2$ of R_2^j and the vertex v are separated by the virtual vertices $w_1^3, w_2^3, \ldots, w_{q_3}^3$ of R_3^j.

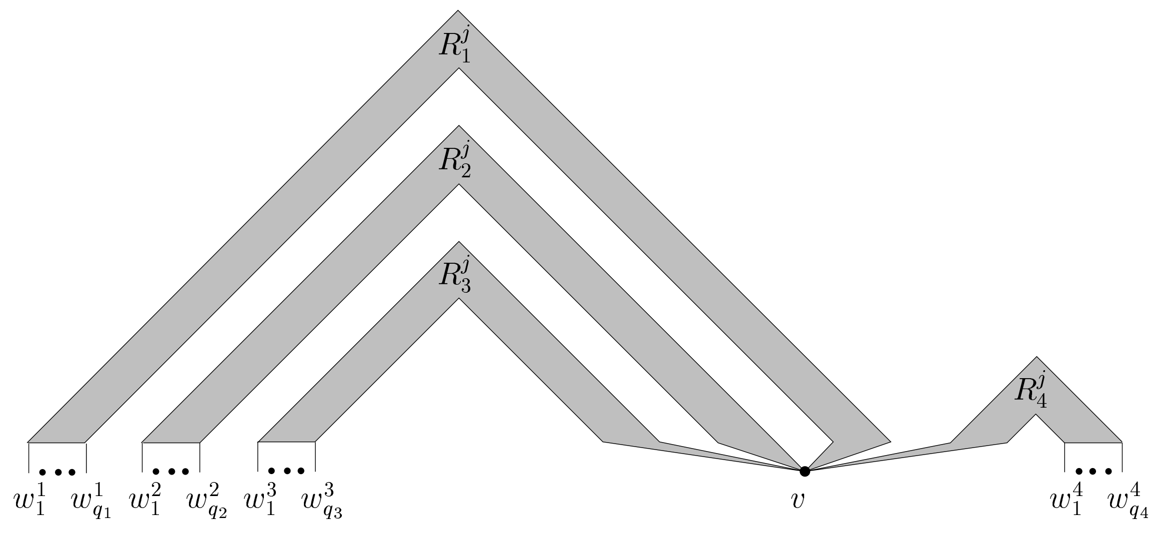


Figure 17: A Level planar embedding of the components $R_1^j \cup_v R_2^j \cup_v R_3^j \cup_v R_4^j$ where the virtual vertices $w_1^2, w_2^2, \ldots, w_{q_2}^2$ are separated from v by $w_1^3, w_2^3, \ldots, w_{q_3}^3$.

If we want to use the merge operations, we have to guarantee that in all level planar embeddings of two v-merged forms, the virtual vertices of the smaller form appear consecutively on level $j+1$. As the counterexample shows, this does not necessarily hold for every merge operation.

The following two lemmas show that if there are two or more partially reduced extended forms that have to be v-merged, there exists an ordering such that pairwise v-merging the forms according to this ordering guarantees the following. When v-merging two forms, the virtual vertices of the smaller form always form a consecutive sequence in all level planar embeddings of the merged form. The ordering is obtained by sorting the forms according to their LL-values. We merge the two partially reduced extended forms with lowest LL-value (that is, we merge the two largest forms). This constructs a new form, say F, and we then start merging the largest remaining form into F until all forms are merged into F.

Since the order of merging the forms is very important, we expand our terminology. Let $R_1^j, R_2^j, \ldots, R_p^j$, $p \geq 2$, be partially reduced extended forms of G^j such that $S_i^v \neq \emptyset$ for all $i \in \{1, 2, \ldots, p\}$. Assume without loss of generality that

$$\mathrm{LL}(R_1^j) \leq \mathrm{LL}(R_2^j) \leq \mathrm{LL}(R_3^j) \leq \cdots \leq \mathrm{LL}(R_p^j) \ .$$

Let F be the subgraph constructed by v-merging $R_1^j, R_2^j, \ldots, R_p^j$. Thus, F equals $R_1^j \cup_v R_2^j \cup_v \cdots \cup_v R_p^j$. If for some vertex $w \neq v$ the sets S_i^w and S_l^w of two extended forms R_i^j and R_l^j, $i \neq l$, are not empty, the virtual vertices in these sets are not identified in F. Thus all virtual vertices with common label are kept separate except for the virtual vertices labeled v.

Let $R_{\{1,2,\ldots,i\}}^j = R_1^j \cup_v R_2^j \cup_v \cdots \cup_v R_i^j$ denote the form that is constructed by v-merging $R_1^j, R_2^j, \ldots, R_i^j$ in this order. (In our previous terminology, which is more useful to describe the algorithm, $R_{\{1,2,\ldots,i\}}^j$ is renamed into R_1^j.) Obviously, we have that $R_{\{1,2,\ldots,p\}}^j = F$.

For a partially reduced extended form R_i^j let $\overline{S}_i^v$ denote the set of virtual vertices of R_i^j except for the vertices labeled v. Let $\overline{S}_{\{1,2,\ldots,i\}}^v$ denote the set of virtual vertices except for the vertices labeled v of $R_{\{1,2,\ldots,i\}}^j$. Let $\pi_{\{1,2,\ldots,i\}}$, $i \leq p$, denote a witness to a level planar embedding of $R_{\{1,2,\ldots,i\}}^j$. In the example of Fig. 17 we have $\overline{S}_4^v = \{w_1^4, w_2^4, \ldots, w_{q_4}^4\}$, and $\overline{S}_{\{1,2,3\}}^v = \{w_1^1, w_2^1, \ldots, w_{q_1}^1\} \cup \{w_1^2, w_2^2, \ldots, w_{q_2}^2\} \cup \{w_1^3, w_2^3, \ldots, w_{q_3}^3\}$. The witness of the shown level planar embedding is $\pi_{\{1,2,3,4\}} = [w_1^1, w_2^1, \ldots, w_{q_1}^1, w_1^2, w_2^2, \ldots, w_{q_2}^2, w_1^3, w_2^3, \ldots, w_{q_3}^3, v, w_1^4, w_2^4, \ldots, w_{q_4}^4]$.

In order to prove that the virtual vertices of the smaller form R_i^j (that is merged into the larger form $R_{\{1,2,\ldots,i-1\}}^j$) appear consecutively in any level planar embedding of the new form $R_{\{1,2,\ldots,i\}}^j$, we need to show that $\overline{S}_i^v$ and the vertex v are consecutive. The concept of the proof is to assume the opposite and then to find a path in R_i^j and a path in $R_{\{1,2,\ldots,i-1\}}^j$ that cross each other in $R_{\{1,2,\ldots,i\}}^j$.

Lemma 6.3. *Let $G = (V, E, \phi)$ be a level planar graph with $k > 1$ levels, and let $v \in V^{j+1}$ be an arbitrary vertex, where $j < k$. Let $R_1^j, R_2^j, \ldots, R_p^j$, $p \geq 2$, be partially reduced extended forms such*

(i) $S_i^v \neq \emptyset$ *for all* $i \in \{1, 2, \ldots, p\}$, *and*

(ii) $\mathrm{LL}(R_1^j) \leq \mathrm{LL}(R_2^j) \leq \mathrm{LL}(R_3^j) \leq \cdots \leq \mathrm{LL}(R_p^j)$.

Then the following holds. If $\pi_{\{1,2,\ldots,i\}}$, $i \leq p$, is a witness to a level planar embedding of $R_{\{1,2,\ldots,i\}}^j$, then the vertices of $\overline{S}_i^v$ form a consecutive sequence in $\pi_{\{1,2,\ldots,i\}}$ and the vertex v appears next to $\overline{S}_i^v$ in $\pi_{\{1,2,\ldots,i\}}$.

Proof. Throughout the proof, we will consider $R_1^j, R_2^j, \ldots, R_i^j$ as well as $R_{\{1,2,\ldots,i-1\}}$ as subgraphs of $R_{\{1,2,\ldots,i\}}$. Let $\pi_{\{1,2,\ldots,i\}}$, $2 \leq i \leq p$, be a witness of a level planar embedding $\mathcal{E}_{\{1,2,\ldots,i\}}$ of $R_{\{1,2,\ldots,i\}}$. The lemma holds trivially, if $\overline{S}_{\{1,2,\ldots,i-1\}}^v = \emptyset$ or $\overline{S}_i^v = \emptyset$. Thus assume, there exists an $x \in \overline{S}_{\{1,2,\ldots,i-1\}}^v \cup \{v\}$, such that x appears between two vertices y_1 and y_2 of $\overline{S}_i^v$ in $\pi_{\{1,2,\ldots,i\}}$. By definition, R_i^j is connected. Furthermore, v is not a cut vertex in R_i^j (otherwise R_i^j would be v-connected). Hence, there exists a path P in R_i^j connecting y_1 and y_2 not containing v. Since $\mathrm{LL}(R_{\{1,2,\ldots,i-1\}}^j) \leq \mathrm{LL}(R_i^j)$ and $R_{\{1,2,\ldots,i-1\}}^j$ is connected, there exist a vertex $z \in R_{\{1,2,\ldots,i-1\}}^j$ such that $lev(z) \leq lev(w)$ for all $w \in P$ and a path $\tilde{P}$ in $R_{\{1,2,\ldots,i-1\}}^j$ connecting x and z (see Fig. 18). By construction the paths P and $\tilde{P}$ are disjoint (since $R_{\{1,2,\ldots,i-1\}}$ and R_i^j are identified only in v), but cross each other,. Thus, $\pi_{\{1,2,\ldots,i\}}$ cannot be a witness of a level planar embedding of $R_{\{1,2,\ldots,i\}}^j$, which is a contradiction.

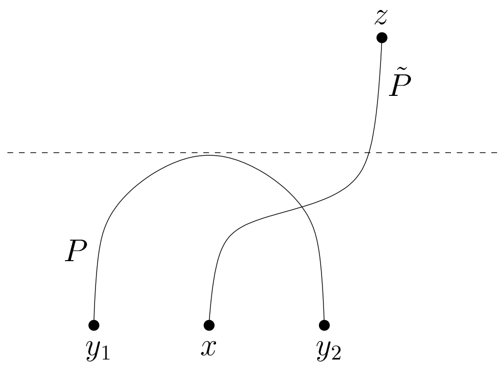


Figure 18: Illustration to the proof of Lemma 6.3. Path P connecting y_1 and y_2 in R_i^j and path $\tilde{P}$ connecting x and z in $R_{\{1,2,\ldots,i-1\}}^j$ cross each other in a level embedding of $R_{\{1,2,\ldots,i\}}^j$ if $x \in \overline{S}_{\{1,2,\ldots,i-1\}}^v \cup v$ appears between $y_1, y_2 \in \overline{S}_i^v$.

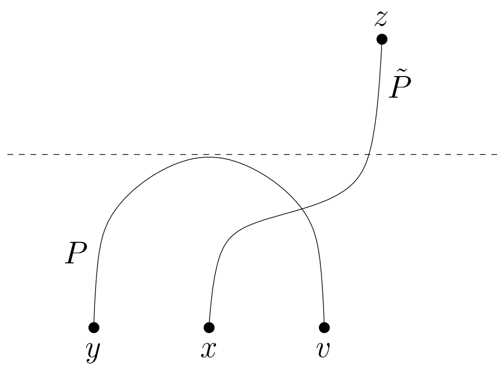


Figure 19: Illustration to the proof of Lemma 6.3. Path P connecting y and v in R_i^j and path $\tilde{P}$ connecting x and z in R_l^j, $l \in \{1, 2, \ldots, i-1\}$, cross each other in a level embedding of $R^j_{\{1,2,\ldots,i\}}$ if $x \in \overline{S}^v_{\{1,2,\ldots,i-1\}}$ appears between $y \in \overline{S}^v_i$ and v.

Assume now that there exists an $x \in \overline{S}^v_{\{1,2,\ldots,i-1\}}$, such that x appears between the vertices of $\overline{S}^v_i$ and v in $\pi_{\{1,2,\ldots,i\}}$, and such that there is a vertex $y \in \overline{S}^v_i$ that appears next to x. Since R_i^j is connected, there exists a path P in R_i^j connecting y and v. By construction $x \in \overline{S}^v_l$ for some $l \in \{1, 2, \ldots, i-1\}$. (Reconsider that v is a cut vertex in $R^j_{\{1,2,\ldots,i-1\}}$ and the cut components are exactly $R_1^j, R_2^j, \ldots, R_{i-1}^j$.) But $\mathrm{LL}(R_l^j) \leq \mathrm{LL}(R_i^j)$ implies that there exist $z \in R_l^j$ such that $lev(z) \leq lev(w)$ for all $w \in P$. Since v is not a cut vertex in R_l^j, there exists a path $\tilde{P}$ in R_l^j connecting x and z that does not contain v (see Fig. 19). Again, since $R^j_{\{1,2,\ldots,i-1\}}$ and R_i^j have only v in common, the paths P and $\tilde{P}$ are disjoint but cross each other, which is a contradiction. □

Using Lemma 6.3, we are able to show Lemma 6.4 which proves the correctness of the merge operations during the second reduction phase. The lemma states that every PQ-tree constructed by v-merging all reduced extended forms with a virtual vertex labeled v according to their size represents exactly all level planar embeddings of the new v-connected form.

Lemma 6.4. *Let $G = (V, E, \phi)$ be a level graph with $k > 1$ levels, and let $v \in V^{j+1}$. Let $R_1^j, R_2^j, \ldots, R_p^j$, $p \geq 2$, be level planar partially reduced extended forms such*

(i) $S_i^v \neq \emptyset$ *for all* $i \in \{1, 2, \ldots, p\}$,

(ii) $|S_i^w| \leq 1$ *for all* $w \in \{1, 2, \ldots, v-1\}$, $i \in \{1, 2, \ldots, p\}$, *and*

(iii) $\mathrm{LL}(R_1^j) \leq \mathrm{LL}(R_2^j) \leq \mathrm{LL}(R_3^j) \leq \cdots \leq \mathrm{LL}(R_p^j)$.

Suppose that the PQ-trees $T(R_1^j), T(R_2^j), \ldots, T(R_p^j)$ represent all level planar embeddings of $R_1^j, R_2^j, \ldots, R_p^j$. Let $T(R^j_{\{1,2,\ldots,p\}})$ be the PQ-tree constructed as described in *the second merge phase of CHECK-LEVEL. Then* $\mathrm{PERM}(T(R^j_{\{1,2,\ldots,p\}}))$ *is exactly the set of permutations of level-*$(j+1)$ *vertices that appear in level planar embeddings of* $R^j_{\{1,2,\ldots,p\}}$.

Proof. We first show that if $\pi \in \mathrm{PERM}(T(R^j_{\{1,2,\ldots,p\}}))$ is a permutation represented by the PQ-tree $T(R^j_{\{1,2,\ldots,p\}})$, then π is a witness to some level planar embedding of $R^j_{\{1,2,\ldots,p\}}$. This can be shown following an idea of Heath and Pemmaraju [1996]. The authors have shown in one of their lemmas the special case of two components $R^j_{\{1,2\}} = R^j_1 \cup_v R^j_2$. We adapt that proof to the more general case and consider v-singular forms.

For all $2 \leq i \leq p$ let $T(R^j_{\{1,2,\ldots,i\}})$ be the PQ-tree constructed in the i-th iteration of the for-loop $(*)$ in the second reduction phase. Now, let $2 \leq i \leq p$ be fixed and assume (by induction) that $T(R^j_{\{1,2,\ldots,i-1\}})$ represents (all) level planar embeddings of $R^j_{\{1,2,\ldots,i-1\}}$. We show that if $\pi_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(T(R^j_{\{1,2,\ldots,i\}}))$, then $\pi_{\{1,2,\ldots,i\}}$ is a witness to a level planar embedding of $R^j_{\{1,2,\ldots,i\}}$.

Let $v_{\{1,2,\ldots,i-1\}}$ be the virtual vertex labeled v in $R^j_{\{1,2,\ldots,i-1\}}$, and let v_i be the virtual vertex labeled v in R^j_i. Two cases may occur, depending on whether R^j_i is v-singular ($\overline{S}^v_i = \emptyset$) or not ($\overline{S}^v_i \neq \emptyset$). We start with the nonsingular case.

1. $\overline{S}^v_i \neq \emptyset$.

   The PQ-tree $T(R^j_{\{1,2,\ldots,i\}})$ has been constructed by reducing the leaves corresponding to $v_{\{1,2,\ldots,i-1\}}$ and v_i in a PQ-tree $\tilde{T}(R^j_{\{1,2,\ldots,i\}})$, and replacing them by the single representative $v_{\{1,2,\ldots,i\}}$ afterwards, where $\tilde{T}(R^j_{\{1,2,\ldots,i\}})$ was the result of the INSERT operation performed on $T(R^j_{\{1,2,\ldots,i-1\}})$ and $T(R^j_i)$. Thus, there exists a $\tilde{\pi}_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(\tilde{T}(R^j_{\{1,2,\ldots,i\}}))$ such that $\pi_{\{1,2,\ldots,i\}}$ arises from $\tilde{\pi}_{\{1,2,\ldots,i\}}$ by identifying the two elements $v_{\{1,2,\ldots,i-1\}}$ and v_i that appear next to each other in $\tilde{\pi}_{\{1,2,\ldots,i\}}$. Since $\overline{S}^v_i \neq \emptyset$, the function INSERT has called the function MERGE. The function MERGE has added the root of $T(R^j_i)$ as a sibling to a node X' in $T(R^j_{\{1,2,\ldots,i-1\}})$. The node X' and its parent X (in case X' was not the root of $T(R^j_{\{1,2,\ldots,i-1\}})$) have been subject to the merge operation in $T(R^j_{\{1,2,\ldots,i-1\}})$. As a result of the merge operation, the leaves of frontier$(T(R^j_i))$ occur consecutively in $\tilde{\pi}_{\{1,2,\ldots,i\}}$, as do the leaves of frontier(X'). Without loss of generality, we assume that the leaves of frontier(X') are immediately followed by the leaves of frontier$(T(R^j_i))$ in $\tilde{\pi}_{\{1,2,\ldots,i\}}$. Hence, the permutation $\tilde{\pi}_{\{1,2,\ldots,i\}}$ can be written as $\tilde{\pi}^a_{\{1,2,\ldots,i\}}\tilde{\pi}^b_{\{1,2,\ldots,i\}}\tilde{\pi}^c_{\{1,2,\ldots,i\}}$ with

   $$\tilde{\pi}^b_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(T(R^j_i))$$

   and

   $$\tilde{\pi}^a_{\{1,2,\ldots,i\}}\tilde{\pi}^c_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(T(R^j_{\{1,2,\ldots,i-1\}}))$$

with $v_{\{1,2,\ldots,i-1\}}$ in $\tilde{\pi}^a_{\{1,2,\ldots,i\}}$ and v_i in $\tilde{\pi}^b_{\{1,2,\ldots,i\}}$ appearing consecutively in $\tilde{\pi}_{\{1,2,\ldots,i\}}$. By assumption, $\tilde{\pi}^a_{\{1,2,\ldots,i\}}\tilde{\pi}^c_{\{1,2,\ldots,i\}}$ is a witness to a level planar embedding $\mathcal{E}^j_{\{1,2,\ldots,i-1\}}$ of $R^j_{\{1,2,\ldots,i-1\}}$ and $\tilde{\pi}^b_{\{1,2,\ldots,i\}}$ is a witness to a level planar embedding $\mathcal{E}^j_i$ of R^j_i. There are two cases that apply depending on whether $\tilde{\pi}^c_{\{1,2,\ldots,i\}}$ is empty or not

(a) $\tilde{\pi}^c_{\{1,2,\ldots,i\}} = \emptyset$. A level planar embedding of $R^j_{\{1,2,\ldots,i\}}$ can be constructed by simply placing R^j_i next to $R^j_{\{1,2,\ldots,i-1\}}$ and then identifying the vertices $v_{\{1,2,\ldots,i-1\}}$ and v_i to a vertex $v_{\{1,2,\ldots,i\}}$. Hence, $\pi_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(T(R^j_{\{1,2,\ldots,i\}}))$ is a witness to a level planar embedding of $R^j_{\{1,2,\ldots,i\}}$.

(b) $\tilde{\pi}^c_{\{1,2,\ldots,i\}} \neq \emptyset$. Let w be the first vertex in $\tilde{\pi}^c_{\{1,2,\ldots,i\}}$ and let Y be the smallest common ancestor of w and $v_{\{1,2,\ldots,i-1\}}$ in $T(R^j_{\{1,2,\ldots,i-1\}})$. Clearly, $w \notin \mathrm{frontier}(X')$. Thus, Y is an ancestor (not necessarily proper) of X (X being the parent of X' before the merge operation). By construction of the merge operation and by Observations 3.1, 3.2, and 3.3 we have

$$\mathrm{ML}(\{v_{\{1,2,\ldots,i-1\}}, w\}) \;<\; \mathrm{LL}(T(R^j_i)) \;.$$

Hence, the level planar embedding $\mathcal{E}^j_i$ of R^j_i can be nested inside the level planar embedding $\mathcal{E}^j_{\{1,2,\ldots,i-1\}}$ of $R^j_{\{1,2,\ldots,i-1\}}$. Merging the virtual vertices $v_{\{1,2,\ldots,i-1\}}$ and v_i to a vertex $v_{\{1,2,\ldots,i\}}$, a level planar embedding $\mathcal{E}^j_{\{1,2,\ldots,i\}}$ of $R^j_{\{1,2,\ldots,i\}}$ is constructed in which the virtual vertices appear according to $\pi_{\{1,2,\ldots,i\}}$. Hence, $\pi_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(T(R^j_{\{1,2,\ldots,i\}}))$ is a witness to a level planar embedding of $R^j_{\{1,2,\ldots,i\}}$.

2. $\overline{S}^v_i = \emptyset$.

There are two possible cases.

(a) The function MERGE was called by INSERT. This case is proven analogously to the case $\overline{S}^v_i \neq \emptyset$.

(b) The function INSERT did not call the function MERGE. Thus either one of the following inequalities holds:

$$\mathrm{PML}(v_{\{1,2,\ldots,i-1\}}) \;<\; \mathrm{LL}(T(R^j_i)) \;,$$

or

$$\mathrm{QML}(v_{\{1,2,\ldots,i-1\}}) \;<\; \mathrm{LL}(T(R^j_i)) \;.$$

It follows from Lemma 5.1 and by construction of the function UPDATE that there exists an interior face or a cavity in some embedding of $R^j_{\{1,2,\ldots,i-1\}}$ that is large enough to level planar embed R^j_i into it. Hence, $\pi_{\{1,2,\ldots,i\}}$ is a witness to a level planar embedding of $R^j_{\{1,2,\ldots,i\}}$.

Thus, one direction of the equivalence stated in the lemma is proved.

To prove the reverse direction, we show (by induction) that for a witness $\pi_{\{1,2,\dots,i\}}$, $2 \leq i \leq p$, of any level planar embedding $\mathcal{E}^j_{\{1,2,\dots,i\}}$ of $R^j_{\{1,2,\dots,i\}}$ the following holds:

$$\pi_{\{1,2,\dots,i\}} \in \mathrm{PERM}(T(R^j_{\{1,2,\dots,i\}})) \ .$$

1. $\overline{S}^v_i \neq \emptyset$.

   The level-$(j+1)$ vertices in $R^j_{\{1,2,\dots,i\}}$ can be partitioned into three sets: $\overline{S}^v_{\{1,2,\dots,i-1\}}$, the set of all level-$j+1$ vertices of $R^j_{\{1,2,\dots,i-1\}}$ except the vertex v, $\overline{S}^v_i$, the set of all level-$(j+1)$ vertices of R^j_i except the vertex v, and the level-$(j+1)$ vertex v. According to Lemma 6.3, the vertices of $\overline{S}^v_i$ appear consecutively in $\pi_{\{1,2,\dots,i\}}$, either immediately followed by or immediately preceded by v. We may assume that the latter case applies. Let $\tilde{R}^j_{\{1,2,\dots,i\}}$ be the graph that consists of $R^j_{\{1,2,\dots,i-1\}}$ and R^j_i, where the level-$(j+1)$ vertices labeled v of the two components are not identified and kept separate. Let $S^v_{\{1,2,\dots,i-1\}} := \{v_{\{1,2,\dots,i-1\}}\}$ where $v_{\{1,2,\dots,i-1\}}$ is the single representative of v in $R^j_{\{1,2,\dots,i-1\}}$ and $S^v_i := \{v_i\}$ where v_i is the single representative of v in R^j_i. "Splitting" in $\pi_{\{1,2,\dots,i\}}$ the vertex v into $v_{\{1,2,\dots,i-1\}}$ and v_i, we get a permutation $\tilde{\pi}_{\{1,2,\dots,i\}}$ that witnesses a level planar embedding $\tilde{\mathcal{E}}^j_{\{1,2,\dots,i\}}$ of $\tilde{R}^j_{\{1,2,\dots,i\}}$.

   The witness $\tilde{\pi}_{\{1,2,\dots,i\}}$ can be written as $\tilde{\pi}^a_{\{1,2,\dots,i\}}\tilde{\pi}^b_{\{1,2,\dots,i\}}\tilde{\pi}^c_{\{1,2,\dots,i\}}$ such that $\tilde{\pi}^a_{\{1,2,\dots,i\}}\tilde{\pi}^c_{\{1,2,\dots,i\}}$ is a witness of a level planar embedding $\mathcal{E}^j_{\{1,2,\dots,i-1\}}$ of $R^j_{\{1,2,\dots,i-1\}}$ and $\tilde{\pi}^b_{\{1,2,\dots,i\}}$ is a witness of a level planar embedding $\mathcal{E}^j_i$ of R^j_i, and such that (without loss of generality) $\tilde{\pi}^a_{\{1,2,\dots,i\}}$ ends with $S^v_{\{1,2,\dots,i-1\}}$, and $\tilde{\pi}^b_{\{1,2,\dots,i\}}$ starts with S^v_i. Since $T(R^j_{\{1,2,\dots,i-1\}})$ and $T(R^j_i)$ correspond to $R^j_{\{1,2,\dots,i-1\}}$ and R^j_i, respectively, it follows by induction that $\tilde{\pi}^a_{\{1,2,\dots,i\}}\tilde{\pi}^c_{\{1,2,\dots,i\}} \in \mathrm{PERM}(T(R_{\{1,2,\dots,i-1\}}))$, and $\tilde{\pi}^b_{\{1,2,\dots,i\}} \in \mathrm{PERM}(T(R^j_i))$. We show that $\tilde{\pi}_{\{1,2,\dots,i\}} \in \mathrm{PERM}(\tilde{T}(R^j_{\{1,2,\dots,i\}}))$, where $\tilde{T}(R^j_{\{1,2,\dots,i\}})$ is the PQ-tree that is constructed by the function INSERT without reducing the PQ-tree with respect to $S^v_{\{1,2,\dots,i-1\}} \cup S^v_i$. There are two cases depending on whether $\tilde{\pi}^c_{\{1,2,\dots,i\}}$ is empty or not.

   (a) $\tilde{\pi}^c_{\{1,2,\dots,i\}} \neq \emptyset$. Suppose that the first vertex in $\tilde{\pi}^c_{\{1,2,\dots,i\}}$ is w. Since according to Lemma 6.3 the vertices of $\overline{S}^v_i$ occur consecutively preceded by v and since the embedding $\tilde{\mathcal{E}}^j_{\{1,2,\dots,i\}}$ of $\tilde{R}^j_{\{1,2,\dots,i\}}$ is level planar the following must hold:

   $$\mathrm{ML}(\{S^v_{\{1,2,\dots,i-1\}}, w\}) < \mathrm{LL}(T(R^j_i)) \ .$$

   Let Y be the node in $\tilde{T}(R^j_{\{1,2,\dots,i-1\}})$ that is the least common ancestor of $S^v_{\{1,2,\dots,i-1\}}$ and w. Then there exists a child Y' of Y such that $S^v_{\{1,2,\dots,i-1\}} \subseteq$

frontier(Y'). Since ML(frontier(Y') $\cup \{w\}$) $\leq$ ML($S^v_{\{1,2,\ldots,i-1\}} \cup \{w\}$), we have according to Observation 3.4 that $\tilde{\pi}_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(\tilde{T}(R^j_{\{1,2,\ldots,i\}}))$.

(b) $\tilde{\pi}^c_{\{1,2,\ldots,i\}} = \emptyset$. According to Observation 3.5 $\tilde{\pi}_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(\tilde{T}(R^j_{\{1,2,\ldots,i\}}))$ holds.

It follows that $\tilde{\pi}_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(\tilde{T}(R^j_{\{1,2,\ldots,i\}}))$ with $S^v_{\{1,2,\ldots,i-1\}} \cup S^v_i$ appearing consecutively in $\tilde{\pi}_{\{1,2,\ldots,i\}}$. This implies that the *PQ*-tree $\tilde{T}(R^j_{\{1,2,\ldots,i\}})$ can be reduced with respect to $S^v_{\{1,2,\ldots,i-1\}} \cup S^v_i$ and therefore $\pi_{\{1,2,\ldots,i\}}$ is contained in $\mathrm{PERM}(T(R^j_{\{1,2,\ldots,i\}}))$.

2. $\overline{S}^v_i = \emptyset$.

There are two cases that may appear.

(a) The set of incoming edges of v in R^j_i (corresponding to S^v_i) separates within the clockwise order of incoming edges of v in $\mathcal{E}^j_{\{1,2,\ldots,i\}}$ the set of incoming edges corresponding to $S^v_{\{1,2,\ldots,i-1\}}$ into two nonempty subsets.

The level-($j+1$) vertices in $R^j_{\{1,2,\ldots,i\}}$ can be partitioned into two sets: $\overline{S}^v_{\{1,2,\ldots,i-1\}}$ the set of all level-($j+1$) vertices of $R^j_{\{1,2,\ldots,i-1\}}$ except the vertex v, and the level-($j+1$) vertex v. Let $\tilde{R}^j_{\{1,2,\ldots,i\}}$ be the form that contains the components $R^j_{\{1,2,\ldots,i-1\}}$ and R^j_i where the incoming edges of v corresponding to $R^j_{\{1,2,\ldots,i-1\}}$ are not identified to v but kept separate. Obviously, $\tilde{R}^j_{\{1,2,\ldots,i\}}$ is level planar. Let $\tilde{\mathcal{E}}^j_{\{1,2,\ldots,i\}}$ be the level planar embedding of $\tilde{R}^j_{\{1,2,\ldots,i\}}$ that is induced by $\mathcal{E}^j_{\{1,2,\ldots,i\}}$. Let $S^{left}_{\{1,2,\ldots,i-1\}}$ be the set of virtual vertices corresponding to the incoming edges of v in $R^j_{\{1,2,\ldots,i-1\}}$ on the left side of R^j_i in $\tilde{\mathcal{E}}^j_{\{1,2,\ldots,i\}}$. Let $S^{right}_{\{1,2,\ldots,i-1\}}$ be the set of virtual vertices corresponding to the incoming edges of v in $R^j_{\{1,2,\ldots,i-1\}}$ on the right side of R^j_i in $\tilde{\mathcal{E}}^j_{\{1,2,\ldots,i\}}$. See Fig. 20 for an illustration. Let $S^v_i := \{v_i\}$, where v_i is the single representative of v in R^j_i. Replacing in $\pi_{\{1,2,\ldots,i\}}$ the vertex v by the set of vertices $S^{left}_{\{1,2,\ldots,i-1\}} \cup S^v_i \cup S^{right}_{\{1,2,\ldots,i-1\}}$ we get a permutation $\tilde{\pi}_{\{1,2,\ldots,i\}}$ that witnesses a level planar embedding $\tilde{\mathcal{E}}^j_{\{1,2,\ldots,i\}}$ of $\tilde{R}^j_{\{1,2,\ldots,i\}}$.

The witness $\tilde{\pi}_{\{1,2,\ldots,i\}}$ can be written as $\tilde{\pi}^a_{\{1,2,\ldots,i\}}\tilde{\pi}^b_{\{1,2,\ldots,i\}}\tilde{\pi}^c_{\{1,2,\ldots,i\}}$ where $\tilde{\pi}^a_{\{1,2,\ldots,i\}}$ ends with $S^{left}_{\{1,2,\ldots,i-1\}}$, $\tilde{\pi}^b_{\{1,2,\ldots,i\}} = \{v_i\}$ and, $\tilde{\pi}^c_{\{1,2,\ldots,i\}}$ starts with $S^{right}_{\{1,2,\ldots,i-1\}}$. Merging the set of vertices $S^{left}_{\{1,2,\ldots,i-1\}}$ and $S^{right}_{\{1,2,\ldots,i-1\}}$ the form $R^j_{\{1,2,\ldots,i-1\}}$ is constructed. By induction, a permutation $\pi^a_{\{1,2,\ldots,i\}}\pi^c_{\{1,2,\ldots,i\}} \in \mathrm{PERM}(T(R^j_{\{1,2,\ldots,i-1\}}))$ is obtained by replacing in

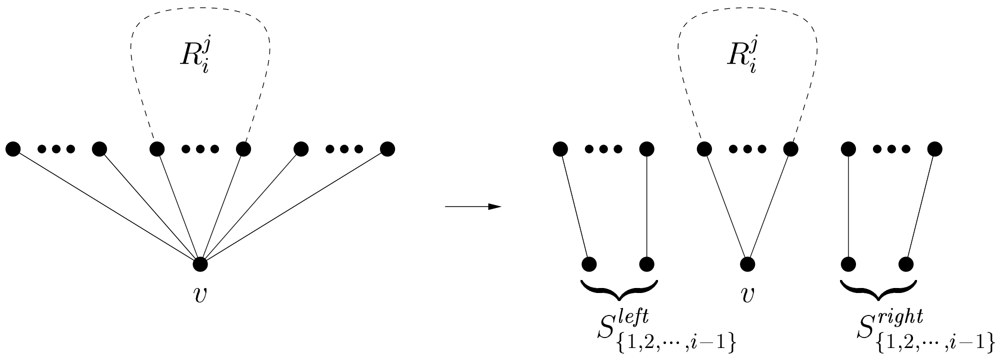


Figure 20: Illustration of the proof of Lemma 6.4. R_i^j is a singular form. The incoming edges of v are partitioned into sets $S^{left}_{\{1,2,\ldots,i-1\}}$, $S^{right}_{\{1,2,\ldots,i-1\}}$ and the edges belonging to R_i^j.

$\tilde{\pi}^a_{\{1,2,\ldots,i\}}\tilde{\pi}^c_{\{1,2,\ldots,i\}}$ the sets $S^{left}_{\{1,2,\ldots,i-1\}}$ and $S^{right}_{\{1,2,\ldots,i-1\}}$ by the single representative $v_{\{1,2,\ldots,i-1\}}$. Since after replacing $v_{\{1,2,\ldots,i-1\}}$ by $v_{\{1,2,\ldots,i\}}$ we have $\pi_{\{1,2,\ldots,i\}} = \pi^a_{\{1,2,\ldots,i\}}\pi^c_{\{1,2,\ldots,i\}}$ this implies that we need to show $T(R^j_{\{1,2,\ldots,i\}}) = T(R^j_{\{1,2,\ldots,i-1\}})$.

Let $v'_{\{1,2,\ldots,i-1\}}$ be the rightmost virtual vertex of $S^{left}_{\{1,2,\ldots,i-1\}}$, and let $v''_{\{1,2,\ldots,i-1\}}$ be the leftmost virtual vertex of $S^{right}_{\{1,2,\ldots,i-1\}}$. Since the embedding of $\tilde{R}^j_{\{1,2,\ldots,i\}}$ is level planar, the following inequality must hold.

$$\mathrm{ML}(v'_{\{1,2,\ldots,i-1\}}, v''_{\{1,2,\ldots,i-1\}}) < \mathrm{LL}(T(R_i^j)) \ .$$

Two possible subcases apply.

i. R_i^j is embedded into an interior face of $R^j_{\{1,2,\ldots,i-1\}}$. Since $v_{\{1,2,\ldots,i-1\}}$ is a cut vertex in $R^j_{\{1,2,\ldots,i-1\}}$, with the cut components being $R_1^j, R_2^j, \ldots, R_{i-1}^j$, the form R_i^j is embedded into an interior face of a form R_l^j, $l \in \{1,2,\ldots,i-1\}$. Thus the virtual vertices $v'_{\{1,2,\ldots,i-1\}}$, and $v''_{\{1,2,\ldots,i-1\}}$ correspond to edges of R_l^j. Let X be the smallest common ancestor of $v'_{\{1,2,\ldots,i-1\}}$ and $v''_{\{1,2,\ldots,i-1\}}$ in $T(R_l^j)$ before the reduction with respect to the leaves labeled v. If X is a P-node we have by proposition 3.3 for the PML-value of the single representative v_l of R_l^j

$$\mathrm{PML}(v_l) \le \mathrm{ML}(X) \le \mathrm{ML}(v'_{\{1,2,\ldots,i-1\}}, v''_{\{1,2,\ldots,i-1\}}) \ .$$

If X is a Q-node, and X' and X'' are the children of X with $v'_{\{1,2,\ldots,i-1\}}$ and $v''_{\{1,2,\ldots,i-1\}}$ in their frontier, respectively, we have by proposition

3.3 for the QML-value of the single representative v_l of R_l^j

$$\mathrm{QML}(v_l) \leq \mathrm{ML}(X', X'') \leq \mathrm{ML}(v'_{\{1,2,\ldots,i-1\}}, v''_{\{1,2,\ldots,i-1\}}) \ .$$

By construction of the function UPDATE it follows that

$$\min\{\mathrm{QML}(v_{\{1,2,\ldots,i-1\}}), \mathrm{PML}(v_{\{1,2,\ldots,i-1\}})\} \leq \mathrm{PML}(v_l)$$

or

$$\min\{\mathrm{QML}(v_{\{1,2,\ldots,i-1\}}), \mathrm{PML}(v_{\{1,2,\ldots,i-1\}})\} \leq \mathrm{QML}(v_l) \ ,$$

and thus INSERT "does nothing".

ii. R_i^j is embedded into a cavity of $R^j_{\{1,2,\ldots,i-1\}}$. Thus i must be at least 3, otherwise no v-cavity exists in $R^j_{\{1,2,\ldots,i-1\}}$. By assumption, we have

$$\mathrm{LL}(R^j_{i-2}) \leq \mathrm{LL}(R^j_{i-1}) \leq \mathrm{LL}(R^j_i) \ .$$

Let X be the root of the pertinent subtree when v-merging R^j_{i-1} into $R^j_{\{1,2,\ldots,i-2\}}$. If X is a P-node, we have by construction $\mathrm{ML}(X) < \mathrm{LL}(T(R^j_{i-1}))$. If X is a Q-node with pertinent adjacent children Y and Z, we have by construction that $\mathrm{ML}(Y, Z) < \mathrm{LL}(T(R^j_{i-1}))$. Let h denote $\mathrm{ML}(X)$ or $\mathrm{ML}(Y, Z)$, respectively. Then we have by construction of the function UPDATE that

$$\min\{\mathrm{QML}(v_{\{1,2,\ldots,i-1\}}), \mathrm{PML}(v_{\{1,2,\ldots,i-1\}})\} \leq h < \mathrm{LL}(R^j_{i-1}) \leq \mathrm{LL}(R^j_i).$$

Thus, again, INSERT "does nothing", and the tree $T(R^j_{\{1,2,\ldots,i-1\}})$ is left unchanged.

(b) Both sets of incoming edges of v corresponding to $S^v_{\{1,2,\ldots,i-1\}}$ and S^v_i form a consecutive sequence within the clockwise order of incoming edges of v in $\mathcal{E}^j_{\{1,2,\ldots,i\}}$. The result follows analogously to the proof of the case $\overline{S}^v_i \neq \emptyset$, with $\tilde{\pi}^b_{\{1,2,\ldots,i\}} = S^v_i$.

□

Theorem 6.5. *The algorithm LEVEL-PLANARITY-TEST tests a given proper level graph $G = (V, E, \phi)$ for level planarity and can be implemented such that the running time is in $\mathcal{O}(|V|)$.*

Proof. The correctness follows immediately from Di Battista and Nardelli [1988] and the Lemmas 6.1, 6.2 and 6.4 by an inductive argument.

The number of operations performed in all calls of the function REDUCE is according to Booth and Lueker [1976] in $\mathcal{O}(|V|)$. The number of steps performed for

v-merging two PQ-trees is proportional to the number of steps that are performed to reduce the pertinent leaves labeled v after a successful merge operation. Thus the total number of operations performed in all calls of MERGE is as well in $\mathcal{O}(|V|)$. The number of steps performed to identify the PQ-trees corresponding to pertinent leaves labeled v is proportional to the number of steps performed in reducing these leaves. Hence, the overall number of operations performed during the identification is bounded by $\mathcal{O}(|V|)$.

We now consider the update operations of the leaves that have to be performed after all merge and reduce operations for a level have been completed. For every PQ-tree we keep its leaves stored in a doubly linked list. Every time two PQ-trees are merged, these lists are merged as well. This can be done without knowing the tree with the lower LL-value. We simply connect the lists at the new single representative that has to be introduced after the merge operation (followed by a reduction) is complete. After finishing all merge and reduce operations we scan for every remaining tree the doubly linked list of its leaves, doing the necessary updates. The total cost of these operations is in $\mathcal{O}(|E|)$ and due to the planarity in $\mathcal{O}(|V|)$. □

7 Remarks

For simplicity, we restricted ourselves in this paper to the level planarity testing of proper level graphs. Of course, every non proper level graph can be transformed into a proper one by inserting dummy vertices. This strategy should not be applied since the resulting number of vertices may be quadratic in the original number of vertices. The following theorem shows that our level planarity test works on non proper level graphs as well as on proper level graphs, having a linear running time for both classes of level graphs.

Theorem 7.1. *The algorithm LEVEL-PLANAR-TEST tests any, not necessarily proper, level graph $G = (V, E, \phi)$ for level planarity in $\mathcal{O}(|V|)$ time.*

Proof. Consider a long edge $e = (v, w)$, $v \in V^j$, $w \in V^l$, $1 \leq j < l - 1 \leq k - 1$, traversing one or more levels. Thus inserting dummy vertices for e in order to construct a proper hierarchy would result in a graph G' such that every dummy vertex u_i^e, $i \in \{j + 1, j + 2, \ldots, l - 1\}$ has exactly one incoming edge and one outgoing edge. However, the reduction of a PQ-tree T with respect to a set S with $|S| = 1$, replacing the set by a new set S' with $|S'| = 1$ is trivial and does not modify the PQ-tree. Hence we do not need to consider the dummy vertices and therefore do not introduce them at all. Therefore, with no change our linear time algorithm correctly tests also nonproper level graphs for level planarity. □

An embedding of a general level planar graph $G = (V, E, \phi)$ can be computed in linear time as follows:

1. Add an extra vertex t on an extra level $k + 1$ and compute a hierarchy by adding an outgoing edge to every sink without destroying level planarity.

2. Add an extra vertex s on an extra level 0 and compute an st-graph by adding the edge (s, t) and an incoming edge to every source without destroying the level planarity.

3. Compute a planar embedding using the algorithm by Chiba et al. [1985].

4. Construct a level planar embedding from the planar embedding.

The difficult part is to insert edges without destroying level planarity. We apply the following strategy (see also Leipert [1998]). The idea is to determine the position of a sink $t \in V_j$, $j \in \{1, 2, \ldots, k-1\}$ by inserting an indicator as a leaf into the PQ-trees. This indicator is ignored throughout the application of the level planarity test and will be removed either with the leaves corresponding to the incoming edges of some vertex $v \in V_l$, $l \in \{j+1, j+2, \ldots, k\}$, or it can be found in the final PQ-tree. However, this strategy is accompanied by a set of difficult case distinctions that are to be discussed in another paper. Nevertheless, the time needed to compute a level planar embedding is bounded by $\mathcal{O}(|V|)$ since the number of extra edges is bounded by the number of sinks and sources in G.

References

K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.

N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa. A linear algorithm for embedding planar graphs using PQ-trees. *Journal of Computer and System Sciences*, 30: 54–76, 1985.

G. Di Battista and E. Nardelli. Hierarchies and planarity theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(6):1035–1046, 1988.

D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific J. Mathematics*, 15:835–855, 1965.

L. S. Heath and S. V. Pemmaraju. Recognizing leveled-planar dags in linear time. In F. J. Brandenburg, editor, *Proc. Graph Drawing '95*, volume 1027 of *Lecture Notes in Computer Science*, pages 300–311. Springer Verlag, 1995.

L. S. Heath and S. V. Pemmaraju. Stack and queue layouts of directed acyclic graphs: Part II. Technical report, Department of Computer Science, Virginia Polytechnic Institute & State University, july 1996.

M. Jünger, S. Leipert, and P. Mutzel. Pitfalls of using PQ-trees in Automatic Graph Drawing. In G. Di Battista, editor, *Graph Drawing '97*, volume 1353 of *Lecture Notes in Computer Science*, pages 193–204. Springer Verlag, 1997.

S. Leipert. *Level Planarity Testing and Embedding in Linear Time.* PhD thesis, Universität zu Köln, 1998.